LNBI 3594

João Carlos Setubal
Sergio Verjovski-Almeida (Eds.)

Advances in Bioinformatics and Computational Biology

Brazilian Symposium on Bioinformatics, BSB 2005
Sao Leopoldo, Brazil, July 2005
Proceedings

Springer

João Carlos Setubal
Sergio Verjovski-Almeida (Eds.)

Advances in Bioinformatics and Computational Biology

Brazilian Symposium on Bioinformatics, BSB 2005
Sao Leopoldo, Brazil, July 27-29, 2005
Proceedings

Springer

Volume Editors

João Carlos Setubal
Virginia Bioinformatics Institute and Department of Computer Science
Virginia Polytechnic Institute and State University, Bioinformatics 1, Box 0477
Blacksburg, VA 24060-0477, USA
E-mail: setubal@vbi.vt.edu

Sergio Verjovski-Almeida
Universidade de Sao Paulo
Instituto de Quimica, Departamento de Bioquimica
Av. Prof. Lineu Prestes 748, 05508-000 Sao Paulo, SP, Brazil
E-mail: verjo@iq.usp.br

Library of Congress Control Number: 2005929321

CR Subject Classification (1998): H.2.8, F.2.1, I.2, G.2.2, J.2, E.1

ISSN	0302-9743
ISBN-10	3-540-28008-1 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-28008-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11532323 06/3142 5 4 3 2 1 0

Preface

The Brazilian Symposium on Bioinformatics (BSB 2005) was held in São Leopoldo, Brazil, July 27–29, 2005, on the campus of the Universidade Vale do Rio dos Sinos (Unisinos). BSB 2005 was the first BSB symposium, though BSB is in fact a new name for a predecessor event called the Brazilian Workshop on Bioinformatics (WOB). WOB was held in three consecutive years: 2002, 2003, and 2004. The change from workshop to symposium reflects the increased reach and quality of the meeting. BSB 2005 was held in conjunction with the Brazilian Computer Society's (SBC) annual conference.

For BSB 2005 we had 55 submissions: 45 full papers and 10 extended abstracts. These proceedings contain the 15 full papers that were accepted, plus 16 extended abstracts (a combination of the accepted abstracts and some full papers that were accepted as extended abstracts). These papers and abstracts were carefully refereed and selected by an international program committee of 40 members, with the help of some additional reviewers, all of whom are listed on the following pages. These proceedings also include papers from three of our invited speakers. We believe this volume represents a fine contribution to current research in bioinformatics and computational biology.

The editors would like to thank: the authors, for submitting their work to the symposium, and the invited speakers; the program committee members and other reviewers for their help in the review process; the Unisinos local organizers, José Mombach and Ney Lemke; Marcelo Walter from Unisinos, coordinator of the SBC conference; Ivan Sendin, from the University of Goiás, who helped with fund raising; Margaret Gabler, from VBI, who helped with the preparation of the proceedings; the symposium sponsors (see list in this volume); Guilherme Telles, Ana Bazzan, Marcelo Brígido, Sergio Lifschitz, and Georgios Pappas, members of the SBC special committee for computational biology; and Springer for agreeing to print this volume.

July 2005

João Carlos Setubal
Sergio Verjovski-Almeida

Organization

BSB 2005 Scientific Program Committee

João Carlos Setubal *Informatics Chair*	(Virginia Bioinformatics Institute, Virginia Tech, USA)
Sergio Verjovski-Almeida *Biology Chair*	(University of São Paulo, Brazil)
Nalvo Almeida Jr.	(Federal University of Mato Grosso do Sul, Brazil)
Ricardo Baeza-Yates	(ICREA-Univ. Pompeu Fabra, Spain & Univ. of Chile)
Valmir Barbosa	(Federal University of Rio de Janeiro, Brazil)
Ana Bazzan	(Federal University of Rio Grande do Sul, Brazil)
Marcelo Brígido	(University of Brasília, Brazil)
Marcelo Briones	(Federal University of São Paulo)
André Carvalho	(University of São Paulo, Brazil)
Julio Collado-Vides	(Autonomous University of Mexico)
Allan Dickerman	(Virginia Tech, USA)
Alan Durham	(University of São Paulo, Brazil)
Carlos Ferreira	(University of São Paulo, Brazil)
James Glazier	(University of Indiana, USA)
Katia Guimaraes	(Federal University of Pernambuco, Brazil)
Lenny Heath	(Virginia Tech, USA)
Victor Jongeneel	(Ludwig Institute, Lausanne, Switzerland)
João Kitajima	(Alellyx, Brazil)
Natalia Martins	(EMBRAPA, Brazil)
Wellington Martins	(Catholic University of Goiás, Brazil)
Marta Matoso	(Federal University of Rio de Janeiro, Brazil)
João Meidanis	(Scylla and University of Campinas, Brazil)
Pedro Mendes	(Virginia Tech, USA)
José Mombach	(Unisinos, Brazil)
Bernard Moret	(University of New Mexico, USA)
Eduardo Jordão Neves	(University of São Paulo)
Ney Lemke	(Unisinos, Brazil)
Sergio Lifschitz	(Catholic University, Rio de Janeiro, Brazil)
Georgios Pappas	(Catholic University of Brasilia, Brazil)
Christian Probst	(Mol.Biol.Institute, Curitiba, Brazil)
Eduardo Reis	(University of São Paulo, Brazil)
Leila Ribeiro	(Federal University of Rio Grande do Sul, Brazil)
Larry Ruzzo	(University of Washington, USA)
Marie-France Sagot	(INRIA, France)
Bruno Sobral	(Virginia Tech, USA)

BSB 2005 Scientific Program Committee (continued)

Siang Song	(University of São Paulo, Brazil)
Osmar Norberto de Souza	(Catholic University of Rio Grande do Sul, Brazil)
Guilherme Telles	(University of São Paulo, Brazil)
Fernando von Zuben	(University of Campinas, Brazil)
Maria Emilia Walter	(University of Brasília, Brazil)

Additional Reviewers

Edson Cáceres
Marcelo Henriques de Carvalho
Jian Chen
Vicky Choi
Lokesh Das
Luciano Digiampietri
Vladimir Espinosa
Katti Faceli
Paulo Roberto Ferreira Jr.
Julio Freyre
Abel González
Marco Gubitoso
Giampaolo Luiz Libralão
Ana Lorena
Sandro Marana
Cleber Mira
Alexey Onufriev
José Augusto Amgarten Quitzau
Cassia Trojahn dos Santos
Marcilio de Souto
Bruno de Souza
Eric Tannier

Local Organizers

José Mombach	(Unisinos, Brazil)
Ney Lemke	(Unisinos, Brazil)

Sponsoring Institutions

Brazilian Computer Society (SBC)
Universidade Vale do Rio dos Sinos
The Brazilian National Council for Research (CNPq)
The Rio Grande do Sul State Research Agency (FAPERGS)
Hewlett-Packard
GE Healthcare
Invitrogen
Microsoft

Table of Contents

Invited Papers

Full Papers

Extended Abstracts

Differential Gene Expression in the Auditory System

Irene S. Gabashvili[1], Richard J. Carter[1], Peter Markstein[1], and Anne B.S. Giersch[2]

[1] Hewlett-Packard Labs, Computational Biosciences Research, 1501 Page Mill Road,
Palo Alto, CA, 94304, USA
{Irene.Gabashvili, Dick.Carter, Peter.Markstein}@HP.com
http://hpl.hp.com/research/cbsr
[2] Department of Pathology, BWH, Harvard Medical School,
75 Francis Street, 02115 Boston, USA
agiersch@rics.bwh.harvard.edu
http://hearing.bwh.harvard.edu/

Abstract. Hearing disorders affect over 10% of the population and this ratio is dramatically increasing with age. Development of appropriate therapeutic approaches requires understanding of the auditory system, which remains largely incomplete. We have identified hearing-specific genes and pathways by mapping over 15000 cochlear expressed sequence tags (ESTs) to the human genome (NCBI Build 35) and comparing it to other EST clusters (Unigene Build 183). A number of novel potentially cochlear-specific genes discovered in this work are currently being verified by experimental studies. The software tool developed for this task is based on a fast bidirectional multiple pattern search algorithm. Patterns used for scoring and selection of loci include EST subsequences, cloning-process identifiers, and genomic and external contamination determinants. Comparison of our results with other programs and available annotations shows that the software developed provides potentially the fastest, yet reliable mapping of ESTs.

1 Introduction

Personalized medicine in the future will be based on the comparison of individual genetic information to reference gene expression, molecular interactions and pathways in tissues and organs, in health and disease. It will be based on advanced genome sequencing, gene expression, proteomic and metabolomic technologies, as well as efficient computational tools for mapping of genes and pathways.

The reliability of computational approaches and models is improving, as "omic" technologies mature and the accuracy of predictions grows with increasing data input. There is a growing need for fast software tools capable of handling massive amounts of data and reanalyzing the data to discover integrated knowledge and identify broken links and wrong connections between intricate processes in individual datasets.

The first step in comparing genomic information is to align DNA sequences, that is, to map nucleotides of expressed sequence tags (ESTs) or full cDNAs to the genome and sequences of known and predicted genes. Sequence alignment is one of

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 1–8, 2005.

the oldest and most successful applications of Computer Science to Biology [1-2]. Many local pairwise alignment methods exist [1-6] and most software tools are freely available. These tools, however, are customized for specific tasks and do not allow enough flexibility for new specialized tasks to external users. The most popular generic programs relevant to EST mapping, BLAST from the National Center for Biotechnology Information [6] and BLAT from U.C. Santa Cruz [4], each have their strengths and weaknesses. The BLAST service offered by NCBI is too slow to use for sets of tens of thousands ESTs. Moreover, it does not handle intron gaps well when used for the whole-genome mappings and works best on expressed sequence databases. The BLAT service offered by UCSC is fast, but its interactive nature and 25-sequence submission limit would prevent its use on a large number of sequences.

To direct and control the process of EST mapping, we needed software with problem-specific intelligence that was not available with existing tools. One of the most important tasks in processing experimental data is estimating the errors and potential sources of errors in measurements [7]. Cloning and sequencing artifacts, for example, could be eliminated using pre-screening procedures. Accordingly, we needed not only to align ESTs, but also check for a number of favorable and detrimental signals, to identify the most likely mapping amongst many possibilities.

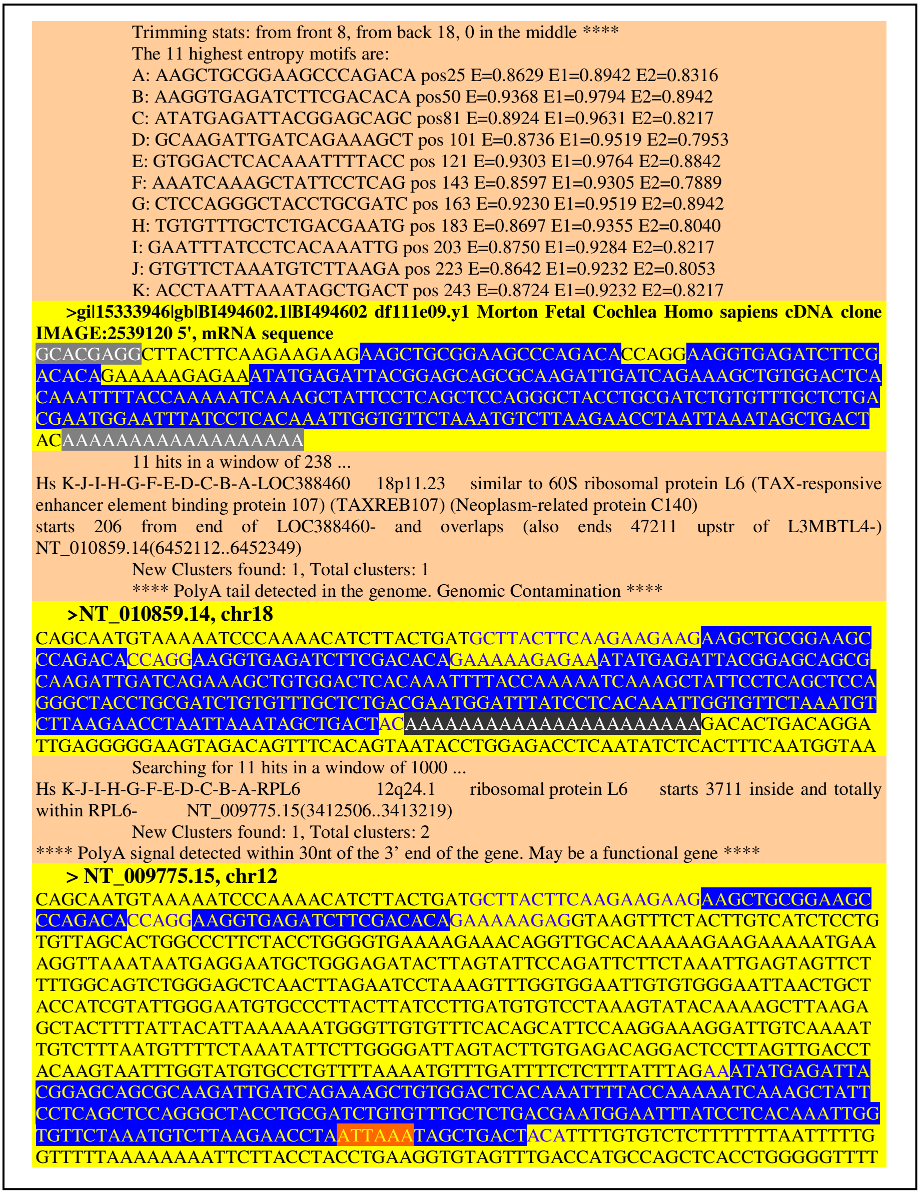
In this work, we have analyzed over fifteen thousand ESTs expressed in the human cochlea. The cochlea is one of the smallest organs in the body located in the inner ear and responsible for auditory transduction (conversion of sound into the language of the brain). Hearing impairment is always the result of damage to either the middle ear, the cochlea or its associated auditory nerve. Over one hundred genes responsible for deafness have been discovered, but many more candidates apparently exist. A much smaller fraction of molecular-level auditory pathways have been identified [8-10], mostly due to the lack of knowledge of human biology in general.

We have mapped and analyzed genes predominantly expressed in the inner ear and their pathways. We have also studied cochlear genes expressed in low numbers. We show that the vast majority of cochlea-unique genes identified by existing tools and servers are either genomic contaminations or can be also found in other tissues. We have selected a small subset of cochlea-specific genes and they are currently being verified by independent experimental methods.

2 Computational Approach

To speed up alignment of ESTs to the genome and improve the scoring of such mappings, we reduced the problem to that of simultaneous exact matching of multiple motifs within ESTs to localized genome regions. Our approach is illustrated on the example of a particular Morton cochlear EST (Fig. 1).

Mapping and selection of ESTs is realized by dynamic interaction of two in-house programs, *Enhancer2* and *BatchSearch*. *Enhancer2* is a 5000-line C++ program that finds exact matches of a number of input search patterns within a database of sequences (whole genomes, mRNAs, etc). The fast exact string prefix matching algorithm (Dick Carter and Peter Markstein, to be published) was applied to other genome search problems in early stages of its development [11]. Some of the features



```
                Trimming stats: from front 8, from back 18, 0 in the middle ****
                The 11 highest entropy motifs are:
                A: AAGCTGCGGAAGCCCAGACA pos25 E=0.8629 E1=0.8942 E2=0.8316
                B: AAGGTGAGATCTTCGACACA pos50 E=0.9368 E1=0.9794 E2=0.8942
                C: ATATGAGATTACGGAGCAGC pos81 E=0.8924 E1=0.9631 E2=0.8217
                D: GCAAGATTGATCAGAAAGCT pos 101 E=0.8736 E1=0.9519 E2=0.7953
                E: GTGGACTCACAAATTTTACC pos 121 E=0.9303 E1=0.9764 E2=0.8842
                F: AAATCAAAGCTATTCCTCAG pos 143 E=0.8597 E1=0.9305 E2=0.7889
                G: CTCCAGGGCTACCTGCGATC pos 163 E=0.9230 E1=0.9519 E2=0.8942
                H: TGTGTTTGCTCTGACGAATG pos 183 E=0.8697 E1=0.9355 E2=0.8040
                I: GAATTTATCCTCACAAATTG pos 203 E=0.8750 E1=0.9284 E2=0.8217
                J: GTGTTCTAAATGTCTTAAGA pos 223 E=0.8642 E1=0.9232 E2=0.8053
                K: ACCTAATTAAATAGCTGACT pos 243 E=0.8724 E1=0.9232 E2=0.8217
     >gi|15333946|gb|BI494602.1|BI494602 df111e09.y1 Morton Fetal Cochlea Homo sapiens cDNA clone
IMAGE:2539120 5', mRNA sequence
GCACGAGGCTTACTTCAAGAAGAAGAAGCTGCGGAAGCCCAGACACCAGGAAGGTGAGATCTTCG
ACACAGAAAAAGAGAAATATGAGATTACGGAGCAGCGCAAGATTGATCAGAAAGCTGTGGACTCA
CAAATTTTACCAAAAATCAAAGCTATTCCTCAGCTCCAGGGCTACCTGCGATCTGTGTTTGCTCTGA
CGAATGGAATTTATCCTCACAAATTGGTGTTCTAAATGTCTTAAGAACCTAATTAAATAGCTGACT
ACAAAAAAAAAAAAAAAAAA
                11 hits in a window of 238 ...
Hs K-J-I-H-G-F-E-D-C-B-A-LOC388460     18p11.23     similar to 60S ribosomal protein L6 (TAX-responsive
enhancer element binding protein 107) (TAXREB107) (Neoplasm-related protein C140)
starts 206 from end of LOC388460- and overlaps (also ends 47211 upstr of L3MBTL4-)
NT_010859.14(6452112..6452349)
                New Clusters found: 1, Total clusters: 1
                **** PolyA tail detected in the genome. Genomic Contamination ****
     >NT_010859.14, chr18
CAGCAATGTAAAAATCCCAAAACATCTTACTGATGCTTACTTCAAGAAGAAGAAGCTGCGGAAGC
CCAGACACCAGGAAGGTGAGATCTTCGACACAGAAAAAGAGAAATATGAGATTACGGAGCAGCG
CAAGATTGATCAGAAAGCTGTGGACTCACAAATTTTACCAAAAATCAAAGCTATTCCTCAGCTCCA
GGGCTACCTGCGATCTGTGTTTGCTCTGACGAATGGATTTATCCTCACAAATTGGTGTTCTAAATGT
CTTAAGAACCTAATTAAATAGCTGACTACAAAAAAAAAAAAAAAAAAAAAAGACACTGACAGGA
TTGAGGGGGAAGTAGACAGTTTCACAGTAATACCTGGAGACCTCAATATCTCACTTTCAATGGTAA
                Searching for 11 hits in a window of 1000 ...
Hs K-J-I-H-G-F-E-D-C-B-A-RPL6          12q24.1     ribosomal protein L6     starts 3711 inside and totally
within RPL6-         NT_009775.15(3412506..3413219)
                New Clusters found: 1, Total clusters: 2
**** PolyA signal detected within 30nt of the 3' end of the gene. May be a functional gene ****
     > NT_009775.15, chr12
CAGCAATGTAAAAATCCCAAAACATCTTACTGATGCTTACTTCAAGAAGAAGAAGCTGCGGAAGC
CCAGACACCAGGAAGGTGAGATCTTCGACACAGAAAAAGAGGTAAGTTTCTACTTGTCATCTCCTG
TGTTAGCACTGGCCCTTCTACCTGGGGTGAAAAGAAACAGGTTGCACAAAAAGAAGAAAAATGAA
AGGTTAAATAATGAGGAATGCTGGGAGATACTTAGTATTCCAGATTCTTCTAAATTGAGTAGTTCT
TTTGGCAGTCTGGGAGCTCAACTTAGAATCCTAAAGTTTGGTGGAATTGTGTGGGAATTAACTGCT
ACCATCGTATTGGGAATGTGCCCTTACTTATCCTTGATGTGTCCTAAAGTATACAAAAGCTTAAGA
GCTACTTTTATTACATTAAAAAATGGGTTGTGTTTCACAGCATTCCAAGGAAAGGATTGTCAAAAT
TGTCTTTAATGTTTTCTAAATATTCTTGGGGATTAGTACTTGTGAGACAGGACTCCTTAGTTGACCT
ACAAGTAATTTGGTATGTGCCTGTTTTAAAATGTTTGATTTTCTCTTTATTTAGAAATATGAGATTA
CGGAGCAGCGCAAGATTGATCAGAAAGCTGTGGACTCACAAATTTTACCAAAAATCAAAGCTATT
CCTCAGCTCCAGGGCTACCTGCGATCTGTGTTTGCTCTGACGAATGGAATTTATCCTCACAAATTGG
TGTTCTAAATGTCTTAAGAACCTAATTAAATAGCTGACTACATTTTGTGTCTCTTTTTTTAATTTTTG
GTTTTTAAAAAAAATTCTTACCTACCTGAAGGTGTAGTTTGACCATGCCAGCTCACCTGGGGGTTTT
```

Fig. 1. Our approach to mapping and scoring of results illustrated on the example of a sequence with accession number BI49460. As a first step, we determined detrimental motifs in this sequence (shaded in grey) and trimmed them off. Blue area represents dynamically selected subsequences used for matching to the human genome. The program found two equally well matching regions in chromosomes 12 and 18. A detrimental signal (polyA tail (black shading), in chromosome 18 and a favorable motif in chromosome 12 determined the best mapping. See text for details

of this algorithm are its ability to handle all IUPAC nucleotide codes with little additional overhead and its high parallelization efficiency.

The other component of our EST-mapping solution is *BatchSearch*, a 2500-line C++ program that interacts with *Enhancer2* by giving it search tasks and dynamically responding to its output. Using the fast exact-matching *Enhancer2* speeds the alignment process since EST-mapping would normally require slower inexact matching to cope with introns and frequent EST sequencing errors or single nucleotide polymorphisms (SNPs). Our idea was to divide an EST into smaller fragments and, using *Enhancer2*, find where some of them occur. Normally the bulk of the fragments would be found clustered within the same locale, thus forming the basis for the reported EST mapping. In the majority of cases, we also observed a very high level of identity, as an entire EST sequence after trimming often exactly matched to a localized region within the genome.

The logic of *BatchSearch* involves a number of steps. First, the input EST is trimmed of bases that are artifacts of the sequencing process (Fig.1). Second, a globally optimal set of high-entropy fragments is chosen from the EST using a dynamic programming algorithm. Then, the formulated exact-match search problem is passed to the waiting *Enhancer2* program. Depending on these results, *BatchSearch* can ask *Enhancer2* to refilter its search results, allowing for more widely dispersed clusters to be reported. In addition, clusters of other detrimental and favorable motifs in the genome are taken into account. Fig.1 demonstrates two such motifs – a polyA tail (*black shading*) that is supposed to be located within 30 nucleotides of the 3' end (larger distance may be allowed in the 5' EST) and a polyA signal (see [12], *orange shading*, not be followed by polyA tail in the genome) Alternatively, *BatchSearch* can redo the genome search with smaller EST subsequences, in an effort to identify the most likely mapping. One search for six 20-nucleotide fragments using *Enhancer2* takes about 2.5 seconds on a 2.8 gHz Xeon CPU with one Giga Byte of RAM. A dual-processor HP XW8000 PC workstation requires 5.5 hours to map the entire library of 15000 cochlear ESTs to the human genome. Datasets with less mapping ambiguity are processed faster.

3 Genes and Pathways of the Human Cochlea

Only from 60% to 95% of all deposited ESTs in tissue- and organ-specific libraries are classified by Unigene. Fig.2 demonstrates the ratio of classified vs. unclassified sequences for fetal cochlear, eyes and brain libraries and adult bone and stomach datasets. Only 11,913 human cochlear sequences out of fifteen thousand deposited (dbEST Library ID.371 [13,14]) are annotated in Unigene. We mapped over 98% (all but 276 – area 3 in inset of Fig.2 showing sequences not available in Unigene) of the ESTs in the Morton fetal cochlear library to specific regions in the human genome and genomes of laboratory organisms. Of the unmapped sequences, most correspond to highly conserved regions that can be exactly matched to dozens of proteins in a variety of organisms. The remaining unmapped ESTs seem to be formed by nonspecific recombination events and cannot be confidently attributed to a specific gene or genome. Non-human contaminations in the dataset (259, area 4 in Fig.2) come from laboratory organisms – mainly yeast, E.coli, phages and cloning vectors, but there are also single occurences of such unexpected species as worm and mouse. Among about five thousand genes identified, almost 2000 genes are represented by single ESTs. Less than 200 genes are supported by ten or more sequences. The most abundant mRNAs were for extracellular matrix genes. This can be explained by the importance of structural support in cochlea. We note that this class of proteins acounts for almost half of nonsyndromic deafness genes.

Less than 10% of all our cochlea sequences were deposited with gene-relevant information in their headers, while 41% of the sequences were annotated based on results of BLAST searches against GenBank databases in early 2000s. Almost 80% from this set are annotated in the latest build of Unigene, although about 8% of these annotations remain hypothetical. We selected many different isoforms among ESTs clustered in the same Unigene clusters. In addition to the 4058 Unigene clusters, we determined almost 1000 additional loci, many of which might represent novel genes or isoforms of known genes (areas 1 and 4 in Fig.2). We found about 20% potential genomic contaminations in the dataset and 1% of sequence flips in EST sequences. Many transcripts corresponding to ESTs present in the dataset might not be expressed as proteins, but instead are degraded by nonsense-mediated mRNA decay or other cell surveillance mechanisms. We revealed a number of incomplete, truncated mRNAs in the library, confirming this possibility.

The inset of Figure 2 shows how sequences extracted from the fetal inner ear and not classified by Unigene are mapped to the human genome and genomes of other species (human pathogens and laboratory organisms). Comparison of our mappings to alignments produced by popular tools, such as BLAST [6] and BLAT [4], shows that our solutions are essentially the same. These other tools, however, offer the best solutions among several other top scoring results, thus requiring post-processing of results, often manually. We note that most of our novel genes are also suggested in the AceView database [15] and are being incorporated into the next build of the human genome. On the one hand, we consider it as another confirmation of the reliability of our findings. On the other hand, we note that the subject of this work is analysis of hearing-specific genes and this was not done by the authors of AceView, GeneScan and other global gene-finding programs.

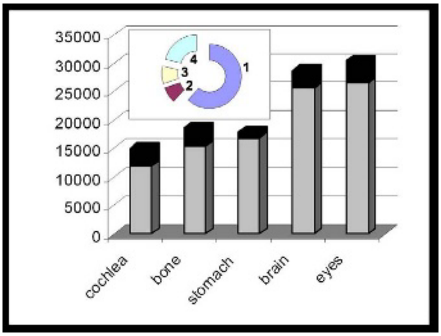


Fig. 2. A bar-chart of sequences of organ-specific libraries classified (white base) and not classified (black top) into Unigene entries. Inset shows our mappings of non-classified cochlear ESTs. Sequences in areas: (1) may be novel isoforms of known genes; (2) are non-human genes; (3) are ambiguous; 4) map to unannotated regions in the human genome

Table 1. The most highly expressed genes and predominant pathways of the human cochlea

Name	EST count, Unigene	EST count, this work	PATHWAYS		
			Ion Transport	Cell Shape Maintenance	Housekeeping
Collagen, type I, alpha 2	314	343		Collagen matrix	
Collagen, type III, alpha 1	153	159		Collagen matrix	
Secreted protein, acidic, cysteine-rich(osteonectin)	125	162		Binds Collagen	
Eukaryotic translation elongation factor 1 alpha 1	81	130		Binds Actin	Protein Synthesis
Vimentin	80	84		Intermediate Filament	Structure and Motiilty
Collagen, type I, alpha 1	70	83		Collagen matrix	
Myristoylated alanine-rich protein kinase C substrate	63	67		Binds Actin	Structure and Motiilty
KIAA1040 protein	55	56	Proton transport		
Tumor protein, translationally controlled	51	53		Extracellular matrix	Structure and Motiilty
Chromosome 5 open reading frame 13	50	56		Cell junctions	
Actin, beta	46	54		Actin filament	Structure and Motiilty
Potassium channel tetramerisation domain containing 12	44	49	Potassium transport		
Actin, gamma 1	42	47		Actin Filament	Structure and Motiilty
Ribosomal protein S20	38	39			Protein Synthesis
Cyclin I	36	37			Cell Cycle Regulation

Comparison of our results to available Unigene assignments shows a very good correspondence as well. Our "new gene" mappings often correspond to "transcribed loci" and most discrepancies in gene names are solely due to different naming of the same genes. For example, "ecotropic viral integration site 2A" is the same gene as "neurofibromin 1 (neurofibromatosis, von Recklinghausen disease, Watson disease)", and ALEX2 is the same as ARMCX2. Less than 1% of our EST mappings do not correspond to Unigene assignments. In half of these cases our results might be better. In several cases old Unigene assignments seem to be better than the latest ones.

In order to normalize the cochlear library to find crucial components of hearing transduction, all housekeeping and cell structure maintenance genes have to be subtracted from the set. This task is not trivial, as many proteins have multiple functions and the difference between cochlear and other existing libraries is statistically significant only for a very small number of relatively highly expressed genes. These are collagens (col1a2, col3a1) and osteonectin (if compared to fetal brain, structural tissues or whole embryo). Comparison with libraries from other tissues points additionally to several other candidates. For example, a protein potentially involved in the assembly of potassium channels is known to be implied in the hearing process ("potassium channel tetramerisation domain containing 12"). Table 1 shows fifteen genes of the human cochlea with the highest level of expression. We note that some of the ESTs appear as genomic contaminations (data not shown) and might not be expressed in the cell. Many such sequences, however, are annotated as legitamate genes in public databases.

We identified a number of pathways including abundant transcripts of the dataset, not-directly related to hearing. They describe cell proliferation, maintenance of ion balance, protein synthesis, splicing, transcription, regulation of actin cytoskeleton, etc. The table shows that certain cellular shape maintenance pathways (extracellular junction and matrix-related) are hearing related, rather than for housekeeping (see [16-17] for lists of housekeeping genes). This can be explained by the importance of maintenance of acoustic resonator structures (on the level of cell assemblies) in the ear.

For genes present in a small number of copies, we can employ a bottom-up approach by focusing on potentially novel genes that seem to be solely or predominantly expressed in the cochlea, then reconstructing pathways involving products of these genes. We selected about 200 clusters of ESTs potentially representing novel genes not classified by Unigene. We have further narrowed this list down by filtering out genomic contaminations and highly repetitive sequences. The candidate genes include possible transcription factors (gene-regulatory pathways), a motor protein (cell shape maintenance), an isoform of collagen (cell shape maintenance) and a transmembrane protein (ion transport). The findings are currently being verified by RT-PCR and other laboratory tests.

4 Concluding Remarks

Crucial processes of life, hearing being one of them, are only partially understood at the molecular level. Important but low-abundant proteins remain elusive. Large-scale sequencing of tissue-specific genes and fast yet reliable mapping of sequences will help to identify the key components of sensory sound transduction pathways. Eventually, this will bring a cure and better treatment to now-incurable deafness and age-related hearing loss.

References

1. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ.: Basic local alignment search tool. *J. Mol. Biol.* (1990), **215**, 403-410.
2. Batzoglou S.: The many faces of sequence alignment *Brief. Bioinform.* (2005), **6**, 6-22.
3. Gemund, C., Ramu, C., Altenberg-Greulich, B., Gibson, T.J.:Gene2EST: a BLAST2 server for searching expressed sequence tag (EST) databases with eukaryotic gene-sized queries. *Nucleic Acids Res* ((2001) , **29,** 1272–1277
4. Kent, W.J.: BLAT—the BLAST-like alignment tool. *Genome Res (2002)*, **12,** 656–664
5. Krüger, J., Sczyrba, A., Kurtz, S., Giegeri, R.: e2g: an interactive web-based server for efficiently mapping large EST and cDNA sets to genomic sequences. *Nucleic Acids Res.* (2004), **32** (Web Server issue), W301-4
6. Altschul, S.F., Madden T.L., Schoffer A.A., Zhang J., Zhang Z., Miller W., Lipman, D.J.: Gapped BLAST and PSI–BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **25**:3389–3402, 1997.
7. Whirl-Carrillo, M., Gabashvili, I.S., Banatao, D.R., Bada, M., Altman, R.B.: Mining biochemical information: lessons taught by the ribosome. *RNA* (2002) **8**, 279-289.
8. Chen, Z.-Y., Corey, D.P.: Understanding Inner Ear Development with Gene Expression Profiling. *Journal of Neurobiology* (2002), **53**, 276-285.
9. Lin, J., Ozeki, M., Javel, E., Zhao, Z., Pan,W., Schlentz, E., Levine, S.: Identification of gene expression profiles in rat ears with cDNA microarrays. *Hear Res.* (2003), **175**, 2-13.
10. McGuire, J.F. ., Casado, B.: Proteomics: a primer for otologists, *Otol Neurotol.* (2004), **25**, 842-849.
11. Markstein, M., Markstein, P., Markstein, V., Levine, M.S.: Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the Drosophila embryo. *Proc Natl Acad Sci U S A* (2002), **99**, 763-768.
12. Beaudoing, E., Freier, S., Wyatt, J.R., Claverie, J.-M., Gautheret, D.: Patterns of Variant Polyadenylation Signal Usage in Human Genes, Gen.Res. (2000), 10, 1001-1010.
13. Robertson, N.G., Khetarpal, U., Gutierrez-Espelata, G.A. , Bieber, F.R., Morton, C.C.: Isolation of novel and known genes from a human fetal cochlear cDNA library using subtractive hybridization and differential screening. Genomics (1994), 23, 42-50.
14. Skvorak, A.B., Weng, Z., Yee, A.J., Robertson, N.G., Morton, C.C.: Human cochlear expressed sequence tags provide insight into cochlear gene expression and identify candidate genes for deafness", *Hum Mol Genet.* (1999), **8**, 439-452.
15. Thierry-Mieg, D., Thierry-Mieg, J-T., Potdevin, M., Sienkiewicz, M.: Identification and functional annotation of cDNA-supported genes in higher organisms using AceView, unpublished. http://www.aceview.org/
16. Hsiao, L.L., Dangond, F., Yoshida, T., Hong, R., Jensen, R.V., Misra, J., Dillon, W., Lee, K.F., Clark, K.E., Haverty, P., Weng, Z., Mutter, G.L., Frosch, M.P., Macdonald, M.E., Milford, E.L., Crum, C.P., Bueno, R., Pratt, R.E., Mahadevappa, M., Warrington, J.A., Stephanopoulos, G., Stephanopoulos, G., Gullans, S.R. A Compendium of Gene Expression in Normal Human Tissues**.** Physiol Genomics. (2001), 21, 97-104
17. Eisenberg, E., Levanon, E.Y.: Human housekeeping genes are compact. Trends Genet. (2003), 19, 362-365.

Searching for Non-coding RNA

Walter L. Ruzzo

University of Washington, Box 352350, Seattle, WA, 981195-2350, USA
http:\\www.cs.washington.edu/homes/ruzzo

Non-coding RNAs (ncRNAs) are functional RNA molecules that do not code for proteins. Classic examples include ribosomal and transfer RNAs, but dramatic discoveries in the last few years have greatly expanded both the number of known ncRNAs and the breadth of their biological roles [1]. In short, ncRNAs are much more biologically significant than previously realized.

The computational problems associated with discovery and characterization of ncRNAs are quite different from, and arguably more difficult than, comparable tasks for protein-coding genes [2]. A key element of this difference is the importance of secondary structure in most ncRNAs. RNA secondary structure prediction is a well-studied problem, and useful tools exist, but they are certainly not perfect. It is generally accepted that the best evidence for stable secondary structure in biologically relevant RNAs is to identify diverged examples exhibiting compensatory base-pair changes that would preserve putative structural elements. Unfortunately, such compensatory mutations interfere with the ability of standard sequence search and alignment tools (e.g., BLAST, ClustalW) to find and align homologs.

This talk will attempt to outline the problems and promises of computational search for ncRNA, with some emphasis on work by my group, including the following. One successful approach to ncRNA homology search that exploits secondary structure conservation employs so-called Covariance Models (CMs), statistical models based on probabilistic context-free grammars [3]. CMs are used, for example, in the important Rfam database [4]. CM searches, although highly accurate, are very slow – years of CPU time. We have developed novel algorithms to make CMs faster, while provably sacrificing none of their accuracy. For most known ncRNA families, this allows genome databases to be scanned in days instead of years, and yields new ncRNAs missed by the heuristics that were necessary for practical CM searches until now [5, 6]. Constructing covariance models is somewhat laborious. We are also developing new methods to automatically learn CM's from a small number of unaligned example RNA sequences [7]. Most importantly, these methods have led us to discovery and/or improved characterization of interesting ncRNAs [8, 9, 10].

References

1. Storz, G.: An expanding universe of noncoding RNAs. Science **296** (2002) 1260–1263
2. Eddy, S.R.: Computational genomics of noncoding RNA genes. Cell **109** (2002) 137–140

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 9–10, 2005.

3. Eddy, S.R., Durbin, R.: RNA sequence analysis using covariance models. Nucleic Acids Research **22** (1994) 2079–2088
4. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S.R., Bateman, A.: Rfam: annotating non-coding RNAs in complete genomes. Nucleic Acids Res **33** (2005) 121–124
5. Weinberg, Z., Ruzzo, W.L.: Faster genome annotation of non-coding RNA families without loss of accuracy. In: RECOMB04: Proceedings of the Eighth Annual International Conference on Computational Molecular Biology, San Diego, CA (2004) 243–251
6. Weinberg, Z., Ruzzo, W.L.: Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. Bioinformatics **20** (2004) i334–i341 ISMB 2004.
7. Yao, Z., Weinberg, Z., Ruzzo, W.L.: Cmfinder: A covariance model based RNA motif finding algorithm. Submitted. (2005)
8. Mandal, M., Lee, M., Barrick, J.E., Weinberg, Z., Emilsson, G.M., Ruzzo, W.L., Breaker, R.R.: A glycine-dependent riboswitch that uses cooperative binding to control gene expression in bacteria. Science **306** (2004) 275–279
9. Barrick, J.E., Sudarsan, N., Weinberg, Z., Ruzzo, W.L., Breaker, R.R.: 6S RNA is a widespread regulator of eubacterial RNA polymerase that resembles an open promoter. RNA **11** (2005) 774–784 [epub 2005 April 5].
10. Trotochaud, A.E., Wassarman, K.M.: A highly conserved 6S RNA structure is required for regulation of transcription. Nat Struct Mol Biol **12** (2005) 313–319

Cyberinfrastructure for PathoSystems Biology

Bruno W.S. Sobral

Virginia Bioinformatics Institute at Virginia Tech (0477), Blacksburg VA USA 92024
sobral@vt.edu
http://www.vbi.vt.edu

Abstract. The application of new information and biotechnologies to infectious disease research provides an opportunity to design, develop and deploy a comprehensive cyberinfrastructure for life sciences. The application of integrative approaches including theory, wet experimentation, modeling and simulation and the leveraging of a strong comparative, evolutionary framework has spawned pathosystems biology. I will show examples of how cyberinfrastructure is being developed and used to support pathosystems biology.

1 Introduction

The application of modern information technologies and biotechnologies (including genome-scale approaches, systems biology, etc.) in the context of infectious diseases has spawned a new way to augment our understanding of infectious diseases, as well as new opportunities to leverage the knowledge and apply it to the development of countermeasures (surveillance, vaccines, therapeutics, diagnostics, etc.) to help protect the global community from attacks by infectious agents (of plants, animals, and humans). This paper will focus on these concepts in the context of the research and development programs I am responsible for implementing.

1.1 Cyberinfrastructure

The Atkins Report on cyberinfrastructure (CI) recalled how infrastructure in general is taken for granted until it stops functioning [1]. For life scientists, thinking about infrastructure is novel in most cases, although the need and power of infrastructure has been shown to most life scientists through the Human Genome Project. Many have pointed out how infrastructure is complex and expensive and should be built specifically by groups capable of developing infrastructure. CI refers to infrastructure based upon distributed computer, information and communication technology. Furthermore, CI is required for a knowledge economy, and biological knowledge is required to support the needs of infectious disease research and development. CI technologies are the components of computation, storage, and communication; also the software programs, services, instruments, data, information, knowledge, and social practices applicable to specific projects, disciplines, and communities, in the case of infectious diseases, microbiology and related bioscience fields (for example those that consider the effects of pathogens on hosts, such as immunology, plant pathology, etc.). Furthermore, there is the layer of enabling hardware, algorithms, software,

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 11 – 27, 2005.

communications, institutions, and personnel. This crucial layer enables specific communities of researchers to innovate and eventually change what they do, how they do it, and who participates. This last layer requires institutions with service-oriented staff and core facilities to provide operational support and services, as well as high-impact applications of CI in relevant areas of science and engineering research and allied education. I believe that infectious diseases provide a high-impact arena in which to develop and deploy CI for life sciences. Infectious disease biology is ready for CI because full deployment of a working system to support public health and biodefense will require grids of computational centers, libraries of digital objects, including software programs and literature, multidisciplinary, well-curated federated collections of scientific data, thousands of online instruments and distributed sensor arrays, convenient software toolkits for resource discovery, modeling, and interactive visualization, and the ability to collaborate with physically distributed teams of people using all of these capabilities, in real-time or quasi-real-time. These are specifically what the Atkins Report characterizes as the vision for CI. Finally, as noted by that report, this "vision requires enduring institutions with highly competent professionals to create and procure robust software, leading-edge hardware, specialized instruments, knowledge management facilities, and appropriate training."

1.2 Pathosystems Biology

Infectious diseases are caused by the interaction of hosts, pathogens, and environmental factors. It is not possible to speak about disease outcomes meaningfully without specifying these factors; thus, a pathogen is not equivalent to a disease and most pathogens are not capable of infecting most organisms (i.e., most organisms are non-hosts of a given pathogen). Therefore, it is common for example in plant pathology to speak of a "pathosystem" when referring to the interaction of hosts, pathogens, and their environments. Some argue that this "disease triangle" (Figure 1) does not apply to animal systems because the environment within the animal is somewhat constant. I would say that even if this is believed to be the case, the epidemiological level clearly involves environmental factors even for animal systems. Systems biology is a relatively new term that can be seen as an extension and modernization of cybernetics [2] Many definitions exist for systems biology, but in my opinion it is characterized by an approach that fully integrates modeling, simulation, theory and wet chemistry experimentation in a unified, multidirectional feedback loop (i.e., theory effects modeling, modeling affects how you design wet chemistry experiments, and so on, in all possible combinations). Taking together the disease triangle as a comparative biological focus area and using a systems biology approach yields the term "pathosystems biology". The comparative aspect is crucial to increase our understanding of pathosystems because evolution re-uses successful components for other needs of the organism (wings may become flippers, for example). At the level of the ongoing molecular arms race that hosts and pathogens engage in, this is well documented [3]. Comparative approaches also may provide crucial benefits because some systems are more tractable to experimentation in the laboratory than others and some of the successful components (of host response or pathogen attack) may be more easily revealed in some systems when compared to others.

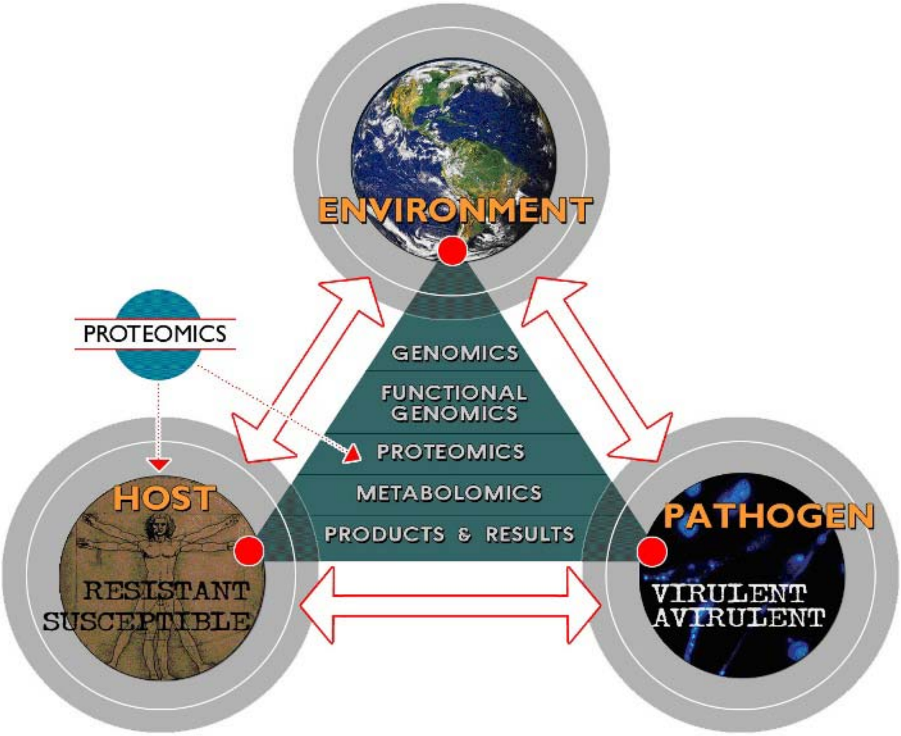


Fig. 1. Host, pathogens, and the environment interact at diverse levels in what is known as the "disease triangle". In the center is a triangle illustrating the use of molecular signatures of DNA, mRNA, proteins and metabolites as but some of the types of data that can be provided through generation, analysis and management of these data, along with the human resources to use the information in pathosystems biology

2 Some Components of Cyberinfrastructure to Achieve Synthesis in Pathosystems Biology

Part of the overall plan for infectious disease monitoring will of necessity reside in data management and analysis capabilities. Coincidentally, as the explosion of types and volume of data occurs, there is an ongoing change in software architectures that support data integration and interoperation. Briefly, in the 1990s, client-server applications changed information systems. Prior to the 90s, mainframes were the norm – these were replaced by client-server architectures with the rise of the PCs. On the software side, vendors released client-server applications, yielding enterprise applications. From the user's perspective, these changes brought end-users into the dialog for the first time. So, IT departments came out to affect all departments in an organization; this is true for scientific organizations as well. Now, there is an evolution from client-server to web-services (see below for characteristics). Web services are enabled because of agreement on standards across a very broad range of hardware and software organizations, (for example, W3C and webservices.org). From the perspective of the mission that must be accomplished to support pathosystems biology, this technological advance is enabling because there is a huge need for information systems interoperation to support collaboration across organizations and real-time information access and analysis, whether it be for public health or biodefense needs.

The catalyzing force behind web services is the agreement by major software and hardware vendors on standards for communication between computer systems, building off the foundation of the Internet (TCP/IP, http and XML). The Internet removed the communication/information bottleneck for information consumers in the client-server model. Web services promise to relieve the information and communication barriers that limit organizational collaboration (because of barriers caused by proprietary, non-interoperable information systems that were independently developed under client-server models). Despite what we see on TV, trans-federal agency or trans-research institution information system interoperability is largely not possible with current architectures without dramatic investments in integration.

By definition, web services are characterized by being: 1) loosely coupled; 2) self-describing (WSDL[1]); 3) accessed programmatically (SOAP[2]); 4) network distributed; and 5) exchange data using platform, vendor and language-neutral protocols. These characteristics provide: flexibility and ease of reconfiguration (1); the software rather than the user determines how to invoke the service and what results the service will return (2); access via Internet protocols and data formats complying with security measures and policies, such as firewalls, allowing deployment and access across intranets as well as Internet (3); data exchange via vendor, platform and language-neutral protocols, due to broad agreement on standards (4).

There are many resources being funded through diverse federal agencies that could be wrapped to become part of a web-services architecture for pathosystems biology. This could be done by other methods, but non-web-services-based integration efforts have been widely used and are appropriate in some mixture with web services, especially in the initial phases of implementation of novel approaches for life sciences data interoperation. Typical approaches include (Marks 2003):

- Ad hoc custom integration – heavily based on individual skills.
- Data warehouses and data marts – develop high quality products based on snapshots of data (frozen in time) and periodic extraction into a common system.
- Enterprise application integration (EAI) – a replication-based middleware approach, tying key systems together.

The above approaches are powerful but can suffer from well-know problems, even outside the technical scope. These are typically (Marks 2003): 1) the requirement for very significant investments in time and money, reducing funds for other, more stra-

[1] Web Services Description Language.

[2] Simple Object Access Protocol.

tegic activities; 2) poor quality data, caused by the lack of definition of standards in the master resources, thereby causing additional time and money investments in cleaning up the data; 3) limited operational visibility, especially in life sciences since there is little understanding and comprehension by most life scientists of the problem at hand and the cost of enterprise integration for example – this has the very negative effect of spending a lot of time trying to get the integration itself right, rather than focusing on the data analysis (the reason for integration); and 4) lack of flexibility, since the above approaches result in tightly coupled systems with reduced operational flexibility – this is perhaps the most severe problem for life scientists since the technologies and underlying data are evolving very rapidly.

Pathosystems biology requires the utilization of diverse types of data that are acquired through standard processes, frequently in distributed locations. Early responses to natural, accidental, or intentional infectious disease outbreaks will require that this information be easily accessed in real-time or near real-time if we are to respond effectively to outbreaks [4]. In addition, technologies for data production are rapidly evolving, especially with respect to machinery and techniques to collect high-resolution data about molecular constituents of living cells (DNA, mRNA, proteins and metabolites, for example, see Figure 4), which may be used to develop signatures of the presence of pathogens. Technologies (laboratory and IT) are thus evolving much more quickly than institutions. Meanwhile, biological knowledge and expertise is distributed organizationally throughout the country and globe, requiring broad community involvement to meet the challenges of infectious diseases in the 21st century. Finally, excellent legacy systems composed of data and analysis/visualization tools are "out there", requiring information system architectures that leverage "old" and enable rapid deployment of "new". All of this argues for flexible, decentralized, modular information system architectures to suit evolving requirements and rapid response – and this is precisely what web services enable.

Distributed data systems, analysis tools and infectious disease expertise require strong collaboration to be in place if we are to respond to infectious diseases rapidly and effectively. In life sciences, collaboration is becoming the norm rather than the exception, although many biologists are still evolving sociologically to accommodate this situation, especially in academia[3]. The goal of collaboration is to establish, maintain and strengthen connections to achieve common objectives. Many of these connections are people to people connections, and these are likely the most important. Yet we must also increase the people to data content, people to applications, and applications to applications to content to applications connections – and these are the ones that web services can enable. In all cases, though, we must not loose sight of the need to understand the social networks[4] [5]

The Internet alone is insufficient to support the type of organization-to-organization collaboration that is needed for pathosystems biology. This is because

[3] NIH Roadmap can be obtained at http://nihroadmap.nih.gov/; more directly related is a subset of the Roadmap developed by the BECON Symposium on Catalyzing Team Science at http://www.becon.nih.gov/symposium2003.htm

[4] The "Atkins Report" on Revolutionizing Science and Engineering through Cyber-Infrastructure can be accessed at http://www.communitytechnology.org/nsf_ci_report/

there is a lack of standards for integration and automation. In addition, manual web browsing and searching does not scale well when there is a need to know about and access diverse information systems – web services provide registry-based applications that find one another and auto-invoke at run time to create larger applications serving specific needs from components that may be used for other purposes and that may reside in distributed machines. Distributed development of biological data sets and analysis tools has been the hallmark of the development of most bioinformatics[5] and computational biology[6] systems thus far – so another advantage of web services approaches is that they leverage what has already been done without the need to invest large sums of money and time into enterprise integration of such components into brittle systems that cannot easily evolve further.

2.1 Bioinformatics, Computational Biology and Community Standards

Bioinformatics and computational biology have grown over the last twenty or so years and through this growth diverse database systems and analytical tools have been developed and deployed, mostly by single investigators or small groups of investigators working together on specific biological problems. Some community resources, such as GenBank, have become key enablers of research on a global scale. The power of this distributed approach to development is that innovation has blossomed at various levels. The challenge is that there have been relatively few concerted efforts to standardize data formats, thus hindering efforts to integrate disparate data types from diverse data sources. Paradoxically, further synthesis in biology largely depends on the capability to access and jointly analyze disparate data. This is especially true for pathosystems biology, since it must deal with data from many types of organisms (pathogens and their hosts) in diverse environments (from intracellular to ecosystems and social networks).

Yet, there are important efforts to develop and deploy community standards for biological data communication. It is important that these efforts be supported and succeed in developing at least data exchange standards for the sake of interoperability across information systems that matter to microbial forensics, whether in existence, or to be (being) developed. The web services stack builds on and extends the standards of the Internet. At the lowest level, there are network protocols (such as TCP/IP, HTTP, FTP, SMTP). The next level is concerned with the meta language (XML). This is where diverse community-based efforts are providing useful standards. Going from DNA through molecules that permit an assessment of the dynamic response of the organism to perturbations, as well as capabilities for modeling and simulation, we have (not meant to be exhaustive):

[5] "Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data." http://www.bisti.nih.gov/

[6] "The development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems." http://www.bisti.nih.gov/

- DNA – DAS-ML[7], BSML[8], MSA-ML[9]
- RNA – MAGE-ML[10] (mRNA profiling)
- Proteins – PEDRo[11] (protein profiling) and ProML[12] (protein sequences, structures and families)
- Molecular models – SMBL[13]
- Cellular levels, including metabolism and signal transduction – CellML[14]
- Organ level – AnatML[15]
- Spatially and temporally varying field information using finite elements – FieldML[16]

Furthermore, there is a need to handle data at the level of phenotypes displayed by organisms. In the case of humans, this is typically placed within clinical records. Fortunately, there are efforts in place to handle these data using XML standards, such as the Clinical Data Exchange Standards Consortium[17] (CDISC). Finally, to handle data from molecular responses to perturbations, through phenotypes and into geographic space (required for epidemiological monitoring and global molecular epidemiologies), there is ArcXML[18] and OpenGIS[19].

There is a danger that of fragmentation of standards through a diversity of non-interacting groups building competing XMLs to represent essentially the same data. Avoiding this will take some vigilance and incentives from funding agencies and requirements for machine-readable interfaces to major resources that are built with federal funding. At some point in the future there will be sufficient advantage through achievement of interoperation that implementations that do not conform to those standards will not be competitive or generally useful.

The network protocol and XML layers are fairly stable technologically and therefore can be thought of as enabling at this point. Above this layer lie three crucial layers that are still undergoing some evolution. These are the services communication layer (SOAP), the services description layer (WSDL), and services publishing and discovery (UDDI[20]/OGSA[21]). These three layers are still evolving and web services implementors need to understand the risks associated with evolution away from the

[7] Distributed Annotation System Markup Language, http://stein.cshl.org/das/.

[8] Bioinformatic Sequence Markup Language, http://www.bsml.org/.

[9] Multiple Sequence Alignment Markup Language, http://xml.coverpages.org/msaml.html.

[10] Microarray Gene Expression Markup Language.
http://www.mged.org/Workgroups/MAGE/mage.html.

[11] http://psidev.sourceforge.net/.

[12] Protein Markup Language, http://www.bioinfo.de/isb/gcb01/talks/hanisch/main.html.

[13] Systems Biology Markup Language, http://sbml.org/index.psp.

[14] http://www.cellml.org/public/about/what_is_cellml.html.

[15] Anatomical Markup Language http://www.physiome.org.nz/anatml/pages/.

[16] http://www.physiome.org.nz/fieldml/pages/.

[17] http://www.cdisc.org/.

[18] http://support.esri.com/.

[19] http://www.opengis.org/.

[20] Universal Description, Discovery and Integration protocol, http://www.uddi.org/about.html.

[21] Open Grid Services Architecture, http://www.uddi.org/about.html.

currently accepted standard. Finally, the most rapidly evolving layers, comprising of still emerging standards, are the business process execution (BPEL4WS[22], WFML[23], WSFL[24], Biztalk, etc.) and additional standards such as WSXL[25].

Another major need is with respect to ongoing curation of data that requires specific biological knowledge, such as much of the microbial data will require. This is especially important because of the distributed nature of biological knowledge in the field. Although funding is limiting in most cases, there are models for supporting distributed curation among specialists.

Sample Cyberinfrastructure for Pathosystems Biology Projects. One model for distributed curation in pathosystems biology has been prototyped on a limited scale in the Pathogen Portal (PathPort[26]) [6]. PathPort project has developed and deployed the Pathogen Information (PathInfo[27]) resource containing data from about 20 of the 50 pathosystems for which acquisition of highly curated data sets referenced from the literature has been requested. One output of the literature curation effort is the Pathogen Information Markup Language or PIML [7], which can now be used further by a distributed community of experts to enter similar data about other pathosystems into a common, machine-readable format. Figure 2 illustrates PIML architecture; figure 3 shows how distributed data acquisition and dissemination is managed in the context of scientific literature and molecular data sets; this is being further developed and deployed under the recently funded Bioinformatics Resource Centers[28] (BRCs) funded by NIAID to develop the capabilities to support genomic data for NIAID category A, B and C pathogens. The goal of the BRCs is to work on the pathogen side of the genomic data management and interoperation issues. To produce, acquire, integrate, manage, analyze and disseminate proteomics data about pathogens, NIAID has recently awarded contracts to establish the Biodefense Proteomics Research Centers[29]. An integral Administrative Resource for Biodefense Proteomic Centers[30] will be responsible for centralized data management for the network.

A number of efforts are now using PathPort's CI (which includes a Core Laboratory[31] and a Core Computational Facility[32] at the Virginia Bioinformatics Institute but they could be anywhere, based on the web-services paradigm). For example, PathPort + Core Computational Facility + Core Laboratory Facility now provide the Bioinfor-

[22] Business Process Execution Language for Web Services. http://www-106.ibm.com/developerworks/library/ws-bpel/.

[23] Windows Forms Markup Language, http://windowsforms.net/articles/wfml.aspx.

[24] Web Services Flow Language, http://xml.coverpages.org/wsfl.html.

[25] Web Services Experience Language. http://www-106.ibm.com/developerworks/library/ws-wsxl/.

[26] https://www.vbi.vt.edu/article/articleview/316.

[27] http://staff.vbi.vt.edu/pathport/pathinfo/.

[28] http://brc.vbi.vt.edu/.

[29] http://www.niaid.nih.gov/dmid/genomes/prc/default.htm.

[30] http://www.niaid.nih.gov/dmid/genomes/prc/administrative.htm.

[31] https://www.vbi.vt.edu/article/articleview/87.

[32] https://www.vbi.vt.edu/article/articleview/88.

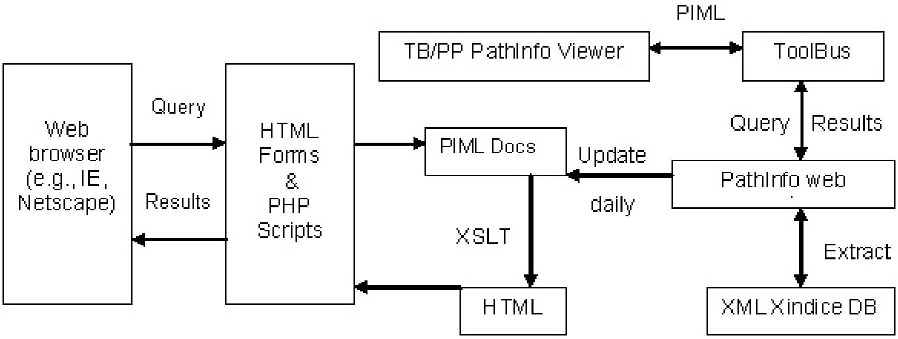


Fig. 2. A web query starts with specifying a particular topic and pathogen(s). Requested pathogen PIML documents are parsed and the results are transformed into HTML by an XSLT script. The PIML documents are updated daily from the Xindice DB via the PathInfo web service. Corresponding viewer is also available via TB/PP system

matics and Genomics Research Core (BGRC[33]) for the Mid-Atlantic Regional Center for Biodefense and Emerging Infectious Diseases (MARCE[34]), funded by NIAID. In this large-multi-institutional, multi-investigator program, part of a national network funded by NIAID this year, the main objective is to develop diagnostics and countermeasures for infectious agents on NIAID category A and B priority lists. The general functional model of the BGRC is illustrated in Figure 4. In the context of MARCE, other CI components, such as the MARCE website[35], supporting external visibility for the project as well as "intranet" functionalities for real-time communication are available as well. These capabilities are meant to support a range of activities, from real-time video conferencing within MARCE and from MARCE to other RCEs as well as interactive tools supporting document preparation, discussion of data, presentations, etc., with the goal of a vibrant, functional CI for pathosystems biology. As different agencies and scientists working on different aspects of infectious diseases use and help evolve the CI, one of the benefits that will come out of the infrastructure, without additional investment, is the possibility of doing joint analyses on data sets that were developed with specific goals in mind but that can be useful to other goals. The success of GenBank in enabling comparative analyses of community sequences because of deposition into a standardized repository is but an example of what can be aspired by the infectious disease CI being developed and deployed.

One of the many reasons for using a web-services, federated approach is the leveraging, with relatively little effort, of key resources being built in the community. It is not possible here to provide an exhaustive review of these, but clearly efforts such as the Microbial Rosetta Stone Database (MRS) project (K.L. Hari, J.A. McNeil, IBIS

[33] http://marce.vbi.vt.edu/cores/bioinformatics_and_genomics_core.

[34] https://www.vbi.vt.edu/article/articleview/426/1/33/.

[35] http://marce.vbi.vt.edu/.

Pharmaceuticals; and J.M. Robertson, FBI; personal communication) are aimed in the right direction. MRS has been motivated by the need to map the landscape of infectious diseases and to assist with microbial forensics needs, specifically. Another interesting resource is Gideon Online[36]. This system has been developed essentially to assist in diagnosing (at the clinical level) infectious agents based on information collected by the clinician and a Bayesian analysis system. It is continually updated and has information on all infectious agents of humans and related mammals, and also a recently released bioterrorism module. It has also been used for training and teaching of physicians. Models could be developed to support further documentation and referencing of the system to the scientific literature and online, real-time update by distributed experts that start to then use the system for data entry to support monitoring.

The PathPort project itself has been federating through web services diverse data sources and analysis tools to support the needs of (currently and primarily) discovery scientists working on developing a more comprehensive knowledge of the mechanisms that infectious agents and their hosts deploy in their interactions (an "arms race"). The client-side interconnect for the federated services, ToolBus (Figure 5), allows users of the system to access and analyze (mostly molecular currently) data of diverse types from diverse sources. The overall architecture of the PathPort system is shown in Figure 6 and the architecture of the client-side interconnect, ToolBus, is shown in Figure 7.

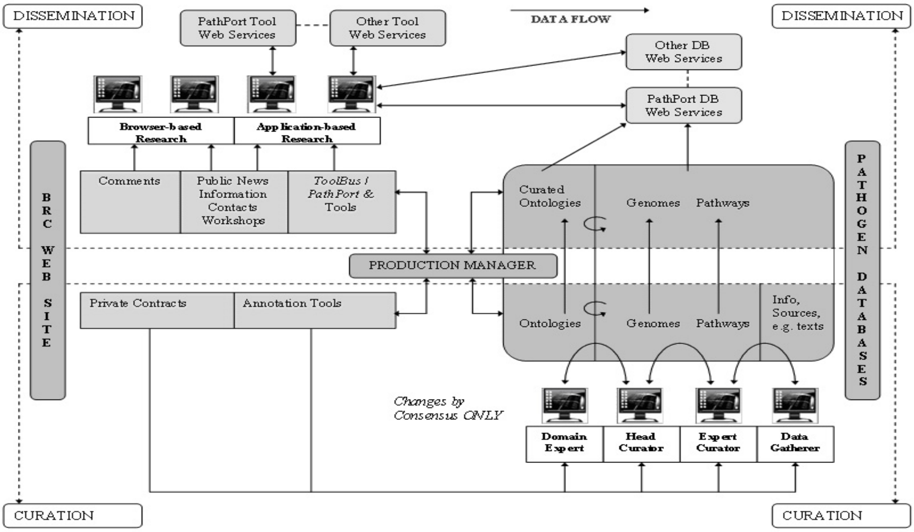


Fig. 3. A model for distributed curation involving subject matter experts throughout the community and showing how many of the (molecular) data types are dealt with, along with the CI needed to ensure that the data are acquired and disseminated appropriately

[36] http://www.gideononline.com/.

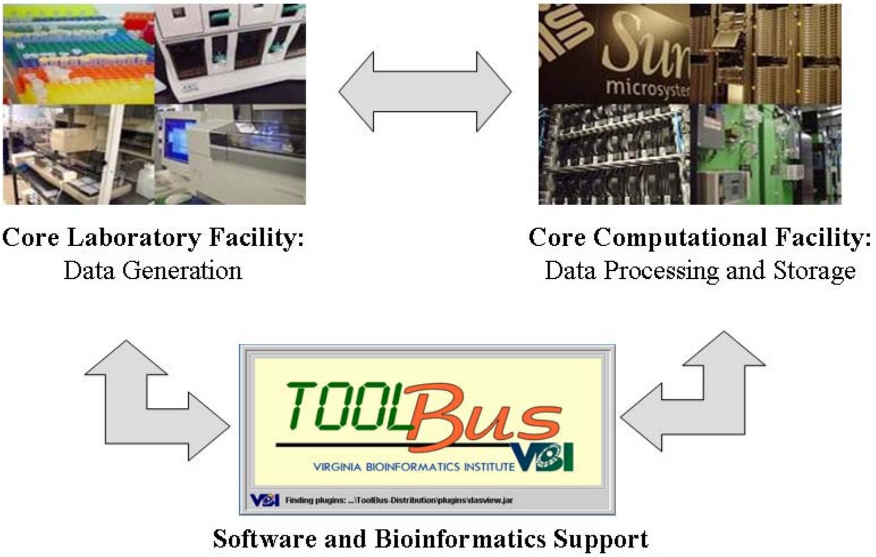

Core Laboratory Facility:
Data Generation
Core Computational Facility:
Data Processing and Storage
TOOLBus
VIRGINIA BIOINFORMATICS INSTITUTE
VBI
Finding plugins: ...\ToolBus-Distribution\plugins\dasview.jar
Software and Bioinformatics Support

Fig. 4. CI supporting data generation, acquisition, analysis, storage for MARCE. Note that although the physical capacities provided by thee cores happen to reside at the same institution in this case, one can envision a number of such facilities distributed, working under standard operating and quality assurance procedures, and supported by the interoperability middleware such as provided by PathPort's implementation of a web services strategy and ToolBus for the client-side interconnect. Also note that analysts providing training support and where appropriate conducting analyses collaboratively with a distributed set of partners is integral to this model but not shown in the figure

PathPort project is following some of the typical phases explored in web services adoption. These are: 1) integration/interoperation, 2) collaboration and 3) innovation [8]. PathPort project is in the first phase, primarily, and exploring the second phase. The first phase typically involves building wrappers around legacy systems and applications. During this first phase, the project has embraced fast cycles of development and deployment[37] with opportunity for community involved in the rapid cycles of learning. The goal has been to deploy early and often to allow users to react and participate effectively with the software development team. This has resulted in sharing of information across collaborators and mutual learning. During this phase the CI team and its collaborators sometime encounter limits based on immature standards and unprepared IT architectures. With the coming of the second phase, collaboration, we eventually expect a reduction in the levels of human intervention required to support collaboration. Finally, as being experienced in the PathPort project, "external" partners start to increase in their sharing and collaboration thus further driving the

[37] See http://staff.vbi.vt.edu/pathport/scrum/ for information about the SCRUM/SPRINT process being employed to agilize software development.

development/implementation/evolution chain. In the innovation phase, we hope to use the lessons learned from the previous phases to drive entirely new processes and models. New, distributed web-services models tend to be disruptive and thereby enable change. We hope, in good CI form, that there will be a redefinition of how research is conducted across organizational boundaries, something I believe the MARCE project, the NIAID RCE Network, the BRC Network and the Biodefense Proteomic Research Centers[38] can help prototype both within their own networks as well as across networks. This redefinition is sorely needed and enabled by exposing specific operational information system elements for dynamic linking to processes of partners/collaborators. The goal is to have organizations operating as a truly inter-

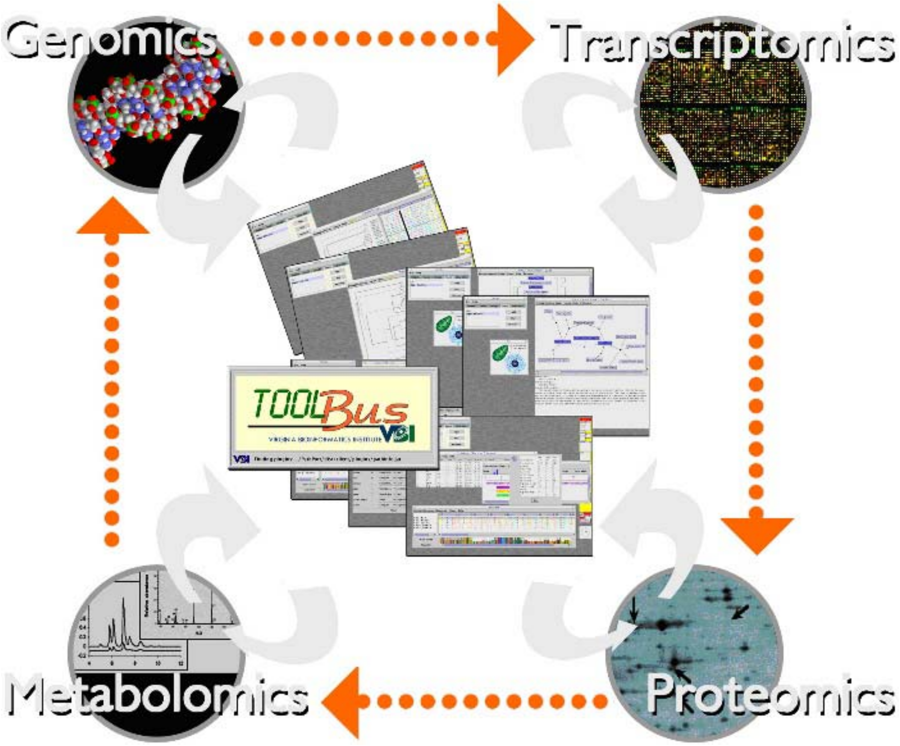


Fig. 5. "-Omics" data provide the opportunity to develop the "parts lists" for pathogens and their hosts (genomics data), along with the contextual "state" data that describe the dynamic molecular responses of living organisms (pathogens and hosts) as they respond to each other in a given environmental condition (transcriptional profiles or transcriptomics data, protein profiles or proteomics data and metabolite profiles or metabolomics data). These data sets not only will allow molecular signatures to be developed, they will also help establish a mechanistic understanding of infectious agents attacking their hosts, thereby enabling development of new countermeasures, such as vaccines and therapeutics

[38] http://www.niaid.nih.gov/dmid/genomes/prc/default.htm.

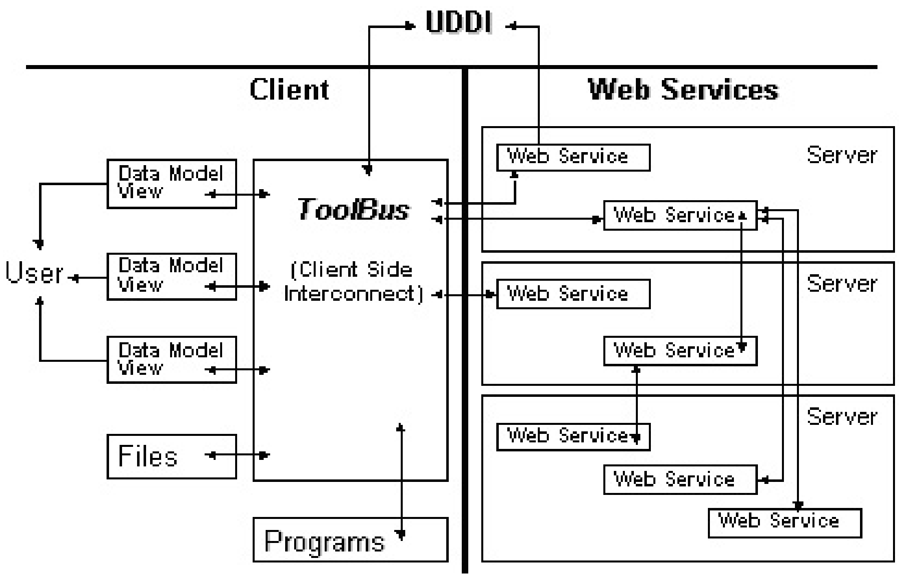


Fig. 6. A simplified view of the architecture employed by PathPort project to allow interoperability of diverse analysis tools and data sources of relevance to infectious diseases. Web services can be either analysis tools, such as BLAST, or a data source, such as GenBank. They can reside anywhere. Local files (available to the local user only), whether programs or data, can be used without making them available to the entire federation if desired

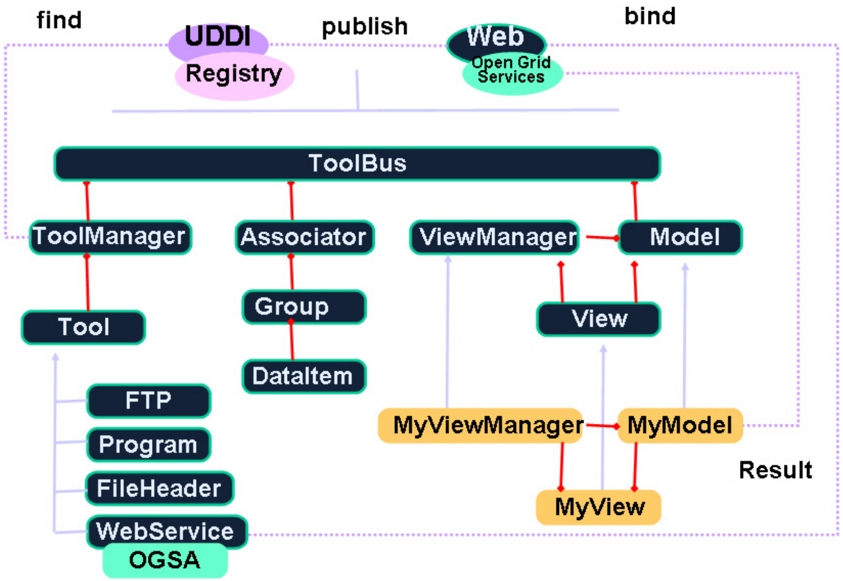
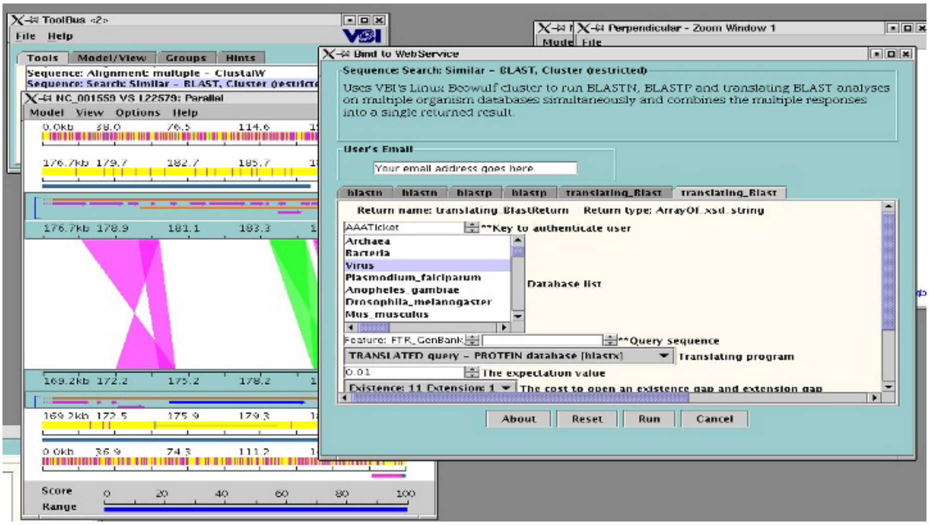


Fig. 7. Architecture of the client-side interconnect, ToolBus, that allows for access of web-services relevant to PathPort project. Note that new data models can be added easily without breaking the system or requiring major re-engineering



Sequence: Alignment: multiple – ClustalW

Sequence: Search: Similar – BLAST, Cluster (restricted)

Uses VBI's Linux Beowulf cluster to run BLASTN, BLASTP and translating BLAST analyses on multiple organism databases simultaneously and combines the multiple responses into a single returned result.

User's Email: Your email address goes here.

Return name: translating_BlastReturn Return type: ArrayOf_xsd_string

AAATicket **Key to authenticate user

Database list: Archaea, Bacteria, Virus, Plasmodium_falciparum, Anopheles_gambiae, Drosophila_melanogaster, Mus_musculus

Feature: FTR_GenBank **Query sequence

TRANSLATED query – PROTEIN database [blastx] Translating program

0.01 The expectation value

Existence: 11 Extension: 1 The cost to open an existence gap and extension gap

Fig. 8. A simple example of interoperability across previously incompatible systems (DAS Viewer, BLAST, MUMer) built by the community as well as of allowing communication between the visualizer for a given analysis result (in this case comparative genomics between *Vaccinia* and *Variola*) through a drag-and-drop approach into another analysis server (in this case BLAST). A web-services architecture separates the building of visualizers from databases such that new visualizations can be achieved easily. The communication across incompatible systems enables a much faster and more efficient workflow for the human knowledge worker/operator/analyst

connected cyber-ecosystem. The newness of these research networks provides a unique opportunity to develop this from the beginning, if this an objective that is adequately and integrally planned.

One question that frequently arises with infectious disease research and data in our post-9/11 world is security. There are many different levels of need to security. From an IT perspective, web-services can provide security via models being developed and implemented, such as the WS-Security[39] or OASIS WS Security TC[40]. Importantly, again, is to leverage community standards for implementation. Although some of the needs may be national security related, it is important to note that most life sciences companies, such as Pharmaceuticals and biotechs, have very stringent security needs because of Intellectual Property requirements. (This is to say that there are meaningful solutions that can leverage web services and be enabling all the same, based on specific requirements.) The Intel community is already implementing prototypical projects in this direction, noting[41]: "In a network-centric envi-

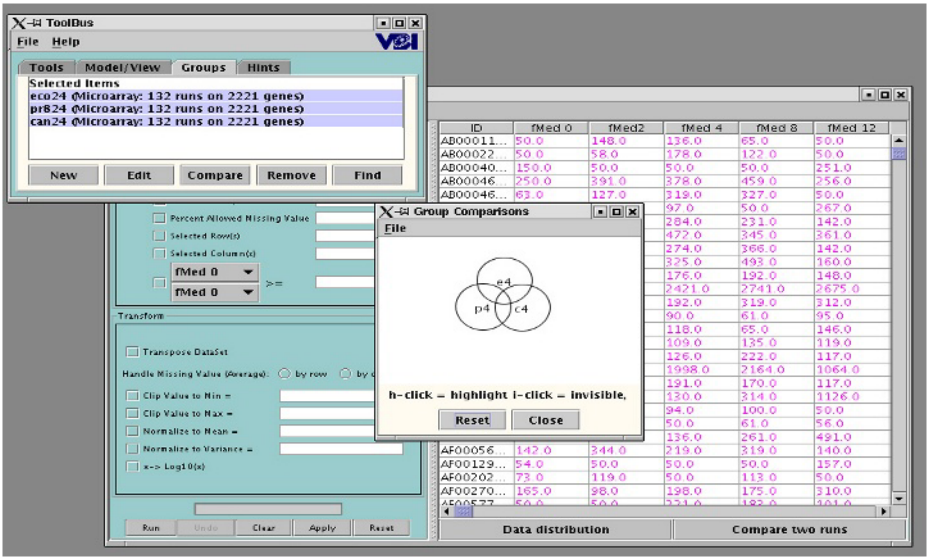
[39] http://www-106.ibm.com/developerworks/webservices/library/ws-secure/.

[40] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss.

[41] See http://fcw.com/fcw/articles/2003/0317/web-nces-03-18-03.asp, for example.

ronment, data would be made available as quickly as possible to those who need it across the organization or on the battlefield. Many DoD systems in the field today use a client/server architecture." and "…would create an infrastructure that will enable users to quickly take advantage of DoD and intelligence community networks, eliminating the system-by-system approach"…"The system will enable users to customize the way they search and actually view information in real-time and display previously unavailable combinations of intelligence, surveillance and reconnaissance data. Access based on individual users' security clearances will be built into the design." Thus there is nothing specific about web-services that will not support security as needed.

In the three years of development experience provided by the PathPort project, interoperability across previously incompatible systems, using web-services, has already been implemented and used by scientists (Figures 8 and 9). In the future, as we move toward the innovation phases of development, ideas and concepts that support large-scale simulations of real-world events pertaining to infectious disease outbreaks (Figure 10) will be possible.



Selected Items
eco24 (Microarray: 132 runs on 2221 genes)
pr824 (Microarray: 132 runs on 2221 genes)
can24 (Microarray: 132 runs on 2221 genes)

□ Percent Allowed Missing Value
□ Selected Row(s)
□ Selected Column(s)
□ fMed 0 ▾ fMed 0 ▾ >=

Transform
□ Transpose DataSet
Handle Missing Value (Average): ○ by row ○ by c
□ Clip Value to Min =
□ Clip Value to Max =
□ Normalize to Mean =
□ Normalize to Variance =
□ x-> Log10(x)

h-click = highlight i-click = invisible,

ID	fMed 0	fMed2	fMed 4	fMed 8	fMed 12
AB00011...	50.0	148.0	136.0	65.0	50.0
AB00022...	50.0	58.0	178.0	122.0	50.0
AB00040...	150.0	50.0	50.0	50.0	251.0
AB00046...	250.0	391.0	378.0	459.0	256.0
AB00046...	63.0	127.0	319.0	327.0	50.0
			97.0	50.0	267.0
			284.0	231.0	142.0
			472.0	345.0	361.0
			274.0	366.0	142.0
			325.0	493.0	160.0
			176.0	192.0	148.0
			2421.0	2741.0	2675.0
			192.0	319.0	312.0
			90.0	61.0	95.0
			118.0	65.0	146.0
			109.0	135.0	119.0
			126.0	222.0	117.0
			1998.0	2164.0	1064.0
			191.0	170.0	117.0
			130.0	314.0	1126.0
			94.0	100.0	50.0
			50.0	61.0	56.0
			136.0	261.0	491.0
AF00056...	142.0	344.0	219.0	319.0	140.0
AF00129...	54.0	50.0	50.0	50.0	157.0
AF00202...	73.0	119.0	50.0	113.0	50.0
AF00270...	165.0	98.0	198.0	175.0	310.0
AF00577	50.0	50.0	[illegible]	[illegible]	[illegible]

Fig. 9. ToolBus use case ToolBus showing the group suggestor function working on a set of transcriptional profiles. It is important to note that ToolBus and the "group suggestor" capability of the system do not "know" about the type of data being analyzed – although in this example all the data are of one type (mRNA expression levels), any type of data that is available in such an interoperable framework could be analyzed with the group suggestor capabilities (for example, transcriptional profiles and GIS coordinates of people or plants from which the profiles were obtained)

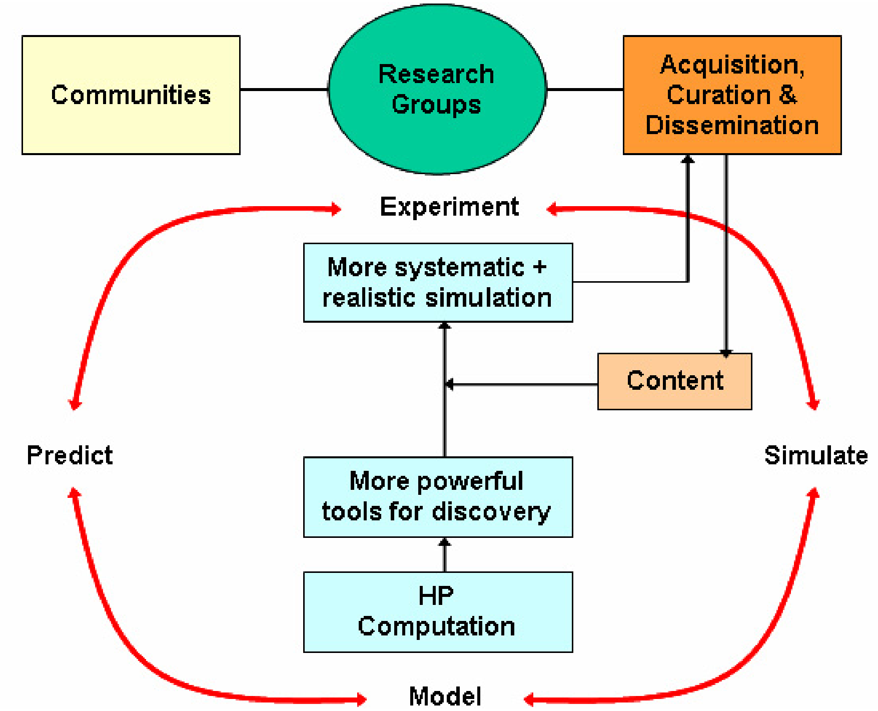

Communities
Research Groups
Acquisition, Curation & Dissemination
Experiment
More systematic + realistic simulation
Content
Predict
Simulate
More powerful tools for discovery
HP Computation
Model

Fig. 10. An illustration of the conceptual, integrative framework for a long-term CI for patho-systems biology

Acknowledgements

I am grateful for funding by the US Department of Defense (contracts DAAD 13-02-C-0018 and W911SR-04-0045), the National Institute of Allergy and Infectious Diseases (contract HHSN266200400035C, HHSN266200400061C and Cooperative Agreement 1 U54 AI057168-01) in support of ToolBus, and PATRIC projects respectively. Special thanks to Dana Eckart, Yongqun He, Ron Kenyon, Dave Sebring and my Cyberinfrastructure Group at VBI for making this work possible. I am also grateful to Darleen Baker for assistance and editorial improvements to this paper. Finally, this work is dedicated to Minnis Ridenour, to whom VBI owes its existence.

References

1. Atkins, D., Droegemeier, K., Feldman, S., Garcia-Molina, H., Klein, M., Messerschmitt, D., Messina, P., Ostriker, J., Wright, M.: Revolutionaizing Science and Engineering Through Cyberinfrastructure: Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure. (2003). http://www.nsf.gov/publications/pub_summ.jsp?ods_key=cise051203
2. Wiener, N.: Cybernetics: or Control and Communication in the Animal and the Machine. 2nd edn.MIT Press (1948) 212

3. Studholme, D. J., Downie, J. A., Preston, G. M.: Protein Domains and Architectural Innovation in Plant-Associated Proteobacteria. BMC Genomics. 6 (2005) 17
4. Eubank, S., Guclu, H., Kumar, V. S., Marathe, M. V., Srinivasan, A., Toroczkai, Z., Wang, N.: Modelling Disease Outbreaks in Realistic Urban Social Networks. Nature. 429 (2004) 180-184. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15141212
5. Ibid
6. Eckart, J.D., Sobral, B.W.: A Life Scientist's Gateway to Distributed Data Management and Computing: The PathPort/ToolBus Framework. Omics. 7 (2003) 79-88. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12831562
7. He, Y., Vines, R. R., Wattam, A. R., Abramochkin, G. V., Dickerman, A. W., Eckart, J. D., Sobral, B. W.: PIML: the Pathogen Information Markup Language. Bioinformatics. (2004) http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15297293
8. Marks, E., Werrell, M. : Executive's Guide to Web Services. John Wiley & Sons, Inc., Hoboken, N.J. (2003)

Analysis of Genomic Tiling Microarrays for Transcript Mapping and the Identification of Transcription Factor Binding Sites

Joel Rozowsky, Paul Bertone, Thomas Royce, Sherman Weissman,
Michael Snyder, and Mark Gerstein

Department of Molecular Biophysics & Biochemistry, Yale University,
New Haven CT, USA

The recently developed technology of genomic tiling microarrays, which can be used for genome annotation, has required the development of new methodologies [Royce et.al] for their design and analysis. Genomic tiling arrays use PCR amplicons or short oligonucleotide probes to *tile* the non-repetitive DNA sequence of a genome in an unbiased fashion for the purposes of detecting novel genomic features. Specifically, they can be used for the identification of novel transcripts, distinguishing between different splice isoforms and for finding transcription factor binding sites using Chromatin-Immunoprecipitation on chip experiments (ChIP-chip).

High density PCR product arrays which has allowed entire human chromosomes to be tiled [Rinn et.al (2003)], and super high density oligonucleotide arrays which has enabled the tiling of the entire human genome or a substantial portion thereof [Kapranov et.al, Bertone et.al, Cheng et.al]. Arrays of this type have also been used for transcript mapping in other genomes: Arabidopsis thaliana [Yamada et.al (2003), Stolc et.al (2005)] and Drosophila melanogaster [Stolc et.al (2004)]. Unlike conventional gene-targeted microarrays, where analysis methods have been well developed, only a small fraction of probes on a tiling array (especially true for mammalian genomes) show-hybridizing signal necessitating a more sophisticated analysis. An additional important factor is that experiments of this type generate vast quantities of data (multiple gigabytes) unlike conventional gene-based arrays.

Tiling arrays are also useful for the identification of transcription factor binding sites [Martone et.al (2003), Cawley et.al (2004) & Euskirchen et.al (2004)] using the so-called ChIP-chip experimental technique. In order to identify binding sites it is essential that all of the non-repetitive genomic DNA sequence is represented in an unbiased manner on the microarray. The analysis methodology for these experiments is different for amplicon and oligonucleotide tiling arrays, as stretches of oligonucleotide probes showing enriched signal compared to the control are required for a statistical significant hits, unlike PCR amplicon arrays where hits are determined on an amplicon-by-amplicon basis.

Another important issue is the design of genomic tiling arrays. Oligonucleotide probes show biases in fluorescent signal intensities due to the variation in binding affinities of different probe sequences. In addition, short sequence motifs can generate disproportionately high signal. Sequences biases and artifacts of these types can be taken into account in the design of arrays of this type.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 28 – 29, 2005.

References

1. Royce et.al, Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping, Trends in Genetics (in press).
2. Rinn et.al, The transcriptional activity of human Chromosome 22, Genes Dev. 2003 Feb 15;17(4):529-40.
3. Bertone et.al, Global identification of human transcribed sequences with genome tiling arrays, Science. 2004 Dec 24;306(5705):2242-6. Epub 2004 Nov 11.
4. Cheng et.al, Transcriptional Maps of 10 Human Chromosomes at 5-Nucleotide Resolution, Science. 2005 Mar 29.
5. Yamada et.al, Empirical analysis of transcriptional activity in the Arabidopsis genome, Science. 2003 Oct 31;302(5646):842-6.
6. Stolc et.al, Identification of transcribed sequences in Arabidopsis thaliana by using high-resolution genome tiling arrays, Proc Natl Acad Sci U S A. 2005 Mar 22;102(12):4453-8. Epub 2005 Mar 8.
7. Stolc et.al, A gene expression map for the euchromatic genome of Drosophila melanogaster, Science. 2004 Oct 22;306(5696):655-60.
8. Martone et.al, Distribution of NF-kappaB-binding sites across human chromosome 22, Proc Natl Acad Sci U S A. 2003 Oct 14;100(21):12247-52. Epub 2003 Oct 3.
9. Cawley et.al, Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNA, Cell. 2004 Feb 20;116(4):499-509.
10. Euskirchen et.al, CREB binds to multiple loci on human chromosome 22, Mol Cell Biol. 2004 May;24(9):3804-14.

Perturbing Thermodynamically Unfeasible Metabolic Networks

R. Nigam[1] and S. Liang[2]

[1] W.W. Hansen Experimental Physics Laboratory,
Stanford University, Stanford, CA 94305-4085, USA
rakesh@quake.stanford.edu

[2] Department of Biostatistics and Applied Mathematics,
The University of Texas M.D Anderson Cancer Center,
1515 Holcombe Blvd, Houston, TX 77030, USA
shoudan@mdanderson.org

Abstract. Reactions within the cell should satisfy the law of mass conservation and the second law of thermodynamics. Networks of reactions violating any of these laws are unphysical and cannot occur in nature. In this paper we describe a technique that perturbs an unfeasible network to produce a metabolic network that satisfies the two fundamental laws. This algorithm has been applied to study the metabolic pathways of *E. coli.*

1 Introduction

With the availability of many completely sequenced genomes and much gene expression data the study of metabolic pathways [1] is entering a phase where constraint-based approaches will be useful in quantitative analysis [7]. These approaches are valuable since they are based on fundamental physical laws and do not make use of any unknown parameters. For a given system of reactions in steady state, mass balance of the reactions restricts the space of possible fluxes or rates to the null space of the stoichiometric matrix [13]. This constraint is used in *flux balance analysis (FBA)*. Formulating the problem in terms of fluxes we do not need detailed knowledge of the kinetic parameters inside the cell. In the absence of detailed knowledge about the kinetic parameters and enzyme concentrations, the FBA assumes that the metabolic flux vector in a biological network optimizes the production of growth or biomass, subject to the mass balance constraint.

The network structure of metabolic pathways imposes an additional thermodynamic constraint on the rates of reactions. Many authors [6] have studied reaction networks from a very theoretical standpoint, but only recently, have applied constraints to metabolic networks, to further constrain the space of feasible fluxes. The thermodynamic constraint is a consequence of the second law of thermodynamics, according to which, the direction of a chemical reaction is from a higher chemical potential to a lower chemical potential [9], like the

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 30–41, 2005.

flow of current from a higher voltage to a lower voltage in an electrical circuit. This constraint is also called *energy balance analysis (EBA)* in the literature [2]. The stoichiometric matrix contains all the information regarding the flux balance and energy balance constraints. Moreover, the chemical potential depends on the concentration of the metabolites, but in the EBA model the sign of the chemical potential difference is used to formulate the thermodynamic constraint.

In [2] FBA and EBA are formulated as non-linear optimization problems for the flux and change in chemical potential, which can lead to errors if the method does not converge. Recently, [3] propose a method of exponential complexity based on matroid theory to solve this problem. In [5] a sign test was proposed which is a necessary condition for testing thermodynamic unfeasibility and which, when combined with *linear programming (LP)*, is both necessary and sufficient, leading to a polynomial algorithm to detect and compute thermodynamically feasible fluxes and chemical potential changes.

In this paper we propose a novel LP based polynomial time algorithm that constructs fluxes and changes in chemical potential which satisfy both the FBA and EBA constraints by perturbing the network, by removing reactions that are thermodynamically unfeasible. We have applied our algorithm to the complete metabolic network of *Escherichia coli* [10], but in this paper we illustrate it on a sub-network of *E. coli* that deals with central metabolism [4]. The reaction network considered in this paper contains 19 metabolites linked by 23 reactions. The algorithm is used to maximize the production of biomass flux, and the resulting flux vector is thermodynamically unfeasible. By appropriately perturbing the network by removing an internal cycle the metabolic pathway becomes thermodynamically feasible.

2 Metabolic Pathways

Biological systems differ from purely chemical systems in the sheer complexity of the reaction schemes and number of chemicals involved. However, their complex kinetic behavior is not merely a result of complex reaction schemes, but is permissible thermodynamically if the system is far from equilibrium. One of the functions of metabolism is to maintain biological systems far from equilibrium, allowing complex behavior to take place. Metabolic processes are controlled by catalysts known as enzymes, which are proteins whose three-dimensional structure is essential to the precision of their operation.

The methods presented in this paper will help biologists to compute the rates of these reactions, without going into detailed kinetics. For example, consider a small subset of the *E. coli* metabolic network, with one unit of oxygen and two units of glucose as inputs. These inputs are boundary or exchange fluxes. From this information we want to know the maximal ATP production rate. To answer this question, we need information about the various reactions associated with the metabolites encoded in the stoichiometric matrix. In the FBA method, the constraints imposed by the stoichiometry of a chemical network at steady state are similar to Kirchhoff's first law for the balance of currents in electric circuits.

By applying just the FBA to find the maximal ATP production, we could produce solutions that violate the second law of thermodynamics. To disallow such unphysical solutions we impose the additional EBA constraint. This constraint is analogous to Kirchhoff's second law in electrical circuits, where the voltage drop around a closed loop in an electrical circuit must be zero, and currents flow from higher to lower voltages. Similarly, in metabolic networks the free energy change around a reaction loop must be zero. In a complicated network, such closed loops can be identified by computing the null space of the stoichiometric matrix corresponding to the internal reaction fluxes. These internal fluxes can for example be associated with the ATPase reaction. These closed loops have non-increasing entropy that violate the second law of thermodynamics, hence they should be detected and removed from the FBA solution. This approach perturbs the original network so as to produce a feasible network that satisfies both FBA and EBA. The EBA approach requires no prior information about the concentrations of the metabolites or any detailed knowledge of the kinetic parameters.

Reference [8] showed that the FBA solution decomposes into weightings of three types of pathways. Type I pathways, for example, deal with the cycling of ATP, which then drives other cellular processes. Type II pathways are those that have exchange fluxes corresponding to metabolites like ATP, NADH, with the rest of the pathway being an internal cycle. These pathways represent futile cycles. Type III pathways have no exchange fluxes, and these represent internal cycles corresponding to internal fluxes. Type III loops in the flux direction violate the second law of thermodynamics [8]. For example, the pyruvate-kinase reaction flux could be part of a type III loop. These loops are detected and removed by our algorithm.

In this paper we consider only steady state solutions, an approximation that holds when metabolites do not accumulate.

3 Flux and Energy Balance Analysis

A metabolic network [12] typically consists of several hundred reactions that are catalyzed by enzymes. A *reaction rate*, or *flux*, is assigned to each metabolic reaction. In flux balance analysis, the law of mass balance is applied to each metabolite, which in steady state implies that the incoming fluxes should balance the outgoing fluxes. The flux balance analysis has been formulated as a linear program by many authors [13], in which a linear objective function Z (Eq.(1)) is to be maximized or minimized. Usually written as a linear combination of the fluxes, the objective function could, for example, be growth rate, ATP production, or glucose intake. The optimization of the objective function is subject to the mass balance constraints in Eq.(2):

$$Z = \boldsymbol{d}^T \boldsymbol{f}, \tag{1}$$

$$S\boldsymbol{f} = \boldsymbol{0}, \tag{2}$$

and,

$$\boldsymbol{l} \leq \boldsymbol{f} \leq \boldsymbol{u} \tag{3}$$

where, $\boldsymbol{f} \in \mathcal{R}^n$ is the vector of n fluxes, $S \in \mathcal{R}^{m \times n}$ is a stoichiometric matrix and m is the number of reactants or metabolites in the network. All vectors by default are column vectors. Also, $\boldsymbol{d}$, $\boldsymbol{l}$ and $\boldsymbol{u}$ are vectors $\in \mathcal{R}^n$ of objective function coefficients, lower and upper bound constraints on the fluxes respectively, and $\mathbf{0}$ is a zero vector of size m. Equation (3) imposes upper and lower bounds on the flux vector, taken componentwise. This constraint is measured experimentally. The lower bound constraint in most cases is either zero or negative infinity. In this paper we will assume these two lower bounds for the fluxes. The upper bound constraint for most of the fluxes is infinity, but for some boundary fluxes has a finite value that is experimentally determined. Also, the vector of objective function coefficients has to be determined experimentally. Usually the objective function depends only on the boundary or exchange fluxes.

In the above formulation the number of fluxes n exceeds the number of metabolites m in the cell, so linear programming is a convenient way to solve the system of underdetermined equations. Due to degeneracy an infinite number of solutions are possible for the flux vector, that satisfy all the constraints and optimize the objective function.

According to the second law, fluxes must flow from reactants of higher chemical potential to ones of lower chemical potential, since the entropy of the reaction is always non-decreasing [9]. The FBA analysis allows an infinite number of fluxes, many of these flux vectors violate the second law and hence are unfeasible. From a network topology point of view, the presence of cycles in the flux direction violates the law of production of entropy. Applying Kirchoff's loop law, eliminates these entropy violating cycles.

From S we remove the columns corresponding to boundary fluxes and keep only the columns of non-redundant internal fluxes, which we define as those between metabolites. The resulting matrix $G \in \mathcal{R}^{m \times n_i}$, where n_i is the number of internal fluxes in the network. Using the reduced row echelon form [11] we can find the null space matrix N of G. The matrix $N \in \mathcal{R}^{n_i \times n_l}$ consists of n_l basis vectors of $\mathcal{N}(G)$, the null space of G. The dimension of $\mathcal{N}(G)$, denoted by $\mathcal{D}(\mathcal{N}(G))$ gives the number of independent loops n_l in the network (Strang, 2003). By taking linear combinations of these basis loops we can generate bigger and compound cycles (see Strang, 2003, page 363). This basis is unique since the reduced echelon form of G is unique.

Associated with each internal flux f_i is a chemical potential difference $\Delta\mu_i$. These potential differences satisfy a law similar to Kirchoff's loop law in electrical circuits, namely [2]:

$$K\boldsymbol{\Delta\mu} = \mathbf{0} \tag{4}$$

where, $K = N^T \in \mathcal{R}^{n_l \times n_i}$ is a matrix whose rows are the basis vectors of the null space of G, $\boldsymbol{\Delta\mu} \in \mathcal{R}^{n_i}$ is a column vector of chemical potential differences for the internal fluxes in the cell, and $\mathbf{0}$ is a zero vector of size n_l.

The second law ensures that entropy increases in each internal reaction i and hence the direction of internal flux f_i is from metabolites of higher chemical potential to ones of lower chemical potential:

$$\begin{cases} f_i \Delta\mu_i < 0 & \text{for } f_i \neq 0 \text{ and } \Delta\mu_i \neq 0, \\ f_i = 0, \Delta\mu_i = 0 \text{ otherwise.} \end{cases} \tag{5}$$

This constraint applies to all the internal fluxes. According to equation (5), if either of f_i and $\Delta\mu_i$ are zero, then both of them must be zero.

Equations (4) and (5) are thermodynamic feasibility constraints that are applied in addition to the flux balance constraints. Equation (5) is a nonlinear constraint which, when incorporated into the FBA, makes it a non-linear programming problem. In this paper we propose a simpler algorithm to solve the problem as a linear programming problem.

In addition to the above constraints we impose upper and lower bound constraints on $\boldsymbol{\Delta\mu}$

$$\boldsymbol{\beta} \leq \boldsymbol{\Delta\mu} \leq \boldsymbol{\alpha} \tag{6}$$

where, $\boldsymbol{\beta}$ and $\boldsymbol{\alpha} \in \mathcal{R}^{n_i}$ represent the lower and upper bounds on the change in chemical potential $\boldsymbol{\Delta\mu}$, and the inequality is componentwise. The absolute values of the components in $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ mean nothing, since equations (4), (5) and (6) can be scaled by a positive constant without changing the linear programming solution.

In the next section we describe the condition for checking the presence of cyclic fluxes, and we introduce some notation here. We will use upper-case indices to denote sets. For example, let F be the set of all fluxes in the network, R be the set of unrestricted fluxes, $F^{\geq 0}$ be the set of non-negative fluxes, $F^{<0}, F^{=0}$ and $F^{>0}$ be the set of negative, zero and positive fluxes, respectively. Denote the ith flux component $f_i \in F$, r_i is an unrestricted flux, $f_i^{\geq 0}$ a non-negative flux etc. The matrix N can be written in terms of its column vectors as $N = [N_{*1}, \ldots, N_{*k}, \ldots, N_{*i}, \ldots, N_{*n_l}]$, where N_{*k} is the kth column vector of N. Also $N_{*k} = [n_{1k}, n_{2k}, \ldots, n_{ik}, \ldots, n_{n_ik}]^T$, where n_{ik} is the (i,k) th entry of the matrix N.

4 No-Cycle Feasibility Constraint

In this section we introduce a simple test to detect the presence of loops in a metabolic network that violate the second law of thermodynamics. To do so we take advantage of the directionality of the flow of fluxes in the cycle. The number of rows n_l of the K matrix gives the number of loops or cycles in the network [11]. These loops are the basis cycles. If in any row j of the K matrix K_{j*} all the entries (more than one should be non-zero) are of the same sign, corresponding to the set of positive fluxes $F_j^{>0}$ for the jth cycle, then the flux vector is thermodynamically unfeasible. If some unrestricted flux r_i belonging to the jth cycle is negative, it can be made positive by reversing the sign of the corresponding entries in the ith column of the G and K matrices, namely G_{*i}

and K_{*i} respectively. If after this transformation, any of the rows of the K matrix still has the same sign, then the flux vector is thermodynamically unfeasible, and we detect the presence of a cycle. By satisfying this condition we can eliminate the non-linear constraint in equation (5), transforming the non-linear problem into a linear one.

The no-cycle feasibility condition is equivalent to solving the FBA problem with constraints (4) and (5). To satisfy equation (4), for a single row of the K matrix corresponding to a single cycle, at least one of the $\Delta\mu_i$ should differ in sign from the other components in the $\boldsymbol{\Delta\mu}$ vector, which, when combined with equation (5), prevents the formation of thermodynamically unfeasible loops in the network. This no-cycle feasibility condition can be applied to basis loops, which can be easily observed in the echelon basis.

We state without proof lemma 1 [5], which summarizes the previous paragraphs, and gives a necessary condition for detecting thermodynamically unfeasible cycles.

Lemma 1. (sign transformation and feasibility lemma)
Transforming the unrestricted internal flux $-r_i$, to r_i, makes the elements of K_{*i} negative. If after this transformation, any row of K is of the same sign, then the metabolic network is thermodynamically unfeasible.

Lemma 2. (pivoting lemma)
Subtracting a multiple of a row of K from the corresponding internal fluxes in a cycle does not change the optimum value of the objective function in equation (1).

Since this pivoting step changes neither the objective function, which contains only boundary fluxes, nor the constraints on the linear program in equations (1)-(3), since the multiple is chosen to respect all the constraints, the application of this step will not change the optimal value of an objective function, such as production of biomass, that is composed of throughput fluxes. The examples discussed in [8] are very restrictive and their flux-zeroing method does not always preserve the optimal value of the objective function, since they set the pivoting vector to be a vector of all ones, which may not always lie in $\mathcal{N}(G)$, the null space of G. Moreover, as a result of this pivoting step, some internal fluxes become zero, forcing the corresponding change in the chemical potentials to be zero. From equation (4), some *additional* change in the chemical potentials are inferred to be zero. These *additional* $\Delta\mu$'s force the corresponding internal fluxes to be zero, to maintain thermodynamic feasibility. These *additional* constraints on the fluxes make the EBA solution sub-optimal. In this paper we use lemma 2 to transform thermodynamically unfeasible loops, into thermodynamically feasible loops by making particular fluxes zero and then removing the reaction corresponding to the zero fluxes. This will perturb the metabolic network.

Definition: An internal flux x_i is called *non-overlapping* if it belongs to only one basis cycle.

Definition: An internal flux x_i is *limiting* if by pivoting on it, the constraints on the other internal fluxes are not violated.

The next section develops a theorem that we will use in the algorithm to break thermodynamically unfeasible cycles.

5 Breaking Cycles: Algorithm to Perturb Thermodynamically Unfeasible Networks

A metabolic network with cycles that violates the second law of thermodynamics is unphysical, hence these unfeasible cycles must be detected and removed. The following theorem gives a method to break such cycles.

Theorem 1. (*Zero Transformation Theorem*)
If the entries of the ith column of the matrix G, G_{*i}, corresponding to the ith internal, *non-overlapping* flux x_i, which *only* belongs to the jth cycle, are set equal to zero, then the jth row of the K matrix K_{j*} is zero everywhere except at the ith column, that is, the (j, i)th entry of the K matrix, k_{ji} is nonzero.

Proof: Since N is the null space matrix of G, we have $GN = 0$, where 0 is a $(m \times n_l)$ matrix of zeros.
From $GN = 0$, we have, on expanding the matrix-matrix product:
$\left[\sum_{p=1}^{n_i} G_{*p}n_{p1}, \ldots, \sum_{p=1}^{n_i} G_{*p}n_{pj}, \ldots, \sum_{p=1}^{n_i} G_{*p}n_{pn_l}\right] = 0$. The jth cycle corresponds to the jth row of K and hence the jth column of the N matrix. Consider the equation corresponding to the jth column of the N matrix:

$$\sum_{p=1}^{n_i} G_{*p}n_{pj} = \mathbf{0} \tag{7}$$

where, $\mathbf{0} \in \mathcal{R}^m$ is a column vector of m zeros, and n_{pj} is the (p, j)th element of the N matrix, which is non-zero if the flux p belongs to cycle j, and is otherwise zero:

$$G_{*1}n_{1j} + \ldots + G_{*n_i}n_{n_ij} = -G_{*i}n_{ij} \tag{8}$$

where on the left side we exclude the ith index, which we bring to the right. Without loss of generality let $n_{ij} = 1$, hence:

$$G_{*1}n_{1j} + \ldots + G_{*n_i}n_{n_ij} = -G_{*i} \tag{9}$$

Since G_{*i} is a zero vector, we have:

$$G_{*1}n_{1j} + \ldots + G_{*n_i}n_{n_ij} = \mathbf{0} \tag{10}$$

From the above equation (10) we see on the left hand side that,
$G_{*1}n_{1j}, \ldots, G_{*n_i}n_{n_ij}$ are at most $n_i - 1$ nonzero vectors, which were a part of the jth cycle along with $G_{*i}n_{ij}$ (which corresponds to the ith internal flux),

setting $G_{*i} = 0$ (whenever $x_i = 0$, we can set $G_{*i} = 0$ without violating the flux conservation constraint $G\boldsymbol{x} = -H\boldsymbol{y}$) breaks the cycle, and the rest of the nonzero column vectors, $G_{*1}, \ldots, G_{*n_i}$ in the broken cycle are linearly independent and since at most n_i internal fluxes in the jth cycle, corresponding to the jth row of the K matrix K_{j*}, are linearly dependent (since they form a cycle), breaking the cycle makes the rest of the fluxes in the jth cycle linearly independent. In our construction of the null space of G we only considered irreducible cycles that form the basis of $\mathcal{N}(G)$. Hence, if breaking the cycle doesn't make the rest of the fluxes in the jth cycle linearly independent, another cycle is present in the jth cycle, which is a contradiction since the jth cycle is an *irreducible* basis cycle. The only way equation (9) holds is when $n_{1j} = \ldots = n_{n_ij} = 0$. Hence, the jth column of the matrix N, N_{*j} is zero except for the n_{ij} element which is nonzero. Since $K = N^T$, the jth row of K, K_{j*} is zero except for $k_{ji} = n_{ij}$.

From theorem 1 we see that when a particular, *non-overlapping* internal flux is zero we can zero out its respective column in the G matrix breaking the cycle which contains the flux that has become zero due to pivoting. So the row and column corresponding to the broken cycle and the zero flux can be deleted from the K matrix, changing the stoichiometric matrix and hence perturbing the metabolic network. The change in chemical potential corresponding to the deleted zero flux is unconstrained. Our algorithm uses this theorem to remove one cycle at a time, by zeroing out *non-overlapping* internal fluxes in the pivoting step.

To construct a flux vector that satisfies both FBA and EBA we impose additional constraints, by setting the *limiting* flux components, of every thermodynamically unfeasible cycle to zero. We then delete these fluxes by setting the corresponding columns of the G matrix to zero. This is described below.

i) Solve the FBA for the flux vector $\boldsymbol{f}$. If the FBA cannot find a solution then the problem has no solution.

ii) Compute the K matrix via the reduced row echelon form of the G matrix.

iii) Scan the flux components of the flux vector computed in step (i) and change the sign of the entries of the columns of the K matrix corresponding to negative fluxes, according to lemma 1, transforming the cycles in the network. Check the K matrix for no-cycle feasibility by applying lemma 1.

iv) If, after the above steps, the K matrix is feasible, we solve the combined linear program to compute the $\boldsymbol{\Delta\mu}$ vector that satisfies the constraints in equations (4), (5) and (6). The components of the $\boldsymbol{\Delta\mu}$ vector in the solution are constrained to satisfy equation (5) by adjusting equation (6).

v) If the K matrix is unfeasible after step (iv), we pivot out the *limiting* flux components in each cycle to zero. We set the columns of the G matrix corresponding to the zero fluxes to zero and compute the K matrix, perturbing the network. In this step, if a column of the G matrix corresponds to a *non-overlapping* flux, then we can apply theorem 1 to compute the K matrix. Otherwise, we compute the K matrix in the usual way. From the K matrix we remove rows corresponding to broken cycles and also remove columns corresponding to zero fluxes. We do not constrain the chemical potential change for these deleted zero fluxes. We check the K matrix for feasibility.

If feasible, we repeat step (i) with the updated G and K matrices and a new set of constraints on the zero fluxes. If the K matrix is unfeasible we report that the flux vector is thermodynamically unfeasible.

The sign test in lemma 1 is a necessary condition for detecting thermodynamic unfeasibility [5]. Combined with the LP in the above algorithm, it becomes a necessary and sufficient test if a given flux is thermodynamically unfeasible.

In the above algorithm, the computation of $\boldsymbol{\Delta\mu}$ decouples from the computation of the flux vector $\boldsymbol{f}$. When implemented in MATLAB (The Mathworks Inc., Natick, MA), however, the two can be combined into single linear program, the details are not discussed in this paper. Also, when trying to identify non-shared fluxes among basic cycles, it is best to use the rational basis of the null space for the G matrix. After a feasible flux vector $\boldsymbol{f}$ is identified, the lower and upper bounds on $\boldsymbol{\Delta\mu}$ in equation (6) can be adjusted, and by using linear programming to produce a $\boldsymbol{\Delta\mu}$ vector that satisfies the constraints in equations (4) and (5). Since the problem can admit multiple solutions, we can perform the MATLAB computation starting from another initial flux vector and repeat all the steps in the algorithm.

The overall complexity of the algorithm comes from computing the null space and solving the LP. Both take polynomial time.

5.1 Perturbing the *E. coli* Central Metabolism Network

We use the stoichiometric matrix S of the model *E. coli* system from Table 1 [4] for our FBA/EBA analysis. The reaction network contains 19 metabolites linked by 23 reactions (Figure 1 [4]). Out of these 23 fluxes, 3 are external or boundary fluxes and 20 are internal fluxes. The network takes glucose as input and produces acetate and carbon dioxide. We applied our algorithm to maximize the production of biomass, which is a linear combination of the different fluxes with experimentally determined stoichiometric coefficients. In the FBA optimization the internal fluxes are unrestricted and only satisfy the flux balance constraint. The $C0_2$ and acetate fluxes come from the literature (references found in [4]). Since only the relative rates matter, the glucose flux is set to 1, and all other fluxes are normalized with respect to it.

The G matrix is formed by considering the columns of the following internal fluxes from Table 1 of Delgado and Liao (1997):

$\boldsymbol{x} = [J_{pgi}, J_3, J_{pep}, J_{pyk}, J_{pdh}, J_{ace}, J_8, J_{ict}, J_{11}, J_{12}, J_{ppc}, J_{14}, J_{15}, J_{16}, J_{tkt}, J_{tal}, J_{resp}, J_{atp}, J_{biomass}, J_{glyox}]$ and the H matrix is formed from the columns of the external fluxes: $\boldsymbol{y} = [J_{gluc}, q_{CO_2}, q_{ace}]$.

The null space of the G matrix is of dimension 1, hence the K matrix consists of one row, corresponding to a single loop in the network:

$K = [0, 0, 0, 1, 1, 0, 0, -1, -1, -1, -1, 0, 0, 0, 0, 0, -1, -3, 0, 1]$.

From the non-zero entries of the K matrix, we see that the following 9 fluxes form a cycle: $[J_{pyk}, J_{pdh}, J_{ict}, J_{11}, J_{12}, J_{ppc}, J_{resp}, J_{atp}, J_{glyox}]$.

We consider the following optimal internal flux vector which satisfies the FBA and optimizes the production of biomass, ($J_{biomass} = 0.0001$ per unit of glucose consumed) but is thermodynamically unfeasible:

$\boldsymbol{x} = [0.87, 0.85, 1.58, 48.51, 49.31, 0.27, 0.56, -47.63, -47.71, -47.71, -47.98, 0.12, 0.09, 0.03, 0.03, 0.03, -44.80, -136.84, 0.0001, 48.19]^T$

To see that this flux vector is thermodynamically unfeasible, we transform columns 8, 9, 10, 11, 17 and 18 of the K matrix corresponding to the negative flux components to obtain a transformed K matrix $[0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 3, 0, 1]$, and see that all components have the same sign. This is unfeasible. We identify the *limiting* internal flux x_{20}, and make $x_{20} = 0$, by subtracting $x_{20} * K^T$ from $\boldsymbol{x}$ to get:

$\bar{\boldsymbol{x}} = [0.87, 0.85, 1.58, 0.33, 1.12, 0.27, 0.56, 0.48, 0.48, 0.21, 0.12, 0.09, 0.03, 3.38, 7.71, 0.0001, 0]^T$. We now delete the reaction corresponding to x_{20}, which corresponds to the flux J_{glyox}, perturbing the unfeasible network. By theorem 1, the cycle breaks and the network becomes thermodynamically feasible. Also, the chemical potential difference $\Delta\mu_{20}$ for the deleted flux x_{20} is no longer restricted. By applying theorem 1, the transformed K matrix is $\bar{K} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$. Since the perturbed network has no cycles, we can easily choose $\Delta\mu_1, \ldots, \Delta\mu_{19}$ to be negative, satisfying the second law of thermodynamics in equation (5).

We have applied our technique to the full metabolic network of *E. coli* [10], and found several cycles that violated the second law of thermodynamics. Perturbing the network can make it feasible, as in this example.

6 Discussion

At the fundamental level metabolic pathways have been represented as complex networks of reactions, which obey the stoichiometric relationship between the different metabolites. The stoichiometric matrix contains all the information for balancing the fluxes and applying the second law of thermodynamics to the overall system. It constrains the metabolic network to satisfy the law of conservation of mass and the thermodynamic law so that the reactions in the network are physically meaningful and can be readily compared with experiments.

If we want to go beyond this model, we need a detailed knowledge of the kinetics of these reactions, the rate constants and the interactions between different reactions. The flux and energy balance analysis provides an alternative route to computing the rates of reaction, without using detailed kinetic information. Recent enhancements to the stoichiometric theory look promising and the contribution in this paper will lead to more complete models of metabolic pathways.

7 Conclusion

This paper gave a simple linear programming algorithm to compute fluxes of reactions that satisfy both flux and energy balance constraints. This technique differs from previous approaches, as it is constructive and it perturbs the metabolic network by deleting reactions to produce feasible solutions. We applied the method to the metabolic network of *E. coli* and computed the fluxes and changes in chemical potentials for the internal reactions. However, FBA together with EBA cannot constrain the metabolic network completely, leading to an infinity of flux and chemical potential difference vectors. More realistic bounds on the values of fluxes from studying the biochemistry of several pathways are required as further constraints. In the future, a more complete formulation could make the change in chemical potential for each internal reaction more understandable and comparable to experiment. Now it just dictates the direction of thermodynamically feasible fluxes in the metabolic network. Linear algebra and thermodynamics together lead to a law of entropy in metabolic networks. Since the analysis and techniques presented here are simple, they can easily be applied to large-scale metabolic networks.

Acknowledgement

Part of this work was done while the first author was visiting the International Computer Science Institute at Berkeley. He would like to thank Prof. Richard Karp for helpful discussions. We also thank Prof. Robert Laughlin of the department of Physics at Stanford for his support of this project. This work was supported by a joint NASA-NCI biotechnology research program.

References

1. Bailey, J. E. (1991) Toward a Science of Metabolic Engineering, *Science* **252**, 1668-1675.
2. Beard, D. A., Liang, S. and Qian, H. (2002) Energy Balance for Analysis of Complex Metabolic Networks, *Biophys. J.* **83**, 79-83.
3. Beard, D. A., Babson, E., Curtis, E. and Qian, H. (2004) Thermodynamic Constraints for Biochemical Networks, *J. Theor. Biol.* **228**, 327-333.
4. Delgado, J. and Liao, J. C. (1997) Inverse Flux Analysis for Reduction of Acetate Excretion in *Escherichia coli*, *Biotechnol. Prog.* **13**, 361-367.
5. Nigam, R. and Liang, S. (2004) Thermodynamic Feasibility of Metabolic Networks, International Conference on Bioinformatics and its Applications, Fort Lauderdale, USA.
6. Oster, G. F., Perelson, A. S. and Katchalsky, A. (1973) Network thermodynamics: Dynamic Modelling of Biophysical Systems, *Quart. Rev. Biophys.* **6**, 1-134.
7. Papin, J. A., Price, N. D., Wiback, S. J., Fell, D. A. and Palsson, B. O. (2003) Metabolic Pathways in the Post-genome Era, *Trends in Biochemical Sciences* **28**, 250-258.
8. Price, N. D., Famili, I., Beard, D. A. and Palsson, B. O. (2002) Extreme Pathways and Kirchhoff's Second Law, *Biophys. J.* **83** 2879-2882.
9. Qian, H., Beard, D. and Liang, S. (2003) Stoichiometric Network Theory for Nonequilibrium Biochemical Systems, *Eur. J. Biochem.* **270**, 415-421.

10. Reed, J. L., Vo, T. D., Schilling, C. H. and Palsson, B. O. (2003) An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR), *Genome Biology*, **4**, R54.1-R54.12.
11. Strang, G. (2003) Introduction to Linear Algebra, *Wellesley-Cambridge Press*, Wellesley, MA.
12. Stephanopoulos, G., Aristidou, A. and Nielsen, J. (1998) Metabolic Engineering: Principles and Applications, *Academic Press*, San Diego, CA.
13. Varma, A., and Palsson, B. O. (1994) Stoichiometric Flux Balance Models Quantitatively Predict Growth and Metabolic By-Product Secretion in Wild Type *Escherichia coli* W3110, *Appl. Environ. Microbiol.* **60**, 3724-3731.

Protein Cellular Localization with Multiclass Support Vector Machines and Decision Trees

Ana Carolina Lorena and André C.P.L.F. de Carvalho

Instituto de Ciências Matemáticas e de Computação (ICMC),
Universidade de São Paulo (USP),
Av. Trabalhador São-Carlense, 400 - Centro - Cx. Postal 668
São Carlos - São Paulo - Brasil
{aclorena, andre}@icmc.usp.br

Abstract. Many cellular functions are carried out in compartments of the cell. The cellular localization of a protein is thus related to its function identification. This paper investigates the use of two Machine Learning techniques, Support Vector Machines (SVMs) and Decision Trees (DTs), in the protein cellular localization prediction problem. Since the given task has multiple classes and SVMs are originally designed for the solution of two class problems, several strategies for multiclass SVMs extension were investigated, including one proposed by the authors.

Keywords. protein cellular localization, Machine Learning, multiclass Support Vector Machines, Decision Trees.

1 Introduction

Proteins may be located at various regions in the cell or transported to the extracellular space [8]. The identification of a protein destination is important to understand its function. Even knowing the protein function, its localization may provide valuable information about enzyme pathways [6].

Several works employed Machine Learning (ML) techniques in this recognition task [6, 9, 10, 11, 14]. ML is a sub-area of Artificial Intelligence that provides techniques which can extract concepts (knowledge) from a given dataset [15]. These techniques are usually applied to the induction of a classifier or predictor, through a process called training. The generated predictor can then be used in the classification of new instances from the same domain.

In [10], a k-nearest neighbor (kNN) classifier, a DT, a Naïve-Bayes (NB) classifier and a probabilistic model were applied in the recognition of *E. coli* and yeast protein locations. In [6] and [11], SVMs were successfully employed in the localization of prokaryotic and eukaryotic proteins. SVMs were also applied recently in the localization of human proteins [9]. In [14], a NB classifier was used in the localization of proteins of five distinct organisms, and the generated predictors are part of a protein analysis web-service.

This work applies two ML techniques in the recognition of prokaryotic and eukaryotic protein locations: Decision Trees (DTs) [17] and Support Vector Ma-

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 42–53, 2005.

chines (SVMs) [5]. These algorithms follow distinct approaches in the learning process, which are generally named symbolic and statistical.

In relation to the SVM classifiers, one issue studied in this paper that differentiates it from previous works [6, 9, 11] is how the extension of the SVMs to the multiclass localization task is performed, since they are originally designed for the solution of two class problems. While the later studies applied one particular method in the multiclass SVMs generalization, the present paper compares several strategies in this extension. Among the tested approaches is one technique proposed by the authors in [13] and expanded here.

This paper is structured as follows. Section 2 describes the materials and methods employed in this work. Section 3 presents experimental results. Section 4 discusses the results obtained. Section 5 concludes this paper.

2 Materials and Methods

2.1 Learning Techniques

Several ML algorithms can be applied to induce a classifier from a set of examples. Given a training set composed of known protein sequences with their corresponding localization, the learning algorithm must induce a classifier that should be able to predict the class of new samples from the same domain. The learning techniques used in this paper are Decision Trees [17] and Support Vector Machines [5].

The Decision Tree (DT) [17] is a symbolic learning technique that organizes information extracted from data in a structure composed of nodes and ramifications. The nodes represent either tests applied to data or classes, when the node is a leaf. The ramifications are possible results of the tests. The classification of a new sample is performed following the nodes and ramifications until a leaf is reached.

The DT induction is conceived iteratively until all training examples are correctly classified. The generated structure is thus subject to overfitting [15], in which the classifier specializes to the training samples, showing poor performance on new data. To avoid this effect, a pruning phase is usually applied to the trained tree. It prunes ramifications that have low expressive power according to some criterion, like the expected error rate [18]. In this process, whole subtrees are replaced by leaf nodes. The replacement is made if the expected error rate in the subtree is larger than in the single leaf.

Support Vector Machines (SVMs) represent a learning technique based on the Statistical Learning Theory [22]. Given a dataset with n samples $(\mathbf{x}_i, y_i)$, where each $\mathbf{x}_i \in \Re^m$ (Euclidean space with m dimensions) is a data sample and $y_i \in \{-1, +1\}$ corresponds to $\mathbf{x}_i$'s label, this technique seeks an hyperplane $(\mathbf{w} \cdot \mathbf{x} + b = 0)$ able of separating data with a maximal margin. To perform this task, it solves the following optimization problem:

$$\textbf{Minimize: } \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$\textbf{Restrictions: } \begin{cases} \xi_i \geq 0 \\ y_i \left(\mathbf{w} \cdot \mathbf{x_i} + b \right) \geq 1 - \xi_i \end{cases}$$

where $\| \cdot \|$ denotes the Euclidean norm, C is a constant that imposes a tradeoff between training error and generalization and the ξ_i are slack variables. These variables relax the restrictions imposed to the optimization problem, allowing some patterns to be within the margins.

When a non-linear separation of the dataset is needed, its data samples are mapped to a higher-dimensional space. In this space, also named feature space, the dataset can be separated by a linear SVM with a low training error. This mapping process is performed with the use of Kernel functions, which compute dot products between any pair of patterns in the feature space in a simple way. Thus, the only modification necessary to deal with non-linearity with SVMs is to substitute any dot product among patterns by a Kernel function. In this work, the Kernel function used was a Gaussian, illustrated in Equation 1 [3].

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (1)$$

It should be noticed that SVMs originally can only deal with binary classifications. However, several strategies can be employed to extend them to multiclass problems, as described next.

2.2 Multiclass Support Vector Machines Approaches

This section describes the strategies used in the extension of SVMs to the solution of the multiclass protein cellular localization problem.

One-against-all. In the one-against-all (1AA) decomposition, given k classes, k binary predictors are generated, each being responsible to distinguish a class i from the remaining classes. The final prediction is given by the classifier with the highest output value [22].

All-against-all. The all-against-all (AAA) decomposition consists of building $k(k-1)/2$ predictors, each differentiating a pair of classes i and j, with $i < j$. For combining these classifiers, a majority voting scheme can be applied [12]. Each AAA classifier gives one vote to its preferred class. The final result is then given by the class with most of the votes.

Directed Acyclic Graphs. The AAA decomposition with majority voting integration presents the drawback of generating unknown classifications, which occur when more than one class receives the maximum number of votes. To overcome this problem, a Decision Directed Acyclic Graph (DDAG) [16] can be employed in the binary classifiers combination.

A Directed Acyclic Graph (DAG) is a graph with oriented edges and no cycles. The DDAG approach uses the classifiers generated through AAA decomposition in each node of a DAG, as illustrated in Figure 1. The prediction of a new pattern class is obtained following the nodes and ramifications of the DDAG until the last level is reached, giving the final classification.

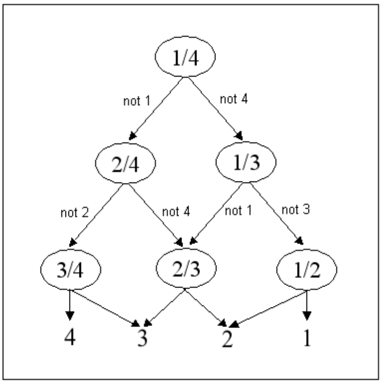


Fig. 1. DDAG for a problem with four classes [16]

Error Correcting Output Codes. In an alternative strategy, Dietterich and Bariki [7] proposed the use of a distributed output code to represent the k classes associated with a multiclass problem. For each class, a codeword of length l is assigned. Frequently, the size of the codewords has more bits than needed in order to represent each class uniquely. The additional bits can be used to correct eventual classification errors. For this reason, this method is named error-correcting output coding (ECOC). A new pattern $\mathbf{x}$ can be classified by evaluating the predictions of the l classifiers, which generate a string $\mathbf{s}$ of length l. This string is then compared to the codeword associated to each class. The sample is assigned to the class whose codeword is closest according to a given distance measure.

In this paper, the decoding function used was extracted from [2]. It considers the margins of each SVM classifier and was more accurate than the Hamming distance applied in [7]. Equation 2 presents the computation of this margin-based measure, where q is a class and m_{qi} represents the ith bit of class q codeword.

$$d_m\left(\mathbf{s}, q\right) = \sum_{i=1}^{l} \max\left\{0, 1 - \left(m_{qi} \cdot s_i\right)\right\} \tag{2}$$

Following the instructions given in [7], the ECOC codes generated in this work are given by all $2^{k-1} - 1$ possible binary partitions of the classes. In this scheme, the codeword of the first class is composed only of ones. For each other class i, where $i > 1$, it is composed of alternate runs of 2^{k-i} zeros and ones.

Hierarchical SVMs and Minimum Spanning Trees. This work also investigates the combination of binary SVMs in a hierarchical structure as illustrated in Figure 2c. Each level of the hierarchy distinguishes two subsets of classes. Based on the decision from the previous levels, new nodes are visited, until a leaf node is reached, where the final classification is given. In general, it can be stated that the hierarchical approaches (that also include the DDAG strategy) have faster prediction times, since usually a lower number of classifiers need to

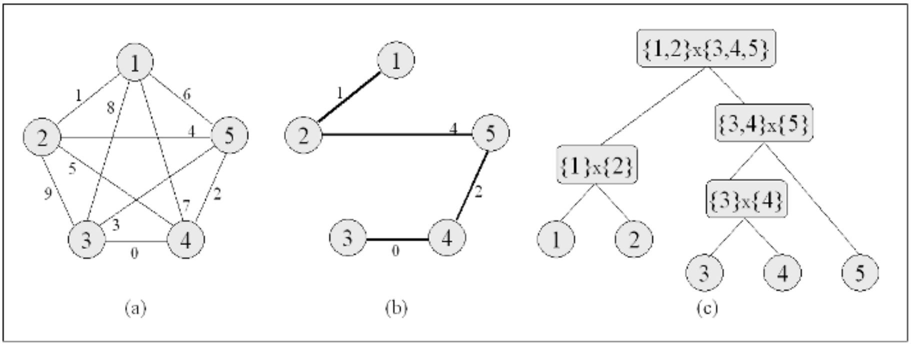


Fig. 2. (a) Graph for a problem with five classes; (b) Minimum Spanning Tree; (c) Multiclass hierarchical structure obtained

be consulted for each prediction. In the hierarchical structure used, a lower number of binary classifiers is also induced ($k-1$), so the time spent on the SVM classifiers training can be reduced too.

This type of hierarchical architecture is also used in [4, 20, 23]. These works differ on the way the binary partitions of classes in each level of the tree are obtained. The present work uses some concepts from these papers to build a Minimum Spanning Tree, which is then used to obtain the hierarchies of classes. This idea was first introduced in [13] and is expanded here.

Given an undirected graph $G = (V, E)$ with $|V|$ vertices, $|E|$ edges and a cost or weight associated to each edge, a Minimum Spanning Tree (MST) T is a connected acyclic subgraph that spans all vertices of G with the smallest total cost of edges [1].

Information collected from the training dataset is used to obtain the weighted graph, which has k vertices and $k(k-1)/2$ edges connecting all pairs of vertices. Figure 2a illustrates an example of a graph for a problem with five classes, while Figure 2b shows the MST extracted from this graph. Various methods can be used to assign costs to the edges. In this work, the following approaches are investigated:

1. *Centroid distances*: each class is first represented by a centroid $\boldsymbol{\mu}_i$. The weight of an arc (i, j) is then given by the Euclidean distance between $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$. Using this criterion, the MST will group in each level of the hierarchy subsets of one or more classes that are similar to each other according to their centroid.
2. *Inverse of centroid distance*: in this case, the weight of an arc (i, j) is given by $1/d_E(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$, where d_E is the Euclidean distance. The MST will group subsets of classes that are more distant according to their centroid.
3. *Balanced subsets*: inspired by ideas presented in [23], this criterion acts by grouping classes that have similar data distribution. The weight of an arc (i, j) is then given by the difference among the number of patterns from classes i and j.

4. *Inverse of balanced subsets*: this criterion weights the connection between classes i and j as the inverse of the measure employed in the balanced subsets method. Using this weighting, classes more distant according to their sample distribution are grouped by the MST algorithm.
5. *Scatter Measure*: using concepts from [4], the weight of an arc (i, j) in this method is given by a scattering measure between classes i and j. This measure is calculated by Equation 3, where $\mathbf{s}_i^2$ and $\mathbf{s}_j^2$ are the variances of data samples from classes i and j, respectively. The MST will group classes considered less separated according to the scattering measure calculated.

$$s_m\left(i,j\right) = \frac{\left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|^2}{\mathbf{s}_i^2 + \mathbf{s}_j^2} \tag{3}$$

6. *Inverse of Scatter Measure*: the application of this method, given by $\frac{1}{s_m(i,j)}$, tends to maximize the distance between group centers and minimize the variance in each group during the hierarchies formation.
7. *Confusion matrix*: given the concept of confusion classes from [20], a confusion matrix can be employed in the graph weights definition. A confusion matrix offers an idea of which classes a classifier has more difficulty to distinguish. For a dataset with k classes, it has kxk dimension, and each element m_{ij} represents the number of examples from class i that were misclassified as belonging to class j [15]. To obtain this matrix, the whole k class problem has to be solved first. In this work, DTs were used in the confusion matrix generation. The weight of an arc (i, j) is then calculated by Equation 4, where n_i is the number of examples from class i. Applied to these weights, the MST algorithm will group subsets of classes that present less confusion with each other.

$$d_{CM}\left(i,j\right) = \frac{m_{ij}}{n_i} + \frac{m_{ji}}{n_j} \tag{4}$$

8. *Inverse of confusion matrix*: in this case, the weight of an arc (i, j) is given by $1/d_{CM}(i, j)$. The MST will group subsets of classes that present more confusion with each other.

The inverse criterions 2, 4, 6 and 8 were employed to explore the contrast of using the dissimilarity between classes in the grouping process against the similarity. They are introduced in this work, as well as the confusion matrix weighting.

Given the obtained weighted graph, an adapted version of the Kruskal algorithm [1] was applied in the multiclass tree determination. The Kruskal algorithm maintains in each interation subsets of grouped vertices. Taking advantage of this characteristic, the proposed algorithm uses these groupings in the hierarchies formation process. The generation of the hierarchical structure operates in a bottom-up iterative way. A pseudocode of this algorithm can be found in [13]. The given algorithm is efficient and allows a totally automatic determination of a multiclass hierarchical classifier structure from binary predictors.

2.3 Datasets

The datasets used in the experiments reported in this paper were extracted from the UCI benchmark database [21]. They were submitted by Horton and Nakai [10]. The first one is used in the localization of *E. coli* proteins, a prokaryotic organism. The second contains samples for the prediction of yeast protein locations, which is an eukaryotic organism. The features contained in these datasets are numerical and were calculated from the amino acid sequences of the proteins. Details about them can be found in [21].

Originally, the *E. coli* dataset has 8 classes of protein locations. However, two of them have only two instances and one has 5 instances. These very low numbers of examples result in problems to the classifiers induction process. Thus, only 5 classes were used. The yeast dataset has one class with only 5 instances, which was also not considered in this study.

Table 1 describes the *E. coli* dataset, showing its number of instances (♯Inst), the number of attributes (♯Attr, all continuous valued), the baseline error (BE), which is the error rate for a classifier that always predicts the class with most instances, and the number of examples in each class. In this table, "cp" refers to cytoplasm, "im" to inner membrane without signal sequence, "pp" to perisplasm, "imU" to inner membrane with uncleavable signal sequence and "om" to outer membrane.

Table 1. *E. coli* dataset description

			♯Inst per class				
♯Inst	♯Attr	BE	cp	im	pp	imU	om
327	7	43.7%	143	77	52	35	20

Table 2. Yeast dataset description

			♯Inst per class								
♯Inst	♯Attr	BE	CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX
1481	8	45.5%	463	429	244	163	51	44	37	30	20

Table 2 describes the yeast dataset. "CYT" refers to cytosolic or cytoskeletal, "NUC" to nuclear, "MIT" to mitochondrial, "ME3" to membrane protein with no N-terminal signal, "ME2" to membrane protein with uncleaved signal, "ME1" to membrane protein with cleaved signal, "EXC" to extracellular, "VAC" to vacuolar and "POX" to peroxisomal.

3 Experiments

In order to obtain estimatives of the accuracy rates of the classifiers generated in this study, the datasets previously described were divided following the 10-fold cross-validation methodology [15]. Accordingly, each dataset was divided in ten disjoint subsets of approximately equal size. In each train/test round, nine subsets were used for training and the remaining was left for test. This makes a total of ten pairs of training and test sets. The accuracy of a classifier on the total dataset was then given by the average of the accuracies observed in each test partition. A stratified approach was adopted in this division process, maintaining the class distribution of the original dataset in each partition.

Some parameters had to be set for the ML techniques. To adjust these parameters, each of the training datasets was further divided with holdout to generate tuning datasets, in a proportion of 70% for training and 30% for validation. For each dataset, the classifiers were then generated on the new training partition and tested on the validation set for all parameters combination. The parameter values were chosen as the ones that lead to a maximum accuracy in the validation set. The final classifier was then generated using the whole training dataset with the parameters determined and tested on the test set. To speed up this process, the same parameters were employed in all binary SVMs induced in each multiclass strategy. It should be noticed that, through out this process, different parameters can be chosen for distinct partitions of the data.

For the DTs, the parameter controlled was the pruning confidence. The values tested were of 0.25, 0.5 and 0.75. For SVMs, different combinations of the C and Kernel σ parameters were tested, being: $C = [10^0, 10^1, 10^2, 10^3]$ and $\sigma = [10^{-3}, 10^{-2}, 10^{-1}, 10^0]$. This gives a total of 16 combinations of parameters for each dataset. The DTs induction was performed with the C4.5 algorithm [18], while the SVMs were generated using the LibSVM library [3].

For SVMs it is also necessary to normalize the datasets, preventing attributes in higher numerical ranges from dominating those in lower levels. All training datasets attributes were then normalized to null mean and unit variance. Their corresponding test and validation sets were also pre-processed according to the normalization factors extracted from the training data.

The DDAG results depend on the sequence of nodes chosen to compose the graph. Thus, for each data partition in *E. coli*, 30 random DDAG structures were generated and the best was chosen based on its validation accuracy. The same procedure was applied in the yeast dataset, with 60 DDAGs.

Table 3 shows the mean and standard deviation of accuracies (Total column) obtained by each technique in the cross-validation partitions for the *E. coli* dataset, as well as the performance in each class. Standard deviation values of the rates are indicated in brackets. The numbers 1 to 8 indicate the MST based techniques. The rows 1AA to 8 refers to SVMs multiclass techniques. The best rates are highlighted in boldface and the worst in italic. Table 4 presents the accuracies rates for the yeast dataset.

As pointed in Section 2.2, the AAA strategy with majority voting, denoted in Tables 3 and 4 by AAA, presents the occurrence of unknown classifications. The total rates of unknown samples in the *E. coli* and yeast dataset were of 0.6 (std 1.3) and 0.7 (std 0.6), respectively.

Table 3. Results on *E. coli* dataset

Tec	cp	im	pp	imU	om	Total
1AA	**97.9**	81.8	86.7	**65.0**	**85.0**	88.1 (4.9)
AAA	**97.9**	79.1	82.7	59.2	**85.0**	86.2 (3.9)
DDAG	**97.9**	81.8	84.7	59.2	80.0	86.9 (3.2)
ECOC	**97.9**	84.3	86.7	61.7	85.0	**88.4 (3.7)**
1	**97.9**	77.9	88.7	59.2	80.0	86.5 (3.9)
2	**97.9**	76.4	88.7	**65.0**	85.0	87.2 (4.2)
3	97.2	83.2	86.7	61.7	**85.0**	87.8 (3.8)
4	**97.9**	83.2	**90.7**	59.2	80.0	88.1 (4.8)
5	*95.8*	79.1	85.0	62.5	85.0	85.9 (4.1)
6	97.1	81.6	84.7	**65.0**	80.0	87.1 (3.5)
7	97.2	79.1	86.7	62.5	**85.0**	86.9 (4.3)
8	97.1	*76.4*	82.7	59.2	80.0	*85.0 (3.6)*
DT	*95.8*	**92.3**	*78.7*	*46.7*	*70.0*	85.6 (5.7)

Table 4. Results on yeast dataset

Tec	CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	Total
1AA	67.7	50.8	**57.7**	**84.8**	33.3	**81.0**	50.0	3.3	45.0	60.1 (1.8)
AAA	72.2	48.2	55.3	78.1	41.3	71.5	61.7	*0.0*	**50.0**	59.9 (2.6)
DDAG	70.2	49.9	54.5	78.6	41.3	71.5	61.7	3.3	**50.0**	59.8 (2.6)
ECOC	69.8	50.1	**57.7**	81.1	39.3	73.5	61.7	**6.7**	45.0	**60.5 (2.8)**
1	71.3	51.3	48.3	75.6	33.7	69.5	47.5	*0.0*	45.0	58.3 (3.1)
2	*56.0*	**59.4**	52.0	78.7	41.3	73.5	61.7	*0.0*	45.0	57.6 (3.3)
3	70.0	50.4	50.4	78.7	**43.3**	75.5	52.5	**6.7**	45.0	59.2 (3.2)
4	**74.7**	49.4	45.8	71.9	26.0	71.5	**65.0**	3.3	45.0	58.3 (2.6)
5	71.7	52.0	51.1	*71.3*	39.0	*51.0*	*40.0*	*0.0*	**50.0**	58.3 (2.7)
6	69.6	*48.0*	*45.8*	83.0	*12.0*	73.5	**65.0**	*0.0*	45.0	57.1 (3.4)
7	61.2	56.4	50.7	79.3	25.0	80.0	52.5	3.3	**50.0**	57.8 (3.6)
8	71.3	51.3	50.3	75.0	33.7	69.5	55.0	*0.0*	45.0	58.8 (2.5)
DT	57.5	50.1	52.4	83.6	41.0	74.5	60.0	3.3	*30.0*	*55.8 (5.1)*

Results concerning training and test times were also collected, but are not presented here due to the lack of space. The DT model was faster in both steps. Among the SVMs strategies, the AAA technique was usually faster on training. Since in this technique each binary classifier involves patterns from only two classes, its induction was faster. Follows the hierarchical strategies, 1AA and ECOC, in this sequence. ECOC showed in general high training times compared to the other techniques (10 times higher in average for the *E. coli* dataset and 200 times higher for the yeast data). In the test phase, all techniques in general showed fast times, although the hierarchical strategies were faster. In the yeast dataset, ECOC again showed high test times compared to the other techniques (60 times higher in average).

4 Discussion

It is interesting to notice that, for both *E. coli* and yeast datasets, the ECOC strategy presented higher total accuracies. The next best results in the *E. coli* dataset were obtained by the hierarchical method 4 and the 1AA decomposition. In the yeast data, the next best strategies were 1AA and AAA. The total accuracies of the pointed strategies were very similar. In the classes, however, some differences can be detected. Each technique tends to present the highest accuracy in one or more classes. Except for the AAA strategy in VAC class (yeast), none of them presented the lowest accuracies on the classes. Their performance was, thus, generally good in all classes.

Comparing the results of all tested techniques in terms of total accuracy with the McNemar test and Bonferroni adjustment [19], a difference with 95% of confidence was found in the following cases: in *E. coli* dataset, between techniques ECOC and 8; in yeast dataset, between 1AA and DT, 1AA and 6, AAA and DT, AAA and 6, ECOC and DT, ECOC and 2, ECOC and 6, DDAG and DT and DDAG and 6. While in the first dataset the hierarchical method 8 showed worst results in comparison to the most accurate technique, in the second dataset the DT and the hierarchical method 6 presented the worst results in relation to the best accuracy techniques. Although the DT technique generally did not present the lowest accuracies for all classes of yeast data, its overall results were worst.

Higher standard deviation rates were vertified for the DT technique in both datasets. This indicates a larger variation among the results in each fold. It is also important to emphasize that all the results presented were above the baseline error of the datasets, indicating that the predictors were able to generalize.

In relation to the hierarchical strategies 1 to 8, 4 was better in the *E. coli* dataset, presenting the second highest total accuracy among all tested techniques, and 3 was better in the yeast dataset. Both represent the balanced subsets criterion, the first one being the inverse variation. In contrast, techniques 8 in *E. coli* and 6 in yeast showed low performances compared to the best accuracy ones, as discussed previously. In the classes, some strategies tend to favour one or more classes, but in general the results were similar to those obtained by other techniques.

The behavior of inverse (2, 4, 6 and 8) against "standard" (1, 3, 5 and 7) criterions in the hierarchical techniques was also observed. In the *E. coli* dataset, except for the matrix confusion criterion, the inverse weightings were better than their standard counterpart. For the yeast dataset, the opposite was verified. The matrix confusion inverse criterion was better than the standard one and the centroid, balanced subsets and scatter standard weightings were better than their inverse. Thus, while in the *E. coli* dataset a dissimilarity between protein classes was in general better in generating the hierarchies, in the yeast dataset the similarity usually showed better effect.

An interesting result is the very good accuracy of DTs in the "im" class of *E. coli* dataset when compared to the other techniques. It suggests that the DTs favoured this class, since in the other classes their results were worse.

In the yeast dataset, all techniques showed low accuracies for the "VAC" class. This aspect was also reported in [10], although their results are not directly comparable to the ones in this paper, since the dataset was modified and the cross-validation partitions are different.

As a general conclusion, the SVMs achieved better results than DTs in the given application, although a statistical significance was found only in the yeast dataset. Among the SVMs multiclass strategies, apart from some exceptions, the results were similar. Other criterions can then be explored to choose a particular technique, like training and prediction times or the size of the multiclass predictors. A smaller number of binary classifiers leads to a simpler multiclass classifier. The hierarchies 1 to 8 present the lowest number of binary classifiers $(k-1)$, followed by the 1AA decomposition (k), AAA decomposition $(k(k-1)/2)$, which includes DDAG, and the ECOC $(2^{k-1}-1)$.

From the Bioinformatics view, the results indicate that the data attributes may be not the most adequate for this application. Thus, the use of new data, with additional features and known proteins, could improve the results observed.

5 Conclusion

This work investigated the use of two ML techniques in the multiclass problem of protein cellular localization, DTs and SVMs. As this is a multiclass problem and SVMs originally perform binary classifications, this work also investigated several techniques to extend them to multiclass applications, including one proposed by the authors, which was expanded in this paper. This study is one of the main contributions of this paper, since previous works with SVMs evaluated the application of only one multiclass strategy.

As a future work, the distinct multiclass SVM strategies tested can be combined. As some complementarity can be observed among the results of these techniques in the classes, a combination can improve the overall results obtained. One issue not investigated in this paper is whether the features used to represent the proteins harm the predictions obtained in some classes. This would be an interesting future work too, since recent studies show that the combination of multiple types of information about the proteins can help the classifier in the localization task [9]. It would be also profitable to observe which patterns each technique has more difficulty to classify.

Acknowledgements

The authors would like to thank the Brazilian research agencies FAPESP and CNPq for their financial support and the reviewers for their suggestions.

References

1. Ahuja, R. K., Magnanti, T. L., Orlin, J. B.: Network Flows: Theory, Algorithms and Applications. Prentice Hall (1993)
2. Allwein, E. L., Shapire, R. E., Singer, Y.: Reducing Multiclass to Binary: a Unifying Approach for Margin Classifiers. In Proc of the 17th ICML (2000) 9–16

3. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/
4. Cheong, S., Oh, S. H., Lee, S.-Y.: Support Vector Machines with Binary Tree Architecture for Multi-Class Classification. Neural Information Processing - Letters and Reviews, Vol. 2, N. 3 (2004) 47–50
5. Cristianini, N., Taylor, J. S.: An Introduction to Support Vector Machines. Cambridge University Press (2000)
6. Cui, Q., Jiang, T., Liu, B., Ma, S.: Esub8: A novel tool to predict protein subcellular localizations in eukaryotic organisms. BMC Bioinformatics, Vol. 5, N. 1 (2004) 66
7. Dietterich, T. G., Bariki, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. JAIR, Vol. 2 (1995) 263–286
8. Feng, Z.-P.: An overview on predicting the subcellular location of a protein. Silico Biology, Vol. 2, N. 3 (2002) 291–303
9. Garg, A., Bhasin, M., Raghava, G. P. S.: Support Vector Machine-based Method for Subcellular Localization of Human Proteins Using Amino Acid Compositions, Their Order, and Similarity Search. J of Biol Chem, Vol. 280, No. 15 (2005) 14427-14432
10. Horton, P., Nakai, K.: Better Prediction of Protein Cellular Localization Sites with k-Nearest Neighbor Classifiers. In: Proc of ISMB, Vol. 5 (1997) 147–152
11. Hua, S., Sun, Z.: Support Vector Machine Approach for Protein Subcellular Localization Prediction. Bioinformatics, Vol. 5, N. 8 (2001) 721–728
12. Kre*β*el, U.: Pairwise Classification and Support Vector Machines. In Advances in Kernel Methods - Support Vector Learning, MIT Press (1999) 185–208
13. Lorena, A. C., Carvalho, A. C. P. L. F.: Minimum Spanning Trees in Hierarchical Multiclass Support Vector Machines Generation. In LNAI 3533, The 18th Int Conf on Ind & Eng Applic of AI & Expert Systems, Springer-Verlag (2005) 422–431
14. Lu, Z., Szafron, D., Greiner, R., Lu, P., Wishart, D. S., Poulin, B., Anvik, J., Macdonell, C., Eisner, R.: Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics, Vol. 20, N. 4 (2004) 547–556
15. Mitchell, T.: Machine Learning. McGraw Hill (1997)
16. Platt, J. C., Cristianini, N., Shawe-Taylor, J.: Large Margin DAGs for Multiclass Classification. In: Solla, S. A., Leen, T. K., Mller, K.-R. (eds.), Advances in Neural Information Processing Systems, Vol. 12. MIT Press (2000) 547–553
17. Quinlan, J. R.: Induction of Decision Trees. Machine Learning, Vol. 1, N. 1, Kluwer Academic Publishers (1986) 81–106
18. Quinlan, J. R.: C4.5 Programs for Machine Learning. Morgan Kaufmann (1988)
19. Salzberg, S. L.: On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. Data Mining and Knowledge Discovery, Vol. 1 (1997) 317–328
20. Schwenker, F.: Hierarchical Support Vector Machines for Multi-Class Pattern Recognition. In: Proc of the 4th Int Conf on Knowledge-based Intell Eng Syst and Allied Tech. IEEE Computer Society Press (2000) 561–565
21. University of California Irvine: UCI benchmark repository - a huge collection of artificial and real-world datasets. http://www.ics.uci.edu/~mlearn
22. Vapnik, V. N.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
23. Vural, V., Dy, J. G.: A Hierarchical Method for Multi-Class Support Vector Machines. In: Proc of the 21st ICML (2004) 831–838

Combining One-Class Classifiers for Robust Novelty Detection in Gene Expression Data

Eduardo J. Spinosa and André C.P.L.F. de Carvalho

Universidade de São Paulo (USP), Instituto de Ciências Matemáticas e de Computação (ICMC),
Av. do Trabalhador São-Carlense, 400, São Carlos, 13560-970, Brasil
ejspin@icmc.usp.br**, andre@icmc.usp.br
http://www.icmc.usp.br/

Abstract. One-class classification techniques are able to, based only on examples of a normal profile, induce a classifier that is capable of identifying novel classes or profile changes. However, the performance of different novelty detection approaches may depend on the domain considered. This paper applies combined one-class classifiers to detect novelty in gene expression data. Results indicate that the robustness of the classification is increased with this combined approach.

1 Introduction

Supervised learning algorithms learn from labeled examples in a training set and later, on a test phase, attempt to classify new unseen examples based on the knowledge acquired in the training phase. In a traditional approach, the absence of good representative examples of a certain class in the training set leads to a poor performance of the classifier on that particular class. In an extreme situation, if a class does not have any examples at all, a traditional classifier will assign objects of that class to one of the known classes, even though it might not be an appropriate choice.

Therefore, the ability to detect a new class or sub-class is an important aspect for a machine learning system. Slight modifications in the data distribution might indicate, for instance, the appearance of a new class, or a profile modification in a class that has already been modeled. The capability to identify these changes is known as *Novelty Detection* (ND) [9], *Outlier Detection* or *One-Class Classification* [11] [12].

The term *One-Class* refers to the key characteristic of ND techniques, which is the fact that the training is carried out based only on examples from a single class that represents the normal profile. In other words, the algorithm learns to identify a novelty profile without having seen any examples of such a class. The power of novelty detection lies exactly on this aspect: in the training phase, no examples of any novel profile are presented. As a consequence, the performance

** Alternate e-mail: ejspin@yahoo.com

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 54–64, 2005.

of one-class classifiers cannot be directly compared to that of two-class classifiers, since the latter uses examples of both classes on the training phase.

Different approaches to ND have been proposed [9] and applied to a variety of tasks. In this paper, some of these approaches are combined to produce a single decision, as explained in Section 2. Section 3 presents and analyzes experiments involving gene expression data. Section 4 reviews the most important conclusions.

2 A Combined Approach to Novelty Detection

The problem of ND consists in the discovery of new profiles that were not present in the training samples. Thus, the classifier is induced based only on positive examples of a target class. All other examples are removed from the training set as these examples are considered outliers.

Of the various approaches to ND described in the literature, five of them have been chosen for this work: *Parzen Window* [10], *K-NN (K-Nearest Neighbor)* [6], *K-Means* [3], *SOM (Self-Organizing Map)* [8] and *PCA (Principal Components Analysis)* [3]. Each of these one-class classifiers uses one of three different strategies, according to the classification proposed by Tax [12]. Other classifications of ND techniques are available in the literature [9].

Parzen Window is a *density estimation* technique that, based on a data distribution scheme, defines a threshold to distinguish between normal and novel profiles. K-NN constructs hypersphere *boundaries* to involve data of the target class, therefore considering outliers any elements that fall outside these boundaries. K-Means, SOM and PCA are classified as *reconstruction* techniques. K-Means is a clustering algorithm that builds a boundary around prototype objects. SOM is based on a Neural Network architecture called Self-Organizing Map, in which prototypes are constrained to a lower-dimensional space in order to be later visualized. PCA performs a transformation of the original input attributes to a smaller number of uncorrelated, thus more meaningful, attributes.

Each of these techniques alone may perform better in a specific domain, and may also depend on a good parameter setting. Therefore, from a user's point of view, it might be hard to discover which approach is more likely to work best when experimenting with a variety of datasets.

The combined approach proposed in this work aims to increase classification robustness by taking into account the opinion of a set of one-class classifiers, instead of relying on a single approach, that might favor one class over the other.

Initially, all classifiers in the set are trained with a set containing only examples of the target class. For the same dataset, each class is considered the target class at one time, and examples of all other classes are labeled as outliers and used for testing purposes only. In the test phase, when target and outlier examples are present, the opinion of each classifier is taken and recorded. The final decision for each example (normal or novelty) is taken by the set of the classifiers. If the majority considers that the example belongs to the target class, then it is labeled *normal*, otherwise it is marked *novelty.*

Many statistic measures are taken throughout this process to ensure a good analysis of the results. A desired situation is one where the classifier is able to detect new profiles with high accuracy, but continues to classify normal examples with a good level of confidence. In other words, the aim is to minimize the false negative and false positive rates. However, this optimum point is not easily achieved, once some classifiers might be more restrictive than others in the definition of the normal profile.

Therefore, the major motivation for the combined approach is the belief that, when the opinions of more than one classifier are considered, the undesirable individual tendencies toward a specific class will be less important in the whole picture, since the final decision is taken by the majority. By doing so, it is expected that the optimum point described previously will be more easily achieved.

Previously, initial good results, not reported here, have been obtained with various standard datasets from the UCI Machine Learning Repository [4]. These results inspired a series of experiments carried out with gene expression data, presented in the following section.

3 Experiments

The main goal of the experiments described in this section is to compare the individual ND performance of each one-class classifier against the performance of the combined approach described previously. All classifiers used are available in DDtools, the Data Description Toolbox for Matlab [13], and this technique has been previously tested on standard datasets from the UCI Machine Learning Repository [4].

The experiments presented in this section have been conducted with the following gene expression datasets:

- breast - Classification of breast tumor samples based on the positive or negative status of the estrogen receptor (ER) [14]. The database is composed of 44 examples with 7129 attributes each.
- colon - Distinction between tumor and normal colon tissue samples based on gene expression [2]. The original database is composed of 62 examples and 2000 attributes.
- leukemia - Identification of two types of Leukemia (ALL and AML) from values of gene expression [7]. The original database contains 72 examples and 7129 attributes.
- lymphoma - Distinction between germinal center and activated diffuse large B-cell lymphoma based on gene expression profiling [1]. The original database is composed of 47 examples and 4026 attributes.

Throughout the analysis, classes are referred with numbers instead of labels, according to the association shown in Table 1.

Table 1. Classes numbers

Base	**Class 1**	**Class 2**
breast	ER-	ER+
colon	Tumor	Normal
leukemia	ALL	AML
lymphoma	Germinal Center	Activated

3.1 Methodology

Stratified 10-fold cross-validation has been used in all experiments to ensure that results represent the average behavior, not a specially successful or unsuccessful case. The same folds were used in all experiments to allow replicability.

According to the number of incorrect predictions, two error rates were calculated: the normal error rate, that considers examples of the normal profile incorrectly classified as outliers, and the novelty error rate, which indicates the percentage of outliers that have been incorrectly considered members of the normal profile. The results obtained are presented and discussed as follows.

3.2 Analysis of the Results

Initially, experiments were performed with 2 original datasets and a set of 5 classifiers: Parzen Window, K-NN, K-Means, SOM and PCA. Table 2 presents these results. In each cell, the mean error rate of the 10 folds tested is followed by the standard deviation. These statistics are available for each classifier alone, and for the combined approach. As previously explained, for all datasets, each class has been considered the normal profile at a time. For example, when class 1 is the normal profile, examples of class 2 are not present in the training phase. In fact, class 2 represents the novelty that the classifier is supposed to identify in the testing phase. Then, the same procedure is carried out considering class 2 as the normal profile.

The first aspect to notice in the results is the poor performance of all classifiers. In general, they consider almost all test examples as being either normal (very high novelty error rate) or novelty (very high normal error rate). For instance, when the Parzen Window technique obtains a novelty error rate equal to 1.00 and a normal error of 0.00, it means that it is classifying all test samples as normal, which is completely inadequate. The opposite is seen with the SOM technique in the *lymphoma* dataset, with normal error rates as high as 0.87. Neither one nor the other behavior is useful, and each shows that the classifier has not been able to estimate the distribution of the data. This situation, i.e. where all data are either considered normal or novelty, can be caused, among other things, by a classifier that is either inadequate for that particular data domain or badly configured. However, in this specific situation, a very high number of attributes (2000 in the *colon* dataset and 4026 in *lymphoma*) could also be the complicating factor.

Table 2. Results with 2 original datasets and a set of 5 classifiers. In each cell, the mean error rate is followed by the standard deviation

Base: colon		Normal Error		Novelty Error	
Normal Class: 1	parzen	0.00	0.00	1.00	0.00
	knn	0.10	0.17	0.95	0.16
	kmeans	0.15	0.17	1.00	0.00
	som	0.15	0.17	1.00	0.00
	pca	0.18	0.26	0.95	0.16
	Combined	**0.13**	**0.18**	**1.00**	**0.00**
Normal Class: 2	parzen	0.00	0.00	1.00	0.00
	knn	0.10	0.21	0.63	0.27
	kmeans	0.13	0.22	0.65	0.27
	som	0.13	0.22	0.63	0.27
	pca	0.20	0.26	0.50	0.26
	Combined	**0.10**	**0.21**	**0.63**	**0.27**
Base: lymphoma		Normal Error		Novelty Error	
Normal Class: 1	parzen	0.00	0.00	1.00	0.00
	knn	0.10	0.21	0.67	0.29
	kmeans	0.18	0.24	0.75	0.27
	som	0.87	0.22	0.00	0.00
	pca	0.27	0.24	0.50	0.34
	Combined	**0.13**	**0.22**	**0.67**	**0.29**
Normal Class: 2	parzen	0.00	0.00	1.00	0.00
	knn	0.08	0.18	0.80	0.26
	kmeans	0.17	0.22	0.80	0.26
	som	0.85	0.24	0.03	0.11
	pca	0.25	0.36	0.50	0.42
	Combined	**0.17**	**0.22**	**0.80**	**0.26**

To investigate that, a preprocessing phase has been added. In that phase, the number of attributes has been reduced to a calculated optimum amount, different for each dataset, based on the same technique used in [7]. This procedure aimed to minimize the error rates of ND. As a positive side effect, it also largely reduced the computational cost.

Table 3 shows the results after attribute reduction, with the same set of classifiers seen previously in Table 2. The *colon* dataset has been reduced to *colon16*, with 16 attributes, and the *lymphoma* dataset has been reduced to *lymphoma32*, with 32 attributes. With a few exceptions, the majority of the error rates decreased, which confirms that the high dimensionality of the original dataset did not allow the induction of reliable ND classifiers. This table also includes results obtained from 2 other reduced datasets, *breast128* and *leukemia64*, with 128 and 64 input attributes respectively.

In this second round of experiments, K-NN and K-Means achieved low error rates, except for the novelty class of *colon16*, which is known to be a difficult dataset. The PCA based classifier obtained good results on all datasets, even for the *colon16* dataset, when the normal examples belong to class number 1. Unfortunately, in that case, most of the classifiers were not as successful. Parzen Window was the worse of all classifiers, displaying the same behavior seen previously in Table 2. However, this negative influence did not have a strong impact on the overall performance of the combined approach.

Table 3. Results with the 4 reduced datasets and 5 classifiers

Base: breast128		Normal Error		Novelty Error	
Normal Class: 1	parzen	0.00	0.00	1.00	0.00
	knn	0.15	0.34	0.03	0.11
	kmeans	0.18	0.34	0.00	0.00
	som	0.18	0.34	0.00	0.00
	pca	0.18	0.34	0.00	0.00
	Combined	**0.15**	**0.34**	**0.00**	**0.00**
Normal Class: 2	parzen	0.00	0.00	1.00	0.00
	knn	0.15	0.24	0.13	0.22
	kmeans	0.15	0.24	0.00	0.00
	som	0.15	0.24	0.00	0.00
	pca	0.18	0.24	0.00	0.00
	Combined	**0.15**	**0.24**	**0.00**	**0.00**
Base: colon16		Normal Error		Novelty Error	
Normal Class: 1	parzen	1.00	0.00	0.00	0.00
	knn	0.10	0.17	0.50	0.34
	kmeans	0.13	0.13	0.53	0.34
	som	0.15	0.13	0.48	0.30
	pca	0.18	0.26	0.27	0.24
	Combined	**0.18**	**0.17**	**0.40**	**0.33**
Normal Class: 2	parzen	1.00	0.00	0.00	0.00
	knn	0.10	0.21	0.85	0.17
	kmeans	0.15	0.24	0.68	0.24
	som	0.15	0.24	0.63	0.27
	pca	0.18	0.24	0.63	0.18
	Combined	**0.15**	**0.24**	**0.65**	**0.21**
Base: leukemia64		Normal Error		Novelty Error	
Normal Class: 1	parzen	1.00	0.00	0.00	0.00
	knn	0.12	0.17	0.07	0.14
	kmeans	0.09	0.11	0.03	0.11
	som	0.11	0.11	0.03	0.11
	pca	0.14	0.13	0.03	0.11
	Combined	**0.13**	**0.11**	**0.03**	**0.11**
Normal Class: 2	parzen	1.00	0.00	0.00	0.00
	knn	0.07	0.21	0.47	0.24
	kmeans	0.08	0.18	0.14	0.19
	som	0.13	0.22	0.17	0.22
	pca	0.28	0.35	0.09	0.15
	Combined	**0.17**	**0.22**	**0.14**	**0.19**
Base: lymphoma32		Normal Error		Novelty Error	
Normal Class: 1	parzen	1.00	0.00	0.00	0.00
	knn	0.08	0.18	0.12	0.19
	kmeans	0.13	0.32	0.05	0.16
	som	1.00	0.00	0.00	0.00
	pca	0.22	0.24	0.28	0.26
	Combined	**0.30**	**0.32**	**0.05**	**0.16**
Normal Class: 2	parzen	1.00	0.00	0.00	0.00
	knn	0.15	0.34	0.23	0.34
	kmeans	0.15	0.24	0.00	0.00
	som	0.97	0.11	0.00	0.00
	pca	0.18	0.24	0.00	0.00
	Combined	**0.23**	**0.34**	**0.00**	**0.00**

The SOM technique only showed difficulty in the *lymphoma32* dataset. However, even with two classifiers providing totally misleading results, the effect on the performance of the combined approach in the *lymphoma32* dataset was little. This shows superior robustness of the combined approach against the choice of a single classification strategy.

Table 4. Results with a set of 3 classifiers, one from each strategy

Base: breast128		Normal Error		Novelty Error	
Normal Class: 1	parzen	0.00	0.00	1.00	0.00
	knn	0.15	0.34	0.03	0.11
	kmeans	0.18	0.34	0.00	0.00
	Combined	**0.15**	**0.34**	**0.03**	**0.11**
Normal Class: 2	parzen	0.00	0.00	1.00	0.00
	knn	0.15	0.24	0.13	0.22
	kmeans	0.15	0.24	0.00	0.00
	Combined	**0.15**	**0.24**	**0.13**	**0.22**
Base: colon16		Normal Error		Novelty Error	
Normal Class: 1	parzen	1.00	0.00	0.00	0.00
	knn	0.10	0.17	0.50	0.34
	kmeans	0.13	0.13	0.53	0.34
	Combined	**0.20**	**0.20**	**0.40**	**0.33**
Normal Class: 2	parzen	1.00	0.00	0.00	0.00
	knn	0.10	0.21	0.85	0.17
	kmeans	0.15	0.24	0.68	0.24
	Combined	**0.15**	**0.24**	**0.68**	**0.24**
Base: leukemia64		Normal Error		Novelty Error	
Normal Class: 1	parzen	1.00	0.00	0.00	0.00
	knn	0.12	0.17	0.07	0.14
	kmeans	0.09	0.11	0.03	0.11
	Combined	**0.19**	**0.15**	**0.03**	**0.11**
Normal Class: 2	parzen	1.00	0.00	0.00	0.00
	knn	0.07	0.21	0.47	0.24
	kmeans	0.08	0.18	0.14	0.19
	Combined	**0.15**	**0.25**	**0.14**	**0.19**
Base: lymphoma32		Normal Error		Novelty Error	
Normal Class: 1	parzen	1.00	0.00	0.00	0.00
	knn	0.08	0.18	0.12	0.19
	kmeans	0.13	0.32	0.05	0.16
	Combined	**0.22**	**0.33**	**0.05**	**0.16**
Normal Class: 2	parzen	1.00	0.00	0.00	0.00
	knn	0.15	0.34	0.23	0.34
	kmeans	0.15	0.24	0.00	0.00
	Combined	**0.20**	**0.35**	**0.00**	**0.00**

To assess the impact of the number of classifiers in the set on the final results, experiments were also performed with a set of 3 classifiers, one representing each of the ND strategies (density estimation, boundary and reconstruction).

The results, displayed in Table 4, show a small increase in the error rates in the *breast128* and *colon16* datasets. On the other hand, for the *lymphoma32* dataset there was a small reduction in the error rates. For the *leukemia64* dataset the results were similar. Considering that the number of classifiers was reduced from 5 to 3, and that the relative influence of the Parzen Window on the overall result was increased, the performance of the combined approach has not been seriously affected.

A different set, in which Parzen Window has been replaced by PCA, has also been tested and the results are presented in Table 5. In most cases, the performance has been improved. However, although Parzen Window, which apparently has not shown any contribution to the combined result, has been replaced in the combination by PCA, a technique which has shown superior performance, the impact on the combined result was not as high as could be expected. In fact, this stability indicates the potential of the combined approach. With the combination, extremes

Table 5. Results with another set of 3 classifiers

Base: breast128		Normal Error		Novelty Error	
Normal Class: 1	knn	0.15	0.34	0.03	0.11
	kmeans	0.18	0.34	0.00	0.00
	pca	0.18	0.34	0.00	0.00
	Combined	**0.15**	**0.34**	**0.00**	**0.00**
Normal Class: 2	knn	0.15	0.24	0.13	0.22
	kmeans	0.15	0.24	0.00	0.00
	pca	0.18	0.24	0.00	0.00
	Combined	**0.15**	**0.24**	**0.00**	**0.00**
Base: colon16		Normal Error		Novelty Error	
Normal Class: 1	knn	0.10	0.17	0.50	0.34
	kmeans	0.13	0.13	0.53	0.34
	pca	0.18	0.26	0.27	0.24
	Combined	**0.15**	**0.17**	**0.40**	**0.33**
Normal Class: 2	knn	0.10	0.21	0.85	0.17
	kmeans	0.15	0.24	0.68	0.24
	pca	0.18	0.24	0.63	0.18
	Combined	**0.10**	**0.21**	**0.73**	**0.18**
Base: leukemia64		Normal Error		Novelty Error	
Normal Class: 1	knn	0.12	0.17	0.07	0.14
	kmeans	0.09	0.11	0.03	0.11
	pca	0.14	0.13	0.03	0.11
	Combined	**0.06**	**0.10**	**0.03**	**0.11**
Normal Class: 2	knn	0.07	0.21	0.47	0.24
	kmeans	0.08	0.18	0.14	0.19
	pca	0.28	0.35	0.09	0.15
	Combined	**0.07**	**0.14**	**0.14**	**0.19**
Base: lymphoma32		Normal Error		Novelty Error	
Normal Class: 1	knn	0.08	0.18	0.12	0.19
	kmeans	0.13	0.32	0.05	0.16
	pca	0.22	0.24	0.28	0.26
	Combined	**0.13**	**0.22**	**0.12**	**0.19**
Normal Class: 2	knn	0.15	0.34	0.23	0.34
	kmeans	0.15	0.24	0.00	0.00
	pca	0.18	0.24	0.00	0.00
	Combined	**0.15**	**0.24**	**0.00**	**0.00**

can be avoided and, consequently, the robustness of the system as a whole can be improved. Although the best possible results may not be achieved, unstable situations in which a classification technique favors one specific profile over the other can be avoided, i.e. normal over novelty or novelty over normal. As mentioned previously, this is an important issue when dealing with one-class classification, since the challenge is to identify new profiles with a high level of confidence while maintaining a good performance on the normal profile.

Finally, to provide a better visualization of the decisions taken throughout the process, individual errors made by each classifier on each example of the test set have been recorded for each fold and later reassembled. Figure 1 displays those errors in a graphical format, where white squares represent examples correctly classified and black squares mark errors. Examples are placed along the horizontal axis and classifiers vertically.

It is easily noticed that the larger number of errors is concentrated in the dataset *colon16* when the second class represents the normal profile. In this situation, all classifiers except Parzen Window make similar mistakes, which can also be confirmed by the error rates displayed in Table 3.

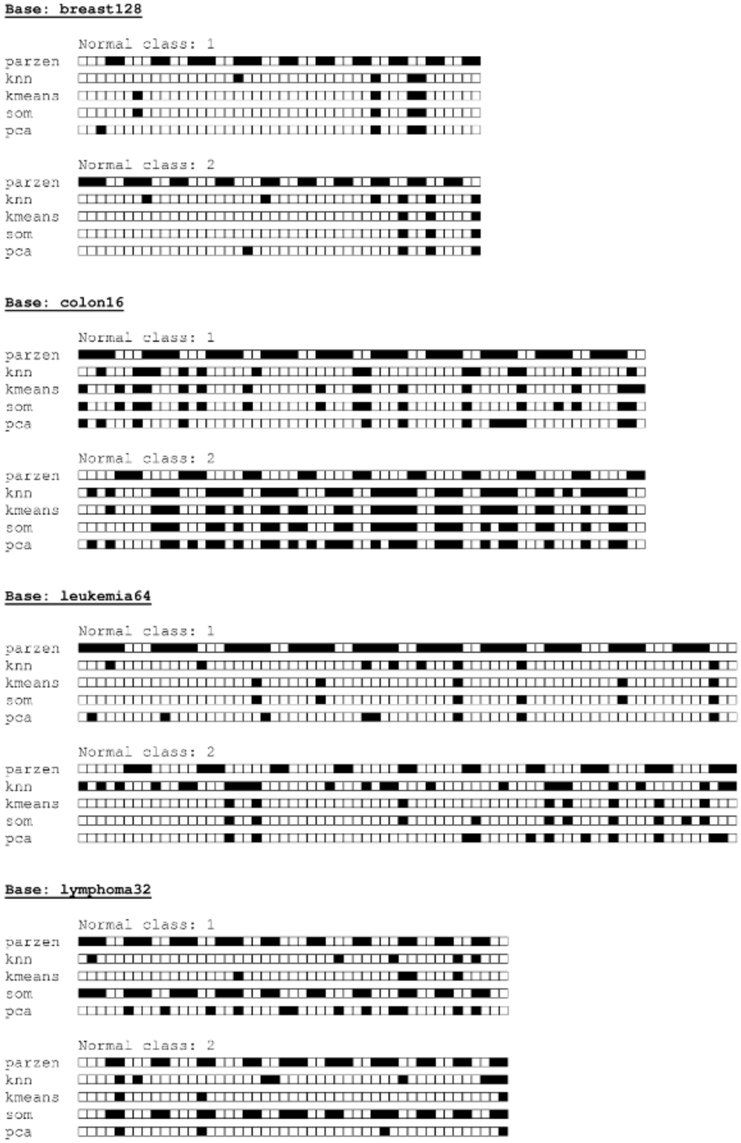


Fig. 1. Individual errors (black squares) of each classifier (vertical axis) on each example (horizontal axis)

Through these graphs it is also clearer to see that a classification strategy that shows good results in one dataset might not be successful in another, even considering datasets of the same domain (gene expression). For example, the horizontal lines which represent the performance of the SOM technique display a very different amount of classification mistakes, depending on the dataset and on the normal class considered. This picture reinforces that the combination of classifiers leads to more robust results, since the decision is always taken by the majority of them.

An example of a desired situation is shown in the first plot of *leukemia64* and in both plots of *breast128*, where low error rates have been achieved. The vertical alignment of the errors indicate that all classifiers are having similar difficulties. These are problematic examples, which can be further investigated with a different technique, or isolated to be analyzed by a specialist.

However, if a larger number of scattered errors is present, the final performance of the combined approach might still be good. This is due to the fact that each classifier is filling-in other classifiers faults, which exemplifies the importance to combine classifiers built with various techniques, since the diversity of classifiers in the set may determine the robustness of the system as a whole.

4 Conclusion

One-class classification techniques are able to, based only on examples of a normal profile, induce a classifier that is capable of detecting novelty.

This paper has shown the use of a simple strategy which combines the opinions of a set of one-class classifiers for the task of ND in gene expression data. The results obtained suggest that the use of such a combined approach improves the robustness of the overall decision. By considering the opinion of the majority of a set of classifiers instead of just one, this technique avoids individual tendencies that certain approaches might present in some datasets or domains.

The improvement achieved so far inspire further investigations. As analyzed, the diversity of classifiers in the decision set seems to be an important aspect in the final performance of the combined approach. Another possible way of improving the results might be the addition of a selection phase, after which only the ND approaches that better fit the problem at hand would be considered. An assessment of the impact of both technical and biological noise on the differential performance of the classifiers has been suggested, and also inspires further experimentation.

Still, other combinations of one-class classifiers are yet to be explored in bioinformatics, following previous initiatives [12], as the authors continue to explore ND techniques for the identification of novel classes and profile changes.

Acknowledgement

The authors wish to thank CNPq for the financial support and Bruno Feres de Souza for the preprocessing of datasets [5].

References

1. A. A. Alizadeh, M. B. Eisen, R. E. Davisintegral, C. Maintegral, I. S. Lossos, A. Rosenwaldintegral, J. C. Boldrick, H. Sabetintegral, T. Tranintegral, X. Yuintegral, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. H. Jr, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.
2. U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of National Academy of Sciences USA*, 96:6745–6750, 1999.
3. C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
4. C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.
5. B. F. de Souza. Seleção de características para svms aplicadas a dados de expressão gênica. Master thesis, Universidade de São Paulo (USP), Instituto de Ciências Matemáticas e de Computação (ICMC), 2005.
6. R. O. Duda and P. E. Hart. *Pattern Classification.* Wiley Interscience, 2nd edition, 2001.
7. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression. *Science*, 286:531–537, 1999.
8. T. Kohonen. *Self-Organizing Maps.* Springer, 3rd edition, 2001.
9. S. Marsland. Novelty detection in learning systems. *Neural Computing Surveys*, 3:157–195, 2003.
10. E. Parzen. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.
11. B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
12. D. M. J. Tax. *One-class classifiers.* PhD thesis, Delf University of Technology, Faculty of Information Technology and Systems, 2001.
13. D. M. J. Tax. DDtools, the data description toolbox for matlab. `http://www-ict.ewi.tudelft.nl/~davidt/dd_tools.html`, March 2005. version 1.1.2.
14. M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. O. Jr., J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of National Academy of Sciences USA*, 98(20).

Evaluation of the Contents of Partitions Obtained with Clustering Gene Expression Data

Katti Faceli[1], André C.P.L.F. de Carvalho[1], and Marcílio C.P. de Souto[2]

[1] Universidade de São Paulo,
Instituto de Ciências Matemáticas e de Computação,
Departamento de Ciências de Computação e Estatística,
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
{katti, andre}@icmc.usp.br
[2] Universidade Federal do Rio Grande do Norte,
Departamento de Informática e Matemática Aplicada - DIMAp
Campus Universitario, 59072-970 - Natal, RN, Brazil
marcilio@dimap.ufrn.br

Abstract. This work investigates the behavior of two different clustering algorithms, with two proximity measures, in terms of the contents of the partitions obtained with them. An analysis of how the classes are separated by these algorithms, as different numbers of clusters are generated, is also presented. A discussion on the use of these information in the identification of special cases for further analysis by biologists is presented.

1 Introduction

Nowadays, gene expression data consists of an important source of information for the understanding of biological processes and diseases mechanisms. Clustering methods are one of the most important tools to support biologists in the analysis of gene expression data. As pointed out by [1], this type of analysis is of increasing interest in the field of functional genomics and gene expression data analysis. One of its motivation is the need for molecular-based refinement of broadly defined biological classes, with implications in cancer diagnosis, prognosis and treatment [1].

There is a huge diversity of clustering techniques described in the literature. Some of them have been employed to gene expression data. Examples are k-means [2], Self-Organizing Maps (SOM) [2], Self-Organizing Tree Algorithm (SOTA) [3] and the hierarchical clustering algorithms [2]. In this paper, k-means and SOTA, with both the Euclidean distance and Pearson correlation, are employed to generate a set of partitions (clusterings). Based on the partitions generated, two types of analysis are developed. First, a high level evaluation and comparison of the quality of the partitions are accomplished. For such, two different validation approaches are used: external validation employing the corrected Rand index [4] and the analysis of the variability of the algorithms by bootstrapping.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 65–76, 2005.

The second type of analysis, which is the main focus of this work, is a finer study of the partitions obtained. More precisely, the best partitions according to the evaluation process from the first step have their contents analyzed in detail. A further analysis of the contents of each cluster in the partitions can bring important insights to the biologists. For example, this analysis can show patterns (samples or genes) that have a different behavior from that expected. These patterns could represent interesting cases to have a detailed investigation in the laboratory.

Furthermore, the analysis of partitions with different numbers of clusters can help in the identification of new subgroups in the data, when main groups are already known, as in the case of cancer classes. This can lead to the discovery of new classes, or subclasses, of cancer. The discovery of new classes of cancer is an issue that has received strong attention recently. Other possible contributions to biologists are discussed in Sect. 5.

2 Experiments

The experiments were carried out by applying two clustering algorithms, k-means and SOTA, to the dataset St. Jude leukemia [5, 1]. This dataset has a multi-class distinction (a phenotype) that will be considered as the gold standard partition, referred also as the true partition of the dataset. Following the conversion used in [1], the groups stated by the gold standard partition are referred as classes, while the notation cluster is reserved for the groups returned by the clustering algorithms.

For the detailed analysis described in Sect. 4, the class label associated to each pattern should be known, otherwise the coloring scheme cannot be applied.

This dataset consists of 248 diagnostic bone marrow samples from pediatric acute leukemia patients corresponding to six prognostically important leukemia subtypes. Each sample is composed of the expression values of 985 genes. Table 1 shows the classes and the number of patterns (samples) of each class present in the dataset. For short, the notation in parenthesis will be employed in the text, when it is the case. In the experiments, the samples were the patterns to be clustered and the genes were their attributes.

Table 1. Classes present in the dataset

Class	Number of patterns
BCR-ABL (BCR)	15
E2A-PBX1 (E2A)	27
‘hyperdiploid>50’ (hyperdip)	64
MLL	20
T-lineage ALL (T-ALL)	43
TEL-AML1 (TEL)	79

The experiments consisted of the generation of partitions having from 2 to 15 clusters, employing k-means and SOTA algorithms with the Euclidean and Pearson proximity measures. This range was chosen because the true number of clusters is six and having as a reference the work in [1], which also investigated such a range for this dataset.

K-means is one of the most traditional clustering algorithms [4]. It is a partitional algorithm that partition the dataset in a predefined number of clusters. In this work, k-means has been chosen as a reference, since it is widely employed in a number of applications, including gene expression analysis. In contrast to partitional features of the k-means, SOTA is a hierarchical divisive algorithm, based in the neural networks Self Organizing Maps (SOM) and Growing Cell Structures (GCS). It is a neural network that grows adopting the topology of a binary tree. Some of the main characteristics of this algorithm, desirable for gene expression data analysis, are its ability in dealing with high-dimensional data, scalability, robustness against noise and outliers and independence from the order of data presentation.

The experiments carried out with SOTA employed default values for the parameters, except for the maximum number of cycles (*max*). This parameter determines the number of clusters to be generated ($max + 1$ clusters). The value of *max* varied from 1 to 14 (2 to 15 clusters). Although SOTA can automatically determine the best number of clusters, the authors forced the algorithm to generate the partitions with the specific numbers of clusters that were being studied. The other parameters of SOTA are the variability and resource thresholds, that define the convergence of the network (default value of 0 for both parameters), the relative error threshold, that defines the convergence of a cycle (default value of 0.0001) and the actualization factors for the winning, mother and sister nodes (default values of 0). Other values for these parameters were not investigated, since the interest were not in the best adjustment of SOTA, but in the comparison among different numbers of clusters in different algorithms and similarity measures. For k-means, the only parameter of the algorithm is the number of clusters, that was varied from 2 to 15.

The algorithm k-means generate different partitions for the same dataset and number of clusters, depending on the random initialization of the centroids. SOTA generates the same partition for a specified number of clusters and just breaks the clusters as a higher number of clusters is specified.

The performance of a clustering method for gene expression data analysis depends on the employment of an appropriate proximity function, according to the properties the researcher wants to focus. As the interest of the authors are in looking for all potentially interesting groups in a dataset, two different proximity measures commonly employed to gene expression data clustering were employed: Euclidean distance and Pearson coefficient [2]. The Euclidean distance measures the absolute distance between two points in an n-dimensional space. According to this metric, similar patterns exhibits the same magnitude and direction. The Pearson correlation coefficient (linear correlation) measures the angular separation of the patterns around their mean. This metric is usually described as a measure of the shape, as it is insensitive to differences in the magnitude of the attributes.

In the following sections, the experiments will be represented by three components. The first one is a letter representing the algorithm: K for k-means and S for SOTA. The second component is also a letter representing the proximity measure employed: E for Euclidean distance and P for Pearson correlation. The last component is the number of clusters generated. For example, the experiment employing the k-means and the Euclidean distance, generating six clusters will be represented by KE6.

3 High Level Evaluation

In this paper, the validation of the results was accomplished by means of two different approaches: external validation employing the corrected Rand index [4, 6] and the analysis of the variability of the algorithms by bootstrapping [7]. The first approach aims to assess how good the clustering techniques investigated are at recovering known clusters. This was performed by using the corrected Rand index (CR for short). In this context, the authors also checked if the partitions generated are valid. A partition can be considered valid, for example, if the value of its CR index is unusually high, according to a reference distribution [4]. In order to do so, the authors followed the procedure described in [6], but employing bootstrap samples as if they were a replication of a Monte Carlo experiment [4]. The number of bootstrap samples, B, considered in this paper was set to 100.

CR measures the agreement between the true partition (the gold standard) and the clustering generated by an algorithm. It can take values from -1 to 1, with 1 indicating a perfect agreement between the partitions, and the negatives or near 0 values corresponding to cluster agreements found by chance.

The other validation approach employed in this paper also uses bootstrapping, but to analyze the variability of each clustering algorithm [7]. The

Table 2. Variablilty and Corrected Rand for the best partitions

Five Best CR			Five Best V_{adj}		
Partition	V_{adj}	CR	Partition	V_{adj}	CR
KP5	0.254802	0.852346	SP3	0.181361	0.287574
KE4	0.260567	0.829675	SP4	0.186054	0.255003
KP6	0.267859	0.829643	KP2	0.191907	0.217778
KP5	0.234667	0.805082	SE3	0.194405	0.380157
SP11	0.204392	0.796235	SE4	0.198795	0.340644

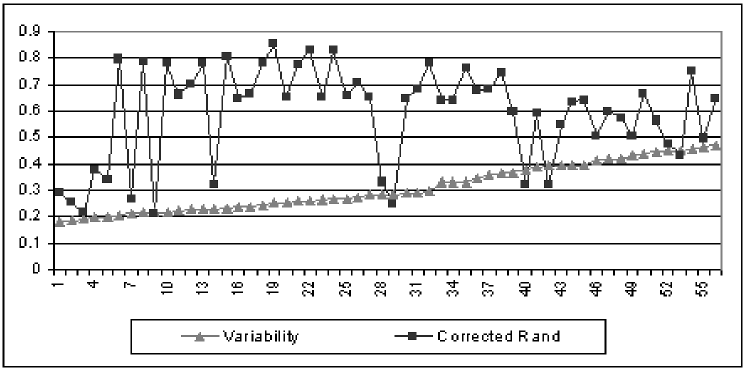


Fig. 1. Variability and Corrected Rand for all partitions generated

variability can be used, for instance, to compare partitions produced by different algorithms, by an algorithm with different parameters values or by an algorithm employing different proximity measures. Such approach sees a clustering algorithm as a point estimator (as in statistical theory) for the partition of the data space and uses bootstrapping to estimate the variability of the estimator. In this context, if the partition is valid, the variability should be low. In order to apply this validation, $B = 100$ bootstrap samples were also generated. The algorithm was run on each sample obtaining a set of partitions. The variability, V, was estimated using CR to calculate the distance between two partitions. Afterwards, 100 random partitions were generated and the variability on them, V_{ran} was also calculated. Finally, the adjusted variability V_{adj} was calculated by $V_{adj} = V/V_{ran}$. V_{adj} is the variability value employed to compare the partitions in the analysis that follows.

Each validation strategy employed led to different best partitions. The five best results according to each strategy are shown in Table 2 - all partitions obtained in the experiments with the external validation employing the corrected Rand index were found to be valid with a significance level of 0.05. Figure 1 is a plot of the values of CR and variability for all partitions obtained, ordered by their variability. Some interesting observations can be made from Table 2 and Fig. 1. Partitions presenting the lowest (best) variabilities show very poor quality according to CR. Variability favors small number of clusters. On the other hand, the best partitions according to CR were obtained for numbers of clusters close to the true number of clusters, six. The partitions presenting high CR values show variabilities slightly above the best variability values obtained. It can be observed that for the 6th to 38th variability values shown in the graphic, most of the corresponding values of corrected Rand lied above 0.6. It was observed that k-means presented the best results according to CR and SOTA showed the best results according to the variability.

4 Partitions Evaluation

This section takes a closer look at the partitions produced in order to evaluate the composition of the clusters and the influence of the clustering algorithms and the number of clusters used. This analysis also considers the true known structure of the dataset (classes).

In order to facilitate the analysis, a coloring scheme was applied to each partition. The first step in the procedure of coloring the partitions was the assignment of a color to each class. Next, the number of patterns from each class present in each cluster for a given partition was determined. Based on this information, the predominant class for each cluster was found (the class that presents more patterns in the cluster). Next, each cluster was labeled with the color of its predominant class. An intensity was also assigned to each cluster, aiming to distinguish the clusters with the same predominant class. An intensity of 0 was assigned to the cluster with the highest number of patterns from the predominant class, an intensity of 1 was assigned to that with the second highest number of patterns of the predominant class, and so on.

With the clusters colored, the partitions to be compared were plotted side by side in a datasheet, with all partitions ordered by the pattern identifier. It should be noticed that the pattern identifier has an indication of the class to which the pattern belongs. Otherwise, an indication of the class should be added to the identifier. This representation associated with the coloring scheme make it possible to readily distinguish the patterns wrongly assigned to a cluster and the most homogeneous clusters.

For a preliminary analysis, the three best partitions of each validation strategy described (Sect. 3) were selected. As the 5th best partition, according to CR, was the 6th best partition, according to the variability, this partition was also selected (SP11). This first analysis originated a question: What does it happen with partitions with a higher number of clusters? Is it possible that a partition with a number of clusters much higher than the true number presents good clusters, together with clusters of poor quality? To check this possibility, the partitions with 15 clusters generated with both algorithms and proximity measures investigated were analyzed. Another issue investigated with 15 clusters was the existence of problematic patterns that can interfere with the clustering result. With a higher number of clusters, these patterns could be isolated, so that the other patterns could be grouped into more homogeneous clusters.

From the coloring scheme and the observation of the clusters contents, useful information was obtained, which is summarized in this section. Table 3 details the amount of patterns from each class in the clusters from the best partition according to CR (KP5), the best partition according to variability (SP3), the partition SP11, described previously and the partition SP15, that present the best CR value among the partitions with 15 clusters.

Table 4 contains a summary of the clusters generated in each experiment considered. The clusters were classified into four types: pure clusters (P), large well defined clusters (LWD), large mixed clusters (LM) and small mixed clusters (SM). The pure clusters contain patterns of only one class. Clusters with one

Table 3. Clusters contents for the best clustering

Partition	Cluster	Class					
		BCR	E2A	hyperdip	MLL	T-ALL	TEL
KP5	1						78
	5	13		62			1
	4					39	
	2	1	27		3		
	3	1		2	16	4	
SP3	3	14	23	62	4		79
	1					39	
	2	1	4	2	16	4	
SP11	10						79
	9	14	1	62	1		
	4					19	
	3					12	
	1					4	
	2					4	
	5					3	
	11		22		3		
	8			2	14	1	
	6	1					
	7		4		2		
SP15	14						79
	13	14	1	62	1		
	3					8	
	6					8	
	8					7	
	1					4	
	2					4	
	7					4	
	4					3	
	9					3	
	5					1	
	15		22		3		
	11		4		2		
	12			2	14	1	
	10	1					

single pattern are also considered pure. Large well defined clusters have the majority of the patterns from the predominant class and just few patterns from other classes. Large mixed clusters have the majority of the patterns from 2 or more classes. Small mixed clusters contain few patterns from more than one class. The table included the number of each type of cluster and, when appropriate, the predominant class of each cluster (in the case of LMC, the classes with a large number of patterns in the cluster).

Table 4. Main structure of the clusters of each clustering

Partition	P	LWD	LM	SM
KP5	2 (T-ALL, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	0
KE4	1 (T-ALL)	1 (TEL)	2 (hyperdip+BCR, E2A+MLL)	0 0
KP6	2 (T-ALL, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	1
SP3	1 (T-ALL)	1 (MLL)	1 (hyperdip+BCR+ E2A+TEL)	0 0
SP4	2 (2 T-ALL)	1 (MLL)	1 (hyperdip+BCR+ E2A+TEL)	0 0
KP2	0	1 (T-ALL)	1 (hyperdip+BCR+ E2A+TEL+MLL)	0 0
SP11	7 (5 T-ALL, BCR, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	1
KE15	9 (5 T-ALL, hyperdip, BCR, MLL, TEL)	5 (2 E2A, BCR, 2 TEL)	0	1
KP15	8 (3 T-ALL, 3 TEL, E2A, hyperdip)	3 (MLL, E2A, hyperdip)	1 (hyperdip+BCR)	3
SE15	9 (8 T-ALL, hyperdip)	6 (MLL, E2A, TEL, hyperdip, 2 BCR)	0	0
SP15	11 (9 T-ALL, BCR, TEL)	2 (MLL, E2A)	1 (hyperdip+BCR)	1

Table 5 shows the number of patterns assigned to a large cluster of another class (wrong assignment), the number of patterns assigned to the small mixed clusters and the number of patterns assigned to small pure clusters (with less than 5 patterns in the cluster), in each clustering analyzed. The patterns in the pure and small mixed clusters are better seen by looking at the clusters composition in Table 3.

Some conclusions can be drawn from the analysis of these data. First, patterns from each class were represented mostly with the same color, but in some cases with different intensities. This means that, even when the patterns from a class were separated in different clusters, they usually were assigned to clusters with the same predominant class. This was also true in the analysis of the partitions with 15 clusters (the highest number of clusters investigated). Even for the partitions with fewer clusters, most of the patterns of each class tended to appear together in the same cluster, even when the clusters were composed of different classes (LM). These were the cases of KP4 and KP2, which presented few wrong assignments due to the large mixed clusters that placed most of the patterns from several classes together.

The best partition according to CR (KP5) generated two pure clusters, two well defined clusters and one mixed cluster (BCR + hyperdip). This is a good partition, but it did not separate the classes hyperdip and BCR. Looking at the partitions of 15 clusters, most of them can separate all classes, including BCR and hyperdip (KE15, KP15 and SE15). The partition KE15 did not generate large mixed clusters and generated just one small mixed cluster, with just four patters. Although the number of clusters was large, the clusters obtained were homogeneous. This partition can also be considered a good partition, in spite of its relatively low value of CR and high variability.

Almost all patterns from the class TEL always grouped together. There were just few cases in some of the partitions where a pattern from this class was associated to another cluster. The TEL patterns also appeared well separated from the other classes, except for the cases where few clusters were generated. The patterns from the class T-ALL formed a well separated cluster too. The algorithm SOTA tended to divide the patterns from the class T-ALL in several small sub-clusters before separating the patterns of the classes TEL, hyperdip-BCR and E2A. This was observed by looking at the clusters of T-ALL for the partitions of three and four clusters generated by SOTA with the Pearson correlation. This trend was confirmed by the analysis of the partitions with 15 clusters, where eight or nine small pure clusters of the class T-ALL were formed.

The patterns from the classes BCR and hyperdip were almost always grouped together in the same cluster. Even when there were clusters with the predominant class BCR and clusters with the predominant class hyperdip, most of these clusters still presented patterns from both classes (BCR or hyperdip). As there are 6 classes, the best solution of 6 clusters found (generated with the algorithm k-means with Pearson - KP6) was analyzed with more attention to compare the clusters with the true classes. This partition did not separate the patterns from the classes BCR and hyperdip, as the other partitions containing a smaller number of clusters. This partition presented a large cluster with most of the hyperdip and BCR patterns and a small cluster containing the other few hyperdip and BCR patterns together with patterns from three other classes The other clusters were similar to those obtained using a smaller number of clusters.

A question arises from the observation of the result obtained in the partition with 6 clusters, KP6. Does the generation of two clusters mixing BCR and hyperdip can indicate that if more clusters were generated, this classes could be separated? It was observed that when a large number of clusters were generated (11 or 15), small pure cluster started to appear. Also, in the analysis of the partitions with 15 clusters, pure clusters of hyperdip, clusters with almost all patterns belonging to this class and clusters with some hyperdip samples, but with the majority of the patterns belonging to the BCR class were found. Such results confirmed that, although the classes hyperdip and BCR are very similar, they have differences that can be found in some way (in this case, generating a higher number of clusters). This observations were valid for both algorithms, SOTA, which generated the same partition for a specified number of clusters, and k-means, which generated a different partition in each run. It should also be observed that the best partition of 15 clusters, according to CR, did not separate the classes BCR and hyperdip, as the other partitions with 15 clusters do. Both partitions of 15 clusters obtained with k-means presented several pure clusters. In the case of SOTA, the class T-ALL was divided into several small pure clusters. Three of these four partitions with 15 clusters separated the class BCR from hyperdip.

Some heterogeneous clusters very similar in many of the partitions analyzed were found. One of this clusters was composed of the majority of the hyperdip and BCR patterns. Another similar case was the cluster composed of 16 MLL patterns, one BCR, two hyperdip and a few other patterns of other classes. This can indicate that the patterns wrongly assigned to this clusters found in all cases are really more similar to the patterns in this clusters than to those of their class, and that the wrong assignments did not occur just because of the variability of the algorithms. Maybe these patterns were either wrongly labeled or contained important information to be investigated, as they should be more similar to patterns from their class.

Other observation is that there were some patterns that were assigned to the same wrong cluster in most of the partitions analyzed. This is the case for the patterns "hyperdip.50.7" and "hyperdip.50.C19", almost always assigned to clusters with the predominant class MLL. Other patterns were also assigned to a wrong cluster, but in only one or two partitions. Table 6 shows the patterns wrongly assigned to at least five partitions. In this table, for each pattern, only the columns of the partitions in which a wrong assignment occurred are marked. This "marking" is made with the predominant class of the cluster to which the pattern was wrongly assigned. For example, the pattern "BCR.ABL.R1" was wrongly assigned to the class E2A in the partition KE4 and to the class MLL in the partitions KP5, KP6, SP3, SP4 and SE15. These wrongly assigned patterns were easily identified with the coloring scheme as they were shown with a different color from the majority of the other patterns from the same class. It should be noticed that for the clusters that encompassed more than one class (the majority of the patterns from more than one class), the patterns from the classes well represented in the clusters were not considered wrong assignment. Thus, for example, in the cluster composed of hyperdip and BCR, neither BCR

Table 5. Number of patterns in each type of cluster

Partition	Wrong assignments	Small mixed	Small pure
KP5	13	0	0
KE4	5	0	0
KP6	13	13	0
SP3	15	0	0
SP4	15	0	0
KP2	7	0	0
SP11	8	6	12
KE15	18	4	6
KP15	16	16	0
SE15	21	0	13
SP15	8	6	20

Table 6. Assignments to wrong clusters in at least 2 clusterings

Pattern	KP5	KE4	KP6	SP3	SP4	KP2	SP11	KE15	KP15	SE15	SP15
BCR.ABL.R1	MLL	E2A	MLL	MLL	MLL					MLL	
Hyperdip.50.7	MLL	E2A	MLL	MLL	MLL	T-ALL	MLL	MLL		MLL	MLL
Hyperdip.50.C19	MLL	E2A	MLL	MLL	MLL		MLL	MLL		MLL	MLL
MLL.C3	E2A		E2A	TEL	TEL		hyp	E2A	E2A	E2A	hyp
MLL.C4	E2A		E2A	TEL	TEL		E2A	E2A	E2A	TEL	E2A
MLL.C5	E2A		hyp	TEL	TEL		E2A	E2A		hyp	E2A
MLL.C6	E2A		E2A	TEL	TEL		E2A	E2A	E2A	TEL	E2A
T.ALL.C5	MLL		MLL	MLL			MLL			MLL	MLL

nor hyperdip patterns were considered wrongly assigned. An assignment was considered an error only if the pattern was assigned to a cluster with only few or no other patterns from its class. The small mixed clusters were not considered wrong assignments either.

Different wrong assignments of a pattern can be due to mixed clusters as they encompasses a large amount of patterns of more than one class and a cluster has a predominant class when this class has more patterns in the cluster than the other classes. For example, "BCR.ABL.R1" was assigned to a cluster of the class E2A in the partition KE4 and to clusters of the class MLL in all other partitions where a wrong assignment occurred. Even when assigned to the cluster from the class E2A, "BCR.ABL.R1" was assigned to a cluster with many MLL patterns, as the cluster E2A is a large mixed cluster of E2A and MLL.

5 Conclusion

This paper investigated two different clustering algorithms and two proximity measures to obtain a series of partitions of a gene expression dataset. For each algorithm and proximity measure, partitions containing from 2 to 15 clusters were generated. Each validation strategy pointed out a different technique as superior. The k-means algorithm presented better results according to CR and SOTA according to variability. The best partitions obtained had their contents analyzed in details.

The analysis carried out in this work can provide useful insights to the area of gene expression analysis. The information outlined can be used to point out new directions for further analysis by biologists. The large mixed clusters can indicate unexpected similarities of the classes. The subdivisions of the classes in smaller clusters can indicate possible important subdivisions of the data, supporting the discovery of new disease subtypes (such as those of great interest in cancer research). The small heterogeneous clusters can have important meaning as they present patterns with different behavior from that expected. They could represent interesting samples that could be further analyzed in laboratory. The samples that were always classified in the same wrong cluster can be either just noisy samples, or can indicate an error in the original classification of these samples. Alternatively, they can occur because these samples really present an unexpected behavior, which may be worth of additional investigation.

Additional experiments are being carried out using other datasets. The results so far have confirmed the potential of the proposed approach. Other clustering algorithms are also being included. The authors also intend to have the support of biologists to identify the true contribution to gene expression data analysis, mainly in the discovery of new subclasses of the data. As a result, more general conclusions can be obtained. Future work includes the application of the same analysis performed in this paper, but comparing all partitions generated with all the different numbers of clusters investigated. The goals are to better analyze the isolation of problematic patterns and their influence in the good separation of the clusters and to investigate the identification of new subclasses in the data.

Acknowledgments

The authors would like to thank financial support of FAPESP and CNPq.

References

1. Monti, E., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning, 52 (2003) 91–118
2. Jiang, D., Tang, C., Zhang, A.: Cluster Analysis for Gene Expression Data: A Survey. IEEE Trans. Knowl. Data Eng. 16(11) (2004) 1370–1386
3. Herrero,J., Valencia,A., Dopazo,J.: A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17(2) (2001) 126–136
4. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall (1988)
5. Yeoh, E. J., et al.: Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell., 1(2) (2002) 133–143
6. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods: Part I. SIGMOD Record 31 (2) (2002) 40–45
7. Law, M. H., Jain, A. K.: Cluster validity by bootstrapping partitions. TR MSU-CSE-03-5, Dept. Comp. Science and Eng., Michigan State University (2003)

Machine Learning Techniques for Predicting *Bacillus subtilis* Promoters

Meika I. Monteiro[1], Marcilio C.P. de Souto[2], Luiz M.G. Gonçalves[1],
and Lucymara F. Agnez-Lima[3]

[1] Department of Computing and Automation,
Federal University of Rio Grande do Norte
{meika, lmarcos}@dca.ufrn.br

[2] Department of Informatics and Applied Mathematics,
Federal University of Rio Grande do Norte
marcilio@dimap.ufrn.br

[3] Department of Cellular Biology and Genetics,
Federal University of Rio Grande do Norte,
59072-970 Natal-RN, Brazil
lfagnez@ufrnet.br

Abstract. One of the most important goals of bioinformatics is the ability to identify genes in uncharacterized DNA sequences. Improved promoter prediction methods can be one step towards developing more reliable *ab initio* gene prediction methods. In this paper, we present an empirical comparison of machine learning techniques such as Naive Bayes, Decision Trees, Support Vector Machines and Neural Networks to the task of predicting *Bacillus subtilis* promoters. In order to do so, we first built a data set of promoter and nonpromoter sequences for this organism.

1 Introduction

The process of mapping from DNA sequences to folded proteins in eucaryotic and prokaryotic organisms involve many steps [1, 2]. The first step is the transcription of a portion of DNA into an RNA molecule, called RNA messenger. Such a process begins with the binding of a molecule called RNA polymerase to a location on the DNA molecule. The exact location where the polymerase binds determines which strand of the DNA will be read and in which direction. Parts of the DNA near the beginning of a protein coding region contain signals that can be recognized by the polymerase: these regions are called **promoters** [2].

Computational promoter prediction methods can be one step towards developing more reliable *ab initio* gene prediction programs [1, 3]. These methods could also be seen as part of the complex process of discovery gene regulatory network activity [4]. However, such a prediction task is hard due to both the large variable distance between various DNA signals that are the substrate of

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 77–84, 2005.

recognition by the polymerase, and a large number of other factor involved in regulation of the expression level [1, 5].

Despite these limitations, a large number of promoter prediction programs have been developed for eukaryotic sequences and are easily accessible [6, 7, 8, 5]. However, up to now, the Neural Network Promoter Prediction program (NNPP) is one of the few available system that can be used as a prokaryotic promoter prediction tool [9]. Some other prokaryotic promoter prediction methods are based on weight matrix pattern searches [10, 11, 12]. In [13], a novel prokaryotic promoter prediction method based on DNA stability is presented.

As pointed out in [5], there is a conceptual difference between trying to recognize all eukaryotic (prokaryotic) promoters, and recognizing only those being active in a specific cell type. Alternatively, the promoter finding problem might be divided into several sub-problems, meaning that it may be necessary to construct specific algorithms to recognize specific classes of promoters. For example, most of the system referred in the previous paragraph were designed for organism specific purposes, such as *Escherichia coli (E. coli).*

Motivated by this, in this paper, we apply machine learning techniques for the analysis of *Bacillus subtilis (B. subtilis)* promoters. We have chosen this bacteria for its wide use by biologists as a model of gram-positive bacteria in genetics studies, as well as for having all its genome already sequenced. Furthermore, there are, in the literature, many experimental analysis confirming several of their promoter sequences [14], which will be fundamental in the construction of our data set.

This paper is organized as follows. Section 2 presents the *B. subtilis* promoter dataset that we built, which is one of our contribution, and how the experimental results will be evaluated. In Section 3, we introduce a brief review of related work. The experimental setting is described in Section 4. Section 5 presents results and discussion. Finally, Section 6 summarizes the work.

2 Material and Methods

We perform an empirical comparison of rule based systems such as Decision Trees and PART, and statistical learning systems such as Naive Bayes, *k*-Nearest Neighbor, Support Vector Machines and Neural Networks to the task of *B. subtilis* promoter prediction. All the learning methods used in our study were obtained from the WEKA machine learning package [15] (http://www.cs.waikato.ac .nz/ ml/weka/).

2.1 Dataset

We built our dataset following the guidelines used in the construction of *E. coli* dataset first used in [17] (ftp://ftp.ics.uci.edu/pub/machine- learningdatabases/ molecular-biology/promoter-gene-sequences/). This *E. coli* dataset contains 53 examples of promoters and 53 of nonpromoters. The 53 examples of promoters were obtained from a compilation produced by [16]. According to [16], negative training examples were derived by selecting contiguous substrings from a fragment of the *E. coli* bacteriophage T7. Each example, either positive or negative, is composed of 57 attributes (i.e., 57 nucleotides).

In the case of the *B. subtilis* dataset, we initially consider for our dataset only experimentally determined promoter sequences presented in the compilation in [14]. Next, we calculated the average length of these sequences, which was 117 nucleotides. Based on this, we fixed the length of our promoters sequences in 117 nucleotides. Putative and shorter sequences than this number were discarded. Promoter sequences equal or longer than 117 nucleotides, on the other hand, were preserved in the dataset. In the latter case, the sequences were first cut on their upstream region such that their final length was kept in 117 nucleotides - this strategy was used to preserve the promoter region of the *B. subtilis* that is often found in the region between -100 (downstream) and +15 (upstream) of the gene. At the end of this process, we ended up with 112 promoter sequences out of 236 originally presented in [14].

In order to create the nonpromoter sequences for our dataset, we select 112 contiguous substrings of 117 nucleotides from the genome of the *B. subtilis* bacteriophage PZA [18] - there is no promoter sequence identified for this phage so far. These 112 nonpromoters sequences were chosen in such way as to maximize the degree of similarity of each of them to the sequences in the promoter set. This resulted on an average degree of similarity between promoter and nonpromoter sequences of 27%. In the case of the *E. coli* dataset in [17] this average is of 24%. In summary, our *B. subtilis* dataset contains 112 examples of promoters and 112 of nonpromoters, each one with 117 nucleotides.

2.2 Evaluation

The comparison of two supervised learning methods is, often, accomplished by analyzing the statistical significance of the difference between the mean of the classification error rate, on independent test sets, of the methods evaluated. In order to evaluate the mean of the error rate, several (distinct) data sets are needed. However, the number of data sets available is often limited. One way to overcome this problem is to divide the data sets into training and test sets by the use of a k-fold cross validation procedure [19, 20, 15].

This procedure can be used to compare supervised methods, even if only one data set is available. The procedure works as follows. The data set is divided into k disjoint equal size sets. Then, training is performed in k steps, each time using a different fold as the test set and the union of the remaining folds as the training set. Applying the distinct algorithms to the same folds with k at least equal to 10, the statistical significance of the differences between the methods can be measured, based on the mean of the error rate from the test sets [19, 20, 15].

In fact, in this work, in order to use the greatest possible amount of data for training, we use a special form of cross-validation called leave-one-out [19, 20, 15]. The leave-one-out cross-validation is simply k-fold cross validation, where k is the number of instances in the dataset. In our case, k is set to 224, that is, the total of promoter and nonpromoter sequences in the dataset.

3 Related Work

In the context of machine learning, the identification of promoter sequences can be stated as follows [3]:

- **Problem:** Identification of promoter sequences.
- **Input:** Set of DNA sequences of fixed length with known promoter regions and sequences without the presence of this signal.
- **Do:** Generate a classifier able to predict if a window of fixed size has or not a promoter region.

The most closely related work to ours is the one developed in [16]. In that work, a hybrid approach based on neural networks and symbolic rules was applied to predicting promoter sequences of *E. coli.* The system used, called KBANN (Knowledge Based Neural Network), uses propositional rules formulated by a biologist in order to determine the network topology and initial weights. By doing so, [16] could observe a decrease of the network training time and an improvement of its generalization.

The dataset used by [16] contained 53 examples of promoters and 53 of nonpromoters, where each example, is composed of 57 attributes (i.e., 57 nucleotides), as described at beginning of Section2.1. In the experiments performed by the author, the results obtained with KBANN were compared to those achieved by using neural network in the form of multi-layer perceptron (as the ones used in our work), decision trees, k-nearest neighbor, among others. The best results achieved was with the KBANN and neural networks - decision trees and k-nearest neighbor had a poorer result. The author do not present in the results the rate of false positives.

In [9], a neural network model of the structural and compositional properties of a eukaryotic core promoter region, the Neural Network Promoter Prediction (NNPP), was developed and applied to analysis of the *Drosophila melanogaster.* The model uses a time-delay architecture, a special case of a feedforward neural network. According the authors, application of this model to a test set of core promoters not only gave better discrimination of potential promoter sites than previous statistical or neural network models, but also revealed indirectly subtle properties of the transcription initiation signal. Such a model was extended to work with prokaryotic promoters. In fact, the (NNPP) tool is one of the few available system that can be used as a prokaryotic promoter prediction program (http://www.fruitfly.org/seq tools/promoter.html).

Recently, [21] developed a comparative analysis on the application of both transductive support vector machines (TSVM) and support vector machines (SVM) to prediction of eukaryotic promoter regions. According to them, TSVM outperformed SVM in this task.

4 Experiments

Our experiments were accomplished by presenting the dataset to the Machine Learning (ML) algorithms. In fact, because of the leave-one-out methodology each method was run 224 times. The values for the parameters of the ML algorithms were chosen as follows. For example, for an algorithm with only one parameter, an initial value for the parameter was chosen followed by the run of algorithm. Then, experiments with a larger and smaller value were also performed.

If with the initially chosen value, the classifier obtained has the best results (in terms of validation error), then no more experiments were performed. Otherwise, the same process was repeated for the parameter value with the best result so far. Of course, this procedure becomes more time consuming with the increasing in the number of parameters to be investigated.

Using the previous procedure, we arrived to the following values for the parameters of the ML algorithms (WEKA implementation [15]):

- k-Nearest Neighbor (k-NN): k was set to 8 and the distance weighting to 1/distance. All other parameters were set to their default.
- Naive Bayes (NB): all parameters were set to their default.
- Decision Tree (DT): all parameters were set to their default.
- PART : all parameters were set to their default.
- Voted Perceptron (VT): all parameters were set to their default.
- Support Vector Machine (SVM): C was set to 1 and the exponent to 4. All the other parameters were set to their default.
- Neural Networks (NN): the number of hidden nodes was set to 50, the learning rate to 10^{-2}, the momentum term to 0.9, the maximum number of iteration to 1000, and the validation set size to 10%. All the other parameters were set to their default.

Each of the previous ML method, as already mentioned, was trained with a leave-one-out methodology, considering the best parameter setting found. Then, for all experiments, the mean of the percentage of incorrectly classified training patterns on independent test sets were measured. Next, these means were compared two by two means of paired t-test, as described in [20, 19].

5 Results

Table 1 presents, for each ML algorithm, the mean and standard deviation of the percentage of incorrectly classified examples (error rate) on independent test sets. According to this table, NB and SVM obtained a lower classification error than the other methods (18.30%). The null hypotheses were rejected in favor of SVM and NB in comparison to k-NN,DT and PART at $\alpha = 0.05$, where α stands for the significance level of the equal means hypothesis test. However, no significance difference was detected between NB and NN ($\alpha = 0.05$) and SVM and NN ($\alpha = 0.05$).

Now, we turn our attention to more detailed comparison. In order to do so, we constructed a confusion matrix as illustrated in the table below Table 2. This table shows a generic confusion matrix, where TP (True Positive) denotes the mean of the correct classification of promoter examples (positive examples).

Table 1. Mean of the Classification Error Rate

Algorithm	Mean	St. Dev.
k-NN	34.82%	47.75%
NB	18.30%	38.76%
DT	30.80%	46.27%
PART	31.25%	46.46%
VP	32.14%	46.81%
SVM	18.30%	38.76%
NN	25.0%	43.40%

True Negative (TN) stands for the mean of correct classification of nonpromoters examples (negative examples). False Positive (FP) represents the mean of the incorrect classification of negative examples into the class of positive examples. Likewise, False Negative (FN) is the mean of positive examples incorrectly classified into the class of negative examples.

Table 2. Confusion Matrix

	Promoter	Nonpromoter
Promoter	TP	FN
Nonpromoter	FP	TN

In order to develop our detailed analysis will consider three best results obtained in our experiments in terms of lower error rate mean, that is, we choose NB, SVM and NN, as shown in Table 3, 4 and 5.

Table 3. Confusion Matrix for NB

	Promoter	Nonpromoter
Promoter	82%	18%
Nonpromoter	19%	81%

Table 4. Confusion Matrix for SVM

	Promoter	Nonpromoter
Promoter	76%	24%
Nonpromoter	12.4%	87.5%

In terms of the previous tables, for the convenience of results comparison, in our evaluation scheme will consider TP and FP. In this case, NB, SVM and NN present, respectively, the following TP/FP relation: 4.31, 6.08, and 2.67. Thus, in such a context, SVM offers the best trade-off between generalization

Table 5. Confusion Matrix for NN

	Promoter	Nonpromoter
Promoter	80%	20%
Nonpromoter	30%	70%

and discrimination. This also implies in a better control of the false positives. Such an issue is important for, as mentioned before, there is a high probability of finding similar sequence elements elsewhere in genomes, outside the promoter regions (false positives).

We also tried our dataset using the NNPP. The NNPP program is available at (http://www.fruitfly.org/seq tools/promoter.html). All the NNPP predictions were carried out at a score cut-off 0.80. In the case of our dataset, the NNPP systems correctly predicted 107 promoter sequences out of 112, that is, a TP of 95.53%. However, for the nonpromoter sequences such a system presented a unacceptable high FP of 74.1% (83 nonpromoter sequences, out of 112, were predicted as true promoter sequences). That is, for this dataset, the TP/FP relation for the NNPP was of 1.29, which is much lower than the rate presented by our SVM (6.08).

6 Final Remarks

In this paper, we presented an empirical comparison of machine learning techniques such as Naive Bayes, Decision Trees, Support Vector Machines and Neural Networks to the task of predicting *B. subtilis* promoters. In order to so, as one of our contributions, we first built a dataset of promoter and nonpromoter sequences for this organism.

From the different learning algorithms analyzed, support vector machines outperformed the others in terms of the relation true positive rate/false positive rate (6.08). In fact, such a method performed better as compared to currently available prokaryotic prediction method, such as NNPP. One reason for this is that we constructed a classifier to recognize a specific classes of promoters (*B. subtilis*), while the NNPP is a general system.

In terms of our results, the classifiers build need to be further improved to reduce the number of false positives. This could be achieved by, for example, combining several classifiers in order to form ensemble [22].

References

1. Baldi, P. and Brunak, S.: the Machine Learning Approach. Bioinformatics MIT Press (1998) second edition
2. Alberts, B. and Bray, D. and Lewis, J. and Raff, M. and Roberts, K. and Watson, J.: The molecular biology of the cell. Garland Publishing New York (1989)

3. Craven, M. W. and Shavlik, J.: Machine learning approaches to gene recognition. IEEE Expert (1994) **9** 2-10
4. Tavazoie, S. and Hughes, J. D. and Campbell, M. J. and Cho, R. J. and Church, G. M.: Systematic determination of genetic network architecture. Nature Genetics (1999) **22** 281-285
5. Pedersen, A. G. and Baldi, P. and Chauvin, Y. and Brunak, S.: The biology of eukaryotic promoter prediction - a review. Comput. Chem. (1999) **23** 191–207
6. Fickett, J. W. and Hatzigeorgiou, A. G.: Eukaryotic promoter recognition. Genome Res. (1997) **7** 861–78
7. Rombauts, S. and Florquin, K. and Lescot, M. and Marchal, K. and Rouze, P. and van de Peer, Y.: Computational approaches to identify promoters and cis-regulatory elements in plant genomes. Plant Physiol. (2003) **132** 1162–1176
8. Werner T.: The state of the art of mammalian promoter recognition. Brief. Bioinform. (2003) **4** 22–30
9. Reese, M. G.: Application of a time-delay neural network to promoter annotation in the drosophila melanogaster genome. Comput. Chem. (2001) **1** 51–56
10. Standen, R.: Computer methods to locate signals in nucleic acid sequences. Nucleic Acids Res. (1984) **12** 505-519
11. Mulligan, M. and Hawley, D. K. and Entriken, R. and McClure, W.: Escherichia coli promoter sequences predict in vitro RNA polymerase selectivity. Nucleic Acids Res.(1984) **12** 789-800
12. Huerta, A. and Collado-Vides, J.: Sigma70 promoters in Escherichia coli: specific transcription in dense regions of overlapping promoter-like signals. Mol. Biol. (2003) **333** 261-278
13. Kanhere, Aditi and Bansal, Manju: A novel method for prokaryotic promoter prediction based on DNA stability. BMC Bioinformatics (2005) **6** 1–10
14. Helmann, J. D.: Compilation and analysis of Bacillus subtilis of extended contact between RNA polymerase and upstream promoter DNA. Nucleic Acids Research (1995) **23** 2351–2360
15. Witten I. H. and Frank E.: Data mining: practical machine learning tools and techniques with Java implementation. USA: Morgan Kaufman Publishers (2000)
16. Towell, G. G.: Symbolic knowledge and neural networks: insertion, refinement and extraction. University of Wisconsin (1991) PhD thesis Computer Science
17. Harley, C. B. and Reynolds, R. P.: Analysis of E. coli promoter sequences. Nucleic Acids Research (1987) **15** 2343–2360
18. Paces, V. and Vlcek, C. and Urbanek, P. and Hostomsky, Z.: Nucleotide sequence of the right early region of Bacillus subtilis phage PZA completes the 19366-bp sequence of PZA genome; Comparison with the homologous sequence of phage phi 29. Gene (1986) **44** 115–120
19. Mitchell, T.: Machine Learning. McGraw Hill New York (1997)
20. Dietterich, T. G.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation (1998) **10** 1895-1923
21. Kasabov, N. and Pang, S.:Transductive support vector machines and applications in bioinformatics for promoter recognition. Neural Information Processing - Letters and Reviews (2004)**3** 31-37
22. Dietterich, T. G.: Ensemble methods in machine learning in Multiple Classifier Systems. Lecture Notes in Computer Science (2000) **1857** 1-15

An Improved Hidden Markov Model Methodology to Discover Prokaryotic Promoters

Adriana Neves dos Reis and Ney Lemke

UNISINOS, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada,
PIPCA Av. Unisinos, 950 – 93.022-000
São Leopoldo, Rio Grande do Sul, Brasil
{adriana, lemke}@inf.unisinos.br
http://www.inf.unisinos.br/~lbbc

Abstract. Gene expression on prokaryotes initiates when the RNA-polymerase enzyme interacts with DNA regions called promoters, where are located the main regulatory elements of the transcription process. Despite the improvement of *in vitro* techniques for molecular biology analysis, characterizing and identifying promoters is a complex task. *In silico* approaches are used to recognize theses regions. Nevertheless, they confront the absence of a large set of promoters to identify conserved patterns among the species. Hence, a methodology able to predict them on any genome is a challenge. This work proposes a methodology based on Hidden Markov Models (HMMs), Decision Threshold Estimation and Discrimination Analysis. For three investigated prokaryotic species, the mainly results are: a reduction in 44.96% of recognition error rate compared with previous works on *Escherichia coli*, an accuracy of 95% on recognition and 78% on prediction for *Bacillus subtilis*. However, it was found a large number of false positives on *Helicobacter pylori.*

1 Introduction

The fundamental mechanism that permits the expression of a gene is the interaction of RNA-polymerase enzyme (RNAp) with a DNA region containing the signals to establish the gene to be expressed and its expression rate. This region is known as the promoter [10].

Prokaryotic promoter sequence can be described by a frame-set where the first nucleotide that belongs to transcript is defined by +1 or transcription start site (TSS), the nucleotides before this are represented by negative integers, while the posterior ones by positive integers. Promoters have three characteristic regions: a conserved sequence of 6 nucleotides (hexamer) centered on -35, another one centered on -10 and finally one region between them. The distance between the two hexamers has on average 17 base pairs and seems to be relevant. This because, even having variable size and showing no conservation on its nucleotide composition, it may be different of other genome regions.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 85 – 94, 2005.

The ideal promoter for *Escherichia coli* (*E. coli*) is characterized by the pattern TTGACAN_{17}TATAAT, where *N* corresponds to any nucleotide (see Fig. 1). However, this pattern cannot be found in any real promoter region [11].

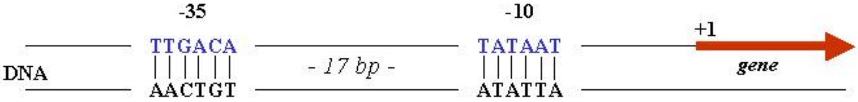


Fig. 1. A representation to the frameset describing conserved regions on prokaryotic promoters

A generic methodology to represent and discover promoters is an essential step toward the understanding of gene expression regulation. Nowadays, we have an extensive dataset with promoter information for *E. coli* and partial information for few other organisms [12]. Most of the previous works on this area do not benefit from recent biochemical data [10].

In this work, we propose a methodology based on HMMs for recognition and prediction of promoter regions, considering: datasets with a great number of sequences, the use of a criterion to determine the critical score to recognize a sequence as a promoter, and a technique to reduce the influence of genome characteristics on identification of promoter patterns.

2 Methodology

Prokaryotic genomes can be classified in two sets of sequences: coding and noncoding regions. A critical task for Bioinformatics has been the classification of sequences on these groups, for that many approaches have been investigated [6].

The most effective ones to find gene sequences are based on Markov models. Genemark [8] uses a fixed-order Markov chain, while Glimmer [14] uses interpolated Markov models. However the success of these approaches on identification of coding regions, the characterization of non-coding regions remains a challenge, despite of *in vitro* and *in silico* researches. Inside of these regions are located the regulatory elements responsible for gene regulation.

The classical studies on promoter recognition are based on hidden Markov models. This technique has been applied successfully to find and interpret different classes of DNA regions, like transcription units, genes and proteins. Observing that promoters are sequences of nucleotides with different conservation levels at each position, HMMs is an adequate model for these sequences. They capture both the strong conserved region, that are modeled with concentrated probability distributions, as nonconserved regions that can be modeled by more uniform distributions [9].

2.1 Standard HMM Methodology

The HMM is a Markov model and can be described as stochastic finite state automata [2], given by $M = (Q, \Sigma, \pi, a, b)$, where:

- $Q = \{q_0, \dots, q_n\}$ is a set with n states;
- $\Sigma = \{\sigma_1, \dots, \sigma_m\}$ is an output alphabet;
- π_i are the initial probabilities;
- a_{ij} are the transition probabilities from state i to j;
- b_{ik} is the probability that i emits a symbol k.

The system evolves visiting the states q and emitting at each one a symbol from Σ. These symbols represent the observable sequences that we wish to model. Since empirically we have no access to the states q, this Markov model is called hidden. The matrices π, a, and b are obtained from a dataset.

HMM used on biological sequence analysis is composed by five types of states that models the sequence alignment analysis: *match*, *insert*, *delete*, *begin* and *end*. These states are organized in an architecture like that presented on Fig. 2.

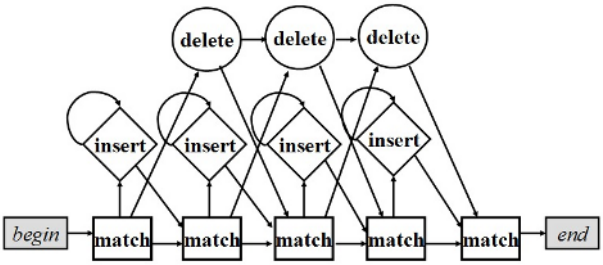


Fig. 2. The classical HMM architecture applied to describe biological sequences like promoters. The number of match states is equal to the length of sequences modeled

An HMM can be trained to recognize a set of sequences belonging to a biological class. The adherence of a sequence to a model can be calculated using *Viterbi*'s algorithm, which computes the likelihood of the best transition path among all the possible states [2]. For convenience, we use instead the negative logarithm of this quantity, denominate *Viterbi score* (S). This implies that sequences with high adherence to the model have smaller scores.

The description above is the standard HMM methodology for any kind of sequence recognition task. On promoters, its application of may be considered problematic because:

- there is no specification of a formal method to determine the threshold among *Viterbi scores* to consider a sequence belonging to the promoter class;
- there is a strong influence of genome properties of each specie, such as A+T content, on the promoter model. This reduces the model applicability to species closely related to the ones used on the training process.

We extended the classical methodology aiming the solution of these problems, resulting in an improved methodology described as follows.

2.2 Improved HMM Methodology

The proposal of an improvement on standard HMM methodology was guided by several experiments to test recognition's tools and available promoter datasets. These allowed the identification of the two cited limitations on the ending of previous section. In this way, we propose an extension to resolve which one of them.

The first extension was the use of Decision Threshold Estimation (DTE), which is based on Bayes Decision Rule. This rule can be applied to resolve the expected value of payoff for two classes of sequences, being chosen that with max value.

Determination of the critical threshold (S_c) is based on calculus of *Viterbi* score (S) for each promoter and gene sequence used along a training process. We observed that the scores values can be modeled as random variables normally distributed. In such case, fitting the each class data set to Gaussians, two distributions are obtained: P_p and P_n, where n, p stands for negative and positive. From these distributions is possible define the S_c value adopting a criterion to be maximized. Our methodology considers as criterion the accuracy defined by:

$$A = \frac{TN + TP}{TN + TP + FN + FP} \, , \tag{1}$$

where T and F stands for true and false and P and N for positive and negative. Assuming that the distributions P represent truly the data, we determine the functions $TN(S)$ and $TP(S)$. These functions represent the fraction of true positives and true negatives as a function of a threshold S_c, which is the estimated value that maximizes TN+TP. Fig. 3 shows both probability distributions.

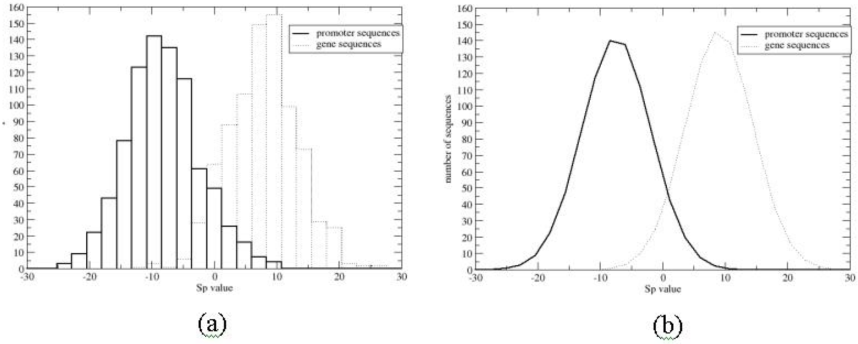


Fig. 3. The score probability distributions to promoter and gene sequences. (a) The histograms for P_p and P_n. (b) The fit of data to Gaussians

On the second extension, we applied a Discrimination Analysis (DA) to minimize the influence of genome properties in the promoter patterns modeled. DA considers two HMMs, one to recognize promoter sequences and another to recognize gene sequences. This is appropriated in promoter case, because it belongs to the class of genome sequence too. However, on prokaryotes, which have genome with a compactable structure, we can reduce the genome sequences to promoters and genes, or coding and no-coding (where probably there is some regulatory element). In this manner, the new classification parameter, S_p, is given by:

$$S_p = S_{promoter} - S_{gene} \quad , \tag{2}$$

where $S_{promoter}$ is the score of the sequence on the promoter HMM and S_{gene} the score on gene HMM. The rationale for this choice is that S_p is the negative logarithm of the ratio between the likelihoods of a sequence being a promoter and a gene. In another words, we are calculating how more high is the probability of a sequence be a promoter of that be a gene. Fig. 4 condenses the main stages of the standard and the improved HMM methodology.

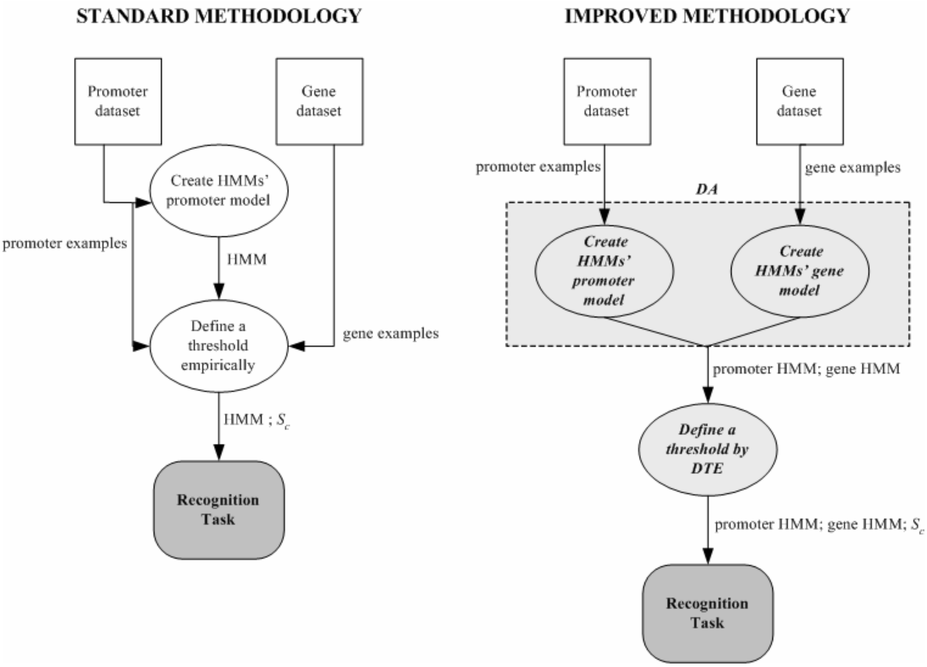


Fig. 4. Global vision of the differences between the standard and the improved methodology

3 Results

The proposed methodology was applied to three prokaryotic species: *Escherichia coli* (*E. coli*), *Bacillus subtilis* (*B. subtilis*) and *Helicobacter pylori* (*H. pylori*). Moreover, it was used to recognize and predict promoters.

For recognition, a dataset to train the promoter model is available, while for prediction there is not enough identified promoters to create the model. In this last case, we use the *E. coli* promoter HMM to predict promoters.

HMMs were trained using the HMMpro, tool designed to simulate and analyze hidden Markov models of biological sequences [1]. Model architecture is linear and permits transitions to all possible states. The models have 81 main states, the same number of nucleotides on the RegulonDB sequences, and each nucleotide has an initial emission probability of 0.25.

Once the structure and the initial parameters are defined, the model was trained during 30 epochs in the on-line mode, period considered adequate for this problem [1]. After the training stage, we have the best model to describe the training dataset for promoters and the best model to describe the training dataset genes.

Since HMMs have a large number of parameters, the new methodology was validated using 10-fold cross validation. For each fold, it was calculated: S_c, the expected accuracy (A_e) and observed accuracy (A_o). A_e is calculated by applying the equation (1) on $TP(S_p)$ and $TN(S_p)$ obtained from the fitted score distributions, while A_o is obtained using *TP* and *TN* from the test datasets.

Since the application of the methodology depends on the available dataset for the organisms, the results are organized accordant them.

3.1 E. coli

The promoter regions of *E. coli* were obtained from the RegulonDB database in the version 4.0 [13]. This database provides biological information on different mechanisms of transcription, regulation, and structural aspects related to gene expression rate of *E. coli* [5].

The 928 promoter sequences (true positives – *TP*) of our dataset with length of 81 nucleotides were filtered and shuffled by a Perl script to build the training and test sets. The database was restricted because some entries have no sequence to the promoter instance. For each promoter set, it was created a coding sequence dataset with the same number of instances, which corresponds to true negatives examples (*TN*) for the experiment.

For a better evaluation of the each extension on the methodology, we execute two distinct experiments: one using just the Decision Threshold Estimation, and another considering the simultaneous use of Decision Threshold Estimation and Discrimination Analysis. The results of both are presented on Table 1.

The comparative analysis of the two experiments shows that the accuracy is higher than 0.9, when we consider both extensions of the methodology. While previous *in silico* studies predicted about 13-54% of the promoters correctly [12].

Table 1. Values of average accuracy for the 10-fold cross validation considering just Decision Threshold Estimation, and this one with Discrimination Analysis on E. coli

Experiment	*measure*	*average*	*standard deviation*
Decision Threshold Estimation	S_c	136.587	0.615
	A_e	0.851	0.011
	A_o	0.817	0.07
Decision Threshold Estimation and Discrimination Analysis	S_c	1.343	0.604
	A_e	0.921	0.005
	A_o	0.919	0.03

Moreover, the recognition performance increased on 12.48% and the error rate decreased to 44.26% when both extensions were associated.

3.2 B. subtilis

Promoter regions of B. *subtilis* were extracted from compilation of Hellman [4], which contains 220 sequences. The application of the same experiments realized on *E. coli* case in this prokaryote results the measures presented on Table 2.

Table 2. Values of average accuracy for the 10-fold cross validation considering just Decision Threshold Estimation, and this one with Discrimination Analysis on *B. subtilis*

Experiment	*measure*	*average*	*standard deviation*
Decision Threshold Estimation	S_c	132.966	1.374
	A_e	0.983	0.004
	A_o	0.95	0.031
Decision Threshold Estimation and Discrimination Analysis	S_c	-22.635	1.454
	A_e	0.986	0.003
	A_o	0.95	0.031

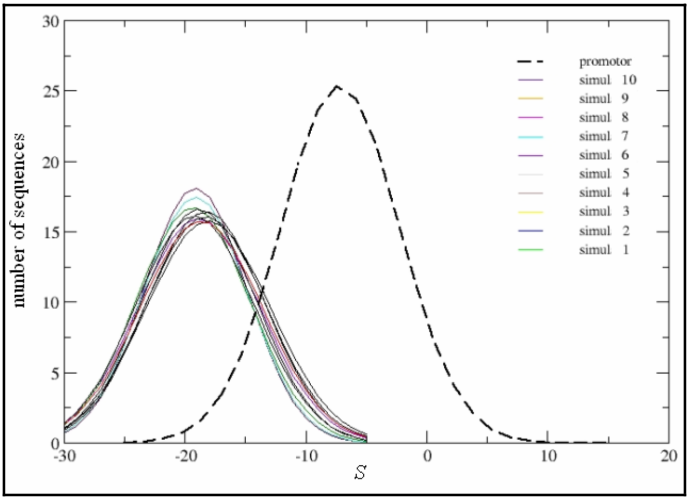


Fig. 5. Distributions of S_p considered on prediction task on *B. subtilis*. The dash line represent the gauss curve for *E. coli* promoters, the others the for *B. subtilis* genes, each one for a fold in the 10-fold cross validation

The accuracies show that both experiments had a better performance than the previous ones. A reason for this may be the high A+T content of *B. subtilis* genome (around 0.56) compared with the *E. coli* (0.49). Thus, it is expected that gene sequences of *B. subtilis* had higher adherence to *E. coli* promoter model than properly *E. coli* promoters

The high A+T content on promoter regions is well known, considering that this characteristic permits the RNAp open the DNA double helix more easily, in function of the two hydrogen bonding of a base-pair A-T.

To determine the performance of the application of *E. coli* model to predict promoters in other organisms, we redefined S_p as the difference between *E. coli* promoter HMM model and *B. subtilis* gene HMM model. In this case, we are supposing that we do not have any promoter dataset to estimate S_c. This value was replaced by the average S_p minus one standard deviation obtained using only gene sequences (see Fig. 5). The Table 3 presents this analysis.

Table 3. Analysis of prediction promoters based on promoter *E. coli* HMM and gene *B. subtilis* HMM. Results from 10-fold cross validation.

Fold	*average S_p*	*standard deviation*	*accuracy*
1	-19.623	4.656	0.775
2	-19.378	4.987	0.784
3	-19.320	4.96	0.775
4	-19.221	5.002	0.773
5	-19.046	4.943	0.775
6	-18.873	4.945	0.777
7	-18.480	4.889	0.786
8	-18.604	4.792	0.784
9	-18.613	4.874	0.777
10	-18.797	4.994	0.773

Observe that the accuracy on prediction is lower than in the recognition task. Nevertheless, in the first case we do not have any idea about promoter nucleotide composition, i.e., values of accuracy around 0.75 is a considerable performance considering that none information about promoter is taken in account.

3.3 H. pylori

The last experiment was the prediction of promoter on an organism that had no available information about their promoter sequences composition. For this case we analyzed the genome of *Helicobacter pylori*.

Using the same methods used on *B. subtilis* case, we concluded that is impossible to apply the same methodology on this organism because its gene sequences were almost totally recognized as promoters. Based on analyses of S_p distribution, this characteristic is confirmed (see Fig. 6).

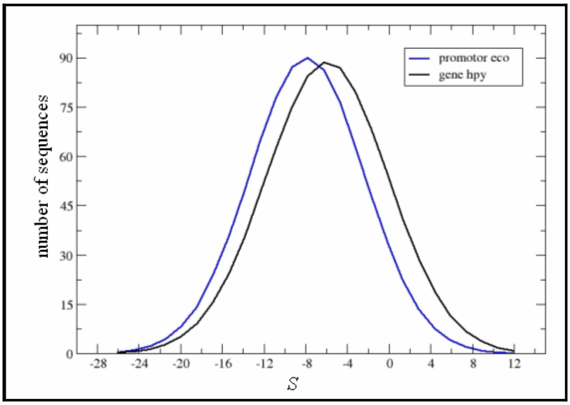


Fig. 6. Distributions of S_p considered gauss curve for *E. coli* promoters and for *H. pylori* genes. Observe that they are practically overlaid

4 Conclusions

In this work we propose a methodology, which extends the standard one by considering Decision Threshold and Discrimination Analysis based on HMMs, for prokaryotic promoter recognition and prediction.

The new methodology was successfully applied to *E. coli*, reducing the error rate in 45%. The mainly reason for this success is the utilization of two HMMs, one for promoters and other for gene sequences. This strategy reduces the number of false positives; those gene regions with high A+T content. This performance was repeated on *B. subtilis* studies on recognition as well on prediction task. However, the methodology failed on *H. pylori* because the genes of this organism had high adherence to HMM *E. coli* promoter model.

We expect that this methodology could be extended to other prokaryotic organisms with A+T content different from 0.5. At this moment, we are investigating extensions to the methodology that will allow its use on prokaryotes to recognize and to predict promoters.

Acknowledgements

We acknowledge the support of CNPq. This work was developed in collaboration with HP Brazil R&D.

References

1. Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M.A.: Hidden Markov models of biological primary sequence information. Proc. Natl Acad. Sci., Vol. 91 (1994) 1059-1063.
2. Clote, P. and Backofen, R.: Computational Molecular Biology: an introduction. John Wiley & Sons, Chichester (2000).

3. Eddy, S. R.: Profile hidden Markov models. Bioinformatics, Vol. 14, n. 9, (1998) 755-763.
4. Helmann, J. D.: Compilation and analysis of Bacillus subtilis □A-dependent promoter sequences: evidence for extended contact between RNA polymerase and upstream promoter DNA. Nucleic Acids Research, vol. 23, n. 13, (1995) 2351-2360.
5. Huerta, A. M., Salgado, H., Thieffry, D., and Collado-Vides, J.: RegulonDB: a database on transcriptional regulation in *Escherichia coli*. Nucleic Acids Res., Vol. 26. (1998) 55-59.
6. Krogh, A.: An Introduction to Hidden Markov Models for Biological Sequences. In: Computational Methods in Molecular Biology, Elsevier, (1998) 45-63.
7. Karp, P.D., Arnaud, M., Collado-Vides, J., Ingraham, J., Paulsen, I.T., and Jr., M.H.S.: The E. coli Ecocyc database: No longer just a metabolic pathway database. ASM News (2004).
8. Lukashin, A. V., Borodovsky, M.: GeneMark.hmm: new solutions for gene finding. Nucleic Acids Research, vol. 26, n. 4, (1998) 1107-1115.
9. Mount, D.W.: Bioinformatics: Sequence and Genome Analysis. CSHL Press, New York (2001).
10. Pedersen, A.G., Baldi, P., Brunak, S., and Chauvin, Y.: Characterization of prokaryotic and eukaryotic promoters using hidden Markov models. Proc Int Conf Intell Syst Mol Biol, (1996) 182-191.
11. Pevzner, P.A.: Computational Molecular Biology: An Algorithmic Approach. Cambridge University, London (2000).
12. Qiu, P.: Recent advances in computational promoter analysis in understanding the transcriptional regulatory network. Biochemical and Biophysical Research Communications, vol. 309, (2003) 495-501.
13. Salgado, H., Gama-Castro, S., Martínez-Antonio, A., Díaz-Peredo, E., Sánchez-Solano, F., Peralta-Gil, M., Garcia-Alonso, D., Jiménez-Jacinto, V., Santos-Zavaleta, A., Bonavides-Martínez, C., Collado-Vides, J.: RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. Nucleic Acids Research, vol. 32, D303-D306 (2004).
14. Salzberg, S. L., Delcher, A. L., Kasif, S., White, O.: Microbial gene identification using interpolated Markov Models. Nucleic Acids Research, vol. 26, n. 2, 544-548 (1998).

Modeling and Property Verification of Lactose Operon Regulation*

Marcelo Cezar Pinto[1], Luciana Foss[1],
José Carlos Merino Mombach[2], and Leila Ribeiro[1]

[1] Instituto de Informática,
Universidade Federal do Rio Grande do Sul,
Porto Alegre, RS, Brasil
{mcpinto, lfoss, leila}@inf.ufrgs.br
[2] Laboratório de Bioinformática e Biologia Computacional,
Universidade do Vale do Rio dos Sinos,
São Leopoldo, RS, Brasil
mombach@exatas.unisinos.br

Abstract. Understanding biochemical pathways is one of the big challenges of the field of molecular biology nowadays. Computer science can contribute in this area in a variety of ways. One of them is providing formalisms and tools to simulate and check properties of pathways. One formalism that is well known and suited for modeling concurrent and distributed systems is Milner's Calculus of Communicating Systems (CCS). CCS is a process algebra and there are many tools that support modeling and automatic verification of properties of systems modeled in terms of CCS processes. This paper describes the regulation of the lactose operon using CCS. We validate our formal model by automatic checking a series of properties that are known for the regulation of the lactose. Thus, we show the viability of using process algebras to model and reason about biochemical networks.

1 Introduction

Biochemical pathways are one of the most studied topics in molecular biology. The behavior of cells is governed and coordinated by biochemical networks that translate external cues (hormones, growth factors, substances) into adequate biological responses such as cell proliferation, specialization and metabolic control. Metabolic and regulatory pathways are two examples of biochemical networks. However, it is very time consuming and expensive to make laboratory experiments to understand how a biochemical pathway works. A better approach would be to simulate these systems using computers, and only make laboratory experiments when the simulations give hints that some expected behavior could occur. The simulation of these networks can answer, for example, whether the

* Project partially supported by CNPq (grant 550042).

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 95–106, 2005.

concentration of some components have increased inside the cell when the cell is put in a glucose-rich environment. To simulate and discover properties of these networks *in silico*, formal models are needed [1]. The most used model to simulate biochemical pathways is based on differential equations and its variants [2]. However, these equation-based models make it very difficult to verify properties of such networks. Therefore, for many applications, hybrid, like XS-Systems [3], and symbolic models, like Petri nets [4] and graphs [5], are preferred.

Bacteria have a simple general mechanism for coordinating the regulation of genes encoding products that participate in a set of related processes: these genes are clustered on the chromosome and are transcribed together. Many prokaryotic mRNAs are polycistronic — multiple genes on a single transcript — and the single promoter that initiates transcription of the cluster is the site of regulation for expression of all the genes in the cluster. The gene cluster and promoter, plus additional sequences that function together in regulation, are called an **operon** [6]. Many of the principles of prokaryotic gene expression were first defined by studies of lactose metabolism in *E. coli*, which can use lactose as its sole carbon source. In 1961, Jacob and Monod published a paper that described how two adjacent genes involved in lactose metabolism were coordinately regulated by a genetic element located at one end of the gene cluster [7]. The genes were those for β-galactosidase, which cleaves lactose to galactose and glucose, and galactoside permease, which transports lactose into the cell. The terms "operon" and "operator" were first introduced in this paper. With the operon model, gene regulation could, for the first time, be considered in molecular terms [6]. In this paper we will focus our modeling in regulation of lactose operon. Besides modeling, we show that our model is a faithful representation of the real system by using model checking techniques (we prove that crucial properties of the real system are satisfied by the formal model).

Recent work by Regev, Silverman and Shapiro suggests that process algebras, like CCS and π-calculus, may become valuable tools in modeling and simulation of biological systems where the interaction and mobility are important features [8]. The field may have an important impact in understanding how biological systems work, giving at the same time a way to describe, manipulate, and analyze them. Ciobanu, Ciubotariu and Tanasă developed a π-calculus model for Albers-Post mechanism to ion (Na^+ or K^+) transport across membrane [9] and Yildirim and Mackey proposed nonlinear differential delay equations to model regulation in the lactose operon and made comparisons with experimental data [10]. A recent work of Chabrier-Rivier, Chiaverini, Danos, Fages and Schächter proposed a formal counterpart of Kohn's compilation on the mammalian cell-cycle control and the use of the Computation Tree Logic (CTL) as a query language for biomolecular networks [11].

Our contribution in this paper is the usage of a process algebra (CCS) associated to a temporal logic (CTL) to analyse the expected behavior of a biological system. To the best of our knowledge it is the first time that a process algebra is used to make qualitative inferences of a biological system, instead of simulations as in [8]. We point out that our approach isn't quantitative, but it complements this one.

This paper is structured as follows: after a short introduction to CCS (section 2) and to the mechanism of regulation of lactose operon (section 3), we present our model (section 4) and show that it is faithful (section 5). Finally, in section 6 we relate our work to previous ones, make some conclusions and show our future work directions.

2 Calculus of Communicating Systems

The Calculus of Communicating Systems (CCS) [12] is a mathematical formalism designed to describe and analyze the behavior of different systems running concurrently. CCS is a process algebra, where all components of the system can be viewed as processes that can interact via message-passing. This interaction is modeled as synchronized communication. A CCS process can be viewed as a black box, which may have a name and has a process interface. This interface describes the channel set that this process can use to interact with other processes in its environment. Each of the channels may be either an input or an output channel. The behavior of a process is given by the actions it can perform. These actions can occur sequentially or in parallel, and there may be non-deterministic choices of which actions shall occur.

For example, the interface for a process named CM (for Coffee Machine) is given by coin (input) and $\overline{\mathsf{coffee}}$ (output) channels. This process may interact with its environment via these channels. Processes that want to interact with CM must have at least one of the complement channels, that is, $\overline{\mathsf{coin}}$ or coffee.

2.1 Syntax and Semantics of CCS

We present a subset of the CCS language that we use in this paper. The most basic process is the process $\mathsf{0}$, that performs no action whatsoever. Another basic construction in CCS is *action prefixing*, that describes the execution of an action after other (sequential behavior). For example, the process $\mathsf{coin}.\overline{\mathsf{coffee}}.\mathsf{0}$ performs an input action in the coin channel, thereafter performs an output action in $\overline{\mathsf{coffee}}$ channel and finally does not perform anything. We can introduce a name for the previous process, and use this name in other process descriptions: $\mathsf{CM} \overset{\mathsf{def}}{=} \mathsf{coin}.\overline{\mathsf{coffee}}.\mathsf{CM}$, where the process CM performs coin and coffee actions, and then behaves like CM again.

A process can choose the action that will perform among several actions (non-deterministic behavior) and this is described by operator $+$. In order to describe a coffee or tea vending machine we can use this operator: $\mathsf{CTM} \overset{\mathsf{def}}{=} \mathsf{coin}.(\overline{\mathsf{coffee}}.\mathsf{CTM}$ $+\overline{\mathsf{tea}}.\mathsf{CTM})$, where, after performing coin action, it can perform either the $\overline{\mathsf{coffee}}$ or the $\overline{\mathsf{tea}}$ actions.

We can describe the behavior of a student that gets some coffee from a vending machine: $\mathsf{S} \overset{\mathsf{def}}{=} \overline{\mathsf{study}}.\overline{\mathsf{coin}}.\mathsf{coffee}.\mathsf{S}$. The student can only get a coffee if he interacts with CM. In order to describe systems consisting of two or more processes running at the same time (parallel behavior), and possibly interacting (synchronizing) with each other, CCS offers the *parallel composition operation* $|$. The expression $\mathsf{CM}|\mathsf{S}$ describes the system where the CM and the S processes are running in parallel. They can interact through their complementary channels named coin and coffee. The S and CM processes have the possibility to communicate in the parallel composition $\mathsf{CM}|\mathsf{S}$, but we do not require that they must communicate with each other. Both processes could use their complementary channels to communicate with other processes in their environment. We can avoid the communication with these other processes through coin and coffee channels using the restriction operator $\backslash$, whose aim is to limit the scope of channel names. For instance, using the operation $\backslash\{\mathsf{coin}, \mathsf{coffee}\}$, we hide the coin and coffee channels from the environment of CM and S processes: $(\mathsf{CM}|\mathsf{S})\backslash\{\mathsf{coin}, \mathsf{coffee}\}$.

During the execution of a CCS process, each time an action is performed, the state is changed. This is called *transition*. The behavior of a CCS process is given by a set of computations that this process can carry out. A computation of a process is a sequence of transitions. A transition is described by an in-state, an out-state and a label α. The in-state (out-state) is the process state before (after) performing an action. The α label describes the action that was performed and can be an output action: $\bar{\mathsf{a}}$; an input action: a; or a synchronization action: τ, where a is a channel. The synchronization actions represent internals executions and are not visible to the environment.

3 Regulation of Lactose Operon

The lactose operon contains three genes related to lactose metabolism. The *lac* Z, Y and A genes encode β-galactosidase, galactoside permease and thiogalactoside transacetylase, respectively. β-galactosidase converts lactose to galactose and glucose or, by transglycosylation, to allolactose. Galactoside permease transports lactose into the cell and thiogalactoside transacetylase appears to modify toxic galactosides to facilitate their removal from the cell.

In the absence of lactose, the *lac* operon genes are repressed — in fact, they are transcribed at a basal level. This negative regulation is done by a molecule called Lac repressor, which binds to some sites near the start of the operon, blocking the activity of RNA polymerase. These sites are called operators. The operator to which the repressor binds most tightly is named O_1. The *lac* operon has two secondary binding sites for the Lac repressor: O_2 and O_3. To repress the operon, the Lac repressor binds to both the main operator and one of the two secondary sites.

When cells are provided with lactose, the *lac* operon is induced. An inducer (signal) molecule binds to a specific site on the Lac repressor, causing a conformational change that results in dissociation of the repressor from the operators. The inducer in the *lac* operon system is allolactose, an isomer of lactose. When unrepressed, transcription of *lac* genes is increased, but not at its higher level.

Other factors besides lactose affect the expression of the *lac* genes, such as the availability of glucose — the preferred energy source of bacteria. Other sugars can serve as the main or sole nutrient, but extra steps are required to prepare them for entry into glycolysis, necessitating the synthesis of additional enzymes.

Clearly, expressing the genes for proteins that metabolize sugars such as lactose is wasteful when glucose is abundant.

The *lac* operon deals with it through a positive regulation. A regulation mechanism known as catabolite repression restricts expression of the genes required for catabolism of lactose in the presence of glucose, even when this secondary sugar are also present. The effect of glucose is mediated by cAMP, as a coactivator, and an activator protein known as cAMP receptor protein, or CRP (sometimes it is called CAP, for catabolite gene activator protein). CRP has binding sites for DNA and cAMP. When glucose is absent, CRP-cAMP binds to a site near the *lac* promoter and stimulates RNA transcription. CRP-cAMP is therefore a positive regulatory element responsive to glucose levels, whereas the Lac repressor is a negative regulatory element responsive to lactose. The two act in concert. CRP-cAMP has little effect on the *lac* operon when the Lac repressor is blocking transcription, and dissociation of the repressor from the *lac* operator has little effect on transcription of the *lac* operon unless CRP-cAMP is present to facilitate transcription; when CRP is not bound, the wild-type *lac* promoter is a relatively weak promoter.

The effect of glucose on CRP is mediated by the cAMP interaction. CRP binds to DNA most avidly when cAMP concentrations are high. In the presence of glucose, the synthesis of cAMP is inhibited and efflux of cAMP from the cell is stimulated. As cAMP declines, CRP binding to DNA declines, thereby decreasing the expression of the *lac* operon. Strong induction of the *lac* operon therefore requires both lactose (to inactivate the Lac repressor) and a lowered concentration of glucose (to trigger an increase in cAMP and increase binding of cAMP to CRP).

There are also another level of regulation called inducer exclusion. The key molecular component in this exclusion is the PTS transport system, a complex of proteins in the bacterial membrane, which phosphorylates and transports sugars into the cell. One of the proteins of this complex (IIA^{Glc}) becomes dephosphorylated as a result of glucose transport. It then binds to galactoside permease and prevents it from importing lactose into the cell [13].

Now we can specify some known properties of lactose operon regulation to be verified in our formal model of the system — shown in Fig. 1.

4 Modeling Regulation of Lactose Operon

The cellular concentration of a protein is determined by a delicate balance of at least seven activities, each having several potential points of regulation:

1. Synthesis of the primary RNA transcript (transcription);
2. Posttranscriptional modification of mRNA;
3. Messenger RNA degradation;
4. Protein synthesis (translation);
5. Posttranslational modification of proteins;
6. Protein targeting and transport;
7. Protein degradation.

A) When glucose concentration is high, will cAMP concentration be low? (yes)

B) When glucose concentration is low, will cAMP concentration be high? (yes)

C) Can cAMP bind to CRP? (yes)

D) When cAMP concentration is low, will it bind to CRP? (no)

E) When cAMP concentration is high, will it bind to CRP? (yes)

F) Can cAMP-CRP complex bind to CRP site? (yes)

G) When cAMP-CRP complex binds to CRP site, will there be an activation of *lac* operon? (yes)

H) Will external lactose interact with galactoside permease? (yes)

I) Will external lactose enter the cell? (yes)

J) Will intracellular lactose react with β-galactosidase? (yes)

K) Can allolactose only be a product of a reaction mediated by β-galactosidase? (yes)

L) Can allolactose be produced when lactose is available? (yes)

M) Will allolactose bind to Lac repressor? (yes)

N) Will *lac* operon be unrepressed when allolactose is bound to Lac repressor? (yes)

O) Can lac repressor bind to operator 1? (yes)

P) Can lac repressor unbind operator 1? (yes)

Q) Can lac repressor bind to operator 2? (yes)

R) Can lac repressor unbind operator 2? (yes)

S) Can lac repressor bind to operator 3? (yes)

T) Can lac repressor unbind operator 3? (yes)

U) Will lac repressor never be bound to operator 2 and 3 simultaneously? (yes)

V) If cAMP-CRP complex is bound to CRP site and lactose operators are free, will the concentration of β-galactosidase increase? (yes)

W) Can glucose only be a product of a reaction mediated by β-galactosidase? (yes)

X) After β-galactosidase concentration is increased, will be possible an increase in glucose concentration? (yes)

Y) After glucose concentration is increased, will be possible an decrease in β-galactosidase concentration? (yes)

Fig. 1. Known Properties of Lactose Operon Regulation

Our model focuses on the regulation of transcription initiation of *lac* operon. We don't include the inducer exclusion in our model to avoid a great number of processes related to glucose transport — which isn't our object of study here. Concerning this we have made some adaptations in the model to abstract some activities that aren't (or aren't known to be) directly related to *lac* regulation, like RNA polymerase activity, decrease of glucose level by cellular activity, production of galactose by metabolism of lactose mediated by β-galactosidase and presence of thiogalactoside transacetylase. Our model of the system is shown in Fig. 2 and Fig. 3.

The modeling uses channel synchronization via lowercase names (silent action) to model relevant biological events and, right after it, message sending (output action) to allow the analysis of system behavior using model checking tools — we need to have output actions (uppercase names) that are observable outside the system to verify properties.

1) System $\overset{\text{def}}{=}$ (Lactose_out|Galactoside_permease|Lactose_in_none|
Allolactose_none|Beta_galactosidase_low|Operator1|
Operator2|Operator3|Lac_repressor_off|CRP_site|CRP_off|
Promoter_iu|C_AMP_low|Glucose_high)\{elac, ilac, rbeta,
iallo, ibeta, iglu, dbeta, ballo, bo1, bo2, rep, bo3, ubo1,
ubo23, act, urep, iact, bs, ubs, bc, ubc, lev, low, high}

2) Lactose_out $\overset{\text{def}}{=}$ elac.$\overline{\text{ilac}}$.$\overline{\text{ILAC}}$.Lactose_out

3) Galactoside_permease $\overset{\text{def}}{=}$ $\overline{\text{elac}}$.$\overline{\text{ELAC}}$.Galactoside_permease

4) Lactose_in_none $\overset{\text{def}}{=}$ ilac.Lactose_in_low

5) Lactose_in_low $\overset{\text{def}}{=}$ ilac.(Lactose_in_low + Lactose_in_high)
+$\overline{\text{rbeta}}$.(Lactose_in_low + Lactose_in_none)

6) Lactose_in_high $\overset{\text{def}}{=}$ $\overline{\text{rbeta}}$.(Lactose_in_low + Lactose_in_high)

7) Beta_galactosidase_low $\overset{\text{def}}{=}$ rbeta.$\overline{\text{RBETA}}$.$\overline{\text{iallo}}$.$\overline{\text{IALLO}}$.Beta_galactosidase_low
+ibeta.$\overline{\text{IBETA}}$.Beta_galactosidase_high

8) Beta_galactosidase_high $\overset{\text{def}}{=}$ rbeta.$\overline{\text{RBETA}}$.$\overline{\text{iglu}}$.$\overline{\text{IGLU}}$.Beta_galactosidase_high
+dbeta.$\overline{\text{DBETA}}$.Beta_galactosidase_low

9) Allolactose_none $\overset{\text{def}}{=}$ iallo.Allolactose_low

10) Allolactose_low $\overset{\text{def}}{=}$ iallo.Allolactose_low + ballo.Allolactose_none

11) Lac_repressor_off $\overset{\text{def}}{=}$ $\overline{\text{bo1}}$.$\overline{\text{BO1}}$.($\overline{\text{bo2}}$.$\overline{\text{BO2}}$.$\overline{\text{rep}}$.$\overline{\text{REP}}$.Lac_repressor_on
+$\overline{\text{bo3}}$.$\overline{\text{BO3}}$.$\overline{\text{rep}}$.$\overline{\text{REP}}$.Lac_repressor_on)

12) Lac_repressor_on $\overset{\text{def}}{=}$ $\overline{\text{ballo}}$.$\overline{\text{BALLO}}$.ubo1.ubo23.$\overline{\text{urep}}$.$\overline{\text{UREP}}$.Lac_repressor_off

Fig. 2. CCS Specification of Lactose Operon Regulation (Part 1)

Sometimes we also need a qualitative measure of substance concentration (or activity) to choose the right behavior for it. So we can have more than one process description to each substance in the model[1]. They are all related since all descriptions can be reached by some channel synchronization.

Our main process is called System (Fig. 2). It contains all relevant processes to *lac* regulation running in parallel. The channel names listed in its description (lowercase names) are restricted to the processes inside it. We start our system with lactose outside the cell, no intracellular lactose and allolactose, a few galactoside permease and β-galactosidase enzymes, high glucose level, low cAMP concentration, all regulatory sites for *lac* operon released and some CRP and Lac repressor proteins[2].

Lactose can be outside or inside cell in our model. For external lactose (Fig. 2) we have the process Lactose_out, which can interact with permease (elac channel) and after that can enter the cell ($\overline{\text{ilac}}$ channel). Intracellular lactose are modeled using three qualitative levels: none, low and high (process descriptions 4 to 6 in Fig. 2). When lactose is available inside cell it can react with β-galactosidase

[1] Experimental results available in [13] were used to choose the number of process descriptions to each substance

[2] Process are all qualitative, that is, one process doesn't means one unit of a substance.

13) $\mathsf{Operator1} \overset{\mathsf{def}}{=} \mathsf{bo1}.\overline{\mathsf{ubo1}}.\overline{\mathsf{UBO1}}.\mathsf{Operator1}$

14) $\mathsf{Operator2} \overset{\mathsf{def}}{=} \mathsf{bo2}.\overline{\mathsf{ubo23}}.\overline{\mathsf{UBO2}}.\mathsf{Operator2}$

15) $\mathsf{Operator3} \overset{\mathsf{def}}{=} \mathsf{bo3}.\overline{\mathsf{ubo23}}.\overline{\mathsf{UBO3}}.\mathsf{Operator3}$

16) $\mathsf{Promoter\_iu} \overset{\mathsf{def}}{=} \mathsf{rep}.\mathsf{Promoter\_ir} + \mathsf{act}.\overline{\mathsf{ibeta}}.\mathsf{Promoter\_au}$

17) $\mathsf{Promoter\_ir} \overset{\mathsf{def}}{=} \mathsf{urep}.\mathsf{Promoter\_iu} + \mathsf{act}.\mathsf{Promoter\_ar}$

18) $\mathsf{Promoter\_au} \overset{\mathsf{def}}{=} \mathsf{rep}.\overline{\mathsf{dbeta}}.\mathsf{Promoter\_ar} + \mathsf{iact}.\overline{\mathsf{dbeta}}.\mathsf{Promoter\_iu}$

19) $\mathsf{Promoter\_ar} \overset{\mathsf{def}}{=} \mathsf{urep}.\overline{\mathsf{ibeta}}.\mathsf{Promoter\_au} + \mathsf{iact}.\mathsf{Promoter\_ir}$

20) $\mathsf{CRP\_site} \overset{\mathsf{def}}{=} \mathsf{bs}.\overline{\mathsf{ubs}}.\mathsf{CRP\_site}$

21) $\mathsf{CRP\_off} \overset{\mathsf{def}}{=} \mathsf{bc}.\overline{\mathsf{bs}}.\overline{\mathsf{BS}}.\overline{\mathsf{act}}.\overline{\mathsf{ACT}}.\mathsf{CRP\_on}$

22) $\mathsf{CRP\_on} \overset{\mathsf{def}}{=} \mathsf{ubc}.\mathsf{ubs}.\overline{\mathsf{UBS}}.\overline{\mathsf{iact}}.\overline{\mathsf{IACT}}.\mathsf{CRP\_off}$

23) $\mathsf{C\_AMP\_low} \overset{\mathsf{def}}{=} \overline{\mathsf{lev}}.(\mathsf{low}.\overline{\mathsf{L\_to\_H}}.\overline{\mathsf{bc}}.\overline{\mathsf{BC}}.\mathsf{C\_AMP\_high} + \mathsf{high}.\mathsf{C\_AMP\_low})$

24) $\mathsf{C\_AMP\_high} \overset{\mathsf{def}}{=} \overline{\mathsf{lev}}.(\mathsf{low}.\mathsf{C\_AMP\_high}$
$+\mathsf{high}.\overline{\mathsf{H\_to\_L}}.\overline{\mathsf{ubc}}.\overline{\mathsf{UBC}}.\mathsf{C\_AMP\_low})$

25) $\mathsf{Glucose\_high} \overset{\mathsf{def}}{=} \mathsf{iglu}.\mathsf{Glucose\_high} + \mathsf{lev}.\overline{\mathsf{high}}.\mathsf{Glucose\_high}$
$+\overline{\mathsf{DGLU}}.\mathsf{Glucose\_low}$

26) $\mathsf{Glucose\_low} \overset{\mathsf{def}}{=} \mathsf{iglu}.(\mathsf{Glucose\_low} + \overline{\mathsf{GLU\_L\_to\_H}}.\mathsf{Glucose\_high})$
$+\mathsf{lev}.\overline{\mathsf{low}}.\mathsf{Glucose\_low}$

Fig. 3. CCS Specification of Lactose Operon Regulation (Part 2)

enzyme ($\overline{\mathsf{rbeta}}$ channel) and its level can decrease. While there isn't high concentration of lactose, it can enter the cell (ilac channel) and its level can increase.

Galactoside_permease (Fig. 2) process just allows the entrance of lactose in the cell ($\overline{\mathsf{elac}}$ channel). The activation of *lac* operon doesn't affect it in our model and, therefore, we don't use concentration levels for it because the only change is in the rate of lactose entering the cell at a given time.

β-galactosidase (processes 7 and 8 in Fig. 2) has two levels: low and high, which are affected by activation and repression of *lac* operon. From low to high concentration we have ibeta channel and from high to low level dbeta channel. When reacting with lactose (rbeta), this enzyme, at low level, can produce allolactose ($\overline{\mathsf{iallo}}$) or, at high level, can produce glucose ($\overline{\mathsf{iglu}}$) and galactose[3]. Since galactose doesn't participate in *lac* regulation we don't include it in our model.

Allolactose (processes 9 and 10 in Fig. 2) can be present at low concentration in the cell or can be absent. When absent, the only action the process can perform is its increase (iallo). When present, besides its production, it can bind to Lac repressor (ballo). After binding, its concentration will reduce.

Lac repressor can be bound (on) or unbound (off) to operators (processes 11 and 12 in Fig. 2). When unbound, it can bind to O_1 ($\overline{\mathsf{bo1}}$) and either O_2 ($\overline{\mathsf{bo2}}$) or O_3 ($\overline{\mathsf{bo3}}$). After that, the *lac* promoter will be repressed ($\overline{\mathsf{rep}}$). When allolactose

[3] We have restricted reaction products at low and high levels in our model because we want to include a preferential product according to enzyme level.

binds to Lac repressor ($\overline{\textsf{ballo}}$), it unbinds the operators (ubo1 and ubo23) and unrepresses the promoter ($\overline{\textsf{urep}}$).

All operator sites (processes 13, 14 and 15 in Fig. 3) can only bind to Lac repressor (bo1, bo2 and bo3) and after that can only unbind from it ($\overline{\textsf{ubo1}}$ and $\overline{\textsf{ubo23}}$).

The *lac* promoter has four states: iu for inactivated and unrepressed, ir for inactivated and repressed, au for activated and unrepressed and ar for activated and repressed (processes 16, 17, 18 and 19 in Fig. 3, respectively). These processes can change the concentration of β-galactosidase from low to high ($\overline{\textsf{ibeta}}$) after it is activated (act) and unrepressed (urep) and from high to low ($\overline{\textsf{dbeta}}$) after it is inactivated (iact) or repressed (rep)[4]. The $\overline{\textsf{ibeta}}$ synchronization abstracts several biological events to one — all transcription and translation steps between operon activation and β-galactosidase production. We don't increase all *lac*-related proteins concentrations to keep only relevant information in our model.

CRP_site (Fig. 3) process can only bind to CRP-cAMP complex (bs) and after that can unbind from it ($\overline{\textsf{ubs}}$).

CRP can be free in the cell (process 21 in Fig. 3) or bound at CRP site (process 22 in Fig. 3). When free, it can bind to cAMP (bc). After that, it binds to CRP site ($\overline{\textsf{bs}}$) and activates the *lac* promoter ($\overline{\textsf{act}}$). When bound, it can unbind cAMP (ubc) and, after that, it unbinds CRP site (ubs) and inactivates the promoter ($\overline{\textsf{iact}}$).

We can have low or high cAMP levels (processes 23 and 24 in Fig. 3). Changes in cAMP level depends on the glucose concentration[5]. So, our cAMP processes always ask glucose its level ($\overline{\textsf{lev}}$). According to the answer (low or high), it can change its concentration. If its level is raised ($\overline{\textsf{L\_to\_H}}$), cAMP binds to CRP ($\overline{\textsf{bc}}$) to start activation of *lac* operon. If its level is reduced ($\overline{\textsf{H\_to\_L}}$), cAMP unbinds CRP ($\overline{\textsf{ubc}}$) and deactivates *lac* operon.

Glucose concentration can be at high or low levels (processes 25 and 26 in Fig. 3). Glucose can have an increase in its concentration via β-galactosidase mediated reaction (iglu) or can be asked for its level by cAMP process (lev followed by $\overline{\textsf{high}}$ or $\overline{\textsf{low}}$). Glucose level can be increased ($\overline{\textsf{GLU\_L\_to\_H}}$) or decreased ($\overline{\textsf{DGLU}}$). We signal these changes in glucose concentration to facilitate the verification of some properties related to glucose influence in *lac* regulation. The decrease of glucose concentration without any apparent reason in process 25 occurs to avoid the usage of more processes in our model for consuming glucose. Instead of it, we abstract the consumption of energy using the $\overline{\textsf{DGLU}}$ channel.

5 Properties Verification of the System

The approach we use to verification of properties of our system is called model checking. This was done using the tool Concurrency Workbench of the New

[4] In fact, we have intermediary levels of transcription (from basal to highest), but for the sake of simplicity we model only basal rate — low — and highest rate — high.

[5] How glucose level affects cAMP level is not entirely known yet [13].

Century [14]. In this approach, we describe the system using CCS (specification language) and the properties using temporal logic.

A temporal logic is an extension of regular predicate logic with modalities and enduring capabilities. This logic gives us the potential to reason about properties for different computations at the same time, not just properties for one computation. Moreover, we can describe properties like "the property is *always* possible" or "the action a will *eventually* happen", that is, valid for several states of the computations.

We validated our formal model by checking the properties in Fig. 1 that are known for the regulation of the lactose. We described these properties using a logic called Computation Tree Logic (CTL) [15], checked them and the obtained results agreed with the known answers. Thus, we showed the viability of using process algebras to model and reason about biochemical networks.

In Fig. 4 some selected properties are shown. The remaining properties were omitted because their formulae are similar to one of those depicted in Fig. 4.

Each CTL operator has a meaning that can translated into an English sentence. The illustrated formulae in Fig. 4 can be translated as follows:

A) Exists one state at one computation where, between $\overline{\mathsf{GLU\_L\_to\_H}}$ and $\overline{\mathsf{DGLU}}$, occurs $\overline{\mathsf{H\_to\_L}}$;

C) Exists one state at one computation where $\overline{\mathsf{BC}}$ will occur;

D) It is similar to A. We selected this property because its result is different from A result;

K) Exists one state at one computation where $\overline{\mathsf{IALLO}}$, followed (preceded) or not by silent actions (τ), occurs after $\overline{\mathsf{RBETA}}$ and does not exist one computation where $\overline{\mathsf{IALLO}}$ occurs before $\overline{\mathsf{RBETA}}$;

L) Exists one state at one computation where $\overline{\mathsf{IALLO}}$, followed (preceded) or not by silent actions (τ), occurs after $\overline{\mathsf{ILAC}}$;

U) For all states at all computations, $\overline{\mathsf{BO2}}$ and $\overline{\mathsf{BO3}}$ do not occur one after another and, at some time, they occur between $\overline{\mathsf{BO1}}$ and $\overline{\mathsf{UBO1}}$.

When we checked our model we were faced to the state explosion problem, where the automata related to the model description have a great number of states. We dealt this problem by adapting our model to each property. First

A) $\mathsf{EF}(\langle\overline{\mathsf{GLU\_L\_to\_H}}\rangle\langle\langle\overline{\mathsf{H\_to\_L}}\rangle\rangle\langle\overline{\mathsf{DGLU}}\rangle\mathsf{tt})$	(yes)
C) $\mathsf{EF}\langle\overline{\mathsf{BC}}\rangle\mathsf{tt}$	(yes)
D) $\mathsf{EF}(\langle\overline{\mathsf{H\_to\_L}}\rangle\langle\langle\overline{\mathsf{BC}}\rangle\rangle\langle\overline{\mathsf{L\_to\_H}}\rangle\mathsf{tt})$	(no)
K) $\mathsf{A}(\neg\langle\overline{\mathsf{IALLO}}\rangle\mathsf{tt}\ \mathsf{W}\ \langle\overline{\mathsf{RBETA}}\rangle\mathsf{tt}) \wedge \mathsf{EF}(\langle\overline{\mathsf{RBETA}}\rangle\langle\langle\overline{\mathsf{IALLO}}\rangle\rangle\mathsf{tt})$	(yes)
L) $\mathsf{EF}(\langle\overline{\mathsf{ILAC}}\rangle\langle\langle\overline{\mathsf{IALLO}}\rangle\rangle\mathsf{tt})$	(yes)
U) $\mathsf{AG}((\neg\langle\overline{\mathsf{BO2}}\rangle\langle\langle\overline{\mathsf{BO3}}\rangle\rangle\mathsf{tt}) \wedge (\neg\langle\overline{\mathsf{BO3}}\rangle\langle\langle\overline{\mathsf{BO2}}\rangle\rangle\mathsf{tt}))\wedge$ $\mathsf{EF}(\langle\overline{\mathsf{BO1}}\rangle\langle\langle\overline{\mathsf{BO2}}\rangle\rangle\langle\overline{\mathsf{UBO1}}\rangle\mathsf{tt})\wedge$ $\mathsf{EF}(\langle\overline{\mathsf{BO1}}\rangle\langle\langle\overline{\mathsf{BO3}}\rangle\rangle\langle\overline{\mathsf{UBO1}}\rangle\mathsf{tt})$	(yes)

Fig. 4. CTL Formulae

of all, we got rid of every non-restricted channel from the model in Fig. 2 and Fig. 3. Thereafter, for each property we included only the non-restricted channels related to it. For instance, the model used to property A contains only $\overline{\mathsf{GLU\_L\_to\_H}}$, $\overline{\mathsf{H\_to\_L}}$ and $\overline{\mathsf{DGLU}}$ channels, exactly at the same places depicted in our model.

6 Conclusion

In this paper we have shown a model of lactose operon regulation using CCS. So, we obtained a formal description for this regulatory system that can be analyzed to verify system properties using model checking techniques and tools. We verify the validity of our model by checking known properties of *lac* regulation and, because of it, we gain confidence to verify other properties of biological systems.

Yildirim and Mckey followed this way to validate their model — nonlinear differential delay equations. But they have found much more data because they have a quantitative model. This kind of model only allows simulation and the discovery of some steady states in the system. They relate that no full stability analysis of steady states was possible in their model[10]. When we use qualitative models, we lose some accuracy to make possible the analysis of system structure.

This was done by Ciobanu, Ciubotariu and Tanasă for Albers-Post mechanism to ion transport across membrane — Na pump. They used π-calculus, a process algebra, to accomplish this. But they have made only deadlock[6] checking in their system[9].

When dealing with process algebra modeling we very often be faced to the state explosion problem, where the automata related to the model description have a great number of states. In our model we have to adapt the processes to each verified property because of this problem — see section 5.

Chabrier-Rivier and colleagues have modeled Kohn's compilation on the mammalian cell-cycle control in a new modeling language[11]. They have used CTL to check system properties related to metabolic pathways. But they don't have an automatic way to transform its own language in a formalism that have available model checking tools.

To proceed our work, some future steps are the inclusion of other kinds of interaction (such as metabolism) and information about where these interactions occur in the cell. Besides, we want to make use of MONET database [16] to translate biological data into CCS language (or another process algebra) in a semi-automatic way. This task can be accomplished by user selection of some data sets from MONET — up to now they have metabolic pathways available. Regulatory and signaling pathways must be fulfilled to make possible the future automation of this task.

A good model must take all relevant biological information into account, and present results that are compatible with the ones reached *in vitro*. The main goal of our work is to do analysis of biochemical processes. Using the CCS and CTL

[6] A process deadlocks if its transition system has states with no successors.

we may be able to verify some properties of biological systems, shedding light on relevant questions of pathways, such as the possibilities of energy generation given a certain substance in the cell.

References

1. Voit, E. O. (2000). *Computational analysis of biochemical systems.* Cambridge University Press, United Kingdom.
2. Jong, H. D. (2002). Modeling and Simulation of Genetic Regulatory Systems: a literature review. *Journal of Computational Biology*, **9**(1):67–103.
3. Antoniotti, M., Policriti, A., Ugel, N. and Mishra, B. (2003). Model Building and Model Checking for Biochemical Processes. *Cell Biochem. Biophys.*, **38**:271–286.
4. Reddy, V. N. (1994). *Modeling Biochemical Pathways: a discrete event systems approach.* Master's thesis, University of Maryland.
5. Lemke, N., Herédia, F., Barcellos, C. K., Reis, A. N. and Mombach, J. C. M. (2004). Essentiality and Damage in Metabolic Networks. *Bioinformatics*, **20**(1):115–119.
6. Nelson, D. L. and Cox, M. M. (2004). *Lehninger Principles of Biochemistry.* 4th Edition, W. H. Freeman and Co., New York.
7. Jacob, F. and Monod, J. (1961). Genetic Regulatory Mechanisms in the Synthesis of Proteins. *Journal of Molecular Biology*, **3**:318–389.
8. Regev, A., Silverman, W. and Shapiro, E. (2001). Representation and Simulation of Biochemical Processes using the π-Calculus Process Algebra. In *Pacific Symposium on Biocomputing*, **6**:459–470, World Scientific Press, Singapore.
9. Ciobanu, G., Ciubotariu, V. and Tanasă, B. (2002) A π-Calculus Model of the Na Pump. *Genome Informatics*, **13**:469–471.
10. Yildirim, N. and Mackey, M. C. (2003). Feedback Regulation in the Lactose Operon: a mathematical modeling study and comparison with experimental data. *Biophysical Journal*, **84**:2841–2851.
11. Chabrier-Rivier, N., Chiaverini, M., Danos, V., Fages, F. and Schächter, V. (2004). Modeling and Querying Biomolecular Interaction Networks. *Theoretical Computer Science*, **325**:25–44.
12. Milner, R. (1989). *Communication and Concurrency.* Prentice Hall, New York.
13. Lewin, B. (2004). *Genes VIII.* Pearson Prentice Hall, USA.
14. The CWB-NC website. http://www.cs.sunysb.edu/˜cwb/.
15. Clarke, E. M., Emerson, E. A. and Sistla, A. P. (1986). Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, **8**(2):244–263.
16. The MONET website. http://www.inf.unisinos.br/˜lbbc/monet.

YAMONES: A Computational Architecture for Molecular Network Simulation

Guilherme Balestieri Bedin and Ney Lemke

UNISINOS, Av. Unisinos 950, São Leopoldo RS, Brazil
lemke@exatas.unisinos.br
http://www.inf.unisinos.br/~lbbc

Abstract. One of the most important challenges for Bioinformatics is the simulation of a single cell, even if we restrict ourselves to simple models of the molecular networks responsible for the behavior of organisms. The challenge involves not only the development of experimental techniques to obtain kinetic parameters that characterize the myriad reactions occurring inside cells, but also computational approaches able to simulate and test the complex models generated. These systems have stochastic behavior; they can take different paths depending on environmental conditions. We can describe them using stochastic models that have a high computational cost, but the simulations can be performed efficiently on distributed architectures like grids and clusters of computers. In this work we describe an implementation of a computational architecture to execute this kind of large scale simulation using a grid infrastructure. We validate the proposed architecture using experiments in order to estimate its performance.

1 Introduction

Bioinformatics has changed from a set of tools to store, manage, visualize and make accessible biological data to a key part of the Biosciences, by developing its own approach to understand the intricacies of life. This approach uses theoretical concepts originating from computer science, such as information and computation, to make sense out of plethora of biological data.

One of the most important challenges in this scenario is the simulation of a single cell, even if we restrict ourselves for simple models of its biochemical networks responsible for the behavior of organisms. The challenge involves not only the development of experimental techniques to obtain kinetical parameters that characterizes the myriad of reactions occurring inside cells, but also computational approaches able to simulate and test the complex models generated. Even though we do not have yet a complete model for even the most simple organism, we know that the computational cost will be high. Given the stochastic nature of the models used in these problems, the algorithms involved can be classified as embarrassing parallel and computer grids can be used efficiently.

In the last years we have witnessed many different approaches to investigate sets of biochemical reactions including new algorithms [1], [2], [3], computational

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 107–117, 2005.

architectures [4], and generic frameworks such as SBW (`http://www.sys-bio.org`) and Biospice (`http://www.biospice.org`). Here, we will not compare these approaches, but instead we will present a new architecture and test its performance.

In this article we describe a grid architecture to run large scale simulations of molecular networks called YAMONES (Yet Another MOlecular NEtwork Simulator). The architecture explores intra and extra-node parallelism considering specific properties of each node. The simulations use different algorithms, including both deterministic ones based on the resolution of Differential Equations and fully stochastic ones based on Gillespie algorithms [5].

This article is divided as follows: in section 2 we present essential concepts about molecular networks, in section 3 we describe the algorithms implemented on YAMONES, in section 4 we discuss the architecture of the system, on section 5 we analyze the architecture performance and on section 6 we draw our conclusions.

2 Molecular Network Models

When we consider an organism as a whole, its metabolism represents all the chemical processes occurring inside its cells. When you consider a given substance, the metabolism is the chemical activity involving this substance on a living organism, when we consider a specific cell the metabolism is the set of all processes occurring in that cell.

The metabolism can make complex chemical conversions in a series of small steps, each step (reaction) is catalyzed by a specific enzyme. The enzymes are also metabolic products, but as catalysts they can be present in small quantities.

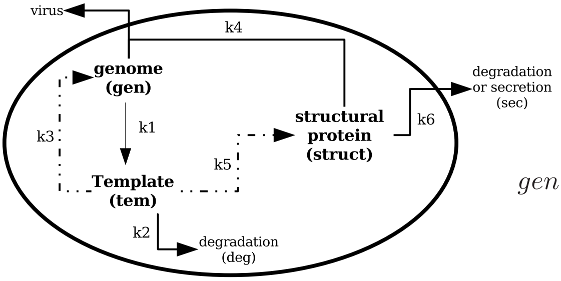
Chemical reactions are the canonical language of biological modeling. Consider the reaction:

$$n_a A + n_b B \xrightarrow{k} n_c C + n_d D. \tag{1}$$

In this example, n_a molecules of the chemical specie A react with n_b molecules of species B and they are transformed into n_c molecules of species C and n_d molecules of species D. The terms on the left are called substrates and the terms on the right are called products. Each reaction can contain an indefinite number of substrates and products. k is the velocity at which the reaction occurs, and its value depends on temperature and volume. The n values are called stoichiometric coefficients.

2.1 Intracellular Viral Kinetics

A simple virus metabolic network model, like the one in Figure 1, was developed to explore differences between the deterministic and stochastic model implementations [6]. The components studied were the viral nucleic acids and a viral structural protein (*struct*). The viral nucleic acids were classified as genomic



$$gen \xrightarrow{k_1} tem \quad (2)$$

$$tem \xrightarrow{k_2} deg \quad (3)$$

$$tem \xrightarrow{k_3} gen \quad (4)$$

$$gen + struct \xrightarrow{k_4} virus \quad (5)$$

$$tem \xrightarrow{k_5} struct \quad (6)$$

$$struct \xrightarrow{k_6} sec \quad (7)$$

Fig. 1. Model of viral replication cycle. The *tem* is only used catalytically in the synthesis of *gen* and *struct*. The catalytic reactions are represented by dashed lines

(*gen*) or template (*tem*). The genome, whether DNA, positive-strand RNA, negative-strand RNA, or some other variant, is the vehicle which transports viral information.

The genome can undergo one of two fates. The first possibility is that it may be modified, whether through integration into the host genome or by some other type of processing (e.g. reverse transcription), to form a template. The template refers to the form of the nucleic acid that is transcribed and is involved on the synthesis of the viral components. Alternatively the genome may be packaged within structural proteins to form progeny virus.

2.2 λ Phage Model

The λ phage is a virus that infects *Escherichia coli* cells, the infected cell has two fates: *lyse* the virus replicates inside the cell and dissolves the cell, freeing about 100 new virus on cellular environment or *lysogeny* in this case the virus genetic material is incorporated in the cell genome, and this material replicate each time the cell replicates. A lysogeny protects the cell against future infections. Under certain conditions a lysogeny can be induced, that is, the virus DNA can be removed from the host DNA and the virus replicate and dissolve the host cell [7].

The λ phage has been extensively studied since it is one of the simplest organism with different final states, to which state the system will evolve depends on the level of infection but is essentially a random choice. These characteristics make it a model organism to study the stochastic nature of gene regulation. Its genome has been sequenced and details about its levels of gene expression are known accurately [8], [9], [7] and [10].

Figure 2 shows the two DNA strands of λ phage. The RNA polymerase can start transcription by binding to any of the promoters. Once transcription starts, RNA polymerase walks through the DNA, each time the final position of a gene is achieved a mRNA molecule is produced. When RNA polymerase arrives at a terminator site, the molecule can leave the DNA with a certain probability. The molecule velocity depends on the DNA position and on the presence or not of the N protein bounded to it.

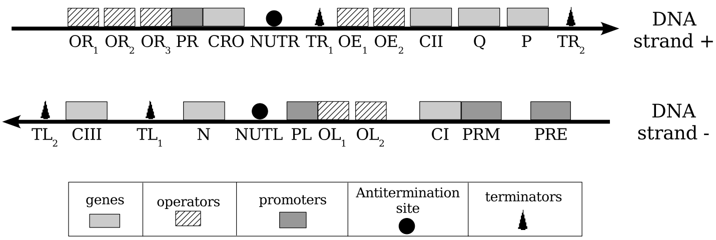


Fig. 2. λ phage DNA and some of its genes and regulatory regions

3 Stochastic Simulation

The time evolution of coupled chemical reactions can be simulated using a stochastic model. Consider the following set of reactions:

$$A + B \xrightarrow{k_1} C \tag{8}$$

$$B + C \xrightarrow{k_2} D \tag{9}$$

$$D + B \xrightarrow{k_3} A \tag{10}$$

To determine the probability of occurrence of a reaction we denote each A molecule as $A_1, A_2, ..., A_{\#A}$, each B molecule as $B_1, B_2, ..., B_{\#B}$, each C molecule as $C_1, C_2, ..., C_{\#C}$ and each D molecule as $D_1, D_2, ..., D_{\#D}$. In this way there are $(\#A) \times (\#B)$ copies of reaction (8), $(\#B) \times (\#C)$ copies of (9) and $(\#D) \times (\#B)$ copies of (10). The copies of reactions (1), (2), (3) have the same occurrence probability since we assume that the chemical system is a homogeneous solution. In this approximation the state of the system is described by the number of molecules of each specie, that is a discrete quantity that changes each time a reaction occurs.

Gillespie [5] proposed two exact stochastic algorithms to simulate systems of r coupled chemical reactions, called the Direct Method and First Reaction Method. Knowing the state of the system, the algorithms determine randomly with the correct probability distribution the next reaction that will occur and when it will occur. The probability of occurrence of a trajectory is the one predicted by the Master Chemical Equation, that is why the algorithms are called exact. When dealing with cells we are in general interested on the behavior of cell populations, in this case we will be interested on average values of the chemical species.

The complexity of Gillespie Methods is $O(rE)$ where r is the number of reactions and E is the number of simulation events [9]. There are other methods that could reduce substantially the computational cost but they are not exact. The Hybrid Method proposed on [11] divides the reactions on two sets, one with the fast reactions and the other with the slow ones. The fast reactions are evolved using deterministic algorithms and are converted into ordinary differential equations, while the slow ones are investigated using stochastic models. The quality

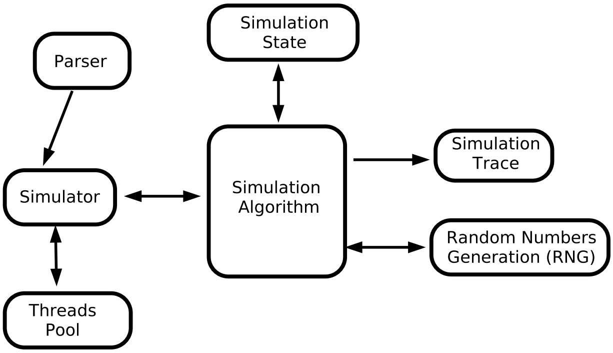


Fig. 3. YAMONES, modules and their interactions

of the approximation on this algorithm depends on a suitable partition criteria that depends on the details of the simulation.

4 Computational Architecture

The YAMONES design divides the main simulation tasks into modules: a configuration file parser (*Parser*), a simulation method computation (*Simulation algorithm*), a simulation trace generation (*Trace*), thread management (*Threads Pool*), random number generation (*RNG*) and simulation state management (*Simulation State*). Figure 3 shows a schematic representation of the modules and their interactions.

The simulator configuration file is in XML. The file is divided into three parts, one with parameters for the simulation method, other specifying the biochemical composites and the last describing the set of chemical reactions. The hybrid method also requires the differential equations for the reactions. These equations are loaded by a dynamic library and are generated automatically from the configuration file.

Complex molecular network models usually have reactions with variable rate. To deploy this feature is possible to specify the dynamic library name and a function that calculate the variable rate in the reaction declaration at the XML configuration file. Another feature of the simulator is the capability of running trajectories in parallel with threads.

4.1 Grid

The Gillespie algorithm is a Monte Carlo algorithm and belongs to the bag of tasks type, being an excellent application for computer clusters and grids [12]. The realizations of a given simulation can be executed asynchronously, the results of N independent trajectories can be consolidated as the result of a single simulation with $N \times T$ realizations, where T is one independent trajectory,

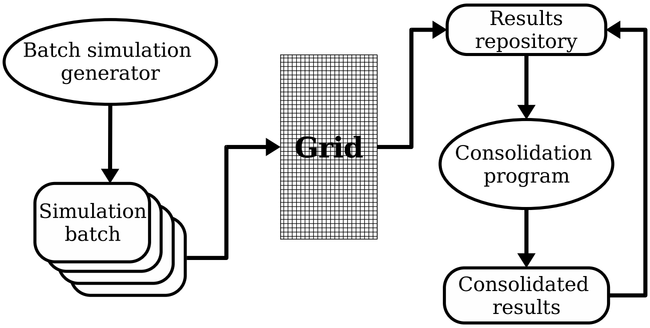


Fig. 4. Execution flow for a grid architecture

reducing the disk space for storing the results of the simulations. To aid in consolidation our architecture implements the merger application that calculates the average of each chemical specie at each time interval.

Figure 4 represents the execution flow of a simulation using YAMONES on a grid architecture. The process is divided in the following steps: preparation of simulation lots, computation of simulations and consolidation of results.

The software that creates the simulations lots is responsible for sending the files to the grid, manage the automatic generation of seeds for the random number generators and periodically verify if the simulation has achieved the stop criteria (predefined precision or number of realizations), if so it stops sending new jobs to the grid, otherwise new jobs are submitted. This execution model is robust to problems on grid nodes. If a node interrupts the simulation of a trajectory, a new job is sent automatically to the grid.

The software which consolidates results also executes periodically, consolidating new results and generating a new file containing the most recent partial results. This file is also used by the software that generates the simulation or by the user to analyze partial simulation results.

The proposed architecture could be implemented using several grid tools. But because we don't have any of them running in our infrastructure, we decided otherwise to validate our proposal by implementing our own scripts in Python and Bash languages. The grids machines can be accessed using Openssh services, that furnishes a complete solution to transfer data and execute jobs on the machines on our architecture. The scripts implement a basic loading balance based on processor clocks. For example, if a node has a clock with frequency v it will compute n_{traj}:

$$n_{traj} = \frac{t_{total}}{\sum v_i} \times v, \tag{11}$$

where t_{total} is the total number of trajectories and $\sum v_i$ is the sum of the clock frequencies of all machines on the grid.

5 Performance Evaluation

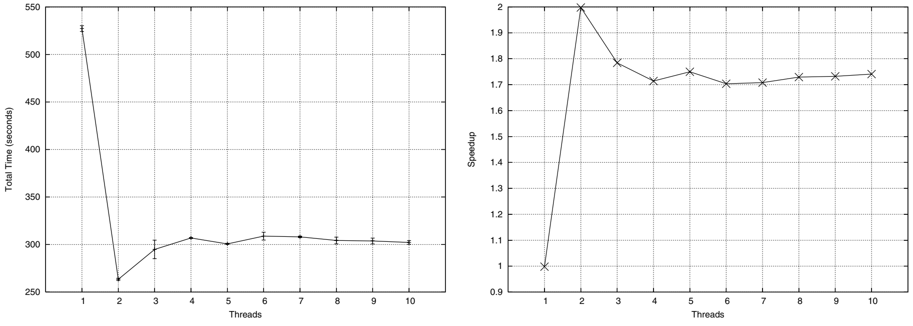
5.1 Intra-node Parallelism

First performance test investigates the number of threads with the best cost-benefit on bi-processed machines with hyperthreading. The target machine hardware configuration is two 2.4 GHz Xeon processors with hyperthread. And software configuration is GNU Linux with 2.6.9 kernel and 2.3.4 glibc.

The number of trajectories are divided equally between all available threads and are computed concurrently. If the division is not exact the rest is also distributed equally between all threads, doing each thread compute the closest number of trajectories possible of each other thread. For example, to calculate 6 trajectories with 3 threads each thread will process 2 trajectories, however if is used 4 threads to compute the same 6 trajectories 2 threads will process 2 trajectories and the other 2 threads will process 1 trajectories each.

Figure 5-(a) presents the dependence of the simulation time on the number of threads for a simulation of 50 trajectories of viral intracellular kinetics with multiplicity of infection 5 using the first reaction method. The best performance is obtained using two threads. The machines with hyperthreading have for each physical processor another logical processor. So there are 4 possible execution flows. Intra-node parallel processing using threads synchronizes when all threads finish their jobs, this allow an efficient exploration of the parallelism.

Figure 5-(b) represents the dependence of the speed-up on the number of threads. The highest value for the speed-up occurs for two threads and is close to two. This result is interesting since we might expect a speed-up of at least 2.6 for bi-processed machines using hyperthreading on Intel CPUs (Intel, `www.intel.com`). Unfortunately the nature of our application, basically the application just executes floating point operations, do not explore the parallel use of the unused CPU resources.



(a) (b)

Fig. 5. (a) Dependence of simulation time to calculate 50 trajectories of the viral intracellular kinetics model on the number of threads on a machine with two processors and hyperthreading. (b) Speed-up in the same situation

5.2 Extra-Node Parallelism

Figure 6 depicted the simulations environment. The grid is composed of two clusters, the first has five machines with two 2.4 GHz Xeon processors and hyper-threading on all processors. They use GNU Linux with kernel 2.6.9, glibc 2.3.4 and Openssh services. The machines are connected by a Gigabit switch. The other cluster has six machines with 1.8 GHz Pentium 4 processor, GNU Linux with kernel 2.6.8, glibc 2.3.4 and Openssh services. The machines are connected by a 100 Megabit switch.

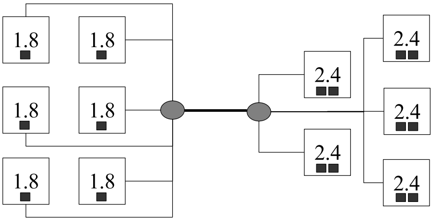


Fig. 6. Grid environment of two clusters; one with five 2.4 GHz two-processor machines and the other with six 1.8 GHz machines

Figure 7 we compare the total simulation time dependence with the number of simulated trajectories on the grid. On this experiment we simulate trajectories for the intracellular infection model with multiplicity of infection 1 using the first reaction method. We can observe that the total simulation time increases linearly with the number of simulated trajectories. The sequential simulation time for 100 trajectories on a 2.4 GHz Xeon processor is 9.8 minutes with a standard deviation of 0.8 minutes, on the grid, the time is 1.31 minutes with a standard deviation of 0.16 minutes. The grid speed-up is 7.5.

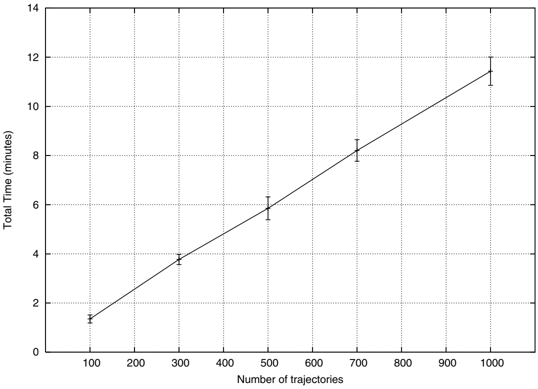
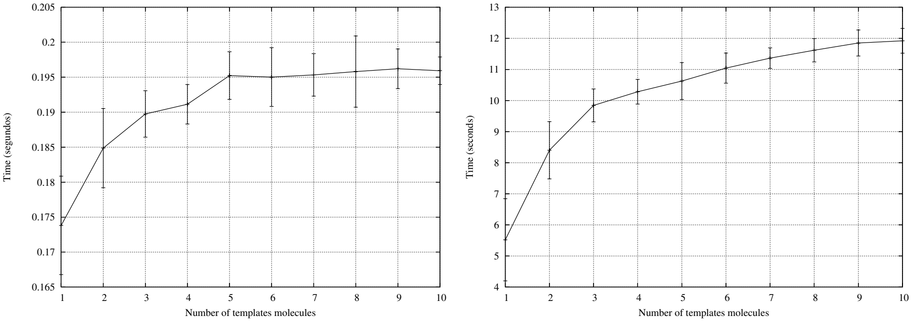


Fig. 7. Dependence of total simulation time on the number of trajectories, on a grid with 11 machines

5.3 Hybrid Method

Figure 8 compares the performance of hybrid and first reaction methods for intracellular viral kinetics for different infection levels. Figures 8-(a) and 8-(b), present, respectively, the performance of the hybrid and first reaction methods.

On both methods there is a monotonical increase of total simulation time with the level of infection. We can observe that the hybrid method is 100 faster than the first reaction method. This performance increase entails a marginal loss of precision. Unfortunately these results are model dependent, and can not be extrapolated to other models.



(a) Hybrid

(b) First Reaction

Fig. 8. Performance comparison of the hybrid (a) and first reaction method (b) using the viral intracellular kinetics at different levels of infection. For each case we simulated 10 trajectories

5.4 λ Phage Model

We applied our architecture also to a realistic model, the λ phage infection model proposed by Arkin on [8]. The description of this complex model, which involves 9000 reactions is beyond the scope of this article. For further details the reader should see the related literature. To generate the list of reactions necessary to describe the model we developed Python scripts.

Figure 9 presents the dependence of the simulation time with the level of infection using the first reaction method, not including the initialization time. The figure shows an almost linear increase on simulation time with the level of infection. The simulations were executed on a 2.4 GHz Xeon processor.

The investigation of this model demands data with statistical significance, we estimate that we need at least $1,000$ trajectories for each level of infection to achieve this goal. Using this environment will be necessary 9 days to obtain a single point, with level of infection 11. In such a case the use of a grid environment is compulsory.

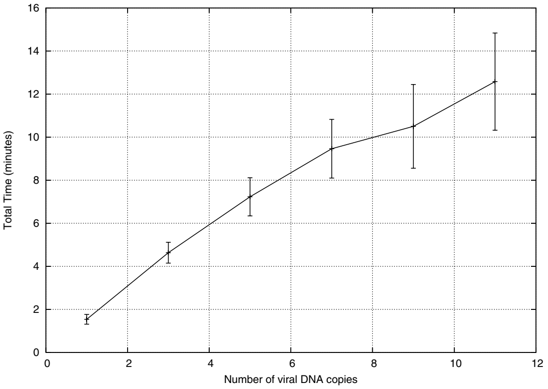


Fig. 9. Simulation time for the λ phage model using the first reaction method

6 Conclusion

In this article we proposed an architecture for simulation of sets of coupled chemical reactions called YAMONES. The performance of this architecture was tested trough a series of experiments showing that we obtain suitable speed-ups on the exploration of intra and extra-node parallelism.

The grid architecture was implemented using scripts written on Python and Bash languages, we plan to extend the model and use some grid tool such as OurGrid and Globus to make further improvements.

Acknowledgements

We acknowledge the support of CNPq and CAPES. This work was developed in collaboration with HP Brazil R&D.

References

1. Dhar, P., Meng, T.C., Somani, S., Ye, L., Sairam, A., Chitre, M., Hao, Z., Sakharkar, K.: Cellware - a multi-algorithmic software for computational system biology. Bioinformatics **20** (2004) 1319–1321
2. Kiehl, T.R., Mattheyses, R.M., Simmons, M.K.: Hybrid simulation of cellular behavior. Bioinformatics **20** (2004) 316–322
3. Gillespie, D.T., Petzold, L.R.: Improved leap-size selection for accelerated stochastic simulation. Chemical Physics **119** (2003) 8229–8234
4. You, L., Hoonlor, A., Yin, J.: Modeling biological systems using dynetica - a simulator of dynamic networks. Bioinformatics **19** (2003) 435–436
5. Gillespie, D.T.: Exact Stochastic Simulation of Coupled Chemical Reactions. J. Phys. Chem. **81** (1977) 2340–2361

6. Srivastava, R., You, L., Yin, J.: Stochastic vs. Deteministic Modeling of Intracellular Viral Kinetics. Theory Biology **218** (2002) 309–321
7. Bower, J.M., Bolouri, H., eds.: Computational modeling of genetic and biochemical networks. Computational molecular biology. MIT Press (2001)
8. Arkin, A., Ross, J., McAdams, H.H.: Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. Genetics **149** (1998) 33–48
9. Gibson, M.A., Bruck, J.: Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels. Physical Chemistry **104** (2000) 1876–1889
10. Santillán, M., Mackey, M.C.: Why the lysogenic state o phage λ is so stable: A mathematical modeling approach. Biophysical **86** (2004) 75–84
11. Haseltine, E.L., Rawlings, J.B.: Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. Chemical Physics **117** (2002) 6959–6969
12. Newman, M.E.J., Barkema, G.T.: Monte carlo methods in statistical physics. Oxford: Clarendon (1999)

Structure Prediction and Docking Studies of Chorismate Synthase from *Mycobacterium Tuberculosis*

Cláudia Lemelle Fernandes[1], Diógenes Santiago Santos[2,3],
Luiz Augusto Basso[2,4], and Osmar Norberto de Souza[1,2,3]

[1] Laboratório de Bioinformática,
Modelagem e Simulação de Biossistemas - LABIO
Pontifícia Universidade Católica do Rio Grande do Sul
[2] Centro de Pesquisa em Biologia Molecular e Funcional, TECNOPUC, PUCRS
[3] Instituto de Pesquisas Biomédicas, PUCRS
[4] Dep. de Biologia Molecular e Biotecnologia, UFRGS,
Porto Alegre, RS, Brasil
osmarns@inf.pucrs.br

Abstract. The enzymes of the shikimate pathway constitute an excellent target for the design of new antibacterial agents. This pathway is found in bacteria, fungi, plants and apicomplexan parasites but is absent in mammals. Chorismate Synthase (CS) catalyzes the last step of this pathway, the product of which is utilized in other enzymatic transformations like the biosynthesis of aromatic amino acids, folate, vitamin K and ubiquinone. This reaction is the most unusual of the entire pathway and is unique in nature. It converts EPSP to chorismate in the presence of a reduced FMN cofactor. Structure prediction used the comparative protein structure modeling methodology. The three-dimensional (3D) structure prediction of the enzyme was performed using the crystal structure (PDB ID: 1QX0) of CS from *Streptococcus pneumoniae* as template ($\approx$ 42% identity), and the MODELLER6v2 package. Additionally, in order to understand the possible binding modes of substrate and cofactor to the enzyme EPSP and FMN, respectively, were geometrically docked to CS. FMN binding to CS of *M. tuberculosis* (MTB) is similar to that of the *S. pneumoniae* template despite the change of Asn251 in *S. pneumoniae* to Gln256 in MTB. The longer side chain of Gln256 is overlapping with the FMN cofactor and a small conformational change is needed in order to properly accommodate this interaction. EPSP binding mode is also very similar to that of the template with three hydrogen bonds missing. This could be due to artifacts from the simple geometric docking we performed. Refinement with energy-based docking algorithms should relax the enzyme and substrates, thus promoting the expected interactions between them. Understanding the structure of MTB CS together with its cofactor and substrate binding modes should facilitate the search for inhibitors of this enzyme as alternative agents to treat tuberculosis.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 118–127, 2005.

1 Introduction

The shikimate pathway is the common way for the production of various products including folic acid, vitamin K, ubiquinone and the three aromatic amino acids. In bacteria, fungi, plants and apicomplexan parasites, chorismate, the final product of the shikimate pathway, is the branch point in the biosynthesis for all these products that are essential for these species. The absence of the shikimate pathway in all other species makes it an attractive target for the development of new antibacterial agents [3, 16].

Chorismate Synthase, the seventh and final step of the shikimate pathway, catalyses the conversion of 5-enolpyruvylshikimate 3-phosphate (EPSP) to chorismate in the presence of a reduced flavin mononucleotide (FMN) as a cofactor [12]. The reaction mechanism of the shikimate pathway has been studied extensively and revealed that the reaction of CS is unique in nature. The reaction involves a 1,4 elimination of phosphate and the loss of a proton of the C-6 hydrogen. This consists in the formation of the second out of three necessary double bonds to build an aromatic ring (Fig. 1). The enzyme activity requires a reduced FMN molecule which is not consumed during the reaction [3, 11].

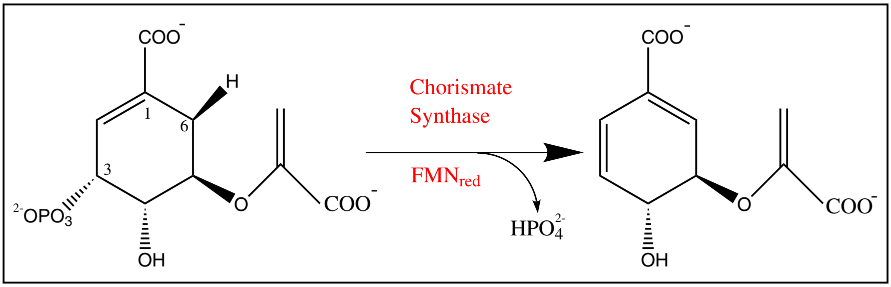


Fig. 1. Reaction catalyzed by Chorismate Synthase. The elimination of the 3-phosphate and the loss of a proton in C-6 introducing a second double bound in the ring

The function of the reduced FMN in catalysis was extensively studied. The most accepted mechanism suggests a direct role of reduced FMN in the elimination reaction. FMN transfer the electron transiently to phosphate and the substrate donates an electron to regenerate the FMN. This reaction does not involve an overall change in the redox state [5, 12].

Recently, with the first high-resolution X-ray structure of CS from *S. pneumoniae* with the substrate and the cofactor in the oxidized form, [13] the structure of CS from *Saccharomyces cerevisiae* [15], and the structure of CS from *Helicobacter pylori* with the cofactor in the reduced form [1], studies on the binding mode of substrate and cofactor in the active site has started.

How reduced FMN is obtained divides the CS into two classes, monofunctional and bifunctional. Bifunctional CS has an intrinsic ability to reduce flavin (specifically FMN) using NADPH. In monofunctional CS this catalytic activity is not present. The bifunctional enzyme is present in fungi and the monofunctional form in plants and bacteria [12].

The active site of *S. pneumoniae* CS is very hydrophobic and extremely basic, with six arginine and two histidine residues. The FMN cofactor is deeply buried into the active site with EPSP blocking any possible exit. FMN makes one hydrogen bond with EPSP and a few polar interactions with the protein. On the other hand, EPSP makes several polar interactions and a few hydrophobic contacts with the protein [13]. The two histidines are present in both classes of CS and mutation of these residues to two alanines reduces the activity of both bifunctional and monofuctional enzymes to 5% [8].

M. tuberculosis, the etiological agent of tuberculosis, is responsible for widespread human morbidity and mortality. The development of new effective chemotherapy should aid in the treatment and control of the disease [21].

Sequencing of the MTB genome has revealed a large number of individual enzymes potentially useful in drug design [6], including CS. Understanding the structure of MTB CS, together with its cofactor and substrate binding modes, should facilitate the search for inhibitors of this enzyme as possible alternative agents to treat tuberculosis.

In this work we present 3D structural models for CS from MTB and evaluate their interactions with the substrate EPSP and the cofactor FMN by docking simulations.

2 Materials and Methods

The starting point of homology modelling is the identification of proteins in the Protein Data Bank (PDB) [4] that are related to the target sequence and then select the templates. In this case, the structure prediction of CS from MTB was based on 3D structures for the homologous *S. pneumoniae* CS protein (PDB ID: 1QXO) [13] found using Blastp [2].

The next step is the multiple sequence alignment comparisons. The objective of this alignment is to improve the sensitivity of the search and to find the regions with high similarity. Possible templates and target sequences alignments were performed with ClustalW [18] and required a small gap, of four residues (insertions and/or deletions).

The program MODELLER6v2 [19] was used to build the protein models, using the standard protocol for comparative protein structure modelling methodology [14]. The best model of each enzyme was evaluated and selected according to their stereochemical quality analyzed with PROCHECK [9]. Validation of the models 3D profiles was performed with VERIFY 3D [10].

The structures of EPSP and FMN were geometrically and manually docked to the CS model of MTB using SwisPdbViewer [7].

The schematic diagrams of protein-ligand interactions of the best docking results were performed with LIGPLOT [20]. All figures were prepared with SwissPdbViewer [7].

3 Results

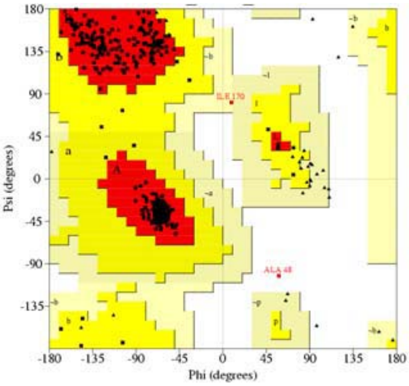
3.1 Homology Modelling

Corismate Synthase is a protein of about 360 to 400 amino acid residues, except in apicomplexan parasites, where it has about 500 amino acids. It has a high degree of sequence conservation among species. The protein has three signature patterns (Fig. 2) from conserved regions rich in basic residues (mostly arginines) [17].

In the search for templates, CS from *Aquifex aeolicus* (PDB ID: 1Q1L) had the best score. However, some residues were missing from the crystal structure which then had to be discarded. Therefore, the second highest score structure,

```
                     10        20        30        40        50        60        70
                      |         |         |         |         |         |         |
TARGET       MLRWITAGESHGRALVAVVEGMVAGVHVTSADIADQLARRRLGYGRGARMTFERDAVTVLSGIRHGSTLG
TEMPLATE     -MRYLTAGESHGPRLTAIIEGIPAGLPLTAEDINEDLRRRQGGYGRGGRMKIENDQVVFTSGVRHGKTTG
              :*::*******  *.*::**: **: :*: ** ::* **: *****.**.:*.* *.. **:***.* *

                     80        90       100       110       120       130       140
                      |         |         |         |         |         |         |
TARGET       GPIAIEIGNTEWPKWETVMAADPVDPAELADVARNAPLTRPRPGHADYAGMLKYGFDDARPVLERASARE
TEMPLATE     APITMDVINKDHQKWLDIMSAEDIEDRLKSKRK----ITHPRPGHADLVGGIKYRFDDLRNSLERSSARE
             .**:::: *.:  **  :*:*: ::    :.      :*:******* .* :** *** *  ***:****

                    150       160       170       180       190       200       210
                      |         |         |         |         |         |         |
TARGET       TAARVAAGTVARAFLRQALGVEVLSHVISIGASAPYEGPPPRAEDLPAIDASPVRAYDKAAEADMIAQIE
TEMPLATE     TTMRVAVGAVAKRLLAELDMEIANHVVVFGGKEIDVPENLTVAEIKQRAAQSEVSIVNQEREQEIKDYID
             *: ***.*:**: :* :     .   *:  * .       . **       * *   ::  * ::   *:

                    220       230       240       250       260       270       280
                      |         |         |         |         |         |         |
TARGET       AAKKDGDTLGGVVEAVALGLPVGLGSFTSGDHRLDSQLAAAVMGIQAIKGVEIGDGFQTARRRGSRAHDE
TEMPLATE     QIKRDGDTIGGVVETVVGGVPVGLGSYVQWDRKLDARLAQAVVSINAFKGVEFGLGFEAGYRKGSQVMDE
               *:****:*****:*. *:******:.. *::**::** **:.*:*:****:* **::. *:**:. **

                    290       300       310       320       330       340       350
                      |         |         |         |         |         |         |
TARGET       MYPG-PDGVVRSTNRAGGLEGGMTNGQPLRVRAAMKPISTVPRALATVDLATGDEAVAIHQRSDVCAVPA
TEMPLATE     ILWSKEDGYTRRTNNLGGFEGGMTNGQPIVVRGVMKPIPTLYKPLMSVDIETHEPYKATVERSDPTALPA
             :  .  ** .* **. **:*********: **..****.*: :.* :**: * :   *  :***  *:**

                    360       370       380       390       400
                      |         |         |         |         |
TARGET       AGVVVETMVALVLARAALEKFGGDSLAETQRNIAAYQRSVADREAPAARVSG
TEMPLATE     AGMVMEAVVATVLAQEILEKFSSDNLEELKEAVAKHRDYTKNY---------
             **:*:*::** ***:  ****..*.* * :. :* ::  . :

Identity (*) : 172 is 42.79 %
Strongly similar (:) : 77 is 19.15 %
Weakly similar (.) : 33 is 8.21 %
Different : 120 is 29.85 %
```

Fig. 2. ClustalW pairwise sequence alignment between the target (*M. tuberculosis* CS) and template (1QXO). The amino acid residues of the CS signatures (G-[DES]-S-H-[GC]-x(2)-[LIVM]-[GTIV]-x-[LIVT]-[LIV]-[DEST]-[GH]-x-[PV], [GE]-x(2)-S-[AG]-R-x-[ST]-x(3)-[VT]-x(2)-[GA]-[STAVY]-[LIVMF], R-[SHF]-D-[PSV]-[CSAVT]-x(4)-[SGAIVM]-x-[IVGSTAPM]-[LIVM]-x-E-[STAHNCG]-[LIVMA]) are underlined



Plot Results

Residues in most favoured regions (red)	318	94.9%
Residues in additional allowed regions (dark yellow)	15	4.5%
Residues in generously allowed regions (light yellow)	1	0.3%
Residues in disallowed regions (white)	1	0.3%
Number of non-glycine and non-proline residues	335	100.0%
Number of end-residues (excl. Gly and Pro)	1	
Number of glycine residues (shown as triangles)	43	
Number of proline residues	22	
Total number of residues	401	

Fig. 3. Ramachandran plot for the best model of MTB CS

CS from *S. pneumoniae* (PDB ID: 1QX0), was used as template to model the 3D structure of MTB CS. 1QX0 turned to be a very attractive template for its structure contained not only the enzyme, but the cofactor FMN in the oxidized form and the EPSP substrate. In addition, as in MTB, CS from *S. pneumoniae* is monofunctional.

Fig. 2 shows a pairwise alignment between the target and template sequences, whose identity is over 40%, well above the 30% limit usually required for comparative protein structure modelling [14].

Ten models of the enzyme were built. They were evaluated by PROCHECK [9] and VERIFY 3D [10] in order to choose the best one. Out of 335 non-glycine and non-proline residues, 318 or 94,9%, were located in the most favored regions of the Ramachandran plot (Fig. 3). The root mean-square (RMS) of planar atoms from best-fit plane are less than 0.02 for rings and 0.01 otherwise. The PROCHECK results are given in Table 1. The best model of MTB CS is illustrated in Fig. 4.

3.2 Docking Studies

The cofactor FMN and the EPSP substrate were geometrically and manually docked into the active site of MTB CS. Based on the assumption that the binding is similar to *S. pneumoniae*, we were able to manually place FMN and EPSP in their respective binding sites [13].

Table 1. Quality of main-chain and side-chain parameters of modelled MTB CS. The model is verified at 2 Å resolution

		Comparison Values			
					Number of
	Number of	Parameter	Typical		bandwidths
Stereochemical parameters	data pts	value	value	Band width	from mean
Stereochemistry of main-chain					
%-tage residues in A, B, L	335	94.9	83.8	10.0	1.1
Omega anglest dev	400	3.8	6.0	3.0	-0.7
Bad contacts 100 residues	3	0.7	4.2	10.0	-0.3
Zeta angle st dev	358	1.3	3.1	1.6	-1.1
H-bond energy st dev	238	0.7	0.8	0.2	-0.5
Overall G-factor	401	0.0	-0.4	0.3	1.2
Stereochemistry of side-chain					
Chi-1 gauche minus st dev	56	5.1	18.1	6.5	-2.0
Chi-1 trans st dev	89	9.1	19.0	5.3	-1.9
Chi-1 gauche plus st dev	121	6.2	17.5	4.9	-2.3
Chi-1 pooled st dev	266	7.0	18.2	4.8	-2.3
Chi-2 trans st dev	72	11.8	20.4	5.0	-1.7

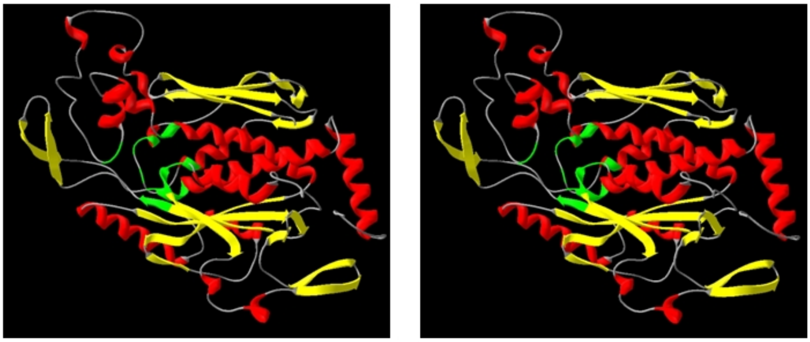


Fig. 4. Stereoview of *M. tuberculosis* CS three-dimensional structure (ribbon representation) looking across the active site. The structure contains 9 α-helices (red) and 15 β-strands (yellow). The secondary structures making up the active site are colored green

Thus, we have similar interactions except one missing contact to Gln256 which in *S. pneumoniae* corresponds to Asn251. The longer side chains of the non-conserved Gln256 and the conserved Lys315 are too close to FMN and thus need a conformational change to be properly accommodated.

All H bonds to FMN in *S. pneumoniae*, with exception of that involving Thr315, are present in MTB, including the H bond with EPSP. The H bonds to FMN involving Asn251 in *S. pneumoniae* were not reproduced in the MTB model for its structural equivalent in MTB, Gln256, is far too close to the FMN molecule. However, Gln256 makes a H bond with the FMN phosphate. Lys315 in

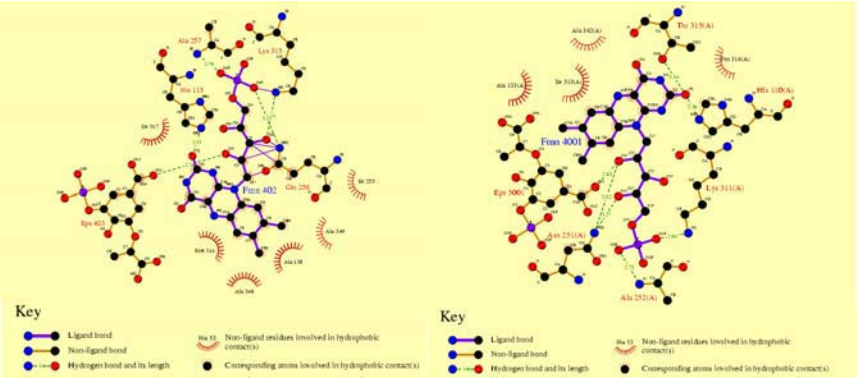


Fig. 5. Ligplot representation of the FMN binding pattern to CS from MTB (Left, Fmn 402) and *S. pneumoniae* (Right, Fmn 4001)

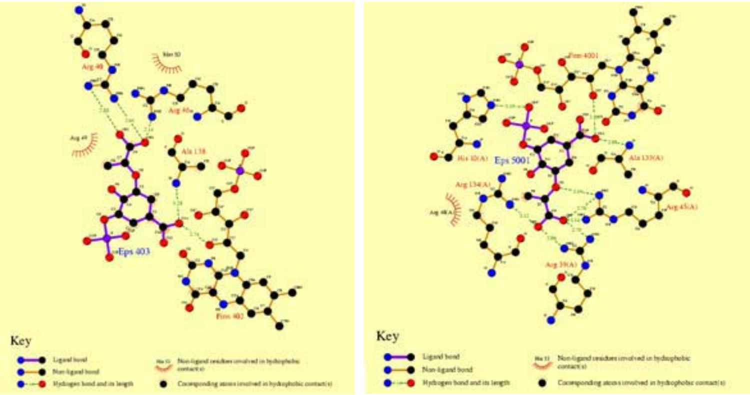


Fig. 6. Ligplot representation of the binding pattern of EPSP to MTB CS (Left, Eps 403) and *S. pneumoniae* (Right, Eps 5001)

M. tuberculosis made a H bond with O3 in the aliphatic portion of FMN which was not present in *S. pneumoniae*. MTB has more hydrophobic contacts (six) then the *S. pneumoniae* (four) (Fig. 5).

The H bonds to EPSP in MTB CS are also very similar to those found in *S. pneumoniae*. Two H bonds, involving amino acids His11 and Arg139, were missing. They correspond to His10 and Arg134, respectively, in the *S. pneumoniae*. Furthermore, while Arg45 makes two H bonds to EPS in *S. pneumoniae*, in MTB, its equivalent, Arg46, makes only one H bond. The hydrophobic contacts were very similar, only one in the *S. pneumoniae* (Arg48) and two in MTB, Arg49 and Met50 (Fig. 6).

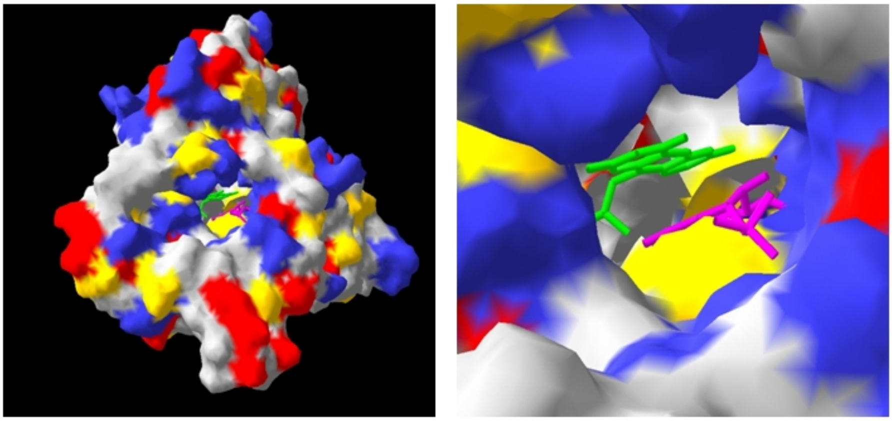


Fig. 7. Molecular surface representation of MTB CS colored by amino acid type. Apolar, gray; polar, yellow; acidic or negatively charged, red; and basic or positively charged, blue. (Left) The enzyme subunit looking into the active site. (Right) A close-up view of the active site entrance. The binding site entrance is composed mainly of arginines (blue). The cofactor FMN (green) and substrate EPSP (pink) are bound deep inside

The molecular surface of MTB CS (Fig. 7) shows that the active site entrance is very hydrophilic with many basic amino acids that are involved in EPSP binding.

4 Discussion and Conclusions

We have obtained the 3D structure of MTB CS based on the crystal structure of an orthologous enzyme from *S. pneumoniae*. In addition, we modelled the interactions of the cofactor FMN and the EPSP substrate with the enzyme using a simple, geometric, docking approach.

The quality of the MTB CS model is good and appropriate for docking studies.

The geometric docking we used is adequate for an initial study only. As observed, the side chains of Gln256 and Lys315 need to undergo some conformational changes to better accommodate the FMN cofactor in its binding site. Nonetheless, our docking analysis showed that the binding mode of EPSP and FMN is similar in both *S. pneumoniae* and MTB CS, as we should expect based on sequence homology.

The hydrophilic amino acids making up the binding site of EPSP are conserved, but the interactions between enzyme and substrate need some additional studies. His11 and Arg139 are important for enzyme activity, but the H bonds their side chains make to the EPSP substrate are missing in MTB. Further docking refinements with energy-based docking algorithms should relax the enzyme and substrates, hence promoting the expected interactions between them.

Understanding the structure of *M. tuberculosis* Chorismate Synthase, together with its cofactor and substrate binding modes, should provide a working model to be used in high throughput virtual screening of small-molecule public libraries so as to accelerate the search for inhibitors of this enzyme as alternative agents to treat tuberculosis.

Acknowledgments

The authors wish to thank Afonso Sales for valuable technical support.

We are grateful to the referees for careful reading of our manuscript and valuable suggestions.

This project was supported by grants from CAPES and FAPERGS to O.N.S., and FINEP and Millennium Institute, CNPq/MCT, to D.S.S. and L.A.B.

C.L.F. is supported by a MSc scholarship from CAPES.

References

1. Hyung Jun Ahn, Hye-Jin Yoon, Byung II Lee, and Se Won Suh. Crystal Structure of Chorismate Synthase: A novel FMN-binding Protein Fold and Funcional Insights. *Journal of Molecular Biology*, 336:903–915, 2004.
2. Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25:3389–3402, 1997.
3. Ronald Bentley. The Shikimate Pathway - A metabolic Tree with Many Branches. *Critical Reviews in Biochemistry and Molecular Biology*, 25(5):307–384, 1990.
4. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.
5. Stephen Bornemann, David J. Lowe, and Roger N. F. Thorneley. The Transient Kinetics of *Escherichia coli* Chorismate Synthase: Substrate Consuption, Product Formation, Phosphate Dissociation, and Characterization of a Flavin Intermediate. *Biochemistry*, 35(30):9907–9916, 1996.
6. S. T. Cole, R. Brosch, J. Parkhill, T. Garnier, C. Churcher, D. Harris, S. V. Gordon, K. Eiglmeier, S. Gas, C. E. Barry, F. Tekaia, K. Badcock, D. Basham, D. Brown, T. Chillingworth, R. Connor, R. Davies, K. Devlin, T. Feltwell, S. Gentles, N. Hamlin, S. Holroyd, T. Hornsby, K. Jagels, A. Krogh, J. McLean, S. Moule, L. Murphy, K. Oliver, J. Osborne, M. A. Quail, M.-A. Rajandream, J. Rogers, S. Rutter, K. Seeger, J. Skelton, R. Squares, S. Squares, J. E. Sulston, K. Taylor, S. Whitehead, and B. G. Barrell. Deciphering the biology of *Mycobacterium tuberculosis* from the complete genome sequence. *Nature*, 393:537–544, 1998.
7. Nicolas Guex and Manuel C. Peitsch. SWISS-MODEL and The Swiss-PdbViewer: An environment for comparative protein modeling. *Electrophoresis*, 18:2714–2723, 1997.

8. Karina Kitzing, Sigrid Auweter, Nikolaus Amrhein, and Peter Macheroux. Mechanism of Chorismate Synthase: Role of the Two Invariant Histidine Residues in the Active Site - *In Press*. *Journal of Biological Chemistry*, December, 10 2003.
9. Roman A. Laskowski, Malcolm W. MacArthur, David S. Moss, and Janet M. Thornton. PROCHECK: a program to check the stereochemical quality of protein structures. *Journal of Appl. Cryst.*, 26:283–291, 1993.
10. Roland Lüthy, James U. Bowie, and David Eisenberg. Assessment of protein models with three-dimensional profiles. *Nature*, 356:83–85, 1992.
11. Peter Macheroux, Stephen Bornemann, Sandro Ghisla, and Roger N. F. Thrneley. Studies with Flavin Analogs Provide Evidence That a Protonated Reduced FMN is the Substrate-induced Transient Intermediate in the Reaction of *Escherichia coli* Chorismate Synthase. *The Journal of Biological Chemistry*, 271(42):25850–25858, 1996.
12. Peter Macheroux, Jürg Schmid, Nikolaus Amrhein, and Andreas Schaller. A Unique reaction in a Common Pathway: Mechanism and Function of Chorismate Synthase in the Shikimate Pathway. *Planta*, 207:325–334, 1999.
13. John MaClean and Sohail Ali. The Structure of Chorismate Synthase Reveals a Novel Flavin Binding Site Fundamental to a Unique Chemical reaction. *Structure*, 11:1499–1511, 2003.
14. Marc A. Martí-Renom, Ashley C. Stuart, András Fiser, Roberto Sánchez, Francisco Melo, and Andrej Šali. Comparative Protein Structure Modeling of Genes and Genomes. *Annual Reviews Biophis. Biomol. Structure*, 29:291–325, 2000.
15. Sophie Quevillon-Cheruel, Nicolas Leulliot, Philippe Meyer, Marc Graille, Michael Bremang, Karine Blondeau, Isabelle Sorel, Anne Poupon, and Joël Janin nad Herman van Tilbeurgh. Crystal Structure of the Bifunctional Chorismate Synthase from *Saccharomyces cerevisiae*. *The Journal of Biological Chemistry*, 279(1):619–625, 2004.
16. Fiona Roberts, Craig W. Roberts, Jennifer J. Johnson, Dennis E. Kyle, Tino Krell, John Coggins, Graham H. Coombs, Willbur K. Milhous, Saul Tzipori, David J. P. Ferguson, Debopam Chakrabarti, and Rima McLeod. Evidence for the Shikimate pathway in Apicomplexan Parasites. *Nature*, 393:801–805, 1998.
17. C.J.A. Sigrist, L. Cerutti, N. Hulo, A.Gattiker, L. Falquet, M. Pagni, A. Bairoch, and P.Bucher. PROSITE: a documented database using patterns and profiles as motif descriptors. *Bioinform.*, 3:265–274, 2002.
18. Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.
19. Andrej Šali and Tom L. Blundell. Comparative Protein Modelling by Satisfaction of Spatial Restraints. *Journal of Molecular Biology*, 234(3):779–815, 1993.
20. Andrew C. Wallace, Roman A. Laskowski, and Janet M. Thornton. LIGPLOT: a program to generate schematic diagrams of protein-ligand interactions. *Protein Engineering Design and Seletion*, 8:127–134, 1995.
21. World Health Organization, USA. Global Tuberculosis Control Report 2004. http://www.who.int/tb/en, 2004.

Analysis of the Effects of Multiple Sequence Alignments in Protein Secondary Structure Prediction

Georgios Joannis Pappas Jr.[1] and Shankar Subramaniam[2]

[1] Biotechnology and Genomic Sciences program,
Universidade Católica de Brasília
gpappas@bioinformatica.ucb.br

[2] Departments of Bioengineering, Chemistry, and Biochemistry,
University of California at San Diego, La Jolla, California
shankar@ucsd.edu

Abstract. Secondary structure prediction methods are widely used bioinformatics algorithms providing initial insights about protein structure from sequence information. Significant efforts to improve the prediction accuracy over the past years were made, specially the incorporation of information from multiple sequence alignments. This motivated the search for the factors contributing for this improvement. We show that in two of the highly ranked secondary structure prediction methods, DSC and PREDATOR, the use of multiple alignments consistently improves the prediction accuracy as compared to the use of single sequences. This is validated by using different measures of accuracy, which also permit to identify that helical regions benefit the most from alignments, whereas β-strands seem to have reached a plateau in terms of predictability. Also, the origins of this improvement is explored in terms of sequence specificity, secondary structure composition and the extent of sequence similarity which provides the optimal performance. It is found that divergent sequences, in the identity range of 25–55% provide the largest accuracy gain and that above 65% identity there is almost no advantage in using multiple alignments.

1 Introduction

One of the earliest utilization of computational methods to solve complex biological problems can be considered protein secondary structure prediction, dating from 1960's. The well established concept that the three-dimensional structure of a protein is dictated by its amino acid sequence [1] prompted the quest for methodologies aiming to predict structure solely from sequence. However, due to the intrinsic complex structural atomic arrangement, a widely used approach is to predict the secondary structure as string of generally three symbols representing α-helices, β-strands and coil (other non-regular structures) [2, 3].

Several methodologies were developed for this task, with varying degrees of success, but not overcoming 75% accuracy on average [4, 5]. The main concep-

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 128–140, 2005.

tual framework in the first generation of algorithms was the idea of establishing propensities for each amino acid to adopt a secondary structure, in a local sequence context. In the second generation accuracy improvement was attained by the adoption of pairwise residue statistics over sequence blocks of limited size, instead of single residue statistics [2, 6]. In recent years, several alternative procedures have been employed to increase the accuracy levels. Amongst these, the one that has been most successful was the utilization of evolutionary information given in multiple alignments of similar sequences [6, 7, 8, 9, 10, 11] instead of a single sequence.

The rationale behind that relies on the observation that protein structures diverge more slowly than sequences. Generally speaking, a lower bound of 30% in sequence identity is considered sufficient for two polypeptide chains to share the same fold [12, 13]. The fact that the possible number of protein folds is limited [14] implies that the sequence to structure mapping is highly degenerated and favors the formation of a single fold by several sequences. This tendency turns out to be a desirable trait for molecular evolution, since a structure can accommodate varying degrees of mutations in the sequence without compromising the structural core or the protein active site. Since the plasticity of residue substitution between structurally homologous peptides does not occur randomly, one can envision a mechanism where the pattern of amino acid substitution conveys specific information regarding the conformation [7].

Several observations provide a link between multiple alignments and protein structure. For instance, after a detailed analysis of several protein families, it was observed that sites of insertions (deletions) in multiple alignments are good indicators of surface loops, since these are more likely to tolerate mutations without disrupting the protein core [15]. Conversely, sites that display strict conservation may represent important structural regions.

The incorporation of aligned sequences in the prediction can be achieved in several ways. One is to combine the homologous sequences representation them as a single consensus sequence, which is subject to prediction [16]. Another approach is to predict the secondary structure for each individual sequence in the alignment and then combine then into a consensus prediction [7]. Following the same line the DSC algorithm [10], deals with multiple alignment inputs by averaging the structural propensities over the aligned sequences.

The PREDATOR method [11], instead of using pre-computed multiple alignments in the input, performs a custom pairwise optimal local alignment between the query sequence and each of the other ones in the input. The alignment procedure will try to find only non-gapped patches in a pair of sequences that will have the best local alignment. Therefore, the alignment regions are discontinuous, as opposed to the ones generated by global alignment procedures, but these reflect more clearly a significant structural relationship. After all local alignments with regard to the query sequence are defined, the prediction utilizes the usual averaging of propensities to generate the final model. The authors report a 74.8% accuracy value for a test set of 125 proteins.

Other methodologies use evolutionary information derived from PSI–BLAST searches [17] of the query against large sequence databases and transform the results in sequence profiles (position-specific scoring matrices), which in turn, are feed to neural networks for prediction [18, 19, 20].

The reports of accuracy increase using multiple alignments when compared to single sequence input in secondary structure prediction, range from 5 to 9% on average, and for most of the methods it is claimed that the 70% accuracy barrier is surpassed [6, 10, 11, 21] even reaching figures close to 80% [18, 19, 20]. This an indication that these methods are maturing to be useful predictors for genomic scale analises.

In this scenario, there is a motivation for a detailed analysis of the factors that cause this accuracy improvement. Previous studies investigated the effects of methodologies employed to create multiple alignments [22] or the size of the databases providing homologous sequences [23]. On the other hand, several questions remain to be answered such as what is the optimal level of sequence identity in the alignments most suitable for the prediction, and if all secondary structure types equally benefit from the alignments. The present work addresses these questions by re-evaluating two of the most popular methods with a new test database, subject to several quality measures.

2 Methods

2.1 Computer Programs

To measure the effect of using multiple aligned sequences, the most appropriate algorithms are these that can be executed with single sequences as well. This enables quantitative characterization of improvements in prediction due to use of alignments. Programs like PHD [21] which only work with multiple sequences or PSIPRED [18] that need homologues in public sequence databases were not used in this work. Two of the best performing methods were selected: DSC [10] and PREDATOR [11]. Their computer implementations were obtained from the respective authors. To simplify the nomenclature, PREDATOR will be designated by the authors initials F_A.

2.2 Testing Database

Starting from a data set of high-resolution protein structures (better than 2.5 Å) from the PDB database (Protein Data Bank, [24]), and with chain size larger than 50 residues, a non-redundant sequence subset was selected using the Obstruct program [25], with the constraint that no chains have more than 25% of global identity.

Based on this initial set of proteins, the HSSP database [26] was used as a source of multiple sequence alignments. HSSP merges structure and sequence information by creating, for each known protein structure from PDB, a multiple alignment of suitable hits from similarity searches against the SWISS–PROT protein sequence database.

Since the number of aligned proteins and the extent of alignment varies considerably between HSSP entries, some limits were imposed on the selection of proteins for analysis. Each member of the aligned sequence set is mandated to be aligned with at least 75% of the amino acids in the reference PDB sequence and an entry in HSSP is considered for further analysis only if it contains at least 3 aligned sequences other than the PDB sequence. The combination of this and the PDB filter above results in 231 HSSP entries constituting our test set.

Secondary structure assignments were taken from DSSP program [27] and transformed to three-states (α-helix, β-strand and coil) according to [23].

2.3 Accuracy Measures

An important issue in the analysis of prediction performance is the definition of prediction success measurements. Historically, the Q_3 measure is the simpler and most widely used, representing the overall fraction of correct predictions. It is also possible to define this quantity for each structural class. For example, Q_3^α means the percentage of correctly predicted helical residues [6].

In addition to Q_3, two additional measures were utilized given their improved sensitivity to mispredictions: the Matthews' correlation coefficients [28] and the Rost–Sander information measure (RSI, [21]). Matthews' correlation coefficients ($C_{\alpha,\beta,c}$) probe specific secondary structures and take into account both true and false positives, as well the correct predictions for the particular structure. The RSI measure on the other hand is consolidated for all types of secondary structure elements and represents well the overall success rate of the prediction.

For an additional assessment of the prediction accuracy, the prediction methods were analyzed by their capacity to identify entire segments of α-helices and β-strands, as opposed to residue-wise measures described previously. In many cases it is useful to get an estimate of the number of secondary structure types present in a protein even if their precise location is not well-predicted. Two measures are used for this purpose. The first (O_{strict}) is the percentage of predicted segments that overlap exactly with the experimental segments. This is a very stringent measure, not allowing any mismatches between the predicted and correct segments. To be more permissive, the percentage of predicted segments overlapping with at least 50% of the segment in the experimental structure is also employed and is called O_{loose}.

3 Results and Discussion

3.1 Evaluation of Secondary Structure Prediction Accuracy from Multiple Alignments

The first task in evaluating the effect of using multiple sequences is the prediction of secondary structure using only the reference PDB sequence (single) and the respective multiple alignment from HSSP (aligned). For a detailed analysis, several accuracy measures are utilized to probe both the global and structure

Table 1. Average accuracy values for prediction with DSC and F_A methods with (aligned) and without (single) the use of multiple sequence alignments. The column labeled Δ is the difference between aligned and single sequence measures for the method

Measure	DSC			F_A		
	aligned	single	Δ	aligned	single	Δ
Q_3	0.6798	0.6294	+0.0504	0.6956	0.6355	+0.0601
Q_3^α	0.6548	0.5305	+0.1243	0.6284	0.5474	+0.0810
Q_3^β	0.5162	0.5513	−0.0351	0.5160	0.4059	+0.1101
Q_3^c	0.7779	0.7436	+0.0343	0.8340	0.8140	+0.0200
C_α	0.5492	0.4527	+0.0965	0.5827	0.4705	+0.1122
C_β	0.4786	0.4112	+0.0674	0.5012	0.3847	+0.1165
C_c	0.4531	0.3989	+0.0542	0.4794	0.4012	+0.0782
RSI	0.2238	0.1585	+0.0653	0.2590	0.1704	+0.0886

specific behavior of the prediction algorithms. Table 1 shows the average accuracy values for predictions of the 231 sequences using DSC and F_A algorithms.

The most important observation is that for both methods, regardless of the accuracy measure utilized, there is a consistent increase in the success level of predictions when a multiple sequence alignment is employed. The only exception to this is the Q_3^β measure for the DSC method which decreased by an absolute value of 3.5% from single to aligned sequences. Another important feature is that the increased values of RSI and the Matthews correlation indicate that not only more residues are being correctly predicted but there are also less under and over-prediction errors.

The overall accuracy gain in using multiple alignments is in the order of 5–6% using the general Q_3 measure, which was also originally reported using a smaller dataset of proteins [10, 11]. However, the newly calculated Q_3 values of 67.9% for DSC and 69.5% for F_A are inferior to these claimed in the original reports, 70.1% and 74.8%, respectively. This disparity suggests that the generalization power of the methods is still somewhat limited, since the new dataset of 231 proteins is about twice as large as the original ones. Additionally, reveal the considerable qualitative dependence on the sequences and alignments used for evaluation.

On the other hand, it is clear that the use of alignments consistently improves the overall performance of the methods. Hence, it is important to explore where this additional information is incorporated in terms of predicting secondary structure elements.

Owing to the fact that gaps in multiple sequence alignments are most likely to occur in coil regions, it is anticipated that predictions for this structural category would benefit the most from multiple alignments. Nevertheless, the inspection of the measures reflecting the coil prediction, namely Q_3^c and C_c, reveals that the increase in accuracy for the coil structure is clearly overshadowed by the improvement in α-helix prediction. The absolute increase in the number of correctly predicted residues in α-helix (Q_3^α) is 12.4% for DSC and 8.1% for F_A, whereas the values for Q_3^c are 3.4% and 2.0% respectively (Table 1). The positive variation of C_α values by 9.6 (DSC) and 11.2% (F_A) clearly reflects a better resolution in the prediction of α-helices.

Conversely, the prediction of β-strands does not improve markedly with the use of multiple alignments. As seen before, in the case of DSC, there is even a loss of performance for β-strands judged by the negative variation in Q_3^β. For the same measure, F_A predictions using multiple alignments significantly improve in comparison to single sequence predictions. However, it should be noticed that the value of predicted accuracy for the single sequence is already excessively low (40.5%), and the multiple alignments partially overcome this deficiency. For both algorithms, the final value of 51.6% is still not impressive, showing that only about half of the residues in β-strands are correctly predicted and that this particular structural type deserves special treatment by the predictive methods.

To further investigate the effect of the alignments on α-helices and β-strands the predictions were analyzed in terms of the ability to locate entire segments of these conformations based on the segment overlap measures (2). Table 2 displays the number of correct region predictions as percentages of the total number of segments in the test database (1630 α-helices and 2277 β-strands segments).

Table 2. Comparative accuracy of the methods with and without the use of multiple sequence alignments based on secondary structure segment location measures

Measure	DSC			F_A		
	aligned	single	Δ	aligned	single	Δ
O^α_{strict}	4.85	2.88	+1.97	4.51	2.66	+1.85
O^β_{strict}	7.33	8.21	−0.88	9.15	6.03	+3.12
O^α_{loose}	69.08	55.03	+14.05	65.06	58.32	+6.74
O^β_{loose}	57.40	61.35	−3.95	57.22	45.73	+11.49

The results for α-helices confirm what was observed earlier with the residue-wise measures, namely, a marked improvement in the identification of helical regions. The number of absolutely correctly predicted α-helices (O^α_{strict}) almost doubled for both algorithms, but these comprise only about 4.5% of the total number of α-helices. When using a less stringent measure (O^α_{loose}), the DSC method shows a substantial improvement in identification of helical segments (14.0% more α-helices than that based on a single sequence). To a lesser extent, F_A method also benefits from multiple alignments (6.7% increase).

The behavior for β-strands also parallels the results in Table 1. The DSC method suffers a consistent decrease in the ability to predict β-strand segments when using multiple alignments as opposed to using only the single sequence. The F_A method, on the other hand, gains accuracy with the aligned sequences, but could only equal DSC predictions overall in terms of O^β_{loose} (57%) and be slightly better in terms of O^β_{strict} (9.1%).

3.2 Analysis of the Influence of Alignment in the Prediction

It is clear that the use of multiple alignments has an overall positive impact on secondary structure prediction. Consequently, it is important to verify what are the beneficial characteristics of the multiple alignments for prediction.

First, the goal is to probe if the amino acid substitution patterns found in the alignments really convey useful structural information than can be assimilated by the prediction methods. As a negative control, alignments were artificially generated by adding noise to the pre-existing alignments for each of the proteins in the test set. This was attained by randomly substituting a percentage of the amino acids by any of the other 19 amino acids or the gap symbol, with equal probability. All alignments had 15 proteins and were created for specific degrees of sequence identity. For example for an identity of 45%, each of the 15 proteins in the alignment had 55% of the amino acids randomly mutated, creating a different sequence for each aligned protein.

The results of the simulation with random alignments using the DSC program is shown in Fig. 1. As expected, the random alignments with low levels of sequence identity (i.e., high noise), produced low average accuracy values for Q_3 measure compared to the average value of the predictions using single sequences alone ($\approx$ 65%). The same behavior was observed using the F_A program, as well

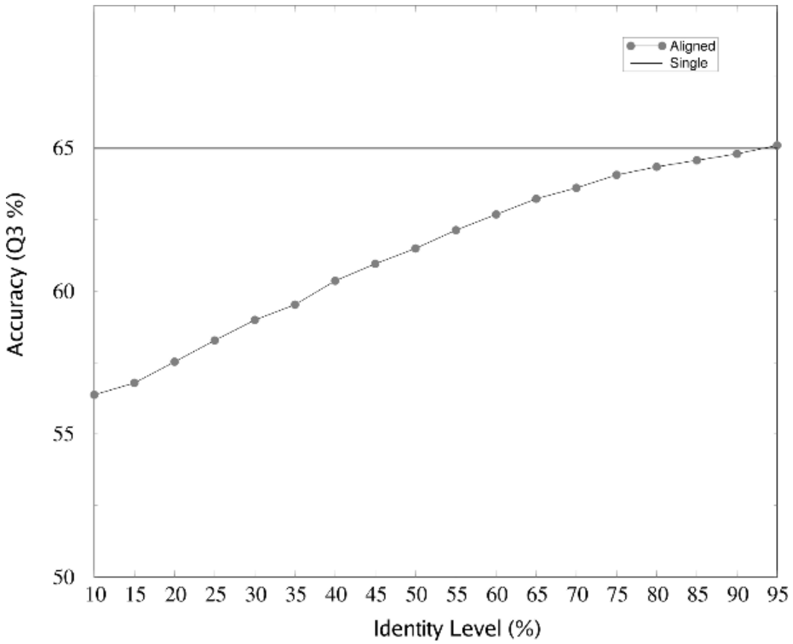


Fig. 1. Prediction accuracy of DSC method using randomly created alignments with varying identity levels compared to the original sequence. The line labeled "single" corresponds to the prediction without aligned sequences

with other accuracy measures (not shown). This result indicates that indeed the information contained in the multiple alignments is relevant to the accuracy of prediction. In other words, if the alignment quality is poor, then the single sequence prediction provides better results. This apparently contradicts previous findings for PHD method were it was found that the inclusion of distant homologues (from BLAST searches) in the alignment, and even false positives, was beneficial for the prediction [23]. However, there is no discrepancy whatsoever since the random alignments do not contain evolutionary reliable substitution patterns, which is not the case for BLAST searches.

3.3 Effect of Alignment Identity Levels

Next we address the optimal level of homology between proteins within the multiple alignment that confers the best prediction, given the indication that the more divergent the sequences the better the prediction [23]. Toward this end, the following protocol was devised:

1. For each HSSP alignment in the test set, create subsets with percent identity ranging from 95% to 25% in intervals of 10%
2. If the number of proteins selected in the identity range is greater than a threshold (3 sequences) the alignment subset with the reference PDB protein is saved
3. For each saved alignment, the secondary structure predictions are performed using DSC and F_A methods for the reference PDB protein alone (single) and for the corresponding multiple sequence alignment subset (aligned)
4. For all proteins in the specified identity subset, the prediction accuracies for the single and aligned input are calculated and their average computed.

This procedure results in a dissection of the original alignment, splitting it into several new ones, with a more homogeneous distribution in terms of similarity. This aims at quantifying the amount of sequence variation in the input alignment that is optimal for secondary structure prediction. This can be observed in Fig. 2, where the average accuracy values for each identity range are compared.

For the F_A method (Fig. 2 b,d), it is clear that the higher gains in accuracy occur when low identity alignments (25–45%) are used as input. The gain in accuracy is inversely proportional to the increase in sequence identity. In fact, in the case of RSI, there is virtually no gain in predictive accuracy when the input alignment contains proteins with identity values higher than 65% in relation to the reference PDB protein. Also, in terms of Q_3, for identity range high than this value, there are only small accuracy improvements ($< 1\%$).

In the case of DSC method (Fig. 2 a,c) the overall trend is maintained, but with the highest absolute Q_3 value appearing in the 45–55% identity range. However, in terms of RSI, the behavior is similar to the one observed using the F_A method, with the accuracy values decreasing monotonically with the increase in sequence identity.

It should be noted that the compositional heterogeneity between the generated alignments renders the direct comparison of the average accuracy values

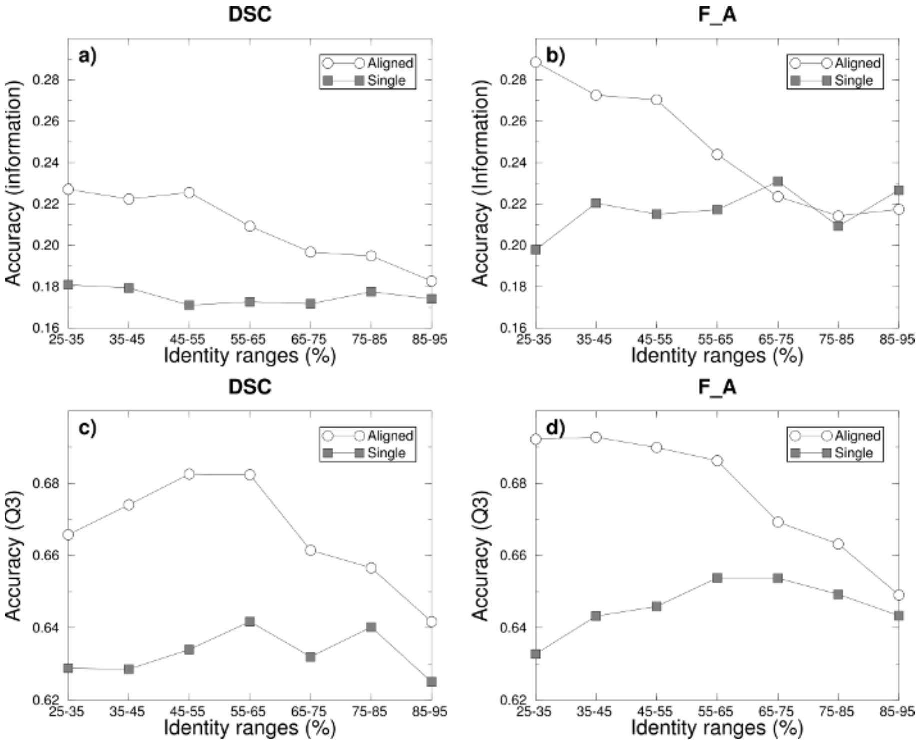


Fig. 2. Variation of the prediction accuracy as a function of the identity level of the aligned proteins for the methods DSC and F_A. The results compare the accuracy values at varying sequence identity levels by using only the reference PDB protein (Single) or the multiple alignment (Aligned). Graphs *a*-*b* use RSI measure, whereas *c*-*d* use Q_3 measure

between identity levels somewhat less rigorous. Nevertheless, since what is sought is the measurement of the gain obtained by using multiple alignments in relation to single sequence input, this problem is less critical, given that the prediction for a single sequence serves as the reference point.

In order to quantitatively assess the amount of prediction improvement and to provide a better qualitative visualization of how the methods benefit from the alignments, a simple measure of the increase in the accuracy, named accuracy gain (G), was defined. This term is the difference between aligned and single sequence accuracies normalized by the maximum difference value between the series, as shown in the formula:

$$G_i = \frac{A_i - S_i}{\mathrm{Max}(\mathbf{A})}, \quad \begin{cases} i & = \text{Identity level} \\ A_i & = \text{Accuracy value for the aligned input} \\ S_i & = \text{Accuracy value for the single input} \\ \mathrm{Max}(\mathbf{A}) & = \text{Maximum value of } A_i \text{ values} \end{cases} \tag{1}$$

In this measure, negative G values imply that the prediction accuracy using a single sequence is actually higher than that obtained using a multiple alignment. With this definition in place, all accuracy measures can be compared across the prediction methods, thus consolidating the plots in Fig. 2 onto a single one, which is shown in Fig. 3. It can be observed that there is a marked difference in the predictions of DSC and F_A methods. For DSC, the highest prediction gain is attained by utilizing aligned proteins in the 25–35% identity range, whereas for F_A the best gains come from the protein identity set in the 45–55% range.

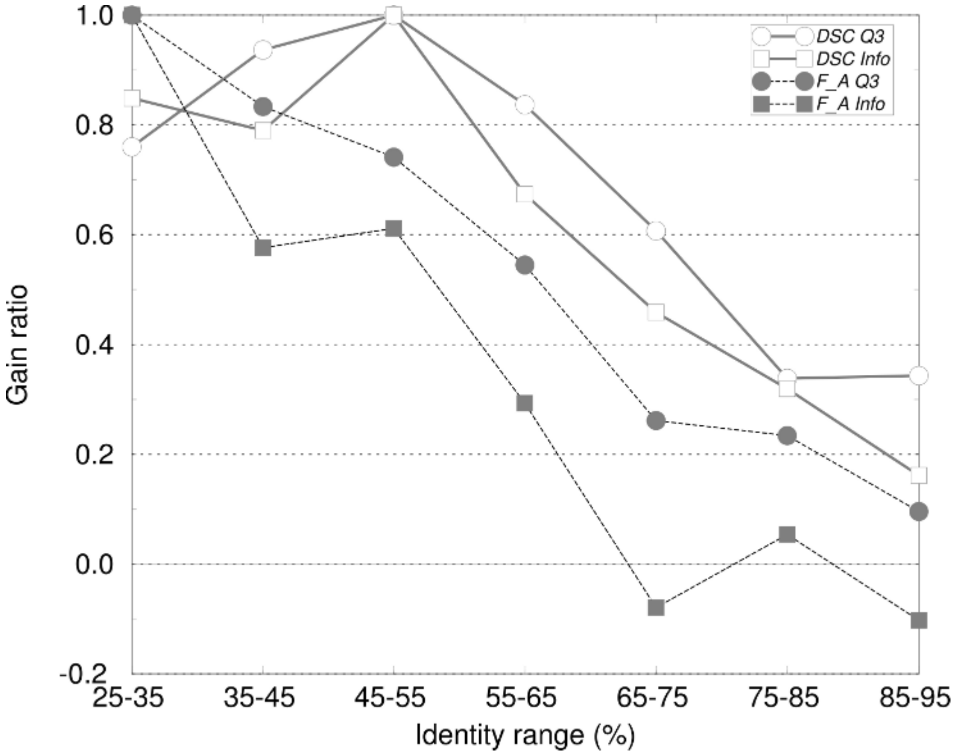


Fig. 3. Prediction improvement as measured by the gain ratio (1), that reflects the increase in accuracy due to use of multiple alignments for the two prediction methods using Q_3 and RSI accuracy measures

An important feature shared by both methods is that the prediction gain continuously decreases for alignments with increasing identity to the reference protein. This trend is more evident for the F_A method using RSI, where identities higher than 65% have an imperceptible or, in some cases, deleterious effect on the prediction.

4 Conclusion

The comparisons of the accuracy values obtained from single sequence and multiple alignment inputs indicate that significant improvements in secondary structure prediction can be obtained by using sets of homologous proteins. Nevertheless, this is not reflected uniformly across all types of secondary structure. In fact, the results show that α-helices tend to be better predicted using multiple alignments, whereas β-strands seem to have reached a plateau in terms of predictability.

The prediction methods that use evolutionary information do not take into account the degree of sequence similarities within the multiple alignments. As pointed out by earlier [29], such methods effectively consider the sequences as independent realizations of some stochastic process. However, based on the results of the analysis of the influence of protein similarity on the prediction of secondary structure, it is clear that this is not the case. The results suggest that the prediction methods could obtain more information about the secondary structure by using only the fraction of related proteins that have low identities with the query sequence. This has not been taken in consideration earlier, and is seen to lead to improvements in the prediction.

The difference between the optimal identity level for DSC (45–55%) and F_A (25–35%) methods stems primarily on how the input alignments are manipulated by the algorithms. DSC does not modify the alignment given in the input, whereas F_A discards the input alignment structure and re-creates alignments by searching for the best local alignments.

However, the critical question that remains to be answered is the prediction improvement with low levels of identity in the alignments. One possible explanation involves the fact that the secondary structure prediction algorithms assign static structural propensities for the amino acids, rendering predictions somewhat rigid and displaying a strong bias toward the assigned propensities. The inherent plasticity in the sequence–structure relationship [5], hence, cannot be effectively accounted for in these methods.

The use of multiple alignments in prediction overcomes this bias by providing a mechanism to create position-specific propensities as opposed to fixed ones, in the case of single sequence prediction. This occurs through the averaging process over the columns of the alignment, which in reality, creates context-dependent values reflecting more faithfully the sequence substitutions allowed for the particular structural environment. The inclusion of very similar sequences will not generate the necessary variability, and as consequence exhibits lower accuracy gains. On the other hand, divergence in low identity alignments translates in better prediction.

Given the fact that short identical sequences can adopt entirely different conformations [30, 31] and the present results, alternative ways to deal with the structure prediction problem should be employed. One approach is to disregard the common view of amino acids having a strict propensity to form a particular secondary structure, in favor of a setting in which the structure acts upon actively selecting the optimal sequence. Multiple alignment incorporation proved to be a successful alternative to the old vision, but still more work is needed to raise even more the accuracy levels.

References

1. Anfinsen, C.: Principles that govern the folding of protein chains. Science **181** (1973) 223–30
2. Rost, B.: Prediction in 1D: secondary structure, membrane helices, and accessibility. Methods Biochem Anal **44** (2003) 559–87
3. Rost, B.: Review: protein secondary structure prediction continues to rise. J Struct Biol **134** (2001) 204–18
4. Garnier, J., Levin, J.: The protein structure code: what is its present status? Comput Appl Biosci **7** (1991) 133–42
5. Rackovsky, S.: On the existence and implications of an inverse folding code in proteins. Proc Natl Acad Sci U S A **92** (1995) 6861–3
6. Kloczkowski, A., Ting, K.L., Jernigan, R., Garnier, J.: Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. Proteins **49** (2002) 154–66
7. Zvelebil, M., Barton, G., Taylor, W., Sternberg, M.: Prediction of protein secondary structure and active sites using the alignment of homologous sequences. J Mol Biol **195** (1987) 957–61
8. Rost, B., Sander, C.: Combining evolutionary information and neural networks to predict protein secondary structure. Proteins **19** (1994) 55–72
9. Salamov, A., Solovyev, V.: Protein secondary structure prediction using local alignments. J Mol Biol **268** (1997) 31–6
10. King, R., Sternberg, M.: Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. Protein Sci **5** (1996) 2298–310
11. Frishman, D., Argos, P.: Seventy-five percent accuracy in protein secondary structure prediction. Proteins **27** (1997) 329–35
12. Abagyan, R., Batalov, S.: Do aligned sequences share the same fold? J Mol Biol **273** (1997) 355–68
13. Rost, B.: Twilight zone of protein sequence alignments. Protein Eng **12** (1999) 85–94
14. Chothia, C.: Proteins. One thousand families for the molecular biologist. Nature **357** (1992) 543–4
15. Pascarella, S., Argos, P.: Analysis of insertions/deletions in protein structures. J Mol Biol **224** (1992) 461–71
16. Di Francesco, V., Garnier, J., Munson, P.: Improving protein secondary structure prediction with aligned homologous sequences. Protein Sci **5** (1996) 106–13
17. Altschul, S., Madden, T., Schffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res **25** (1997) 3389–402
18. Jones, D.: Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol **292** (1999) 195–202
19. Cuff, J., Barton, G.: Application of multiple sequence alignment profiles to improve protein secondary structure prediction. Proteins **40** (2000) 502–11
20. Petersen, T., Lundegaard, C., Nielsen, M., Bohr, H., Bohr, J., Brunak, S., Gippert, G., Lund, O.: Prediction of protein secondary structure at 80% accuracy. Proteins **41** (2000) 17–20

21. Rost, B., Sander, C.: Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol **232** (1993) 584–99
22. Cuff, J., Barton, G.: Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins **34** (1999) 508–19
23. Przybylski, D., Rost, B.: Alignments grow, secondary structure prediction improves. Proteins **46** (2002) 197–205
24. Bernstein, F., Koetzle, T., Williams, G., Meyer, E., Brice, M., Rodgers, J., Kennard, O., Shimanouchi, T., Tasumi, M.: The Protein Data Bank: a computer-based archival file for macromolecular structures. J Mol Biol **112** (1977) 535–42
25. Heringa, J., Sommerfeldt, H., Higgins, D., Argos, P.: OBSTRUCT: a program to obtain largest cliques from a protein sequence set according to structural resolution and sequence similarity. Comput Appl Biosci **8** (1992) 599–600
26. Sander, C., Schneider, R.: Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins **9** (1991) 56–68
27. Kabsch, W., Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers **22** (1983) 2577–637
28. Matthews, B.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta **405** (1975) 442–51
29. Goldman, N., Thorne, J., Jones, D.: Using evolutionary trees in protein secondary structure prediction and other comparative sequence analyses. J Mol Biol **263** (1996) 196–208
30. Argos, P.: Analysis of sequence-similar pentapeptides in unrelated protein tertiary structures. Strategies for protein folding and a guide for site-directed mutagenesis. J Mol Biol **197** (1987) 331–48
31. Cohen, B., Presnell, S., Cohen, F.: Origins of structural diversity within sequentially identical hexapeptides. Protein Sci **2** (1993) 2134–45

Tests of Automatic Annotation Using KOG Proteins and ESTs from 4 Eukariotic Organisms

Maurício de Alvarenga Mudado, Estevam Bravo-Neto, and José Miguel Ortega

Universidade Federal de Minas Gerais, Avenida Antonio Carlos, 6627,
Belo Horizonte MG, Postal Code 486, Brazil
miguel@icb.ufmg.br
http://www.biodados.icb.ufmg.br

Abstract. BLAST homology searches have been largely used to annotate function to novel sequences. Secondary databases like KOG can be used in this intention since their sequences have functional classification. We devised an experiment where public ESTs from four eukariotic organisms, which protein sequences are present in the KOG database, are classified to functional KOG categories using tBLASTn. First we assigned the ESTs from one organism to KTL (KOG, TWOG and LSEs) proteins and then we searched the database depleted of the same organism's proteins to simulate a novel transcriptome. Data show that classification was correct (assignment equals annotation) 87.2%, 96.8%, 92.0%, 88.7% for *A. thaliana*(Ath), *C. elegans*(Cel), *D. melanogaster*(Dme) and *H. sapiens*(Hsa) respectively. We have estimated identity cutoffs for all organisms to use with tBLASTn. These cutoffs trim the same amount of events that a BLASTn in order to minimize false positives in consequence of sequence errors. We found values of 80%, 78%, 78% and 84% for amino-acid identity cutoff for Hsa, Dme, Cel and Ath, respectively. We then evaluated our system by comparing the KTL categories of the assigned ESTs with the KTL categories that the ESTs were classified without the organism's KTL proteins. Moreover, we show the potential of annotation of the KOG database and the ESTs used. Suplementary Information can be found at: http://www.biodados.icb.ufmg.br

1 Introduction

Homology searches have been largely used to annotate the putative function of novel described sequences, either nucleotides or aminoacids. Usually software from the BLAST package [8] is used in this type of search [2] and the best hit (higher bit score) associated with a cutoff requirement of low E value is sufficient to establish a relationship of homology between query and subject [12][13]. Quality of annotation remarkably depends on the quality of the database that is being used as subject in homology searches. Secondary databases are currently available where sequences are not only deposited, but classified into functional categories. These databases are being widely used in the categorization of ESTs [4][18][5]. One of these databases is KOG [17], at NCBI, which organizes protein

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 141–152, 2005.

entries from seven organisms with complete sequenced genome into three classes of occurrence: KOG, TWOG and LSE, which occur in three, two or only one organism, respectively. Each protein has received a KOG ID - e.g. enolase is KOG0047. All orthologs, and eventually occurring paralogs, are classified under the same ID, and there are IDs for the three classes of KOG. Thus, this database is an attractive subject for testing automated annotation procedures. ESTs and transcriptome projects have showed its importance not only for gene discovery [1] but also for analysis of differential expression of genes [14][6][16]. ESTs are known to bear up to 4% of sequencing errors due to its single-pass nature. Development of automated annotation for ESTs is already being issued [3][15]. Annotation is mostly solved with the use of best hit to the subject database, but the identity of a nucleotide sequence, that contains errors, to an aminoacid sequence of the proper organism have not been addressed yet. It has been largely accepted (e.g. UniGene database - Lukas Wagner, personal communication,[19]) that 96% identity at nucleotide level is sufficient to assign an EST to the correspondent nucleotide cDNA sequence. Errors occurring in the third base of the codons tend to be silent in either tBLASTn or BLASTx searches. However, errors in the first two bases of the codon are expected to be hazardous to the alignment. In this work we set up to define a cutoff in BLASTx / tBLASTn searches that would be equivalent to 96% identity cutoff in nucleotide to nucleotide comparisons (BLASTn).Then we devised an experiment where we initially assigned ESTs to proteins from the KOG database of the proper organism and further annotated the ESTs with the entire KOG database lacking the proteins from the organism whose ESTs were used to query the database. This procedure simulates the annotation of a novel transcriptome. Furthermore, we evaluated our procedure verifying if the annotation was either correct, resulting in the same database ID, changed to a different one or even speculative (ESTs not assigned to any organism's protein but annotated by other organism's sequences).

2 Material and Methods

2.1 Vector Sequences

The pUC18 sequences used in this work have been provided by 3 laboratories from *Universidade Federal de Minas Gerais* (UFMG) that integrate the network *Rede Genoma de Minas Gerais*. The reactions were made in a single pool and divided into tubes for the PCR sequencing reaction. After the reaction, the sequences were joint again in the same tube, mixed, and then divided into three 96 sequencing well plates. Each plate was run 3 times on a MegaBASE sequencing equipment, yielding a total of 864 reads. From those, 846 processed ESD files were obtained.

2.2 Other Sequences

The EST sequences were downloaded from dbEST database [9] at Mai/2003. All KTL proteins and KOG conserved domains were downloaded from NCBI homepage [7] from the "kyva" file. We then selected the 88,613 classified KTL proteins found at the "kog, "twog" and "lse" files at the same address, to use in the BLAST searches. The KTL proteins are divided in 60,758 KOG, 4,451 TWOG and 23,404 LSE proteins. To retrieve the 50 CDS relative to the 50 KOG proteins from the four organisms, we selected the 100 more expressed KOG proteins that were hit by the ESTs from the four organisms (data not shown). We chose the 50 ones that have only one representing ortholog protein, to avoid ESTs being aligned to paralogs. We downloaded the respective mRNA sequences from these 50 KOG proteins (NCBI provides a list of proteins from KOG database assigned to their relative mRNAs - called "kyva=gb"). We then selected only the CDS of these mRNAs and removed the stop codons, by parsing the genbank file with a PERL script, to assure the proportion of identity between the alignments of ESTs to its proteins/nucleotides.

2.3 Data Processing

All data were processed using MySQL version 3.23.58 and scripts wrote in PERL language, version 5.8.0. The BLAST software package version 2.2.8 was obtained from NCBI. PHRED software version 0.020425.c was obtained (see [10]), thanks to Phill Green. All processing was made on a Linux Red Hat 9 machine, Pentium IV HT, 2.4 GHz and 1 GB RAM. The BLAST searches were run additionally on four other machines with similar power of processing and same operational system.

2.4 BLASTs

The tBLASTn/BLASTn were run with the following parameters: -m 8 -b 10e6 -e 1e-10 -F f . These parameters activate the tabular output of BLAST, allows up to 10 million hits to one protein (the default is 250) and deactivates the low complexity filter, respectively. The low-complexity filter was deactivated in order to permit tBLASTn to achieve 100% identity in the alignments.

2.5 PHRED

The software PHRED was run with the following parameters: -trim_alt "" -st fasta -trim_cutoff <n> Which activates the trimming algorithm selects the file type and activates the trimming with error cutoff (n) respectively.

2.6 Statistics

When necessary data were reported as means ± SEM (standard error of the mean).

3 Results

3.1 Defining Cutoffs

To define a cutoff for tBLASTn that is equivalent to 96% for BLASTn, we took advantage of 846 sequence reads of pUC18 (see material and methods) and aligned these sequences with either BLASTn to the published nucleotide sequence (genbank access number L09136) or with tBLASTn to a single frame translation starting at the first nucleotide downstream to the primer. We solved the problem of alignments to stop codons by representing the respective positions with the "*" character, what leads to 100% identity to tBLASTn alignments (not shown). Reads were trimmed with PHRED basecalling software under increasing error acceptance, using trim_alt PHRED internal algorithm. (E.g. 1% of error corresponds to PHRED 20 , 10% to PHRED 10). Data presented in figure 1A show that, for all error densities used, alignments of single-pass pUC18 reads (here simulating controlled ESTs) to the nucleotide sequence result, in average, to more than 96% identity , while alignments to the aminoacid sequences yielded lower levels of identity.

In order to investigate the behavior of the actual cDNA sequences we downloaded large sets of ESTs (Table 1) from the four organisms present in the KOG database (ath: *A. thaliana*; Cel : *C. elegans*; Dme: *D. melanogaster*; *hsa: H. sapiens*) from dbEST. We then selected 50 KOG proteins from each organism requiring that they corresponded to the most occurring ESTs and did not have paralogs, thus hits should probably point to a single protein. Data in figure 1B show that ESTs, aligned with tBLASTn to the amino-acid sequences, consistently show average levels of identity lower than the correspondent complete CDS nucleotide sequences. Moreover, the identities observed for each EST collection seem to represent the error density characteristic of the collection, as judged by comparison with the data presented in figure 1A.

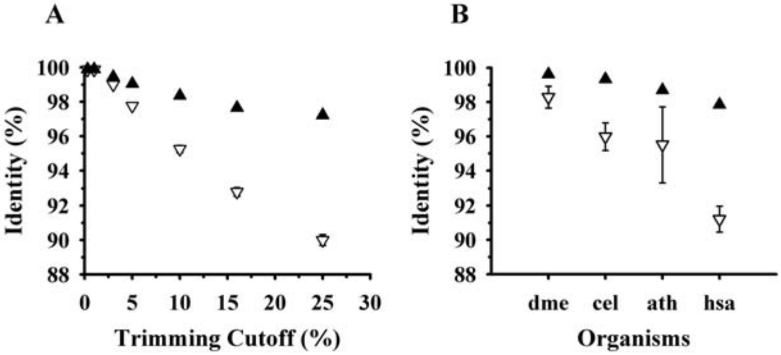


Fig. 1. A. Plot of mean identity ± mean standard error obtained from tBLASTn and BLASTn experiments with PUC18 using variable PHRED trim cutoff parameters. B. Plot of average identity and mean standard error obtained from tBLASTn and BLASTn experiments with a set of ESTs and 50 nucleotide/aminoacid sequences from the four organisms. Inverted open triangles are tBLASTn experiments and straight full triangles are BLASTn experiments. The n for pUC18 ranged from 846 reads with 25% of error to 673 reads with 0.3% of error. The n for Dme, Cel, Ath and Hsa was 23,630, 14,096, 1,891 and 14,484 sequences respectively

Table 1. Organisms and the respective ESTs, KOGs and proteins used in this work

Organisms	ESTs	KOGs	Proteins
Arabidopsis thaliana	178,538	4,872	24,154
Caenorhabditis elegans	215,200	5,306	17,101
Drosophila melanogaster	261,404	5,145	10,517
Homo sapiens	1,941,556	6,572	26,324
pUC18*	846 **	-	1 ***

* pUC18 stands for the commercial vector (see GenBank accession number L09136). ** pUC18 reads obtained by sequencing. *** The nucleotide sequence of pUC18 was translated into 1 protein sequence.

To estimate a cutoff for tBLASTn that would correspond to the 96% cutoff for BLASTn and to find the amount of events that these two cutoffs represent, we performed BLASTn and tBLASTn alignments using the pUC18 reads and its respective nucleotide/aminoacid sequences (Figure 2A). Reads were processed with 16% trim_alt cutoff (this procedure yields a maximum score plateau when aligning pUC18 reads to its nucleotide/amino-acid sequences - data not shown). Figure 2B shows the same tuples as in figure 2A but grouped by number of events, so it is possible to conclude that 96% of identity cutoff, when aligning nucleotides, corresponds to 93% of the totality of tuples. Therefore, the identity cutoff value that retrieves the same amount of tuples when using aminoacid

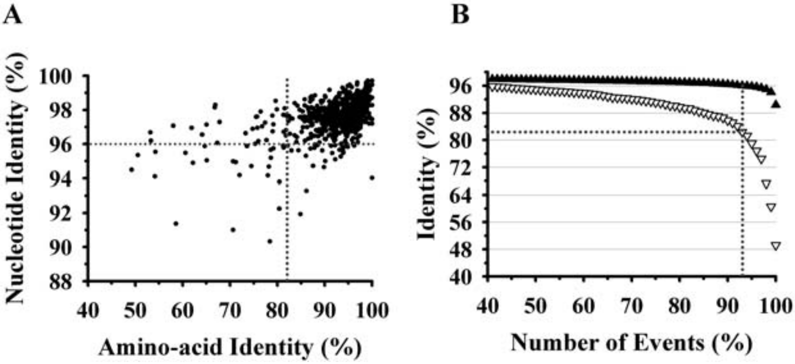


Fig. 2. A. Plot of tBLASTn - BLASTn tuples result of BLASTs performed with the translated and the nucleotide sequence of pUC18 with the reads obtained by automatic sequencing and PHRED 8. The dotted lines limit 96% and 82% of identity cutoff for BLASTn and tBLASTn. B. Same plot organized by number of events. The open inverted triangles represent tBLASTn plots and the full straight triangles BLASTn. The horizontal dotted line shows the cutoff for tBLASTn when 96% of identity for BLASTn is the reference. The vertical dotted line shows the ammount of events that the two cutoffs are representing

sequences as a target is 82.3% (depicted by the dotted lines in figure 2A). This same procedure was performed with the 4 organisms sequences and we found values of 80%, 78%, 78% and 84% for amino-acid identity cutoff for Hsa, Dme, Cel and Ath, respectively (data not shown). The mean cutoff value found for the 4 organisms (80% ± 2.8) is very close to the one found for pUC18 (82.3%). The percentage of events that these values collect ranged from 84% to 99%, suggesting that these cutoffs are discarding a minority of correct events when using tBLASTn.

3.2 Simulating Novel Transcriptomes

To test if these cutoff values are adequate, and to verify the accuracy of an automatic annotation experiment using ESTs and KTL proteins with tBLASTn, we conducted the pipeline as explained in figure 3.A. First, all ESTs from one organism (eg. Dme) are searched against the KTL proteins from the same organism. The best matches from this experiment are therefore assigning Dme ESTs to KTL proteins. Second, all ESTs from Dme are searched against the KOG database, but lacking Dme proteins, simulating in this way an annotation of a novel transcriptome. The best matches from this second experiment can be classified into 3 groups: correct annotation, when an EST from the second experiment is assigned to the same KOG ID as in the first experiment; speculative annotation, when an EST has not been assigned to a KOG ID in the first experiment, but it found a hit to a KOG ID in the second experiment; changed annotation, when an EST points to a KOG ID in the second experiment that is different from the KOG ID it was assigned to in the first experiment. The first experiment was conducted using the respective cutoff value for each of the four organisms and the accuracy of the annotation measured with the second experiments. There is two further possibilities of missing annotations (figure 3B), where ESTs are assigned but miss annotations in the simulation of a novel transcriptome (assigned but no hit), and where ESTs have no hit at all in neither experiments (no hit). When analyzing the totality of annotated ESTs, this methodology is able to correctly process around 90% (87.2%, 96.8%, 92.0%, 88.7% for *A. thaliana*, *C. elegans*, *D. melanogaster* and *H. sapiens* respectively), using the cutoffs determined. The percentage of changed annotation remains very small for all organisms, never overscoring 5%. The speculative annotation is more expressive in Hsa and Ath, but with values below 10% (data not shown).

We tested if annotability is altered by using different identity cutoffs when assigning ESTs to KOG IDs. We performed rounds of annotation, starting from 45% up to 100% of identity cutoff. Figure 4 shows the percentage of ESTs that are found in the 3 categories, when using these cutoffs and the 4 organisms sequences. Together these 3 categories and all assigned ESTs forms the group of ESTs that are potentially annotable. We found that, in most cases, the use of low cutoff values augments the group of correct annotation relative to the other groups. Moreover, the changed annotation stays at low values (below 2% of the ESTs in most cases). Changed annotation slightly diminishes when the cutoff is raised, probably because fewer errors are permitted in the alignment.

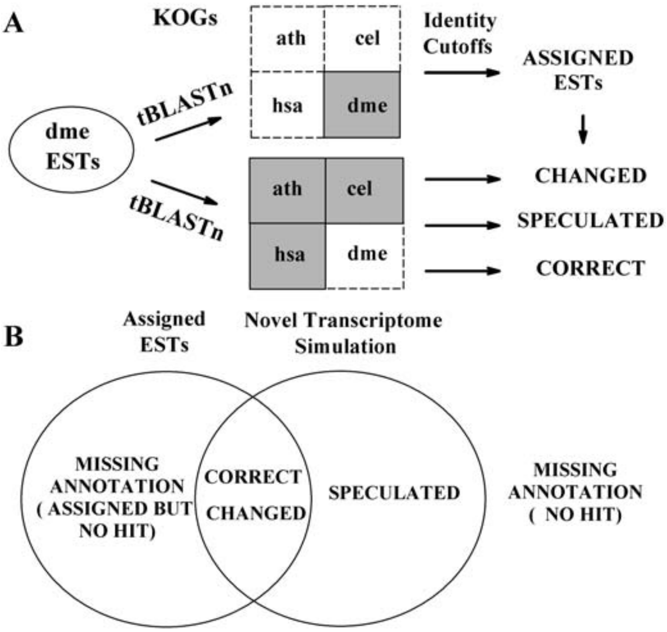


Fig. 3. Schema of the experiment devised to test the annotation of the ESTs from the four organisms with the KOG database. A. The experiment is made using ESTs and KTLs (KOGs, TWOGs and LSEs) from Dme as an example. It is divided in two steps, first a tBLASTn is made against KTL proteins only from Dme (grey square), to assign ESTs to KTL classes. The next step is a tBLASTn of ESTs from Dme against KTL proteins from the other organisms (the tree grey squares) but not from Dme, then simulating a novel transcriptome. The classification of the annotation is obtained by comparing the classes of KTL that the ESTs were assigned in the first and second experiments. Three classes are possible: changed, speculated and correct annotations. B. The products obtained from A. 5 classes are possible: correct, changed and speculated and 2 classes of missing annotations: the "no hit" and "assigned but no hit" ESTs

On the contrary, when raising the cutoff values above 80%, the percentage of speculative annotation raises because less ESTs are being assigned to a KOG ID in the first experiment (less alignments pass this filter). This can be assumed by the diminishment of the correct annotation class in the same proportion of the increase of the speculative class. We found that, for Dme and cutoff values below 90%, annotation of almost 50% of the total ESTs is to correct KOG IDs. This is followed by Cel annotation, with 40% of total ESTs. Distinctly, Ath and Hsa annotation might be classified as poorer, with only 20% and 13% of all ESTs being correctly annotated.

In order to show the potential of annotation that is gained or lost by using different identity cutoffs, we plotted in figure 5 the amount of ESTs that are potentially annotable: correct, changed, speculated and assigned but with miss-

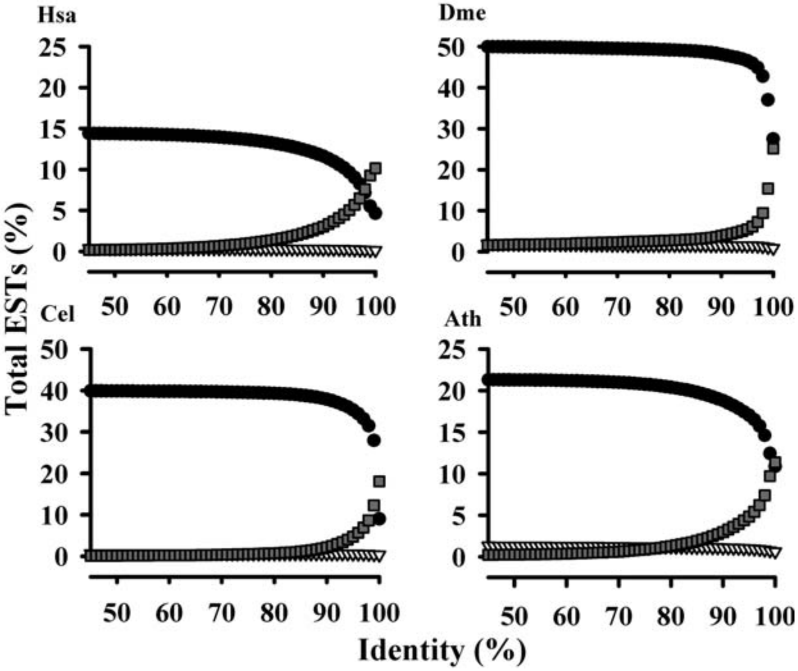


Fig. 4. Testing the annotation with KOG using different identity cutoffs. Full circles represent the correct annotation. Gray squares are representing speculations of a right KOG and inverted open triangles represent changed annotations. Hsa: *Homo sapiens*. Cel: *C. elegans*. Dme: *D. melanogaster*. Ath: *A. thaliana*

ing annotation when simulating new transcriptomes (the black + grey areas). The definition of potentially annotable is used because these ESTs have had a hit to a KOG protein in any of the experiments. The amount of ESTs that this methodology was unable to classify (no hit to any database when assigning and when simulating novel transcriptomes), can be observed by calculating the complementary area of the black + grey areas. The black area represent the correct, speculated and changed annotations. In other words, the ESTs that had been annotated by any KOG protein in the second experiment. The grey areas represent the ESTs that had been assigned to a KOG ID but had no annotation in the second experiment. This can be caused by assignments to unique proteins of the organism (dark grey areas) in the first experiment. Using lower cutoff values, Dme ESTs have the best potential of annotability with around 77% of all ESTs being annotable and less than 23% unable to be classified. It is followed by Cel (75% and 25%) and Ath with a good anotability (around 80%) but with a poor potential of classification by the system (57% loss). This loss can be explained by a large amount of assignments to LSE proteins (discussed below). *H. Sapiens* is a special case, with a very low efficiency of around 35% and with almost 20% of its ESTs unable to be classified by our system.

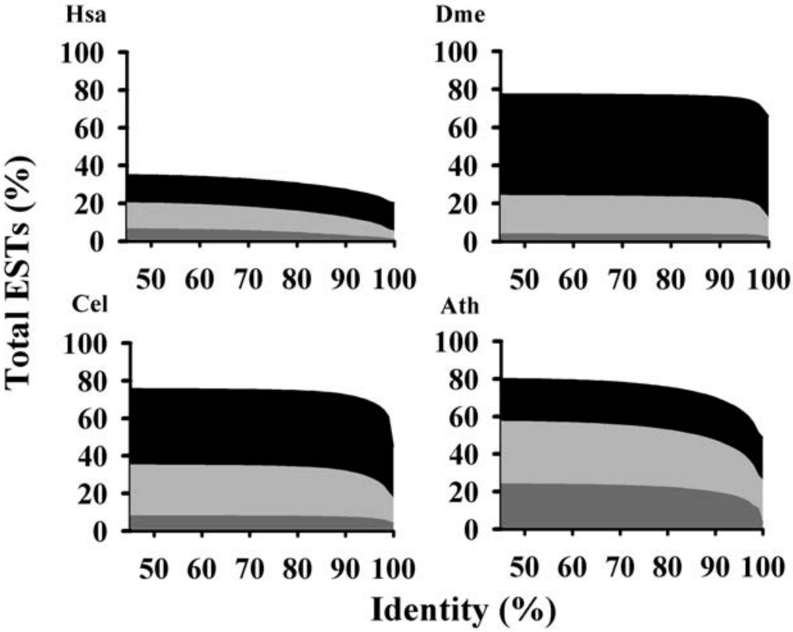


Fig. 5. Test of annotation with KOG using different identity cutoffs showing the percentage of ESTs that are annotable and the missing assigned ESTs. The black + grey areas show the total percentage of annotable ESTs. The black areas are representing the correct, speculated and changed annotations. The light grey areas are representing the assigned ESTs to KOG proteins but with no hit when simulating a new organism transcriptome. The dark grey areas are representing the amount of ESTs from the light grey areas that were assigned to LSE proteins. Hsa: *Homo sapiens.* Cel : *C. elegans.* Dme: *D. melanogaster.* Ath: *A. thaliana*

3.3 Searching LSEs

When simulating a novel transcriptome, a low potential of annotability and wrong KOG ID classification can be explained by ESTs being assigned to LSE proteins, genes that are present only in the organism being annotated. In this case, annotation can result in two cases: a no hit to any KOG, TWOG or LSE from other organisms or an undesired changed annotation. We did a survey into all assigned EST sequences using the organism's cutoffs, to test if assignment to LSE proteins were biasing these results. We found that Ath, as expected for being the only plant in the database, has the higher number of ESTs assigned to LSE proteins (41% of the total 129,100 EST sequences assigned), followed by Hsa (16% of 574,091), Cel (11% of 160,065) and Dme (6% of 194,838). Furthermore we surveyed all set of changed annotation and missed assigned ESTs to obtain the percentages of these groups that were initially assigned to LSE proteins. We found that, from the group of changed annotation, Hsa and Ath have 30.8% and 35.5% of sequences assigned initially to LSE proteins respectively. Cel and

Dme have a smaller amount with 10.3% and 16.6% respectively. From the set of sequences that were assigned but are missing an annotation, Hsa and Ath have again the greater percentage with 29.0% and 42.4% belonging to LSE proteins respectively. Cel has 23.2% followed by Dme with only 7.0%. Altough in some cases, like Hsa and Ath that have around 30% of all changed annotation caused by EST sequences assigned to LSE proteins, these numbers are representing a very small portion of the total EST sequences used. Thus, the most part of sequences that were initially assigned to LSE proteins are not being classified as a changed annotation. This result indicates that ESTs assigned to LSE proteins are not causing a strong bias on changed annotation. In most cases, less than 3% of all EST sequences assigned to LSE proteins are contributing to this group. Except by Ath, the group of EST sequences that had been assigned but is missing an annotation is not greatly increased by assignments to LSE proteins. The plant shows a significant ammount of LSE proteins being assigned but are missing an annotation.

4 Discussion and Conclusion

Assuming a cutoff value for identity when using tBLASTn is necessary since the big volume of data is diminished, requiring less computational effort and storage space.

Our system was able to correctly classify around 90% of all annotable ESTs which passed the first 10^{-10} BLAST E-value cutoff.

The identity cutoff values found for the 4 organisms are therefore suitable as changed annotation is almost not altered for all cutoffs and always have low values, representing less than 5% of all ESTs. Our results also show that the use of high identity cutoffs can be harmful to an automatic annotation procedure. This is depicted by the raise of speculative annotation and diminishment of correct annotation, when using high identity cutoffs (above 80%), in figure 4.

Hsa lacks a good potential of annotability and we speculate two possible explanations. First, probable low quality EST sequences in the database: sequences with low lengths and high error rates (see Fig.2B). We are currently investigating this fenomena.

The second explanation is that KOG is not yet a complete database to annotate Hsa and it may lack more than 60% of the necessary proteins to annotate a human transcriptome. To explain this, we'll perform a future annotation experiment with larger secondary databases like Uniprot [11] or the NCBI's nr database.

However, all annotable Hsa ESTs showed a high rate of correctness (above 87%). Furthermore, Dme, Cel and Ath showed a better automatic annotation potential with around 80% of annotability.

We conclude that KOG is a reliable database for EST annotation depicted by the results obtained with the four organisms studied. Suplementary information was made available with APACHE/PHP and can be found at http://www.biodados.icb.ufmg.br .

References

1. Adams, M.D., Kelley, J.M., Gocayne, J.D., Dubnick, M., Polymeropoulos, M.H., Xiao, H., Merril, C.R., Wu, A., Olde, B., Moreno, R.F. et al.: Complementary DNA sequencing: expressed sequence tags and human genome project. Science **252** (1991) 1651–1656
2. Altschul, S.F., Madden, T.L., Schaffer, AMINO-ACID, Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25** (1997) 3389–3402
3. Cuff J.A., Birney E., Clamp M.E., Barton G.J.: ProtEST: protein multiple sequence alignments from expressed sequence tags. Bioinformatics. **16(2)** (1999) 111–6
4. Faria-Campos A.C., Cerqueira G.C., Anacleto C., Carvalho C.M.B., Ortega J.M.: Mining microorganism EST databases in the quest for new proteins. Genet. Mol. Res. **2(1)** (2003) 169–177
5. Felipe M.S., Andrade R.V., Petrofeza S.S., Maranhao A.Q., Torres F.A., Albuquerque P., Arraes F.B., Arruda M., Azevedo M.O., Baptista A.J., Bataus L.A., Borges C.L., Campos E.G., Cruz M.R., Daher B.S., Dantas A., Ferreira M.A., Ghil G.V., Jesuino R.S., Kyaw C.M., Leitao L., Martins C.R., Moraes L.M., Neves E.O., Nicola A.M., Alves E.S., Parente J.A., Pereira M., Pocas-Fonseca M.J., Resende R., Ribeiro B.M., Saldanha R.R., Santos S.C., Silva-Pereira I., Silva M.A., Silveira E., Simoes I.C., Soares R.B., Souza D.P., De-Souza M.T., Andrade E.V., Xavier M.A., Veiga H.P., Venancio E.J., Carvalho M.J., Oliveira A.G., Inoue M.K., Almeida N.F., Walter M.E., Soares C.M., Brigido M.M.: Transcriptome characterization of the dimorphic and pathogenic fungus Paracoccidioides brasiliensis by EST analysis. Yeast. **20(3)** (2003) 263–71
6. Franco G.R., Rabelo E.M., Azevedo V., Pena H.B., Ortega J.M., Santos T.M., Meira W.S., Rodrigues N.A., Dias C.M., Harrop R., Wilson A., Saber M., Abdel-Hamid H., Faria M.S., Margutti M.E., Parra J.C., Pena S.D.: Evaluation of cDNA libraries from different developmental stages of Schistosoma mansoni for production of expressed sequence tags (ESTs). DNA Res. **4(3)** (1997) 231–40
7. ftp://ftp.ncbi.nih.gov/pub/COG/KOG/
8. http://www.ncbi.nlm.nih.gov/BLAST/
9. http://www.ncbi.nlm.nih.gov/dbEST
10. http://www.phrap.org
11. http://www.uniprot.org
12. Koonin E.V., Fedorova N.D., Jackson J.D., Jacobs A.R., Krylov D.M., Makarova K.S., Mazumder R., Mekhedov S.L., Nikolskaya A.N., Rao B.S., Rogozin I.B., Smirnov S., Sorokin A.V., Sverdlov A.V., Vasudevan S., Wolf Y.I., Yin J.J., Natale D.A.: A comprehensive evolutionary classification of proteins encoded in complete eukaryotic genomes. Genome Biol. **5(2)** (2004) R7
13. Koonin, Eugene V. and Galperin, Michael Y.: Sequence - Evolution - Function Computational Approaches in Comparative Genomics. Norwell (MA) 2003
14. Lee N.H., Weinstock K.G., Kirkness E.F., Earle-Hughes J.A., Fuldner R.A., Marmaros S., Glodek A., Gocayne J.D., Adams M.D., Kerlavage A.R., et al.: Comparative expressed-sequence-tag analysis of differential gene expression profiles in PC-12 Cel ls before and after nerve growth factor treatment. Proc Natl Acad Sci. **92(18)** (1995) 8303–7
15. McCallum J, Ganesh S.: Text mining of DNA sequence homology searches. Appl Bioinformatics. **2(3 Suppl)** (2003) S59–63

16. Stekel D.J., Git Y., Falciani F.: The Comparison of Gene Expression from Multiple cDNA Libraries. Gen. Res. **10** (2000) 2055–2061
17. Tatusov R.L., Fedorova N.D., Jackson J.D., Jacobs A.R., Kiryutin B., Koonin E.V., Krylov D.M., Mazumder R., Mekhedov S.L., Nikolskaya A.N., Rao B.S., Smirnov S., Sverdlov A.V., Vasudevan S., Wolf Y.I., Yin J.J., Natale D.A.: The COG database: an updated version includes eukaryotes. BMC Bioinformatics. **4(1)** (2003) 41.
18. Vettore A.L., da Silva F.R., Kemper E.L., Souza G.M., da Silva A.M., Ferro M.I., Henrique-Silva F., Giglioti E.A., Lemos M.V., Coutinho L.L., Nobrega M.P., Carrer H., Franca S.C., Bacci Junior M., Goldman M.H., Gomes S.L., Nunes L.R., Camargo L.E., Siqueira W.J., Van Sluys M.A., Thiemann O.H., Kuramae E.E., Santelli R.V., Marino C.L., Targon M.L., Ferro J.A., Silveira H.C., Marini D.C., Lemos E.G., Monteiro-Vitorello C.B., Tambor J.H., Carraro D.M., Roberto P.G., Martins V.G., Goldman G.H., de Oliveira R.C., Truffi D., Colombo C.A., Rossi M., de Araujo P.G., Sculaccio S.A., Angella A., Lima M.M., de Rosa Junior V.E., Siviero F., Coscrato V.E., Machado M.A., Grivet L., Di Mauro S.M., Nobrega F.G., Menck C.F., Braga M.D., Telles G.P., Cara F.A., Pedrosa G., Meidanis J., Arruda P. Telles G.P., Braga M.D.V., Dias Z., Lin T. Quitazau J.AMINO-ACID, da Silva F. R., Meidanis J. Analysis and functional annotation of an expressed sequence tag collection for tropical crop sugarcane. Genome Res. **13(12)** (2003) 2725–35
19. Wheeler D.L., Church D.M. ,Federhen S., Lash A.E., Madden T.L., Pontius J.U., Schuler G.D.,Schriml L.M., Sequeira E., Tatusova T.A, Wagner L.: Database Resources of the National Center for Biotechnology. Nucl Acids Res **31** (2003) 28–33

Diet as a Pressure on the Amino Acid Content of Proteomes

Francisco Prosdocimi[1] and J. Miguel Ortega[2]

[1] Departamento de Biologia Geral, ICB-UFMG
franc@icb.ufmg.br
[2] Departamento de Bioquímica e Imunologia, ICB-UFMG
miguel@icb.ufmg.br

Abstract. Whether diet has been influencing the genomic and proteomic constitution of the organisms along the evolution is an interesting and not answered question. Here, we investigate the hypothesis that essential amino acids – the ones that are not produced by the organisms – have being replaced in proteins by non-essential ones. We compare the amino acid composition of the proteome from human, worm and fly, that cannot synthesize all amino acids, with the ones from plant, baker yeast and budding yeast, capable to synthesize all of them. The analysis was made with 190,074 proteins composed of 87,175,891 amino acids. Our data seems to evidence a little bias on the usage of non-essential amino acids by the metazoan organisms, except for the worm. Thus, the preliminary results shown here support the thesis that non-essential ones have replaced essential amino acids.

1 Introduction

Whether diet has been influencing the genomic constitution of the organisms along evolution is an interesting and not answered question. Considering this topic, two different and opposite evolutionary scenarios can be conceived.

In the first one, diet has not been made particular selection pressure in the proteome of ancestral organisms. This first scenario could be though in two ways: (1) the high-fitness organisms capable to reproduce have always been well fed and the bad fed individuals could not reproduce; or (2) the need of essential amino acids (EAA) is too small and even the worst fed organisms were capable to reproduce. This way the diet has never pressure for the genome or proteome modification.

A second scenario can be described if we consider diet as a putative mechanism for genome and proteome modification. In this case, ancestral organisms presenting a genome full of EAA would need to ingest a great amount of food and, if they could not get the nutrients, their proteins would not be produced appropriately. So, these organisms would produce few offspring. This way, mutations happening in ancestral proteins leading to the substitution from essential to non-essential amino acids (NEAA) would be positive selected and the organisms harboring them would be capable to produce a greater and fertile offspring.

The study of the diet influence in the genome modification can be evaluated through the analysis of the EAA content in proteins from different species. Consider-

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 153 – 159, 2005.

ing that many metazoan organisms (MO) cannot synthesize some amino acids, the essential ones, they need to be obtained through the diet. So, if diet has been acting as a selection pressure for the proteomic modification along evolution, one can imagine that MO will be changing their amino acid composition of proteins from essential to non-essential amino acids. This way, organisms would be each time less dependent of the amino acid composition of the proteins ingested for their perfect metabolism and functioning. So, the current work intends to verify if diet has been acting as a selective pressure for the proteome modification and, if so, in which scale it has been happening. Therefore, our main goal is try to find evidences for the substitution of EAA to NEAA in the metazoan organisms and investigate if this kind of substitution is happening all over the proteome or only in particular group of proteins.

Here, we report the analysis of the EAA content from all proteins of 6 genome-completed organisms. Three of them are MO -- *Homo sapiens (hsa)*, *Drosophila melanogaster (dme)* and *Caenorhabditis elegans (cel)* -- and, therefore, present 8 EAA; and the other three are non-metazoan organisms (NMO) holding enzymes to synthesize all amino acids – *Arabidopsis thaliana (ath)*, *Saccharomyces cerevisiae (sce)* and *Schizosaccharomyces pombe (spo)*. Two well-curated secondary genomic databases were used to evaluate the differences on the EAA usage in proteins: COG and RefSeq. The NCBI eukaryotic cluster of orthologous groups (KOG) was used to allow the comparison between evolutionary related proteins, in order to investigate if the amino acids have been changing in proteins with the same origin and function (Tatusov et al., 2003). The Reference Sequence database (RefSeq) was also used to evaluate the amino acid composition of proteins throughout the complete proteome of the selected organisms (Pruitt et al., 2000; Pruit et al., 2005), since KOG just contain proteins conserved by three organisms at least.

2 Methodology

2.1 Essentiality Index Ranking

For each KOG orthology group, it was created an index called "essentially index" (EI). This index represents the proportion of EAA in that KOG. The amino acid arginine was removed from the index since it is frequently called a semi-essential amino acid and it can be produced in some phases of organisms' life. So, the essential index was calculated as shown on (1).

The essentiality index:

$$EI = \frac{N(aa\_ess)}{N(aa\_total) - N(R)} \qquad (1)^*$$

* In KOGs presenting more than one gene (paralogs), the number of all amino acids were considered to generate the EI. The R, in formula, represents the number of arginines that were removed from the total.

EI pair-wise comparisons were made between MO and NMO for all KOGs. The total number of events and the number of times where NMO KOGs presented greater index than MO were counted.

2.2 Amino Acidic Index Clustering

For each organism, it was calculated amino acid usage for all its proteome based on RefSeq database. Once more, pair-wised comparisons were done between MO and NMO, this time to verify which amino acids one or other group are preferentially using. If diet has been influencing proteome modification, we would expect that the most dissimilar used amino acids between the groups should be the essential ones. So, hierarchical clustering of amino acidic indexes were produced using cluster software from Michael Eisen (Eisen et al., 1998) and normalized by $\log_2$.

$$PI_{AA} = \log_2 \frac{\%AA\_Usage_{MO}}{\%AA\_Usage_{NMO}} \tag{2}$$

3 Results

3.1 Download Data

Protein sequence data were downloaded from RefSeq and eukaryotic COG database (for *Schizosaccharomyces pombe* only KOG data was analyzed).

3.2 Raw Data Analysis

The first analysis performed was simply the calculation of EAA percentage in the proteome (table 1).

The EAA percentage in KOG appears to be greater than in RefSeq for all organisms analyzed, as well as lower in MO than NMO (with exception of *cel*), although these differences did not seem to be statistically significant.

Table 1. General project information

Database	Organism	# proteins	# aa	% EAA	Std EAA
KOG	Ath	24,155	9,981,732	46%	5,4%
	Sce	4,842	2,423,755	47%	5,6%
	Spo	4,234	1,915,466	46%	5,5%
	Cel	17,102	7,397,061	47%	6,7%
	Dme	10,518	5,214,193	44%	6,6%
	Hsa	26,325	11,431,714	44%	6,5%
	Total KOG	**87,176**	**38,363,921**		
REFSEQ	Ath	29,157	12,120,473	43%	5,3%
	Sce	5,868	2,913,021	45%	5,1%
	Cel	21,136	9,221,024	44%	6,8%
	Dme	18,759	10,563,721	41%	6,2%
	Hsa	27,978	13,993,731	41%	6,6%
	Total RefSeq	**102,898**	**48,811,970**		
	TOTAL	**190,074**	**87,175,891**		

3.3 Voting of KOGs Essentiality Index

The EI for each KOG orthology group shared by a couple of organisms (MOs and NMOs) was taken on account. So, the percentage of MO KOGs presenting higher EI than NMO ones was generated (Table 2).

Table 2. Percentage of KOGs with higher EI

	Ath	*Sce*
Hsa higher	45%	34%
Dme higher	45%	33%
Cel higher	60%	46%

Such as expected, *hsa* and *dme* has shown lower number of high-EI KOGs than *ath* and *sce*. Although the results have not shown an outstanding difference between MO and NMO they point out in the direction of the existence of a pressure. Once more, *cel* was an exception when compared to *ath* and considering missing information about amino acid biosynthesis pathways in this worm, further analysis was done preferentially with *hsa* and *dme*.

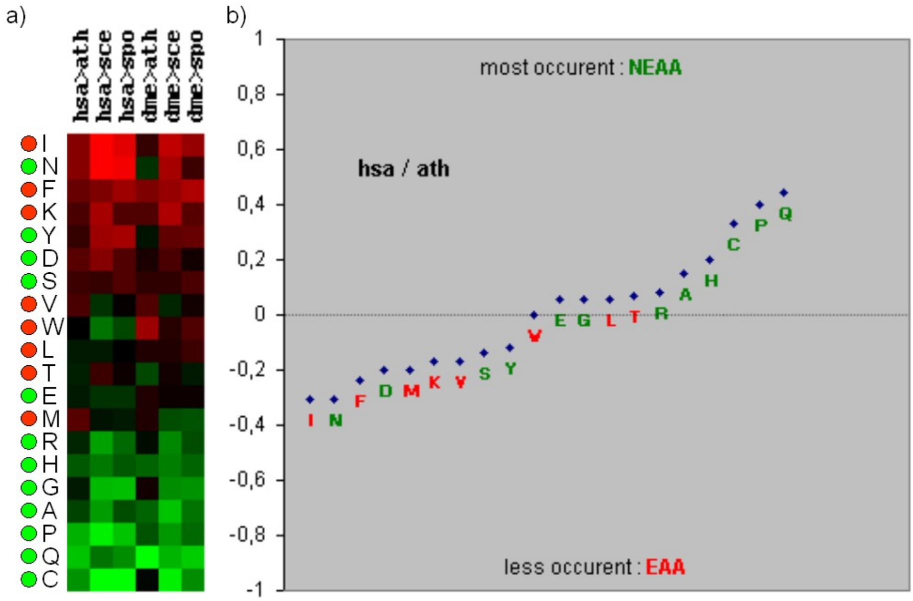


Fig. 1. Clustering analysis of amino acidic indexes between the indicated MO and NMO. A) Green and red filled circles indicate, respectively, NEAA and EAA. The colors in the plot represent the tendency of the amino acids to be present rather in MO (green) than NMO (red). The color intensity is the representation of amino acid PI. B) PI plot of amino acids for *hsa-ath*. EAA are shown in red and NEAA in green

3.4 Amino Acidic Index Clustering

In order to verify if non-essential amino acids (NEAA) are supposed to occur preferentially in MOs, an amino acid percentage usage was derived based on RefSeq data. A preference index ratio (PI) was defined as the percentage of each amino acid in MO divided by its the percentage in NMO normalized by $\log_2$.

Data for the comparison between *hsa* and *ath* is shown (Figure 1b). Remarkably, all the amino acids occurring preferentially in human than plant (A, H, C, P and Q) are NEAA. Data for other MO/NMO comparison also support this observation (figure 1a).

3.5 The First Hungry-Failed Proteins

It is also interesting to analyze which proteins would be mainly affected in an organism with a restrictive diet. This is the same of analyzing the proteins presenting the highest EI. The top ten proteins found for human genome are shown (Table 3).

These high-EI protein groups have not shown any particular relationship amongst them and they seem to be unrelated by means of molecular function or biological process.

Table 3. Human KOGs with higher EI

KOG	Description
KOG1721	FOG: Zn-finger
KOG0613	Projectin/twitchin and related proteins
KOG3594	FOG: Cadherin repeats
KOG3656	FOG: 7 transmembrane receptor
KOG3544	Collagens (type IV and type XIII), and related proteins
KOG2177	Predicted E3 ubiquitin ligase
KOG0619	FOG: Leucine rich repeat
KOG0516	Dystonin, GAS (Growth-arrest-specific protein), and related proteins
KOG3595	Dyneins, heavy chain
KOG1217	Fibrillins and related proteins containing Ca2+-binding EGF-like domains

3.6 The Proteins Under Selection Pressure

The identification of the KOGs harboring the higher number of expected amino acid changes (EAA to NEAA) between NMOs and MOs was also performed. The most different KOGs between *hsa* and *ath* are shown (Table 4). So, the EI of each KOG from both organisms was observed and we select the 10 top KOGs presenting the most different EI. These proteins have shown direct modification in amino acidic structure, changing their EAA to NEAA during evolution (considering they were the same on the common ancestor between *hsa* and *ath*).

Table 4. Hsa-Ath most different KOGs on EI

KOG	DIFF*	Description
KOG4752	38%	(J) Ribosomal protein L41
KOG0002	24%	(J) 60s ribosomal protein L39
KOG3491	19%	(S) Predicted membrane protein
KOG3445	17%	(J) Mitochondrial/chloroplast ribosomal protein 36a
KOG3500	15%	(C) Vacuolar H+-ATPase V0 sector, subunit M9.7 (M9.2)
KOG1793	14%	(S) Uncharacterized conserved protein
KOG4293	14%	(T) Predicted membrane protein, contains DoH and Cytochrome b-561/ferric reductase transmembrane domains
KOG3423	14%	(K) Transcription initiation factor TFIID, subunit TAF10 (also component of histone acetyltransferase SAGA)
KOG2346	13%	(S) Uncharacterized conserved protein

Interestingly, the first four proteins with most dissimilar EI KOGs between *ath* and *hsa* represent highly expressed ribosomal proteins.

4 Discussion

As far as we know this is the first attempt to investigate the hypothesis that diet has been influencing the proteome modification of the organisms. Since amino acids substitutions can be conservative, the genome of complex organisms might be modified as a response for a selection pressure on the preferentially usage of non-essential amino acids. We have shown indicatives of the occurrence of this kind of modification along evolution, although it seems to be happening mainly in the most expressed proteins (Table 4). At this moment, we are currently extending our investigation to a large number of organisms that might have any information about the requirement of EAAs. It would be highly desirable to study position-specific substitutions in orthologous proteins, since we could find which substitutions are more frequent, and if those ones happen using minimum pathways of nucleotide substitutions based on the genetic code. However, it is very probable that the substitutions are happening freely in the non-conservative regions of the proteins (that ones do not matched by local sequence alignment software). Moreover, the investigation of EAA proportion in specific processes and pathways should give us a glimpse on where these substitutions seem to be more relevant. Thus, the preliminary results shown here support the thesis that EAA has been replaced by non-essential ones at least in a long term way.

References

1. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A. 1998 Dec 8;95(25):14863-8.
2. Pruitt KD, Katz KS, Sicotte H, Maglott DR. Introducing RefSeq and LocusLink: curated human genome resources at the NCBI (2000). Trends Genet 16(1):44-47.

3. Pruitt KD, Tatusova, T, Maglott DR. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins (2005). Nucleic Acids Res 33(1):D501-D504.
4. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, Koonin EV, Krylov DM, Mazumder R, Mekhedov SL, Nikolskaya AN, Rao BS, Smirnov S, Sverdlov AV, Vasudevan S, Wolf YI, Yin JJ, Natale DA. The COG database: an updated version includes eukaryotes (2003). BMC Bioinformatics 4(1):41.

A Method for Comparing Three Genomes*

Guilherme P. Telles[1,**], Marcelo M. Brigido[2], Nalvo F. Almeida[3],
Carlos J.M. Viana[3], Daniel A.S. Anjos[4], and Maria Emilia M.T. Walter[4]

[1] University of São Paulo, Institute for Mathematical and Computer Sciences,
São Carlos, Brazil
gpt@icmc.usp.br

[2] University of Brasília, Institute of Biology, Brasília, Brazil
brigido@unb.br

[3] Federal University of Mato Grosso do Sul, Department of Computing and
Statistics, Campo Grande, Brazil
{nalvo, cjmv}@dct.ufms.br

[4] University of Brasília, Department of Computer Science, Brasília, Brazil
{dasanjos, mariaemilia}@unb.br

Abstract. The large amount of data from complete genomes gave rise to the need for computational tools to analyze and compare them. In this work, we propose a method for comparing three genomes simultaneously, at the basic level of their sequences. This comparison can indicate the set of genes shared by genomes, giving interesting clues about the metabolic pathways and proteins related to particular issues. The input for the method is three sets of gene coding sequences or products and the output are the sequences exclusive to each genome, the sequences common to pairs of genomes, and the sequences common to the three genomes. Because each sequence in a genome may be similar to many sequences in the other two genomes, some complicated situations may arise. The main feature of our method is the ability to avoid such situations. We used our method to compare genomes of two pathogenic and five non-pathogenic fungi, and made a biological analysis based on one of these results.

1 Introduction

The increasing availability of complete genomes has created the need for computational tools to analyze and compare them. Genome comparison is useful to investigate common functionalities of corresponding organisms and to get a better understanding of how genes or groups of genes are involved in particular functions and characteristics.

Different methods for genomic comparison have been described in the literature. They are based on sequence comparisons of raw genomic DNA, coding

[*] This project was partially funded by Finatec-DF, Fundect-MS and MCT/CNPq.

[**] Author to whom all correspondence should be addressed.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 160–169, 2005.

sequences or gene products [1, 5, 11, 10, 6]. Some of them make comparative analysis of pathogenic and non-pathogenic organisms trying to identify genes that contribute to infection and disease [4, 7]. There are also many computational tools available on the Internet that can be freely used [8, 14, 15, 9].

In this work we propose a method for comparing three genomes simultaneously, at the basic level of their sequences. The input for the method is a set of gene coding sequences or products and the output are the sequences exclusive to each genome, the sequences common to pairs of genomes, and the sequences common to the three genomes. This output corresponds to the regions in a Venn-Euler diagram, as the one shown in Figure 1. Determining the sequences in each region of the diagram may indicate the set of genes shared by genomes, giving interesting clues about shared metabolic pathways and proteins related to some particular issues.

We used our method to compare genomes of pathogenic and non-pathogenic fungi. Particularly, we compared two pathogenic fungi with five different non-pathogenic fungi, three at a time, trying to identify genes involved on pathogenicity. We also make a biological analysis from the results obtained by one of these comparisons.

The rest of the paper is organized as follows. In Section 2, we discuss some issues on comparing three genomes at the same time. In Section 3 we describe our method. Experiments with fungal genomes appear in Section 4. A biological analysis of one of the results obtained from the experiments is presented on Section 5. Finally, in Section 6, we make our concluding remarks.

2 3-Genome Comparison

Suppose that we have three genomes, and that a genome is a set of gene sequences that can be either coding DNA or peptidic gene products. If we are able to unambiguously assign the genes from each genome to a single region of a Venn-Euler diagram as shown in Figure 1, the regions in the diagram would represent the sequences exclusive to each genome, the sequences common to pairs of genomes, and the sequences common to the three genomes.

In order to produce the diagram, we use sequence similarity to select the sequences that are going to be assigned to each region of the diagram. There are a number of clear cut situations. Such cases are illustrated in Figure 2, where an edge connecting two genes means that the sequences are similar.

It can often be hard to decide in which region a sequence has to be included because a sequence may be similar to many other sequences in different genomes. This can lead to complicated relations among the sequences, whose biological meaning is unclear (Figure 3).

We propose a method for finding the sequences in each region of the diagram that avoids dealing with complicated cases. Our method starts finding similar sequences among the genomes. Then it finds as much sequences common to the three genomes as possible, taking into consideration a score given to every sequence triplet. Finally it finds as much sequences common to pairs

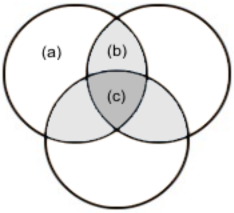


Fig. 1. A Venn-Euler diagram representing (a) sequences exlusive to one genome, (b) sequences restricited to two genomes, and (c) sequences common to the three genomes

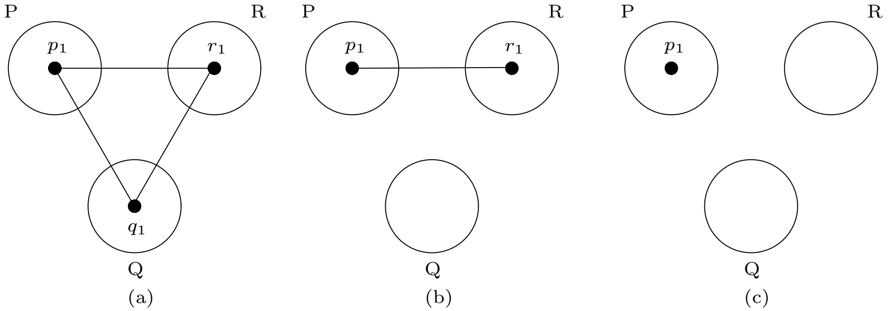


Fig. 2. Genes that can be unambiguously assigned to a region of the Venn-Euler diagram. (a) Genes in all the three genomes (triangles). (b) Genes in genomes P and R only (edges). (c) Genes exclusive to P. Other possibilities are symmetric and are not shown

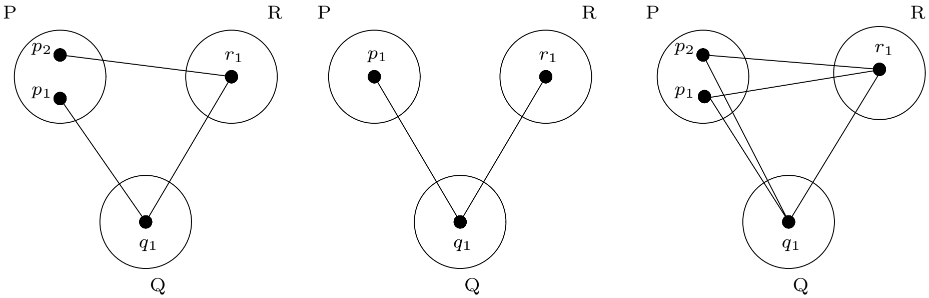


Fig. 3. Genes with not clearly defined biological relations, which makes hard their inclusion on the Venn-Euler diagram regions. Other situations similar to these are easy to find

of genomes as possible, also taking into consideration a score given to every sequence pair.

Our method is a general framework, that is, we leave some measures and thresholds unspecified. This allows the method to be specialized depending on many issues, such as processing power availability, size of the genomes, specificity and sensitivity.

3 Method Description

In our method a **sequence** is a juxtaposition of letters. A sequence may be either DNA or protein. A **genome** is a set of sequences. We restrict a genome to be a set of sequences of the same kind. We denote by $G(p)$ the genome to which sequence p belongs. Our method lays on two main concepts: edge and triangle.

- Given two genomes P and Q, an **edge** is a pair of sequences (p, q), $p \in P$ and $q \in Q$. The **weight** of an edge, $w(p, q)$, is any measure of similarity that can be evaluated given p and q.
- Given three genomes P, Q and R, a **triangle** is a triplet of sequences (p, q, r), $p \in P$, $q \in Q$ and $r \in R$, such that there are edges (p, q), (p, r) and (q, r).

```
Method 3GC
Input: genomes $P$, $Q$ and $R$
Output: sequences in $P$, $Q$ and $R$ attributed to Venn-Euler diagram regions

Evaluate $w(a, b)$ for every pair of sequences such that $G(a) \neq G(b)$.
For every triangle $(p, q, r)$, $p \in P$, $q \in Q$ and $r \in R$, such that $w(p, q, r) \geq T_t$ do
    Add $(p, q, r)$ to list L;
Sort L non-increasingly on the weights of the triangles.
while $L \neq \emptyset$ do
    Take the first triangle from L, and call it $t = (p, q, r)$.
    Add $t$ to the proper region in the diagram.
    Remove any triangle in $L$ that has $p$, $q$ or $r$ as a member.
    Remove $p$, $q$ and $r$ from $P$, $Q$ and $R$, respectively.
For every edge $(p, q)$, $p \in P$ and $q \in Q$, such that $w(p, q) \geq T_e$ do
    Add $(p, q)$ to list L;
For every edge $(p, r)$, $p \in P$ and $r \in R$, such that $w(p, r) \geq T_e$ do
    Add $(p, r)$ to list L;
For every edge $(q, r)$, $q \in Q$ and $r \in R$, such that $w(q, r) \geq T_e$ do
    Add $(q, r)$ to list L;
Sort L non-increasingly on the weights of the edges.
while $L \neq \emptyset$ do
    Take the first edge from L, and call it $e = (a, b)$.
    Add $e$ to the proper region in the diagram.
    Remove any edge in $L$ that has $a$ or $b$ as a member.
    Remove $a$ and $b$ from $G(a)$ and $G(b)$, respectively.
for every sequence $s$ in $P$ do
    add $s$ to the region corresponding exclusively to $P$ in the diagram;
for every sequence $s$ in $Q$ do
    add $s$ to the region corresponding exclusively to $Q$ in the diagram;
for every sequence $s$ in $R$ do
    add $s$ to the region corresponding exclusively to $R$ in the diagram;
```

Fig. 4. The method description

The weight of a triangle, $w(p, q, r)$, is any measure that can be evaluated from $w(p, q)$, $w(p, r)$ and $w(q, r)$.

Our method evaluates the weight of every pair of sequences from distinct genomes. Then the triangles are processed in non-increasing order of weights. One by one the triangles are assigned to the common region of the diagram, until no triangles with weight greater or equal to T_t are left. When a triangle (p, q, r) is included in the diagram, no more triangles or edges involving p, q or r are taken into consideration. After that, the edges are processed in non-increasing order of weights. One by one the edges are assigned to the proper region of the diagram, until no edges with weight greater or equal to T_e are left. The general description of the method appears in Figure 4.

Our method has been described taking into account that the measure used in weight evaluation is based on sequence similarity [13]. By making only minor changes, it can also be described with other kinds of measures, like Blast expectation [2], or edit distance [13]. Proper changes on the measures and thresholds will lead to a family of algorithms that can be applied for comparing three genomes at the same time.

The running time for an algorithm that follows the steps in our method is loosely $O(|P||Q|\alpha + |P||R|\alpha + |Q||R|\alpha + |P||Q||R|\beta)$, that corresponds to the number of sequence comparisons at cost α per comparison, plus the maximum number of triangles that can be generated among the genomes at cost β per triangle weight evaluation. Of course the number of triangles is going to be smaller in practice. Indeed, our experiments have shown that the processing time is affordable.

4 Experiments

We used our algorithm to compare the genomes of fungi *Aspergillus nidulans* (9541 sequences), *Candida albicans* (6165 sequences), *Criptococcus neoformans* (6578 sequences), *Fusarium graminearum* (11640 sequences), *Magnaporte grisea* (11109 sequences), *Neurospora crassa* (10082 sequences) and *Saccharomyces cereviseae* (6305 sequences). *C. neoformans* and *C. albicans* are human pathogens; the other fungi are not. We compared every non-pathogenic genomes with *C. neoformans* and *C. albicans*, giving rise to the five Venn-Euler diagrams that appear in Figure 5. Every gene located in the diagram regions and the input genomes are available at `http://egg.dct.ufms.br/3gc/`.

We set our algorithm as follows. There is an edge between two sequences if they form a Blast bidirectional hit. Given a sequence p from genome P and a sequence q from genome Q, we say that p and q are a **bidirectional hit** if

- q is found by the Blast search of p against Q with expectation less or equal to e^{-5}, and
- p is found by the Blast search of q against P with expectation less or equal to e^{-5}.

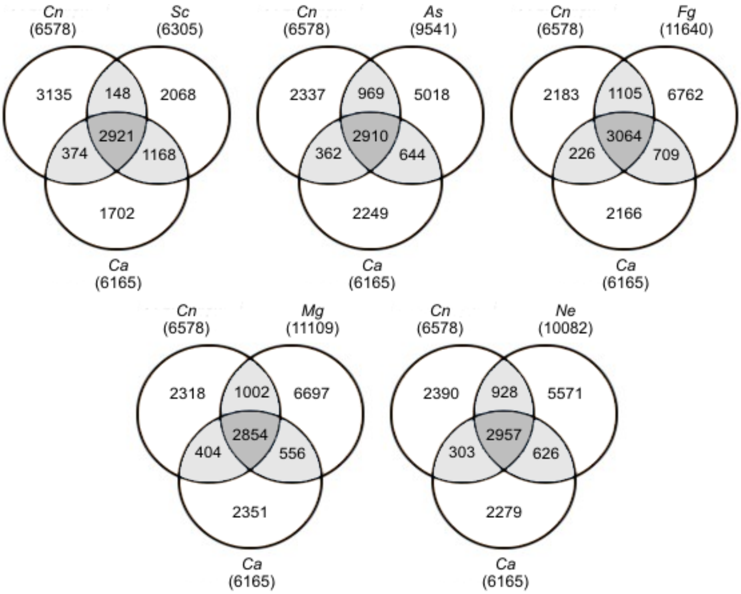


Fig. 5. Venn-Euler diagrams for two pathogenic and five non-pathogenic fungi, compared three at the same time

The weight of an edge (p, q) is given by the average percent coverage of the sequences by the alignments produced by the Blast searches. We believe that using coverage it is possible to avoid adding an edge between two sequences sharing only small subsequences, like small protein domains. Triangles are weighted by the average of the weights of its edges.

In order to set T_t, we conduced the following experiment. We executed the algorithm for $1 \leq T_t \leq 100$, in steps of 1. For each value of T_t, we recorded the number of triangles with three sequences belonging to the same Pfam family [3] and the number of triangles with two sequences belonging to the same Pfam family. Pfam was used since it is based on Hidden Markov Models that detect family/domain features in a given protein sequence. Pfam models were considered assuming that orthologous sequences in a given triangle would share the same Pfam family. Triangles sharing two or three sequences with the same Pfam result were counted and plotted considering only steps of 10 (Figure 6). This curve suggests a reduced effect of coverage cutoff on the efficiency of detecting sequences with similar Pfam result.

In Figure 7 we show the number of resulting triangles with increasing coverage cutoff. As expected, there was a marked decrease of triangles as the stringency increases, falling close to zero when sequence pairs must share 100% coverage. Although the existence of such sharp drop, the number of triangles are rather constant for coverages below 50%, a comprehensive coverage for interspecies orthologous detection. It is note worth that the method is rather insensitive to a low coverage cutoff. It is possibly a feature of the algorithm in use, which

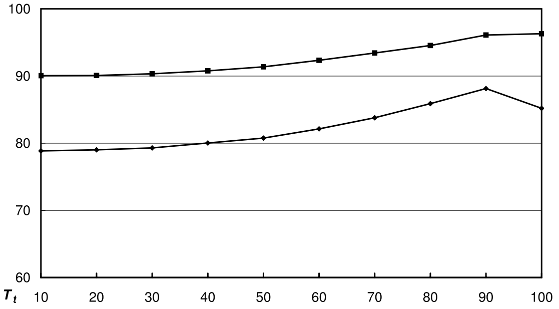


Fig. 6. Percentages of triangles with at least two coincident (upper curve) and percentages of triangles with three coincident Pfam families (lower curve), for T_t varying from 10 to 100

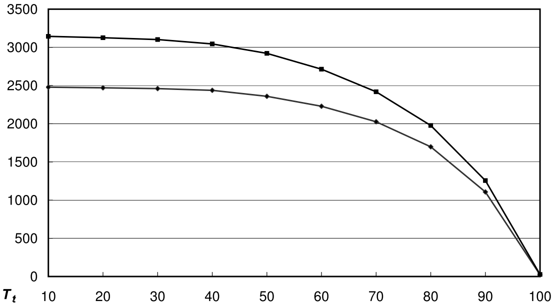


Fig. 7. Number of triangles with increasing coverage cutoff. The upper curve represents the total number of triangles for a given cuttoff. The lower curve represents the number of triangles having three sequences matching with the same Pfam family

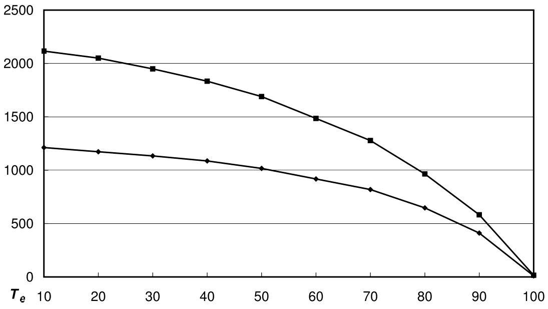


Fig. 8. Number of edges with increasing coverage cutoff. The upper curve represents the total number of edges for a given cuttoff. The lower curve represents the number of edges having two sequences matching with the same Pfam family

eliminates the high coverage triangles progressively reducing the search space at each step. This approach apparently prevents low similarity hits from getting into triangles, improving the number of correctly detected orthologous among tested species. We selected $T_t = 50$.

In order to set T_e we conduced a similar test. We executed the algorithm for $1 \leq T_e \leq 100$, in steps of 1, with $T_t = 50$. For each value of T_e, we recorded the number of edges with two sequences belonging to the same Pfam family. The results are plotted considering steps of 10 in Figure 8. The percentage of edges in the same Pfam family had similar behavior of triangles. So, we selected $T_e = 50$.

It took 35 minutes on the average to run the program for every fungal comparison in a Pentium 4 at 2 GHz, 1 GB RAM and an ultra ATA/100 40 GB hard disk. Running Blast took less than 22 hours in the same machine. Our program was written in Perl.

5 Biological Inference Based on Our Experiments

We selected seven complete genomes to test the 3-genome comparison algorithm. In order to focus on genes related to human pathogenesis, we compared *C. albicans* and *C. neoformans* to any of the other five non-pathogenic species. The numbers resulting from this analysis are shown in Figure 5. There is a striking constant number of triangles among comparison (2941 ± 79). This set may reflect shared genes involved in the informational and central metabolism, a set of indispensable genes for fungi.

We conduced a biological analysis with three fungi genomes among the analyzed genome set, chosen because they are biological representative systems. *S. cereviseae* was the first determined eukaryote genome and now is object of intensive annotation validation by many groups around the world. This non-pathogenic fungus is standard for genome comparison due to the large availability of biochemical and genetic data. *C.albicans* is a human opportunistic pathogen that had its genome recently dissected. Its genome had been intensively annotated and, likewise *S. cereviseae*, is a well studied fungi. These two genomes were compared to the newly described genome of *C. neoformans*, the etiological agent of an opportunistic mycosis. Cryptococcosis is an opportunistic fungal infection that most frequently causes pneumonia and/or meningitis. The comparison of *C. neoformans* genome to other fungi may reveal pathogenic related genes and biochemical peculiarities yielding to alternative therapeutic approaches.

The 3-genome comparison resulted in the first Venn-Euler diagram shown on Figure 5. The genomes harbor an equivalent number of genes (*S. cereviseae*: 6305 genes; *C. albicans*: 6165 genes; *C. neoformans*: 6578 genes). There are 2921 triangles corresponding to orthologous genes among genomes. According to this, roughly half of the genes in the three genomes could be assigned to triangles. Edges representing genes in two but absent in the third genome is small (less then 6%) except for those presented in *S. cereviseae* and *C. albicans* and absented in *C. neoformans*. This number is followed by a 48% of exclusive genes in the *C. neoformans*. This region represents genes with no orthologous genes on the other compared genomes. This result suggests that *S. cereviseae* and *C. albicans* share more orthologous than any of them with *C. neoformans*.

Histones are conserved proteins involved in the assembly of DNA into chromatin. Histones are coded by a set of conserved genes that can be found in any eukaryote. They are classified by histone H1, H2a, H2b, H3 and H4, where H1 performs an associative role to keep together the nucleosomes particles. The triangles revealed the presence of 8 orthologous among *S. cereviseae*, *C. albicans* and *C. neoformans*. There were three copies of H3 and two copies of H4; H2a and H2b appears as single copies. There is also the regulatory histone variant H2A F/Z found on the three analyzed genomes. Interestingly, we did not find a triangle for the Histone H1. We only found a single edge between *S. cereviseae*, and *C. neoformans*, suggesting the absence of this histone in *C. albicans*. Indeed there is no evidence for a H1 homologous in *C. albicans* in the literature, and it was shown that for other fungi, it was dispensable [12]. So, this analysis could easily reveal the hole set of orthologous histones among these three genomes.

Fungal pathogenesis depends on a series of genes to yield fungi cell growth inside host cell and to evade immune system. The glyoxilate cycle help fungi survive inside macrophage, an immunity cell that phagocyte foreign bodiesLBF04. The invading cell must be able to survive inside macrophages to succeed in infecting. It had been shown that the activation of the glyoxilate cycle is fundamental for *S. cereviseae* and *C. albicans* survival in this condition. The enzymes isocitrato lyase and malate synthase together with malate dehydrogenase, citrate synthase and aconitase complete the cycle. All these enzymes were found on triangles among the three genomes and support the existence of the cycle in the poorly characterized *C. neoformans* genome. This fact suggests new targets for *C. neoforms* drug development, based in the absence of this metabolic pathway in animal (host) cell.

6 Concluding Remarks

The need for computational tools to analyze and compare the large amount of data from complete genomes is posed. In this work, we proposed a method for comparing three genomes simultaneously, at the basic level of their sequences. The input for the method is a set of gene coding sequences or products and the output is sequences exclusive to each genome, sequences common to pairs of genomes, and sequences common to the three genomes. We used our method to compare genomes of two pathogenic (*Candida albicans* and *Criptococcus neoformans*) and five non-pathogenic fungi (*Aspergillus nidulans*, *Fusarium graminearum*, *Magnaporte grisea*, *Neurospora crassa* and *Saccharomyces cereviseae*). Finally, we chose the results from (*C. albicans*, *C. neoformans* and *S. cereviseae*) to make a biological analysis.

We believe that our method performed well in affordable time, since considering the percentage of triangles with at least two coincident Pfam families, more than 90% of all triangles had that condition, and this ratio increased to more than 95% in a high coverage match cutoff. This data showed that the most part of the triangles had coincident Pfam results, suggesting that they are indeed structurally/functionally related, a *sine qua non* condition for assigning orthology to a pair of related sequences.

Other possibilities for comparing three genomes include a previous clustering step in each genome. Like our method does, this also could help to avoid those complicated cases involving paralogous genes, reported in Section 2. By analyzing intersections of interest across diagrams, keeping a pair of genomes fixed, highly conserved pathways in fungi could be revealed. Analyzing specific genes in each genome could reveal particular pathways of secondary metabolism. Our method is general in the sense that one can use any measures and/or thresholds, so its use in conjunction to other methods found on the literature could lead to new interesting biological findings.

References

1. N.F. Almeida. *Tools for genome comparison.* PhD thesis, IC-Unicamp, Campinas-SP, Brazil, 2002. in Portuguese.
2. S.F. Altschul, T.L. Madden, A.A. Schäffer, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.
3. A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S.R. Eddy, S. Griffiths-Jones, K.L. Howe, M. Marshall, and E. L. L. Sonnhammer. The Pfam protein families database. *Nucleic Acids Research*, 30(1):276–280, 2002.
4. B. Birren and Initiative Fungal Genome. A white paper for fungal comparative genomics, october 2003. Whitehead Institute MIT Center for Genome.
5. E.L. Braun, A.L. Halpern, M.A. Nelson, and D.O. Natvig. Large-scale comparison of fungal sequence information: mechanisms of innovation in Neurospora crassa and gene loss in Saccharomyces cerevisiae. *Genome Research*, 10:416–4300, 2000.
6. A.L. Delcher, S. Kasif, R.D. Fleischmann, O. White J. Peterson, and S.L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27(11):2369–2376, 1999.
7. M. Kellis, N. Patterson, B. Birren, B. Berger, and E.S. Lander. Methods in comparative genomics: genome correspondence, gene identification and motif discovery. *Journal of Computational Biology*, in press. Special issue dedicated to Proceedings of the 7th International Conference on Research on Computational Biology.
8. G. Kurapakt and C. Sutter-Crazzorala. International Plant and Animal Genome IX Conference, 2001. http://www.intl-pag.org/9/abstracts/W15_03.html.
9. lagan. http://lagan.stanford.edu/lagan_web/index.shtml.
10. M. Li, J.H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154, 2001.
11. Y. Liu, X.S. Liu, L. Wei, R.B. Altman, and S. Batxoglou. Eukariotic regulatory element conservation analysis and identification using comparative genomics. *Genome Research*, 14:451–458, 2004.
12. A. Ramon, M.I. Muro-Pastor, C. Scazzochio, and R. Gonzalez. Deletion of the unique gene encoding a typical Histone H1 has no apparent phenotype in Aspergillus nidulans. *Molecular Microbiology*, 35:223–233, 2000.
13. J.C. Setubal and J. Meidanis. *Introduction to computational molecular biology.* PWS Publishing Co., 1997.
14. srs. http://www.lionbioscience.com.
15. vista. http://genome.lbl.gov/vista/index.shtml.

Comparison of Genomic DNA to cDNA Alignment Methods

Miguel Galves[1,2] and Zanoni Dias[1,2]

[1] Instituto de Computação - Unicamp, Campinas - SP - Brasil
{miguel.galves, zanoni}@ic.unicamp.br
[2] Scylla Bioinformática, Campinas - SP - Brasil
{miguel, zanoni}@scylla.com.br

Abstract. Aligning cDNA sequences to genomic sequences is a very common way to study expressed sequences, find their genes, and study alternative splicing. Several computer programs address this problem, using heuristics to define exon regions. Usually, standard alignment algorithms are not used to align ESTs to genomic DNA, due to the existence of large regions of introns. This paper compares the EST-to-genomic alignments produced by *sim4*, *est_genome*, *Spidey* and standard sequence aligners using an appropriate score. Surprisingly, standard aligners performed quite well with sequences having few errors.

1 Introduction

Identifying genes in non-characterized DNA sequences is one of the great challenges in genomics. One of the most common methods for this task is aligning expressed sequence tags (EST) to genomic sequences.

ESTs are key to understanding the inner working of an organism. However, in order to fully understand the function of an expressed sequence, it must be put in its genomic context. Estimates show that the human being has a number of genes between 30000 and 35000 [1], and therefore alternative splicing may be an important factor in generating transcriptional diversity, so EST-to-genomic alignment will be crucial to our understanding of the genome. Generic sequence alignment algorithms are not usually used to perform EST-to-genomic alignment, mostly because of the high amount of introns that may occur in the genomic sequence. The main goal of this paper is to compare alignments between genomic DNA and cDNA produced by a conventional aligner, using a custom set of scores, with the results produced by publicly available software (*sim4*, *est_genome* and *Spidey*) that use heuristics to find exon boundaries.

In Section 2 we describe the classic algorithms to align any given pair of sequences. In Section 3 we describe the aligner used in this paper. In Section 4, we discuss the strategies adopted by *sim4*, *est_genome* and *Spidey* to find good EST-to-DNA alignments. In Section 5 we describe the data set used as testbench for the paper. In Section 6 we describe the test methodology, whose results are further analyzed in Section 7. Finally, in Section 8 we conclude and propose possible further work.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 170–180, 2005.

2 Classic Algorithms to Align a Pair of Sequences

An alignment of a pair of sequences is defined as an operation in which gaps are inserted in both sequences in order to make them have the same length, allowing comparison between bases [8]. For a given alignment, it is possible to define a score that measures the quality of the obtained result. The simplest score system consists of a penalty given to base aligned to a gap, (gap penalty), points given to alignment of different bases (mismatch) and points given to alignment of identical bases (match).

This type of score system does not differentiate one-base gaps from contiguous (multi-base) gaps. However, it is known that k-length gaps are much more common that k one-base gaps [8]. Therefore a strategy had to be developed, in which one-base gaps suffer a bigger penalty than contiguous gaps. With that, it is expected that most gaps are joined together. To accomplish this, the gap penalty was replaced by an affine function $w(k) = g + hk$, where k it is number of contiguous spaces, g it is the cost opening a new gap, and h is the cost of extending an open gap.

Throughout this paper, we will always deal with 4 parameters: g, which will be called *opengap*, h, or *extendgap*, *match* and *mismatch*. The goal of alignment algorithms is to achieve optimum alignment, i.e, the one that receives the highest possible score. Global alignment intends to achieve the best possible alignment for two sequences. Spaces may be inserted at any position of the sequence, in order to get optimal score. Semi-global alignment intends to group the highest amount of spaces at the beginning and at the end of the alignment, at no penalty cost, with the sole goal of getting the best alignment between a prefix of one sequence prefix and a suffix of the other sequence or between one sequence and a subsequence of the other sequence.

3 Global and Semi-global Aligners

For this paper, we developed global and semi-global aligners with affine score systems. The implemented aligners use linear space, considering that space is crucial to aligning long sequences. The global aligner is an implementation of the algorithm for global alignment proposed by Miller and Myers [4]. The semi-global aligner is based on the global algorithm.

Both aligners were implemented using the Java programming language. Comparative tests between the implemented aligners and **fasta** [6] package aligners were made to validate the implementation.

4 ESTs to Genomic DNA Alignment Packages

One of the main goals of aligning ESTs to genomic DNA is to detect exon (coding sequences) and intron (non-coding sequences) boundaries in the latter. Some heuristic-based algorithms have been developed to address this problem.

In this paper we will compare our aligner's results with those produced by three of theses softwares: *est_genome* [3], *sim4* [2] and *Spidey* [10].

5 Test Dataset

In order to compare the aligners, a large human genome dataset was built, containing relevant data about chromosomes, genes, mRNAs and CDSs. The data used to build such dataset was extracted from FASTA files and GENBANK flat files. Detailed description of how the files are organized can be found on the NCBI website [5]. All files used were obtained at the NCBI's FTP repository and were made available in the repository on October 9, 2004. To achieve consistency, it was necessary to filter the dataset.

5.1 Data Filtering Criteria

An mRNA is a single stranded RNA molecule that specifies the amino acid sequence of one or more polypeptide chains. This information is translated during protein synthesis when ribosomes bind to the mRNA. CDS (Coding sequence) is the region between mRNA's start and stop codon that effectively is translated by the ribosome and code protein sequence. UTR (untranslated sequence) are sections of the RNA before the start codon and after the stop codon that are not translated. These come from the template DNA strand that the RNA was transcribed from. These regions, known as the 5'UTR and 3'UTR, code for no

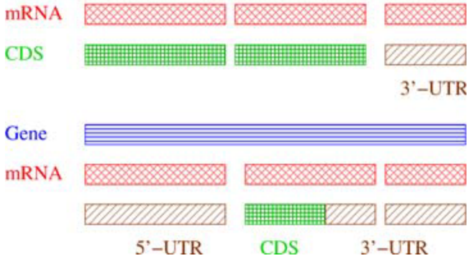


Fig. 1. Accepted mRNA and CDS patterns: mRNA composed by a number of CDSs without UTR; mRNA composed by two CDSs and one region of 3'-UTR ; mRNA composed by one CDSs, one region of 5'-UTR and two segments of one region of 3'-UTR

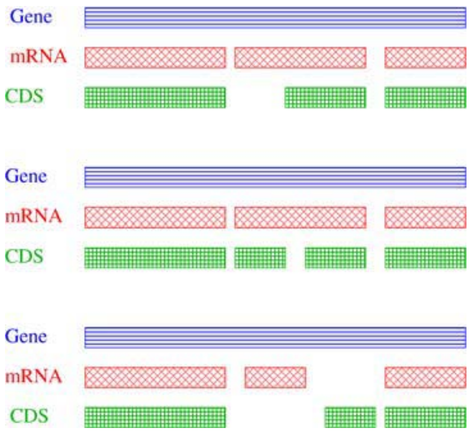


Fig. 2. Discarded mRNA and CDS patterns: an inner region of the mRNA is not covered by the CDS; one of the inner regions of the mRNA is partially covered by two regions of CDS; mRNA regions not present in the CDS and vice-versa

protein sequences. An mRNA must be fully covered by CDSs and UTRs and UTRs and CDSs must not have regions outside the mRNA. Examples of the accepted mRNAs and CDSs patterns in the database are showed in Fig. 1.

The following elements were removed from the database:

- Genes, mRNAs and CDSs with a */pseudo* tag.
- CDSs matching the patterns shown in Fig. 2.
- mRNAs without any CDS.
- Genes without any mRNA.

5.2 Results After Filtering

From all the genes extracted from the files, 6.72% were removed for being incomplete (without mRNAs), and 6.78% were removed for being pseudo-genes. Only 0.06% of the mRNAs were removed for being pseudo-mRNAs. From all the CDSs, 0.17% were removed for being pseudo-CDSs and 1.7% were removed for not matching the patterns mentioned earlier.

By the end of the filtering process, 23124 genes (86.5% of the genes) and 27448 mRNAs were stored in the database. Among the mRNAs inserted in the database, 9.48% do not contain any UTR, 4.74% contain 5'-UTR only, 4.33% contain 3-'UTR only and 81.45% contain both 3'-UTR and 5'-UTR.

6 Test Methodology

To evaluate the many score systems considered, we defined gene subsets, in order to perform a large number of evaluation experiments. The subsets defined are:

Subset 1. Data set composed by 66 mRNAs extracted from 66 genes from chromosome Y with less than 100000 bases.

Subset 2. Data set composed by 50 mRNAs extracted from 40 complete genes (without N symbols) from chromosome Y with less than 100000 bases.

Subset 3. Data set composed by 8056 mRNAs extracted from 7376 complete genes from the whole human genome database with less than 10000 bases.

Subset 4. Data set composed by modified subsequences extracted from 493 complete genes from chromosome 6 with less than 10000 bases. For each selected gene, 10 subsequences were randomly extracted, with size ranging from 200 to 1000 bases: in 5 of them we introduced errors (insertion, deletions and base substitution) at a rate of 1%, and in the other 5, at a rate of 10%. The errors were introduced using a random number generator. The final subset is composed by 4930 simulated ESTs to be aligned with 493 genes. The values of 1% and 10% are justified by the fact that the error rate of base-calling programs can be as high as 3% [9].

To determine the more appropriate alignment strategy (global or semi-global) and score system to produce good EST-to-genomic alignments using a standard aligner, we started producing alignments with data sets 1 and 2. For each gene and mRNA, alignment algorithm and score system, we produced 4 types of alignments: mRNA (CDS+UTR) to gene, CDS to gene, mRNA to gene+200 bases (100 bases appended to each extremity) and CDS to gene+200 bases.

Once the alignment strategy and score system were defined, we performed the 4 alignments defined above using the data from sets 3 and 4. Besides, we used the external programs *sim4* [2], *Spidey* [10] and *est_genome* [3], with their default configuration, to align all the data from the 4 data sets.

6.1 Evaluation Methods of the Obtained Alignments

The following metrics were defined to compare the alignments produced by our simple aligner and those produced by the external programs.

Gaps introduced in the aligned gene sequence. Simply counts the number of gaps inserted by the aligner in the genomic sequence. Using mRNAs and CDS without errors, the expected value for all alignments is 0. All values must be positive.

Delta exons. Subtracts the number of exons created by the aligner from the number of exons of the original mRNA. Positive values mean that the aligner created less exons than expected. Ideally, the value must be 0 for all produced alignments. It is important to notice that the generic aligner does not define exon/intron boundaries. However, we consider a contiguous region of bases in the aligned mRNA or EST as being exon regions.

Bases similarity percentage. Calculates the similarity percentage between the expected gapped cDNA and the gapped cDNA produced by the aligners. This is done comparing each base from the predicted gapped cDNA to the base in the same position in the produced gapped cDNA.

Mismatch percentage. Counts the number of mismatches created by the aligners and divides it by the size of the cDNA. In exact alignments (data sets 1, 2 and 3), the expected value is 0 for all produced alignments.

7 Results and Comparative Analysis

The comparative analysis of the alignments produced by a conventional aligner and those produced by carefully crafted programs, designed to produce EST-to-genomics alignments, was divided in two stages: first, the tests were designed to identify the most appropriate algorithm (global or semi-global) and the most appropriate score system to produce good EST-to-genomics alignments. Second, we ran tests to compare our aligner with *sim4*, *Spidey* and *est_genome*, using both exact sequences and sequences with introduced errors.

7.1 Alignment Algorithm and Score System Definition

Using the aforementioned methodology, we tested 15 different score systems, using data sets 1 and 2. The score system $match = 1$, $mismatch = -2$, $opengap = -1$,

Table 1. Minimum, maximum, average and standard deviation for base similarity percentage of the exons produced by *sim4*, *est_genome*, *Spidey*, global and semi-global aligner using dataset 3, and percentage of alignments with the expected base similarity score of 100%

Base similarity percentage (Dataset 3)

Aligner	Alignment type	Min	Max	Avg	σ	% Score 100%
Semi-global	Gene x mRNA	80.85%	100.00%	99.89%	0.49%	53.56%
Semi-global	Gene+200 x mRNA	80.31%	100.00%	99.83%	0.63%	53.56%
Semi-global	Gene x CDS	80.31%	100.00%	99.83%	0.63%	59.35%
Semi-global	Gene+200 x CDS	80.85%	100.00%	99.89%	0.49%	59.35%
Global	Gene x mRNA	80.70%	100.00%	99.85%	0.52%	53.62%
Global	Gene x CDS	80.31%	100.00%	99.78%	0.63%	46.97%
Global	Gene+200 x mRNA	80.31%	100.00%	99.76%	0.67%	38.87%
Global	Gene+200 x CDS	90.01%	100.00%	99.90%	0.28%	43.61%
sim4	Gene x mRNA	36.00%	100.00%	99.39%	1.34%	22.72%
sim4	Gene x CDS	10.29%	100.00%	99.08%	2.29%	33.19%
sim4	Gene+200 x mRNA	36.00%	100.00%	99.39%	1.34%	22.72%
sim4	Gene+200 x CDS	10.29%	100.00%	99.08%	2.29%	33.19%
est_genome	Gene x mRNA	1.80%	100.00%	53.83%	35.09%	18.11%
est_genome	Gene x CDS	2.68%	100.00%	62.45%	35.09%	31.05%
est_genome	Gene+200 x mRNA	1.80%	100.00%	53.80%	35.10%	18.11%
est_genome	Gene+200 x CDS	2.68%	100.00%	62.44%	35.09%	31.05%
Spidey	Gene x mRNA	0.00%	100.00%	80.34%	36.49%	44.25%
Spidey	Gene x CDS	0.00%	100.00%	81.47%	37.06%	50.92%
Spidey	Gene+200 x mRNA	0.00%	100.00%	80.19%	36.75%	44.19%
Spidey	Gene+200 x CDS	0.00%	100.00%	81.53%	37.02%	50.93%

Table 2. Minimum, maximum, average and standard deviation for delta exons produced by *sim4*, *est_genome*, *Spidey* and semi-global aligner using dataset 3, and percentage of alignments with the expected delta exons score of 0

Delta exons (Data set 3)

Aligner	**Alignment type**	**Min**	**Max**	**Avg**	σ	**% Score 0**
Semi-global	Gene x mRNA	0	0	0.00	0.00	100.00%
Semi-global	Gene x CDS	0	1	0.00	0.03	99.91%
Semi-global	Gene+200 x mRNA	0	0	0.00	0.00	100.00%
Semi-global	Gene+200 x CDS	0	1	0.00	0.03	99.91%
Global	Gene x mRNA	-2	0	-0.27	0.45	99.55%
Global	Gene x CDS	-2	1	-0.22	0.42	78.15%
Global	Gene+200 x mRNA	-2	1	-0.27	0.45	72.80%
Global	Gene+200 x CDS	-1	0	0	0.07	73.01%
sim4	Gene x mRNA	-3	8	-0.01	0.23	97.46%
sim4	Gene x CDS	-3	13	-0.05	0.33	94.02%
sim4	Gene+200 x mRNA	-3	6	-0.01	0.22	97.44%
sim4	Gene+200 x CDS	-3	13	-0.05	0.33	94.00%
est_genome	Gene x mRNA	-4	0	-0.14	0.38	76.79%
est_genome	Gene x CDS	-4	0	-0.21	0.48	80.24%
est_genome	Gene+200 x mRNA	-4	0	-0.14	0.38	76.85%
est_genome	Gene+200 x CDS	-4	0	-0.21	0.48	80.24%
Spidey	Gene x mRNA	-27	-1	-4.04	3.13	0.00%
Spidey	Gene x CDS	-27	-1	-3.60	3.05	0.00%
Spidey	Gene+200 x mRNA	-27	-1	-4.04	3.13	0.00%
Spidey	Gene+200 x CDS	-27	-1	-3.60	3.05	0.00%

$extendgap = 0$ was the first to produce consistent results for both global and semi-global aligners. Table 1 shows that the average similarity percentage is greater than to 99.8% and standard deviation less than 1%.

We can see that the results are very similar for the global and semi-global aligners. Mainly, two factors determined the choice of the semi-global aligner: the fact that in general, more alignments have 100% of similarity with the expected gapped mRNA, as shown in Table 1, and the fact that the semi-global aligner got slightly better results with the delta exons evaluation, as showed in Table 2. Indeed, one expects the semi-global aligner to be more appropriate for EST-to-genomics alignments, since it ignores the beginning and ending spaces.

7.2 mRNA-to-Genomic and CDS-to-Genomic Alignments Analysis

Analyzing the results in Table 1, Table 2 and Table 3, we can see that the semi-global aligner with the chosen score system ($match = 1$, $mismatch = -2$, $opengap = -1$, $extendgap = 0$), in comparison to the other tools, produced quite good alignments. Semi-global alignments got base similarity percentages closer to 100%, are as good as alignments produced by *sim4*, and much better than those produced by *Spidey* and *est_genome*. Moreover, in more than 99.9% of the semi-global alignments results, the number of produced exons was equal

Table 3. Minimum, maximum, average and standard deviation for extra gaps in genomic DNA produced by *sim4*, *est_genome*, *Spidey* and semi-global aligner using dataset 3, and percentage of alignments with the expected extra gaps score of 0

Extra gaps (Data set 3)

Aligner	**Alignment**	**Min**	**Max**	**Avg**	σ	**% Score 0**
Semi-global	Gene x mRNA	0	0	0	0	100.00%
Semi-global	Gene x CDS	0	1	0	0.03	99.91%
Semi-global	Gene+200 x mRNA	0	0	0	0	100.00%
Semi-global	Gene+200 x CDS	0	1	0	0.03	99.91%
sim4	Gene x mRNA	0	14	1.11	1.69	54.56%
sim4	Gene x CDS	0	16	0.96	1.63	60.85%
sim4	Gene+200 x mRNA	0	14	1.11	1.69	54.51%
sim4	Gene+200 x CDS	0	16	0.96	1.64	60.82%
est_genome	Gene x mRNA	0	201	16.99	21.49	27.84%
est_genome	Gene x CDS	0	201	14.13	21.12	41.66%
est_genome	Gene+200 x mRNA	0	201	17.00	21.49	27.84%
est_genome	Gene+200 x CDS	0	201	14.13	21.12	41.66%
Spidey	Gene x mRNA	0	36	0.15	1.39	97.43%
Spidey	Gene x CDS	0	23	0.10	1.03	98.01%
Spidey	Gene+200 x mRNA	0	36	0.15	1.34	97.18%
Spidey	Gene+200 x CDS	0	23	0.10	1.02	98.03%

Table 4. Minimum, maximum, average and standard deviation for mismatch percentage in genomic DNA produced by *sim4*, *est_genome*, *Spidey* and semi-global aligner using dataset 3, and percentage of alignments with the expected mismatch percentage score of 0%

Mismatch percentage (Data set 3)

Aligner	**Alignment type**	**Min**	**Max**	**Avg**	σ	**% Score 0%**
Semi-global	Gene x mRNA	0.00%	0.00%	0.00%	0.00%	100.00%
Semi-global	Gene x CDS	0.00%	0.00%	0.00%	0.00%	100.00%
Semi-global	Gene+200 x mRNA	0.00%	0.00%	0.00%	0.00%	100.00%
Semi-global	Gene+200 x CDS	0.00%	0.00%	0.00%	0.00%	100.00%
sim4	Gene x mRNA	0.00%	2.15%	0.17%	0.21%	36.68%
sim4	Gene x CDS	0.00%	3.23%	0.24%	0.33%	45.47%
sim4	Gene+200 x mRNA	0.00%	2.15%	0.17%	0.21%	36.67%
sim4	Gene+200 x CDS	0.00%	3.23%	0.24%	0.33%	45.47%
est_genome	Gene x mRNA	0.00%	7.25%	1.19%	1.26%	21.55%
est_genome	Gene x CDS	0.00%	13.36%	1.48%	1.70%	35.18%
est_genome	Gene+200 x mRNA	0.00%	7.25%	1.19%	1.26%	21.56%
est_genome	Gene+200 x CDS	0.00%	13.36%	1.48%	1.70%	35.18%
Spidey	Gene x mRNA	0.00%	23.69%	0.15%	0.98%	90.65%
Spidey	Gene x CDS	0.00%	20.56%	0.20%	1.28%	89.67%
Spidey	Gene+200 x mRNA	0.00%	28.68%	0.20%	1.38%	90.75%
Spidey	Gene+200 x CDS	0.00%	20.06%	0.21%	1.30%	89.59%

to the expected number of exons in the original mRNA. The software *sim4* again produced the best results among the third-party software packages with at most 97% of the produced alignments having the correct number of exons. *Spidey* did not produce any correct number of exons (it always produced more exons than expected).

In almost all of the semi-globals alignments, no gaps were inserted in the genomic sequence, which is expected considering that all the CDSs and UTRs were extracted from the gene. *Spidey* and *sim4* produced good results, with the average number of extra gaps ranging from 0.1 to 1.1 (Table 3). Finally, our aligner did not create any mismatch (Table 4).

7.3 EST-to-Genomic Alignments Analysis

Analyzing the results produced by the four aligners with artificial ESTs, we can see that the results are very similar. In Table 5, we can see again that the results from our aligner and those from *sim4* are pretty much equivalent. In this evaluation, *est_genome* shows the worst results, with an average similarity of 27% for ESTs with error rate of 1% and 17% for ESTs with error rate of 10%.

Table 5. Minimum, maximum, average and standard deviation for similarity percentage from exons produced by *sim4*, *est_genome*, *Spidey* and semi-global aligner using dataset 4, and percentage of alignments with the expected similarity percentage score of 100%

Base similarity percentage (Data set 4)

Aligner	Error rate	Min	Max	Avg	σ	% Score 100%
Semi-global	1%	4.04%	100.00%	54.14%	30.42%	0.45%
Semi-global	10%	2.99%	95.88%	48.20%	27.64%	0.00%
sim4	1%	4.04%	100.00%	53.98%	30.33%	0.41%
sim4	10%	3.55%	95.88%	50.41%	28.54%	0.00%
est_genome	1%	1.36%	100.00%	27.78%	22.11%	1.54%
est_genome	10%	1.24%	67.23%	17.77%	12.18%	0.00%
Spidey	1%	0.00%	100.00%	47.43%	32.61%	0.49%
Spidey	10%	0.00%	95.28%	41.47%	29.67%	0.00%

Besides, Table 6 shows that *sim4* created the smallest number of extra exons in average, compared to the expected number of exons, and in almost half of the alignments the number of generated exons was in accordance with expected results. In this delta exons evaluation, *est_genome* and *Spidey* did not produce the correct number of exons in any alignment.

7.4 Performance Comparison

All the tests were performed on a 1.7GHz Intel Pentium IV system with 512Mb of RAM running Fedora Core Linux 3. Table 7 shows comparative running times of each aligner used in this work.

Table 6. Minimum, maximum, average and standard deviation for delta exons produced by *sim4*, *est_genome*, *Spidey* and semi-global aligner using dataset 4, and percentage of alignments with the expected delta exons score of 0

Delta exons (Data set 4)

Aligner	**Error rate**	**Min**	**Max**	**Avg**	σ	**% Score 0**
Semi-global	1%	-15	16	-2.46	3.27	49.90%
Semi-global	10%	-94	2	-33.89	15.56	46.00%
sim4	1%	-17	2	-1.37	2.13	45.80%
sim4	10%	-16	2	-1.34	2.08	49.13%
est_genome	1%	-17	0	-1.48	2.17	0.00%
est_genome	10%	-16	0	-1.48	2.15	0.00%
Spidey	1%	-18	-1	-3.58	2.75	0.00%
Spidey	10%	-18	-1	-3.58	2.75	0.00%

Table 7. Comparison of aligner's running times, in seconds per alignment

Comparison of Running Times

	EST-to-DNA	**mRNA-to-DNA**
	(s/alignment)	**(s/alignment)**
sim4	0.013	0.017
Spidey	0.066	0.140
est_genome	0.640	3.400
Semi-global	0.670	5.170

Table 8. Average running time to align two sequences of the same size, using a Java sequence aligner and the fasta package implementation

Semi-global aligner implementation comparison

Sequence Size	**Java aligner**	**fasta align**
(bases)	**(s/alignment)**	**(s/alignment)**
1600	0.523	0.452
3200	2.055	1.203
6400	9.974	4.091

The packages *sim4* and *Spidey* are faster than our simple semi-global aligner, while *est_genome*'s running time is equivalent to ours. Still, it is worth of notice that the aligner used in this work is written in Java, while the others are written in C, which produces faster programs.

We made some running time comparisons with the fasta package [6], which performs semi-global alignments using linear space and is written in C: the results are shown in Table 8. We can see that the C implementation improves the running time.

8 Conclusion and Further Work

Based on the results shown above, it is possible to say that our semi-global aligner with score system ($match = 1$, $mismatch = -2$, $opengap = -1$, $extendgap = 0$) produces very acceptable results in cDNA-to-genomic alignments. Considering alignments with few or no errors, the results are very close to the results produced by third party software especially designed to address this kind of alignment. The best external aligner was *sim4*, both in tests with ESTs with errors and in alignments with error-free cDNA. It is worth of notice that our aligner produced alignment results as good as those produced by *sim4* with error-free data.

Further improvements to exon detection could be possible. One way to do that would be to first try to define regions more likely to be exon or intron regions, and then, to define different score systems for those regions. One possible criterion to define exon regions would be the to find high-GC percentage regions. Moreover, it would be interesting to perform tests with more realistic ESTs data. One possible solution would be to use SeqGen [7] to generate artificial data.

References

1. The Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.
2. L. Florea, G. Hartzell, Z. Zhang, G. Rubin, and W. Miller. A computer program for aligning cDNA sequence with genomic DNA sequence. *Genome Research*, 8:967–974, 1998.
3. R. Mott. EST_GENOME: A program to align spliced DNA sequences to unspliced genomic DNA. *Computer Applications in the Biosciences*, 13:477–478, 1997.
4. E. W. Myers and W. Miller. Optimal alignment in linear space. *Computer Applications in the Biosciences*, 4(1):11–17, 1988.
5. National Center for Biotecnology Information. http://www.ncbi.nlm.nih.gov/.
6. W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America*, 85(8):2444–2448, 1988. ftp://ftp.virginia.edu/pub/fasta/.
7. A. Rambaut and N. C. Grassly. Seq-gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Computer Applications in the Biosciences*, 13:235–238, 1997.
8. J. C. Setubal and J. Meidanis. *Introduction to Computional Molecular Biology.* PWS Publishing Company, 1997.
9. R. Sorek and H. M. Safer. A novel algorithm for computational identification of contaminated EST libraries. *Nucleic Acids Research*, 31(3):1067–1074, 2003.
10. S. J. Wheelan, D. M. Church, and J. M. Ostell. Spidey: A Tool for mRNA-to-Genomic Alignments. *Genome Research*, 11:1952–1957, 2001. Disponfvel em http://www.ncbi.nlm.nih.gov/spidey.

Segmentation and Centromere Locating Methods Applied to Fish Chromosomes Images

Elaine Ribeiro de Faria, Denise Guliato, and Jean Carlo de Sousa Santos

Faculdade de Computação,
Universidade Federal de Uberlândia,
Av. João Naves de Ávila, 2121, Uberlândia-MG
{elaine, guliato, jeancarlo}@lcc.ufu.br

Abstract. The objective of this paper is to describe a new approach for locating the centromere of each chromosome displayed in the digitalized photomicrography of fish cells. To detect the centromere position, the authors propose methods for both image segmentation and split touching chromosomes based on the fuzzy sets theory and a method for the rotation of chromosomes. These methods were applied to two species of fish chromosomes: *Astyanax scabripinnis* and *Astyanax eigenmanniorum.* Using a database with 40 images including metacentric, submetacentric and subtelocentric chromosomes, and comparing the centromere locating obtained by the proposed algorithm with the manual results obtained by two expert cytogeneticists, the average accuracies were 81.79% and 82.54% respectively.

1 Introduction

Chromosome analysis is an essential task for the detection of some diseases, abnormal cells and numerical variations. The process of visualization and classification of chromosomes is called karyotyping. The chromosomes are classified in four classes: metacentric, submetacentric, subtelocentric and acrocentric, depending on the centromere location. Then, they are paired in pre-defined classes, according to their similarity and displayed in a decreasing order.

The karyotype is used to detect abnormalities in cells which have structural defects and numerical variations. Many diseases can be foreseen through chromosome analysis, as an example, acute promyelocytic leukemia which can be detected analysing the extremity of the chromosome [1].

The process for obtaining the karyotype manually is very repetitive, time consuming and requires an experienced professional, making it an expensive procedure [2],[3]. After the acquisition of the digital chromosome image the manual process requires that the cytogeneticist cuts each chromosome into its individual parts and visually determine their centromere location. The centromere is a point at which the chromatides (identical arms) are joined together. After detecting the centromere, the lengths of chromosome arms are assessed. Then the centromeric index is calculated using the ratio between the longer arm (P) and shorter arm (Q). Using the centromeric index, chromosomes are classified in one

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 181–189, 2005.

of four possible classes, mentioned above. The chromosomes of each class are paired according to their similarity (for example: size or banding pattern) and the karyotype is displayed with chromosome pairs in decreasing order of size.

Various systems for the assembling of human karyotype are presented in literature. Pantaleão [1] proposed a system to recognize chromosomes from one unique image to analyze the acute promyelocytic leukemia disease. The proposed classification is based on chromosome area, perimeter and shape factors. The system had problems to rotate chromosomes properly for the displaying of the karyotype and did not treat the problems as centromere locating, overlapping chromosomes or as touching chromosomes. Popescu et al. [4] proposed a system that automatically processes cells that contain overlapping chromosomes. The main features of this system are the use of cross-section sequence graphs (CSSG) to segment overlapping chromosomes and the use of pale-path to separate touching chromosomes. The use of pale-path is a good solution in simple cases. In more complex situations, this solution fails, as there is no path that separates the chromosomes.

Agam and Dinstein [5] proposed a method for chromosome segmentation based on the shape of the contour. Only high concave points on external contours are considered for segmentation. Based on these points, lines are traced to separate the chromosomes and a series of hypotheses are checked to choose the best line. When a hypothesis is verified, an appropriate separation is performed. This process treats both touching and overlapping chromosomes. The system provides good results when there exist clusters with two chromosomes. Clusters with three or more chromosomes do not produce satisfactory results. The method presents problems in chromosome separation when there is a large contact area between bent chromosomes and when there are clusters containing small chromosomes.

Centromere locating is an important task for obtaining the karyotype automatically, however, this problem was not treated in the three previous works. Features as banding pattern, medial axis (MAT) and projection vector are commonly used for locating the centromere. Moradi et al. [6] proposed an approach for locating the centromere of human chromosome based on the horizontal projection vector. This work shows that the centromere locating is the narrowest part of the chromosome on its longitudinal direction. In order to calculate the horizontal projection vector from a binary image, the chromosome pixel values of each row are summed. The position of centromere is the point of global minimum in the central region of the horizontal projection vector. In the case of acrocentric chromosomes this method fails as there is no global minimum and the fact that the centromere is located on the extremity of the chromosome.

Cho [7] proposed centromere locating using medial axis transform (MAT). The profile shape is obtained for measuring the width along a transverse line, perpendicular to the tangent of the medial axis and centered at the unit distance along the medial axis. As the profile shape has peaks, only the first and the least peak are considered. The position of the centromere is the minimum value between the two peaks. The method could not be applied to acrocentric chromosomes. In this case, the centromeres were located manually.

The methods used to assembly the human karyotype cannot be used for the assembling of the fish karyotype due to fact that the fish chromosomes present some particularities such as: i) the number of fish chromosomes varies and the human chromosomes is always 46; ii) the fish chromosome skeleton is more ramified than human chromosome skeleton; iii) the image possess more noise, poorer contrast and the quality of the banding pattern is lower. These particularities make the centromere locating of fish cells a more difficult task.

This paper proposes a method for locating of the centromere of fish chromosomes for future classification and karyotype assembling procedures. The proposed method was applied to *Astyanax scabripinnis* and *Astyanax eigenmanniorum* fish species presenting good results.

2 The Outline of Proposed Centromere Locating Algorithm

For locating the centromere of fish chromosomes the digitalized fish photomicrography is given as input, a fuzzy-set-pre-processing method is executed to remove noise and increase contrast. After that, a segmentation method based on fuzzy set is applied to extract the chromosomes from the background. However, some chromosomes are not separated and continue touching one other. A procedure to separate touching chromosomes is therefore carried out. Then, for determining the position of the centromere, the individual chromosomes are first rotated to a vertical orientation. A schematic algorithm is shown in Figure 1. The following sections will present each phase of the algorithm in detail.

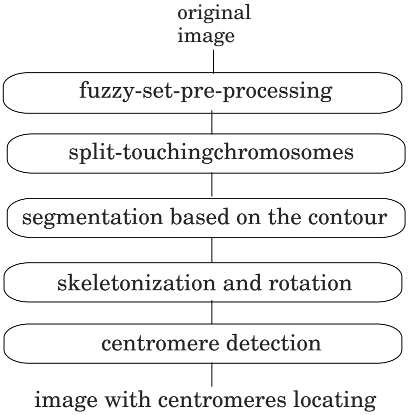

original image
fuzzy-set-pre-processing
split-touchingchromosomes
segmentation based on the contour
skeletonization and rotation
centromere detection
image with centromeres locating

Fig. 1. Outline of the centromere detecting algorithm

2.1 Preprocessing Based on Fuzzy Sets

The image of the fish chromosome presents noise and low contrast. The central part of the chromosomes appears darker and the transaction region between the central part and the background is clearer. The immediate application of thresholding does not yield good results. The objective of this procedure is to improve the contrast of chromosome images and to eliminate the background noises. The fuzzy-set-pre-processing method proposed first uses a quadratic function, defined in equation (1) to increase contrast between chromosomes and the background.

$$P = \frac{(p_j)^2}{255}, \; where\ p_j\ is\ the\ j^{th}\ pixel\ of\ image \tag{1}$$

A fuzzy membership function defined in equation 2 is applied to enhance the chromosomes. This function is based on the work developed by Guliato et al. [8] and defined as $\mu_1 : I \rightarrow [0,1]$, where I is a image, p_j is a j^{th} pixel of image I, L_i and L_s determine the interval for characterizing the center of the chromosome in gray-scale levels and β determines the opening of fuzzy membership function: higher values result in a function with severe behavior and lower values result in a permissive behavior.

$$\mu_1(p_j) = \begin{cases} 1, \text{ if } \quad L_i \le p_j \le L_s \\ \dfrac{1}{1+\beta|L_s - p_j|}, \text{ if } p_j > L_s \\ \dfrac{1}{1+\beta|L_i - p_j|}, \text{ if } p_j < L_i \end{cases} \tag{2}$$

The resulting image displays the chromosomes with higher gray levels and the background with darker gray levels. Now a thresholding algorithm can be used to eliminate the background. The 8-connected regions are labeled and statistical measurements are assessed such as: mean, standard deviation, maximum and minimum gray-scale values. These statistical measurements will be used to guide the following phases in the centromere locating algorithm. Figure 2 shows the partial results obtained after each stage of fuzzy-set-pre-processing image.

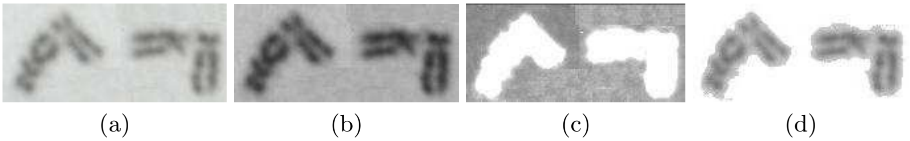


Fig. 2. Partial results obtained of fuzzy-set-processing (a) part of the original image obtained through fish chromosome photomicrography; (b) the resulting image after applying the function defined in eq. (1); (c) the resulting image after applying the fuzzy membership function defined in eq.(2) and (d) the enhanced original image

2.2 Image Segmentation Based on the Contour

The objective of this phase is to extract the contour of the chromosomes concerning touching chromosomes. However, overlapping chromosomes are not treated

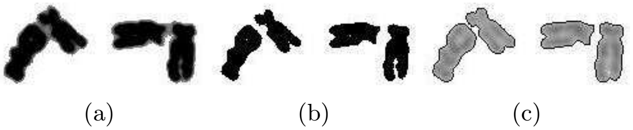


Fig. 3. Results of segmentation (a) result obtained after applying the fuzzy membership function $\mu_2(p_j)$, eq. (3), in image shown in Figure 2 (d); (b) thresholding to separate the touching chromosomes; (c) the resulting contour superimposed to original image

in this work. For obtaining the chromosome contour segmentation each chromosome is first represented by a fuzzy set that preserves the transition information between chromosomes and background. To obtain the fuzzy set, the equation (3) is applied to each chromosome,

$$\mu_2(p_j) = \begin{cases} 1, \text{ if} \quad mvR_k \ \le \ p_j \ \le \ \mu - \frac{\sigma}{2} \\ \\ \dfrac{1}{1+\beta|((\mu-\frac{\sigma}{2})-p_j)|}, \ otherwise \end{cases} \tag{3}$$

where mvR_k, μ and σ are the value of gray-scale level lower, mean and the standard deviation of 8-connected region, respectivally and β is the opening of fuzzy membership function.

At the end of such a process, pixels in and around of each chromosome will be displayed according to their degree of similarity with respect to the feature of central part of chromosome given by μ - $\frac{\sigma}{2}$, where μ and σ are the mean and standard deviation of the chromosome being segmented, as shown in Figure 3 (a).

For separating touching chromosomes, a global threshold is applied to the image, see Figure 3 (b). After that a region growing algorithm is carried out for each chromosome obtained in the last step to detect the chromosome contours, as shown in Figure 3 (c).

2.3 Chromosome Rotation

Before locating the chromosome centromere it is necessary to rotate the chromosome to a vertical position. In order, to realize it automatically the authors proposed first to preprocess the chromosomes making then a convex connected region by filling in the regions among the chromatides. After that, the chromosome skeleton is obtained with one-pixel width [9] and then the inclination coefficient of the principal axis of the skeleton is assessed. The angle related to the inclination coefficient will be used to rotate the chromosomes. However, in some cases the chromosome skeleton presents extra ramifications that make it difficult to detect the principal axis. A posprocessing is applied to the skeleton to eliminate these extra ramifications and to enhance the principal axis. Sometimes it is not possible to find the principal axis of the skeleton and the rotating angle accurately. In this case a manual adjustment is necessary. This procedure yields a binary image with the chromosomes rotated to a vertical position. The results of skeletonization and chromosome rotating are shown in the Figure 4.

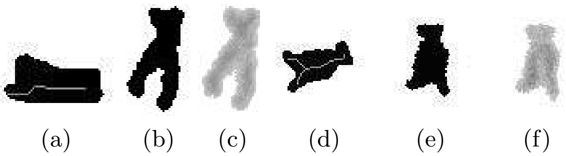
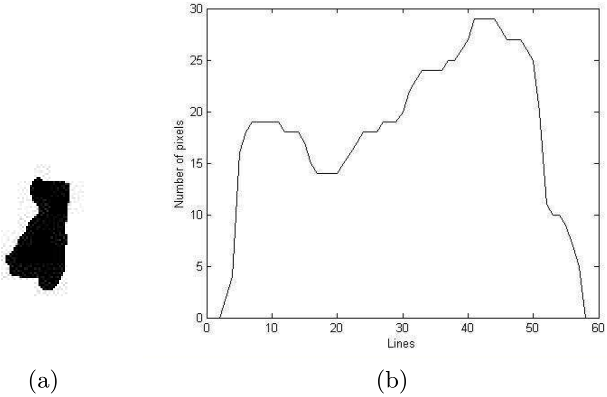

(a) (b) (c) (d) (e) (f)

Fig. 4. Result of skeletonization and rotation (a) skeleton without ramifications superimposed to binary image of chromosome with the region among chromatides filled; (b) rotation based on skeleton of Figure 4 (a), applied in binary image (c) rotation based on skeleton of Figure 4 (a), applied in original image; (d) skeleton with ramifications superimposed to binary image of chromosome with the region among chromatides filled; (e) rotation based on skeleton of Figure 4 (d), in which extra segments were eliminated, applied in binary image and (f) rotation based on skeleton of Figure 4 (d), in which extra segments were eliminated, applied in original image

2.4 Centromere Locating

The chromosome centromere is the point where the chromatides of the chromosomes join together. The acknowledgement of this position is fundamental to the classification of each chromosome in one of the four classes: metacentric, submetacentric, subtelocentric and acrocentric. The centromere is located in the thinner region of the chromosome. For the automatic locating of the centromere, a vector projection is obtained by scanning each chromosome image line and summing the number of the pixels of the scan line that belong to the chromosome. The use of vector projection with this purpose was first presented by Moradi et al. [6], who applied their method to locate the centromere of human chromosomes. The vector projection as proposed by Moradi et al [6] does not work well when applied to fish chromosome as their chromatides are



(a) (b)

Fig. 5. Calculation of projection vector (a) chromosome image after the regions among the chromatides have been filled (b) projection vector used to choose the locating centromere

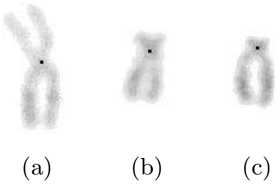


Fig. 6. Results of centromere locating (a) metacentric chromosome; (b) submetacentric chromosome and (c) subtelocentric chromosome

disjoined. For locating the fish chromosome centromere using the vector projection approach each binary rotated chromosome image, as shown in Figure 4 (b) and 4 (e), should have the regions among the chromatides filled in again. This process makes each chromosome a convex region preserving its external morphology as shown in Figure 5 (a). After scanning the binary chromosome image, as shown in Figure 5 (b), the resulting projection vector can be seen as a skyline that possess two or more peaks and some valleys. The position of the centromere is the lowest point located in the largest basin. For defining the largest basin of this skyline an algorithm based on watershed transform was used [10]. This algorithm is successful when the chromosomes are metacentric, submetacentric and subtelocentric. For the automatic locating of the centromere of acrocentric chromosomes the proposed algorithm did not yield satisfactory results. A discussion about the centromere locating results is presented in the next section. Figure 6 shows the three different classes with the position of the chromosome centromere superimposed.

3 Discussion

We proposed a new approach for locating the centromere of fish chromosomes using fuzzy-set-based preprocessing step to enhance the chromosome region, an algorithm based on region growing to extract the contour and a vector projection to detect the position of the centromere.

The segmentation and locating centromeres methods were applied to 40 fish chromosome images, which were chosen randomly. In the segmentation process there were 20 images with occurrence of touching chromosomes. In these images, there were about 2.2 touching chromosomes per image. After the application of segmentation methods this value changes for 0.4 touching chromosomes, resulting in an improvement of 81.82%.

For the rotation process, on average, 15.53 chromosomes must be rotated manually, since the images have about 49.15 chromosomes.

For the locating centromere process, two cytogeneticists were solicited to detect the centromeres of 40 images, manually. In these images the acrocentric chromosomes, the overlapped chromosomes and the chromosomes where the cytogeneticists were not sure about the correct centromere locating were eliminated.

Based in the fact that the number of chromosomes, whose centromere locating caused doubts, varied from one cytogeneticist to another, and these chromosomes were eliminated from the analysis, the results given by the first cytogeneticist for the average number of chromosomes available for analysis were by 40.625 per image and for the second cytogeneticist 38.925 chromosomes per image.

According to the first cytogeneticist, the percentage of chromosomes with correct locating centromeres was 81.79% and for the second cytogeneticist 82.54%.

Based on these results, the following steps help to improve the chromosome rotation and develop an algorithm for locating the centromere of acrocentric chromosomes. After chromosome classification based in centromeric index is made and assembly.

Future directions include to assembly karyotypes and choose automatically the best one to represent the population being studied. Furthermore, to evaluate this system to assembly human and other animals karyotype.

Acknowledgment

This work is supported by CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico. We are grateful to Laboratório de Citogenética of Universidade Federal de Uberlândia, which supplied the images for tests and helped us in centromere locating.

References

1. Pantaleão, C.H.Z.; Pereira E.T.; Azevedo F.M.; Ribeiro M.C. M. Desenvolvimento de um Sistema Computadorizado para Análise e Classificação Citogenética, II Workshop de Informática Aplicada à Saúde - CBComp 2002, 2002.
2. Araujo, A.C.S. Comparação Citogenética de Quatro Populações de Astyanax scabripinnis (Pisces, Characidae) da Região do Triângulo Mineiro, 2003. Dissertação (Mestrado em Genética e Bioquímica) - Universidade Federal de Uberlândia, Uberlândia.
3. Neo D.M.; Ferro D.A.; Moreira-Filho O.; Bertollo L. A. Nucleolar organizing regions, 18S and 5S rDNA in Astyanax scabripinnis (Pisces, Characidae): populations distribution and functional diversity, Genetica 110(1):55-62, 2000.
4. Popescu M.; Garder P.; Keller J.; Klein C.; Stanley J.; Caldwell C. Automatic Karyotyping of Metaphase Cells With Overlapping Chromosomes, Computers in Biology and Medicine, Science, vol. 29, n. 1, jan., 1999.
5. Agam G.; Dinstein I. Geometric separation of partially overlapping non-rigid objects applied to automatic chromosome classification, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-19, pp. 1212-1222, 1997.
6. Moradi M.; Setarehdan S.K.; Ghaffari S.R. Automatic Locating the Centromere on Human Chromosome Pictures, Proceedings 16th IEEE Symposium on Computer-Based Medical Systems (CBMS'03) pp. 56-61, New York, 2003.
7. Cho M.J. Chromosome Classification Using Back propagation Neural Networks, IEEE Eng in Medicine and Biology, pp: 28-33, Jan/Feb 2000

8. Guliato D.; Rangayyan R.M.; Carnielli W.A.; Zuffo J.A.; Leo Desautels J.E. Segmentation of breast tumors in mammograms. Journal of Eletronic Imaging 12(3):369-378, 2003.
9. Facon J. Home Page de Jacques Facon - Processamento e Análise de Imagens: Algoritmo de Afinamento Holt. Pontifícia Universidade Católica do Paraná, Programa de Pós-Graduação em Informática Aplicada, Disponível em: <http://www.ppgia.pucpr.br/∼facon/Afinamento/AfinamentoHolt.pdf> Acesso em 22 de março de 2005.
10. Roerdink J.B.T.M.; Meijster A., The Watershed Transform: Definitions, Algorithms and Parallelization Strategies, Fundamenta Informaticae 41, pp.187-228, 2000.

Sequence Motif Identification and Protein Family Classification Using Probabilistic Trees*

Florencia Leonardi and Antonio Galves

Instituto de Matemática e Estatística, Universidade de São Paulo

Abstract. Efficient family classification of newly discovered protein sequences is a central problem in bioinformatics. We present a new algorithm, using *Probabilistic Suffix Trees*, which identifies equivalences between the amino acids in different positions of a motif for each family. We also show that better classification can be achieved identifying representative fingerprints in the amino acid chains.

1 Introduction

A central problem in genomics is to determine the function of a new discovered protein using the information contained in its amino acid sequence [1]. Nowadays, the most popular methods to generate a hypothesis about the function of a protein are BLAST and Hidden Markov Models (HMM).

Probabilistic Suffix Trees (PST) were first introduced in [2] as a universal model for data compression. A major advantage of PST is its capacity of extracting structural information from the sequences under analysis. Recently, an implementation of PST has been successfully used in protein classification [3], even though its performance decreases with less conserved families. Better results have been obtained using mixtures of PST models for sparse sequences [4, 5]. A major drawback of these algorithms is their high complexity, which makes problematic their application in very large databases.

We present a new algorithm to estimate *Sparse Probabilistic Suffix Trees* (SPST). We also show that the identification of sub-sequences of maximal mean probability (*fingerprints*) increases the classification rates of the SPST algorithm. This is the basis of our F-SPST algorithm.

2 Variable Length Markov Models

It was suggested in the literature to use PST models to fit protein families. A PST is a Variable Length Markov Model (VLMC), that is, a stochastic chain $(X_0, X_1, \ldots)$ taking values on a finite alphabet $\mathcal{A}$ and characterized by two elements. The first element is the set of all contexts. A context $X_{n-\ell}, \ldots, X_{n-1}$

* This work is partially supported by CAPES and is part of PRONEX/FAPESP's Project *Stochastic behavior, critical phenomena and rhythmic pattern identification in natural languages* (grant number 03/09930-9).

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 190–193, 2005.

is the finite portion of the past $X_0, \ldots, X_{n-1}$, for each time, which is relevant to predict the next symbol X_n. The second element is a family of probability transitions associated to each context. Given a context, its associated probability transition gives the distribution of occurrence of the next symbol immediately after the context.

In a PST the set of contexts has the *suffix property*: looking from the present to the past no context is a suffix of another context. This makes it possible to define without ambiguity the probability distribution of the next symbol. The suffix property makes it possible to represent the set of contexts as a tree. In this tree, each context $c = (c_{-k}, \ldots, c_{-1})$ is represented by a complete branch, in which the first node on top is c_{-1} and so on until the last element c_{-k} which is represented by the terminal node of the branch.

In a PST model for a protein family, the alphabet $\mathcal{A}$ represents the set of twenty amino acids and the stochastic chains $(X_0, X_1, \ldots)$ are the sequences of amino acids belonging to the family.

A *Sparse Probabilistic Suffix Tree* (SPST) is a PST in which some contexts are grouped together in an equivalence class. More precisely, the contexts of a SPST model are sequences of the form $A_{n-\ell}, \ldots, A_{n-1}$, with $A_i \subset \mathcal{A}$ for each i. This feature makes SPST models more suitable for sparse sequences like amino acids chains.

3 The SPST and the F-SPST Algorithms

The SPST algorithm works as follows. It starts with a tree consisting of a single root node. At each step, for every terminal node t with depth less than L and for every symbol x, the leaf x is added to t, if the sequence xt appears in the training sequences at least $N_{\min}$ times. For every pair of new leaves of a node, we test their *equivalence* using a log-likelihood ratio test and choose the pair that realizes the minimum between all the tests. If this minimum belongs to the acceptance region, the leaves are merged together in a single leaf. The procedure is iterated with the new set of leaves. It stops when no more leaves can be merged. The acceptance region is defined by $\{c < r_{max}\}$, where c is the value of the test. Clearly, taking the minimum between the tests ensures the independence of the order in which the tests are performed.

To conclude the construction of the SPST we assign to each leaf a transition probability estimated by the usual maximum likelihood procedure. In order to avoid non zero probabilities, the distributions associated to each leaf (context) are smoothed by a constant γ_{min}.

After the construction of the model, we want to decide if a given sequence of amino acids belongs to the family or not. To do this, we calculate the log probability of the sequence in the family model and divide this value by the length of the sequence. If this value is greater than a predefined threshold, the protein is identified as a member of the family.

The *Fingerprint-SPST* algorithm estimates the context tree and the transition probabilities in the same way as the SPST algorithm. However, to classify a new sequence of amino acids, F-SPST starts by identifying fingerprints defined as follows. Given a new sequence of amino acids, we look for the sub-sequence of length M with maximal probability, where M is a parameter which depends on the size of the domains in each family. If this maximum is bigger than a pre-defined threshold, the protein is identified as a member of the family.

4 Statistical Results

In order to test our algorithms and to compare them with PST published results [3] we use protein families of the Pfam database [6] release 1.0. This database contains 175 families derived from the SWISSPROT 33 database [7]. We trained both SPST and F-SPST with 4/5 of the sequences in each family, and then we applied the resulting models to classify all the sequences in the SWISSPROT 33 database. To establish the family membership threshold, we used the **equivalence number criterion** [8]. This method sets the threshold at the point where the number of false positives equals the number of false negatives. The quality of the model is measured by the number of true positives detected relative to the total number of proteins in the family.

Table 1 summarizes the classification rates obtained with our SPST and F-SPST algorithms together with the published results obtained with the PST algorithm [3]. We emphasize that these are preliminary results as no attempt was made to optimize the choice of the parameters. It is clear that SPST and

Table 1. Performance comparison between PST, SPST and F-SPST. The parameters in the SPST and F-PST algorithms where: $L = 20$, $N_{\min} = 2$, $\gamma_{\min} = 0.001$ and $r_{\max} = 3.8$. The length of the fingerprint in the F-SPST algorithm was $M = 80$ for all families

Family	Size	PST	SPST	F-SPST
7tm_1	515	93.0%	96.3%	97.7%
7tm_2	36	94.4%	97.2%	100.0%
7tm_3	12	83.3%	100.0%	100.0%
AAA	66	87.9%	90.9%	93.9%
ABC_tran	269	83.6%	85.9%	89.3%
actin	142	97.2%	97.2%	99.3%
adh_short	180	88.9%	89.4%	92.8%
adh_zinc	129	95.3%	91.5%	95.3%
aldedh	69	87.0%	89.9%	92.8%
alpha-amylase	114	87.7%	91.2%	94.7%
aminotran	63	88.9%	88.9%	90.5%
ank	83	88.0%	86.8%	86.6%
arf	43	90.7%	93.0%	93.0%
asp	72	83.3%	90.3%	91.7%
ATP-synt_A	79	92.4%	94.9%	97.5%

F-SPST improves PST classification rates in all cases except for the *Ankyrin repeat* family. It is interesting to note that this family consists of very short domains (with mean length equal to 28.12), and this could explain the reduction in the classification rate.

Another very interesting feature of SPST appears when we compare the equivalence classes in the estimated trees with the classes obtained by grouping the amino acids by their physical and chemical properties. For instance, the estimated tree for the AAA family identifies as equivalence class the set of amino acids $\{I, V, L\}$ which corresponds exactly to the group of aliphatic amino acids. For more details see `http://www.ime.usp.br/~leonardi/spst/`.

5 Conclusion

The preliminary results presented in this paper strongly suggest that these new algorithms can improve in a significant way the classification rates obtained with the PST algorithm. We are presently applying our algorithms to more families in the Pfam database to confirm this initial encouraging results.

Nevertheless, even at this preliminary stage, it is alredy clear that a Sparse Probabilistic Tree fits protein families well. This is probably due to the fact that the sparse model mimics well the sparse nature of relevant domains in the amino acids chains. It is also worth observing that the complexity of the SPST and F-SPST algorithms is smaller than the complexity of previously presented algorithms for sparse sequences [4, 5].

References

1. Karp, R.M.: Mathematical challenges from genomics and molecular biology. Notices Amer. Math. Soc. **49** (2002) 544–553
2. Rissanen, J.: A universal data compression system. IEEE Trans. Inform. Theory **29** (1983) 656–664
3. Bejerano, G., Yona, G.: Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. Bioinformatics **17** (2001) 23–43
4. Eskin, E., Grundy, W.N., Singer, Y.: Protein family classification using sparse markov transducers. In: Proc. Int'l Conf. Intell. Syst. Mol. Biol. Volume 8. (2000) 134–145
5. Bourguignon, P.Y., Robelin, D.: Modèles de Markov parcimonieux: sélection de modèle et estimation. manuscript (2004)
6. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L.L., Studholme, D.J., Yeats, C., Eddy, S.R.: The Pfam protein families database. Nucl. Acids Res. **32** (2004) D138–141
7. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucl. Acids Res. **31** (2003) 365–370
8. Pearson, W.R.: Comparison of methods for searching protein sequence databases. Protein Sci **4** (1995) 1145–1160

Prediction of Myotoxic and Neurotoxic Activities in Phospholipases A2 from Primary Sequence Analysis

Fabiano Pazzini[1], Fernanda Oliveira[1], Jorge A. Guimarães[1],
and Hermes Luís Neubauer de Amorim[1,2]

[1] Biotechnology Center, Federal University of Rio Grande do Sul – UFRGS,
91501-970, RS, Brazil
fabianopasin@cbiot.ufrgs.br
[2] Department of Chemistry, Lutheran University of Brazil - ULBRA,
92420-280, RS, Brazil

Abstract. We developed a methodology to predict myotoxicity and neurotoxicity of proteins of the family of Phospholipases A2 (PLA2) from sequence data. Combining two bioinformatics tools, MEME and HMMER, it was possible to detect conserved motifs and represent them as Hidden Markov Models (HMMs). In ten-fold cross validation testing we have determined the efficacy of each motif on prediction of PLA2 function. We selected motifs whose efficacy in predict function were above 60 % at the Minimum Error Point (MEP), the score in which there are fewest both false positives and false negatives. Combining HMMs of the best motifs for each function, we have achieved a mean efficacy of 98 ± 4 % on prediction of myotoxic function and 77.4 ± 4.8% on prediction of neurotoxicity. We have used the results of this work to build a web tool (available at www.cbiot.ufrgs.br/bioinfo/phospholipase) to classify PLA2s of unknown function regarding myotoxic or neurotoxic activity.

1 Introduction

One of the most important tasks of the bioinformatics is to give meaning to the large amount of data from genomic and proteomic projects. Part of this task comprises automatic prediction of the function of proteins. However, the most currently used algorithms and databases (such as BLAST [1], PFAM [2] and PROSITE [3]) strive to classify protein sequences into broad families, which not necessarily share the same biological function. The Phospholipase A2 (PLA2) family (E.C. 3.1.1.4), initially classified according to its ability to catalyze the cleavage of membrane phospholipids, represents an interesting challenge. Despite the high level of sequence similarity and structure conservation of the family [4], the proteins of this group are involved in distinct biological functions such as digestion, cell signaling and inflammation. They also present myotoxic, neurotoxic and cytotoxic activities.

Based on the analysis of conserved amino acids and protein motifs and using Hidden Markov Models (HMMs) to capture the particular characteristics of the Multiple Sequence Alignments (MSAs), we developed a methodology to discriminate

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 194 – 197, 2005.

between neurotoxic PLA2 (nPLA2) and myotoxic PLA2 (mPLA2). Based in the results of this work we provided a tool, now available at www.cbiot.ufrgs.br/bioinfo/phospholipase, which allows the identification of PLA2s displaying myotoxicity and neurotoxicity. To our knowledge no other method is available allowing classification of the biological function of these PLA2s.

2 Methodology

We collected sequences which were used to build and test the models representing the biological function of interest. For each biological function there are two main sets of sequences: one represents sequences with biological function and other with sequences without the function (negative control).

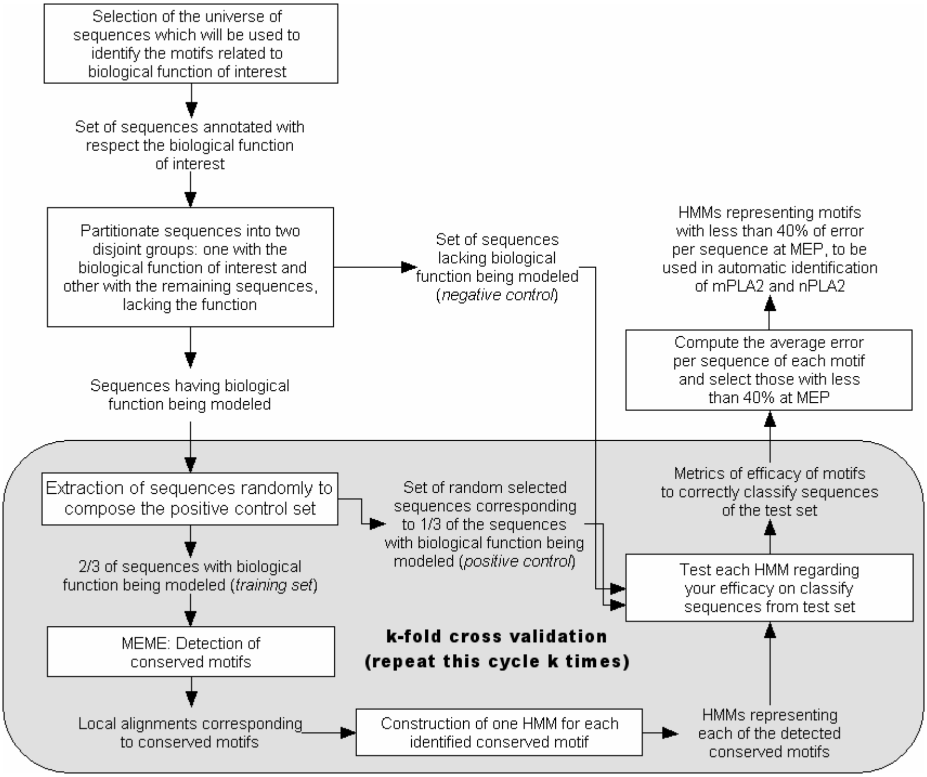

Selection of the universe of sequences which will be used to identify the motifs related to biological function of interest
Set of sequences annotated with respect the biological function of interest
Partitionate sequences into two disjoint groups: one with the biological function of interest and other with the remaining sequences, lacking the function
Set of sequences lacking biological function being modeled (*negative control*)
HMMs representing motifs with less than 40% of error per sequence at MEP, to be used in automatic identification of mPLA2 and nPLA2
Compute the average error per sequence of each motif and select those with less than 40% at MEP
Sequences having biological function being modeled
Extraction of sequences randomly to compose the positive control set
Set of random selected sequences corresponding to 1/3 of the sequences with biological function being modeled (*positive control*)
Metrics of efficacy of motifs to correctly classify sequences of the test set
2/3 of sequences with biological function being modeled (*training set*)
Test each HMM regarding your efficacy on classify sequences from test set
MEME: Detection of conserved motifs
k-fold cross validation (repeat this cycle k times)
Local alignments corresponding to conserved motifs
Construction of one HMM for each identified conserved motif
HMMs representing each of the detected conserved motifs

Fig. 1. Flow diagram of the methodology to detect sequence motifs specific to some biological function

In our approach we used MEME [5] to detect conserved motifs and HMMER [6] to construct HMMs of each motif found. The complete process is depicted in Fig. 1.

On each iteration, the sequences that have the biological function are split in training set (used to construct HMMs) and positive control (which together with negative control forms the *test set*).

The first efficacy metric that is calculated is the *Error Rate* (ER) at each score. It shows how well some HMM classify the sequences of the test set:

$$\text{ER} = (\text{FP} + \text{FN}) \,/\, \text{size of test set} \qquad (1)$$

FP represents false positives, and FN, false negatives.

Based on the ER at each score, it is possible to determine the *Minimum Error Point* (MEP), the score which ER value is minimum. The efficacy value at the MEP is the best possible for the motif. Note that we define Prediction Accuracy (PA) as the complement of the error rate, *i.e.*,

$$\text{PA} + \text{ER} = 1 \qquad (2)$$

The *coverage* measures how much of the true positives (TP) were correctly classified above some score. It is calculated as the ratio between TP above some score and the total number of sequences of the positive control.

As the biological function can be associated to more than one motif, all motifs with PA greater than 60% were selected to be used in function prediction.

In the case of occurrence of multiple motifs to detect the same biological function, it is possible to combine them, improving the PA of the respective function. If each of these motifs recognizes different subsets of true positives, combining their results will increase the coverage, but the impact on PA must be calculated considering both TP and FP of the maximal set composed by all sequences with score above MEP of the respective motif.

3 Results

Table 1 shows the motifs detected and the corresponding average accuracy values, computed after 10-fold cross validation process.

Table 1. Motifs with mean predictive accuracy (PA) greater than 60 % at MEP

Group	Motif	MEP score	PA at MEP (%)	Coverage at MEP (%)
mPLA2	N-terminal	28.03	86	72
mPLA2	C-terminal region	18.17	80	60
nPLA2	N-terminal	82.67	63.5	39.8
nPLA2	near catalytic site	47.67	67.3	75

In order to improve the PA of the final model, all detected motifs related to the same biological function were combined, maximizing their capability to correctly recognize their target biological function. The parameters of the 10-fold cross validation and the mean efficacy for the best motifs are in Table 2.

Table 2. Parameters and results of k-fold cross validation

Group	k	Size of functional set	Size of negative control	Number of motifs with ER < 40%	Coverage for combined motifs (%)	Best mean PA for combined motifs
mPLA2	10	20	8	2	96,0	98,0±4,0%
nPLA2	10	16	13	2	78,6	69.5±7.6%

4 Concluding Remarks

The use of conserved motifs, instead the entire sequences, to construct each HMM helps to minimize the bias induced by the small training sets [7]. Additionally the utilization of Dirichlet Mixtures by HMMER also increases the generalization power of the resulting HMM [8].

Considering the biochemical and pharmacological importance of the PLA2s, especially those exhibiting toxicological effects, we expect that the methodology described here can contribute for the advance of the knowledge in this area of research.

References

1. Scott McGinnis, Thomas L.Madden: BLAST: at the core of a powerful and diverse set of sequence analysis tools. Nucl. Acids. Res.,Vol. 32. (2004) W20-W25
2. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L.L., Studholme, D.J., Yeats, C., and Eddy, S.R.: The Pfam protein families database. Nucl. Acids. Res.,Vol. 32. (2004) D138-D141
3. Sigrist C.J.A., Cerutti L., Hulo N., Gattiker A., Falquet L., Pagni M., Bairoch A., and Bucher P.: PROSITE: A documented database using patterns and profiles as motif descriptors. Briefings in Bioinformatics,Vol. 3. (2002) 265-274
4. Manjunatha Kini, R.: Excitement ahead: structure, function and mechanism of snake venom phospholipase A2 enzymes. Toxicon,Vol. 42. (2003) 827-840
5. Timothy L. Bailey and Charles Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers", Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, pp. 28-36, AAAI Press, Menlo Park, California, 1994.
6. Eddy, S.R.: Profile hidden Markov models. Bioinformatics, Vol. 14. (1998) 755-763
7. Grundy, W.N. *et al*: Meta-MEME : motif-based hidden Markov models of protein families. CABIOS, Vol. 13. (1997) 397-406
8. Haussler, D. *et al*: Dirichlet Mixtures: A Method for Improving Detection of Weak but Significant Protein Sequence Homology. CABIOS, Vol. 12. (1996) 327-345

Genomics and Gene Expression Management Tools for the *Schistosoma Mansoni* cDNA Microarray Project

Venancio, T.M.[1,3], DeMarco, R.[2,3], Oliveira, K.C.P.[2,3], Simoes, A.C.Q.[1,3], da Silva, A.M.[3], and Verjovski-Almeida, S.[2,3]

[1] Laboratório de Bioinformática
[2] Laboratório de expressão gênica em eucariotos
[3] Departamento de Bioquímica, Instituto de Química, Universidade de São Paulo

Schistosoma mansoni, a trematode parasite, is the major causative agent of Schistomiasis, a public health problem in South America and Africa. Recently, as a result of two separate efforts, the transcriptome of *Schistosoma mansoni* [1] and *S. japonicum* [2] were published. Schistosomes possess distinct and differentiated organs and have evolved to adapt to parasitism. Availability of transcriptome data has raised a number of issues regarding the parasite's cell biology and signaling pathways, as recently discussed in a review [3].

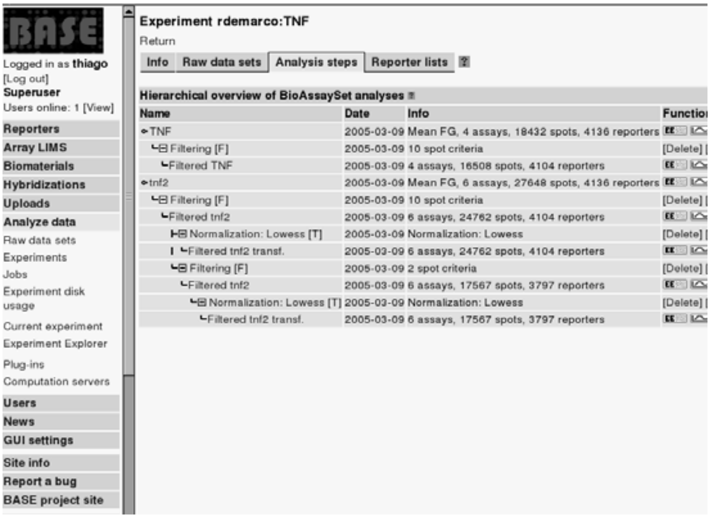
Currently, the *S. mansoni* microarray project, being conducted at Instituto de Quimica, Universidade de São Paulo, is aimed at identifying the genes and pathways involved in the parasite's development. This project raises the need for appropriate tools and databases to manage and analyze gene expression data, integrating these results with genome and sequence analysis information.

In this work we describe the implementation of local copies of two important tools in our project: (i) the BioArray Software Environment (BASE), a platform to manage and analyze microarray data [4] and (ii) the generic genome browser, a web-based tool to visualized genomic information and other features [5]. We have implemented the BASE system (version 1.2.15) to centralize storage and to maintain data integrity, which is a very important aspect in large-scale microarray experiments. The relational database manager used is MySQL. We have deposited information regarding our array design with all reporters, integrating the information of the 96-well re-array plates, the 384-well consolidated cDNA source plates and the position of each reporter in the final array design. A screenshot of our BASE implementation can be seen in figure 1.

The microarray images were previously analyzed with ArrayVision 6.0 in order to extract the raw fluorescence intensity data, which were subsequently corrected for background intensity. Lowess normalization was performed using R scripts adapted from Koide et al. (2004) [6]. An example of this first step normalization is shown in figure 2. We have built a pipeline that processes all data, from the ArrayVision spreadsheets to a user friendly schema that shows the BLAST search results for the *S. mansoni* differentially expressed genes. The first step is data normalization, followed by filtering steps, which permit exclusion of controls and genes with weak signal from the subsequent analysis. We have performed Significance Analysis of Microarray (SAM) [7] to identify the differentially expressed genes in the dataset. The subsequent steps in the pipeline build a multifasta file with sequences from all these se-

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 198 – 201, 2005.

lected genes, submit it to a BLAST search, parse it and build a HTML report with a summary of the results. This pipeline has been used with success in our lab, as we have already identified several gender specific differentially expressed genes in *S. mansoni* (in preparation). Some of these genes have been similarly identified recently by Fitzpatrick et al (2005) [8], but some of them are newly identified. Currently, we are performing wet lab validation steps in order to confirm these results.



Experiment rdemarco:TNF

Hierarchical overview of BioAssaySet analyses

Name	Date	Info	Functio
TNF	2005-03-09	Mean FG, 4 assays, 18432 spots, 4136 reporters	
Filtering [F]	2005-03-09	10 spot criteria	[Delete]
Filtered TNF	2005-03-09	4 assays, 16508 spots, 4104 reporters	
tnf2	2005-03-09	Mean FG, 6 assays, 27648 spots, 4136 reporters	
Filtering [F]	2005-03-09	10 spot criteria	[Delete]
Filtered tnf2	2005-03-09	6 assays, 24762 spots, 4104 reporters	
Normalization: Lowess [T]	2005-03-09	Normalization: Lowess	[Delete]
Filtered tnf2 transf.	2005-03-09	6 assays, 24762 spots, 4104 reporters	
Filtering [F]	2005-03-09	2 spot criteria	[Delete]
Filtered tnf2	2005-03-09	6 assays, 17567 spots, 3797 reporters	
Normalization: Lowess [T]	2005-03-09	Normalization: Lowess	[Delete]
Filtered tnf2 transf.	2005-03-09	6 assays, 17567 spots, 3797 reporters	

Fig. 1. Screenshot of our BASE implementation. The steps of spot filtering can be visualized

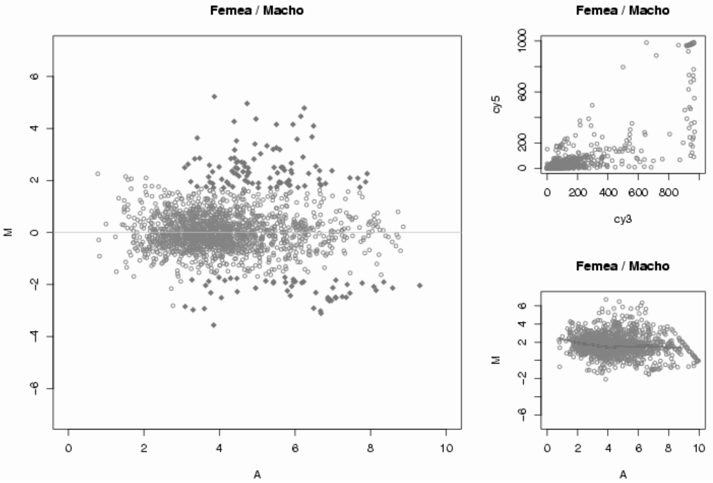
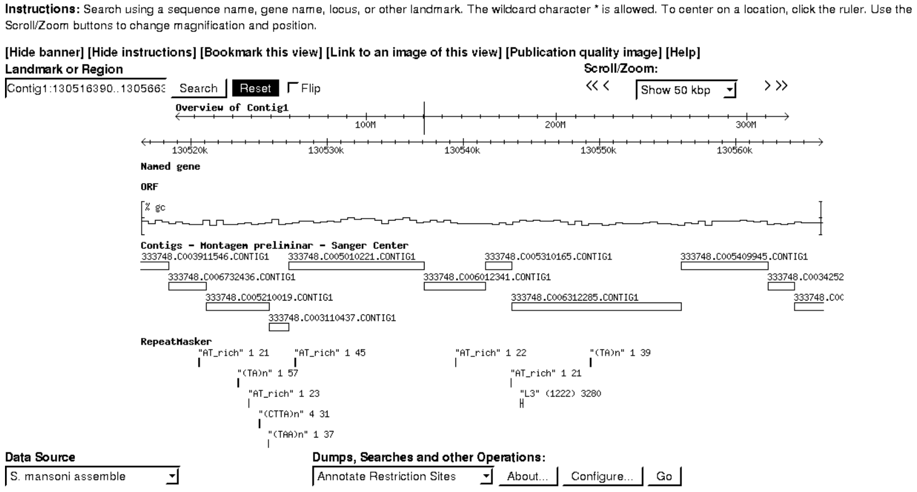


Fig. 2. One of our pipeline steps. Lowess normalization process; the panel on the left shows the normalized data output resulting from the raw input data on the right. “Femea/macho” indicates comparison between female/male gene expression

To facilitate further data integration of microarray results with genomic information, we have implemented the Generic Genome Browser. This browser allows the user to scroll and zoom in different regions of the genome, search for a specific landmark and perform a full search of all features, as well as enable and disable the visualization of some tracks. Some of the tracks implemented are the mapping of our *S. mansoni* EST reads to genomic sequence, BLASTX results of similarity searches against other species, low complexity regions, GC content and the sequence itself. Although the *S. mansoni* genome sequence is not fully determined yet, we have used a preliminary assembly obtained at the Sanger Institute FTP site [9] in which the genome is still fragmented into 70,714 contigs (Jan 14, 2005 release). As an example, this browser permits us to visualize the differentially expressed genes identified in our microarray and to analyze them in a genomic context, checking if any of them overlaps with the low complexity regions, compare to gene predictions, 3′ and 5′ UTR etc. These analyses steps are essential to obtain a detailed picture of the differentially expressed genes for further elucidation of their functions. A screenshot of our Genome Browser implementation can be seen in figure 3.



Instructions: Search using a sequence name, gene name, locus, or other landmark. The wildcard character * is allowed. To center on a location, click the ruler. Use the Scroll/Zoom buttons to change magnification and position.

Landmark or Region: Contig1:130516390..1305663

Scroll/Zoom: Show 50 kbp

Data Source: S. mansoni assemble

Dumps, Searches and other Operations: Annotate Restriction Sites

[Hide banner] [Hide instructions] [Bookmark this view] [Link to an image of this view] [Publication quality image] [Help]
Search
Reset
Flip
<< <
> >>
Overview of Contig1
100M
200M
300M
130520k
130530k
130540k
130550k
130560k
Named gene
ORF
% gc
Contigs - Montagem preliminar - Sanger Center
333748.C003911546.CONTIG1
333748.C005010221.CONTIG1
333748.C005310165.CONTIG1
333748.C005409945.CONTIG1
333748.C006732436.CONTIG1
333748.C006012341.CONTIG1
333748.C0034252
333748.C005210019.CONTIG1
333748.C006312285.CONTIG1
333748.COC
333748.C003110437.CONTIG1
RepeatMasker
"AT_rich" 1 21
"AT_rich" 1 45
"AT_rich" 1 22
"(TA)n" 1 39
"(TA)n" 1 57
"AT_rich" 1 21
"AT_rich" 1 23
"L3" (1222) 3280
"(CTTA)n" 4 31
"(TAA)n" 1 37
About...
Configure...
Go

Fig. 3. Screenshot of our Generic Genome Browser implementation for *S. mansoni* genome visualization. Here we can see the EBI assembly track, the low complexity regions and GC content

References

1. Verjovski-Almeida, S., DeMarco, R., Martins, E., Guimaraes, P., Ojopi, E., Paquola, A., Piazza, J., Nishiyamajr, M., Kitajima, J., Adamson, R., Ashton, P., Bonaldo, M., Coulson, P., Dillon, G., Farias, L., Gregorio, S., Ho, P., Leite, R., Malaquias, L., Marques, R., Miyasato, P., Nascimento, A., Ohlweiler, F., Reis, E., Ribeiro, M., Sa, R., Stukart, G., Soares, M., Gargioni, C., Kawano, T., Rodrigues, V., Madeira, A., Wilson, R., Menck, C., Setubal, J., Leite, L., Dias-Neto, E. Transcriptome analysis of the acoelomate human parasite *Schistosoma mansoni*. Nature Genetics Vol. 35 (2003): 148-157.

2. Hu W, Yan Q, Shen DK, Liu F, Zhu ZD, Song HD, Xu XR, Wang ZJ, Rong YP, Zeng LC, Wu J, Zhang X, Wang JJ, Xu XN, Wang SY, Fu G, Zhang XL, Wang ZQ, Brindley PJ, McManus DP, Xue CL, Feng Z, Chen Z, Han ZG. Evolutionary and biomedical implications of a *Schistosoma japonicum* complementary DNA resource. Nature Genetics Vol. 35(2003):139-147.
3. Verjovski-Almeida, S., Leite, L., Dias-Neto, E., Menck, C., Wilson, A. Schistosome transcriptome: insights and perspectives for functional genomics. Trends in Parasitology Vol. 20 (2004): 304-308.
4. Saal LH, Troein C, Vallon-Christersson J, Gruvberger S, Borg A, Peterson C. BioArray Software Environment (BASE): a platform for comprehensive management and analysis of microarray data. Genome Biology Vol. 15 (2002): SOFTWARE0003 1-8.
5. Stein LD, Mungall C, Shu S, Caudy M, Mangone M, Day A, Nickerson E, Stajich JE, Harris TW, Arva A, Lewis S. The generic genome browser: a building block for a model organism system database. Genome Research Vol. 12 (2002):1599-1610.
6. Koide, T., Zaini, P.A., Moreira, L.M., Vencio, R.Z.N., Matsukuma, A.Y., Durham, A.M., Teixeira, D.C., El-Dorry, H., Monteiro, P.B., daSilva, A.C.R., Verjovski-Almeida, S., daSilva, A.M., Gomes, S.L. DNA Microarray-Based genome comparison of a pathogenic and non-pathogenic strain of *Xylella fastidiosa* delineates genes important for bacterial virulence. Journal of Bacteriology Vol. 186 (2004): 5442-5449.
7. Tusher VG, Tibshirani R, Chu G. Significance analysis of microarrays applied to the ionizing radiation response. Proceedings of the National Academy of Sciences USA Vol. 98(9):5116-5121.
8. 8- Fitzpatrick JM, Johnston DA, Williams GW, Williams DJ, Freeman TC, Dunne DW, Hoffmann KF. An oligonucleotide microarray for transcriptome analysis of *Schistosoma mansoni* and its application/use to investigate gender-associated gene expression. Molecular and Biochemical Parasitology. Vol. 141 (2005) 1-13.
9. ftp://ftp.sanger.ac.uk/pub/pathogens/Schistosoma/mansoni/

SAM Method as an Approach to Select Candidates for Human Prostate Cancer Markers

ACQ Simoes[1,2], AM da Silva[1], S Verjovski-Almeida[1], EM Reis[1]

[1] Departamento de Bioquímica, Instituto de Química
anacarol@iq.usp.br
[2] Programa de Pós-Graduação Interunidades em Bioinformática,
Universidade de São Paulo.
emreis@iq.usp.br

Abstract. In order to select gene markers among differentially expressed transcripts identified from tumoral prostate, we have applied a filter and Significance Analysis of Microarrays (SAM) as the feature selection method on a previously normalized dataset of DNA microarray experiments reported by Reis et al., 2004 (Oncogene 23:6684-6692). Twenty seven samples with different degrees of tumor differentiation (Gleason scores) were analyzed. SAM was run using either two-class, unpaired data analysis with Gleason 5-6 and Gleason 9-10 samples, or multiclass response analysis with an additional category of Gleason 7-8. Both strategies revealed a promising set of transcripts associated with the degree of differentiation of prostate tumors.

1 Introduction

To extract information with experimental significance from DNA microarray data and reduce dimensionality, many methods can be applied for feature selection [1] such as Pearson's correlation [2], Principal Component Analysis [3], SAM (Significance Analysis of Microarrays) [4] and improved SAM [5], which differ on the structure and metric used.

The degree of prostate tumor differentiation is defined by the Gleason Score (GS), which is assigned to the tumor sample by histological examination. Our group has recently initiated a thorough search for transcripts whose expression would correlate to the degree of tumor differentiation. For this purpose, we have generated and analyzed a microarray dataset using Self Organizing Map (SOM) and an unsupervised hierarchical clustering analysis as well as Pearson's correlation [6]. This microarray dataset was generated with samples from 27 human prostate tumors along with samples from adjacent normal prostate tissues. Hybridizations were performed with intronic cDNA microarray slides that were enriched with a collection of partial transcripts that map in the human genome sequence to intronic segments of known genes [6]. In the present work we have re-analyzed this complete microarray dataset using four different filters and SAM as feature selection method.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 202 – 205, 2005.

2 Methodology

The dataset used in this work consisted of a table with the expression levels of 3821 spots representing approximately 3700 ORESTES and ESTs [6] obtained for each of the 27 tumoral prostate samples from 27 patients. The data has been previously normalized as described in Reis et al., 2004 [6] and is available at http://verjo19.iq.usp.br/gec/en/publications/.

A filter was generated to parse this table. It was designed to select transcripts that potentially exhibit differential expression levels and to do so it uses the mean intensity of each transcript across the different samples (line) and its variance, considering that only the transcripts that have their intensity above a certain value in a given number of samples should be selected. Two sets were generated, one without the filter and another using the filter to select samples that had intensity levels out of the range of mean +/- 1.25 standard deviation of the mean in at least 4 samples.

For every set two analysis were performed: (i) a two class, unpaired data analysis was performed considering the labels of Gleason Score (GS) 5-6 as category 1 and 9-10 as category 2; and (ii) a multiclass response analysis, considering the labels 1 for GS 5-6, 2 for GS 7-8 and 3 for GS 9-10. Finally, a hierarchical clustering was performed for each analysis using Ward's method [7] and average value distance as the ordering function.

3 Results and Discussion

Expression profiles of 27 prostate samples were analyzed using SAM. A delta value of 1.45 was used in both two-class, unpaired data analyses. It showed 67 differentially regulated probes with a False Discovery Rate (FDR) equal to 0.91% for the two-class, unpaired data analysis of the set without filtering. For the filtered set, two-class, unpaired data analysis revealed 49 differentially regulated probes with a False Discovery Rate (FDR) equal to 1.22%. The filtered set shares 35 probes with the set without filtering.

In both multiclass response analyses a delta value of 0.42 was used. SAM showed 19 differentially regulated probes with a False Discovery Rate (FDR) equal to 3.88% for the multiclass response analysis of the set without filtering. For the filtered set, multiclass response analysis revealed 12 differentially regulated probes with a False Discovery Rate (FDR) equal to 5.80%. The filtered set shares 9 probes with the set without filtering.

The differentially regulated probes revealed by the two-class, unpaired data analysis for the unfiltered set contain known prostate markers such as PSA, KLK-1 and RPL17 and allow separation of the samples according to their GS. The hierarchical clustering presented in figure 1 shows how well the probes retrieved from the filtered set can separate the samples according to their GS, where the samples with higher GS are in a separate branch from the samples with lower GS. The hierarchical clustering of figure 2 shows how this set of probes classifies the GS 7-8 samples, showing that the latter set of samples exhibits a heterogeneous expression profile and that the higher GS samples still cluster in a separate branch.

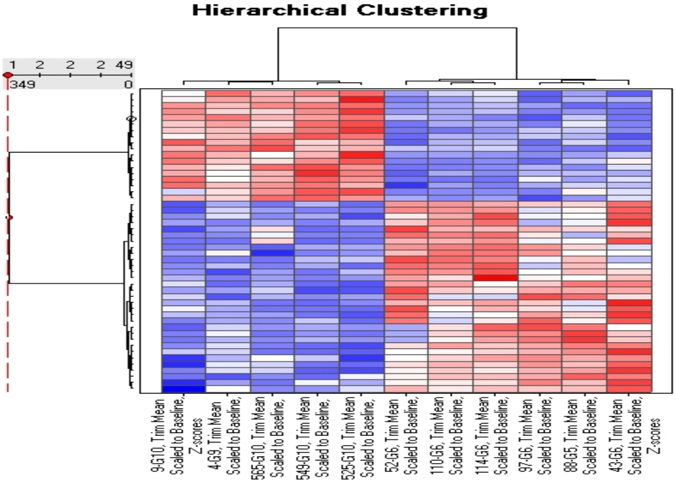


Fig. 1. Hierarchical Cluster using Ward´s method with average value as ordering function for the transcripts obtained with two-class, unpaired data analysis with SAM for the filtered set. SAM analysis was performed using only the samples with GS 5-6 and GS 9-10. Each line represents a transcript, each column represents a sample and the acronym under each column refers to the sample number – GS

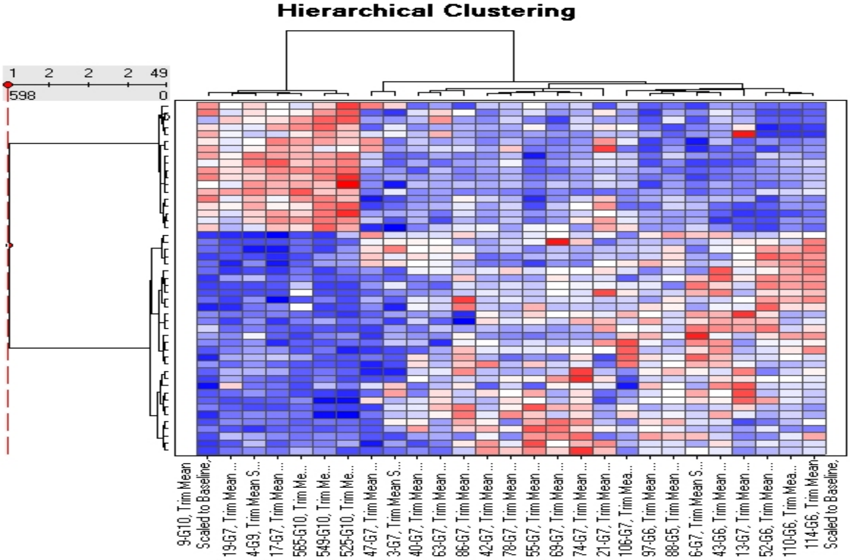


Fig. 2. Hierarchical Cluster using Ward´s method with average value as ordering function. The transcripts selected by SAM analysis with two-class unpaired data for the filtered set, obtained in Fig. 1, were used as a classifier that was applied to all samples with GS 5-6, GS 7-8 and GS 9-10. Each line represents a transcript, each column represents a sample and the acronym under each column refers to the sample number – GS

The result of the analyses reported here share 10 differentially regulated probes with previously reported results [6] where Pearson correlation and SOM were used in the same normalized dataset, suggesting that these are the most robust candidate markers. Among these probes we highlight RAB3B, a member of RAS oncogene family.

Even though no clustering evaluation techniques have yet been applied to the current analysis, the transcripts revealed by SAM allowed a categorization of the samples and are consistent with previous analysis [6]. The identification of additional sets of correlated genes in the current analysis indicates that complementary statistical approaches are required to allow the full description of the transcriptional profile of prostate cancer at different stages of tumor progression.

As a next development of our work we intend to fully automate the statistical analysis of our microarray data and integrate the analysis procedures with the BASE database [8]. This will certainly improve the set of analysis tools available for the BASE database.

References

1. Causton, H.C., Quackenbush, J., Brazma, A.:Microarray Gene Expression Data Analysis: A beginner´s guide. Blackwell Science Ltd, Oxford (2003)
2. van 't Veer L.J., Dai H., van de Vijver M.J., He Y.D., Hart A.A., Mao M., Peterse H.L., van der Kooy K., Marton M.J., Witteveen A.T., Schreiber G.J., Kerkhoven R.M., Roberts C., Linsley P.S., Bernards R., Friend S.H.: Gene expression profiling predicts clinical outcome of breast cancer. Nature Vol. 415(2002):530-536
3. Misra J., Schmitt W., Hwang D., Hsiao L.L., Gullans S., Stephanopoulos G., Stephanopoulos G.: Interactive exploration of microarray gene expression patterns in a reduced dimensional space. Genome Research 12(2002):1112-20.
4. Tusher, V., Tibshirani, R., Chu, C.: Significance analysis of microarrays applied to ionizing radiation response. Proceedings of the National Academy of Sciences Vol. 98(2001):5116-5121
5. Wu, B.: Differential Gene Expression Detection Using Penalized Linear Regression Models: the Improved SAM Statistics. Bioinformatics. 21(2005) :1565-71
6. Reis E.M., Nakaya H.I., Louro R., Canavez F.C., Flatschart A.V., Almeida G.T., Egidio C.M., Paquola A.C., Machado A.A., Festa F., Yamamoto D., Alvarenga R., da Silva C.C., Brito G.C., Simon S.D., Moreira-Filho C.A., Leite K.R., Camara-Lopes L.H., Campos F.S., Gimba E., Vignal G.M., El-Dorry H., Sogayar M.C., Barcinski M.A., da Silva A.M., Verjovski-Almeida S.: Antisense intronic non-coding RNA levels correlate to the degree of tumor differentiation in prostate cancer. Oncogene 23(2004) 6684-6692.
7. El-Hamdouchi A., and Willett P.: Hierarchical document clustering using Ward's method. In Proceedings of the 9th International ACM SIGIR Conference on Research and Development in Information Retrieval (1986) 149-156.
8. Saal L.H., Troein C., Vallon-Christersson J., Gruvberger S., Borg A., Peterson C.: BioArray Software Environment (BASE): a platform for comprehensive management and analysis of microarray data. Genome Biology. 15 (2002): SOFTWARE0003 1-8.

New EST Trimming Strategy

Christian Baudet[1,2] and Zanoni Dias[1,2]

[1] Institute of Computing - Unicamp - Campinas - SP - Brazil
{christian.baudet, zanoni}@ic.unicamp.br
[2] Scylla Bioinformatics - Campinas - SP - Brasil
{christian, zanoni}@scylla.com.br

Abstract. Trimming procedures are an important part of the sequence analysis pipeline in an EST Sequencing Project. In general, trimming is done in several phases, each one detecting and removing some kind of undesirable artifact, such as low quality sequence, vectors or adapters, and contamination. However, this strategy often results in a phase being unable to recognize its target because part of it was removed during a previous phase. To remedy this drawback, we propose a new strategy, where each phase detects but does not remove its target, leaving this decision to a post processing step occurring after all phases. Our tests show that this strategy can significantly improve the detection of artifacts.

1 Introduction

EST Sequencing Projects are developed with the objective of quickly obtain the gene index of an organism.

An EST sequence may contain unwanted regions made up of ribosomal RNA, low quality ends, poly-A/T fragments, vector and adapter fragments and slipped fragments. In this work, we denote these regions as artifacts.

Therefore, EST projects must submit their sequences to the sequence trimming processes before analyzing them. Some projects make use of specific trimming softwares as ESTprep [5] or LUCY [3]. The latter is used by TIGR - The Institute of Genomic Research.

A trimming process is a set of procedures that has the goal of removing regions of low quality and subsequences that do not belong to the project target organism. This cleaning process must be performed because these subsequences can add errors to the data analysis [6]. For example, a simple adapter sequence can determine whether two sequences will be clustered together or not in a clustering process.

Usually, each sequencing project executes its own sequence trimming protocol. Some projects perform a complete analysis, while others execute only low quality and vector trimming. This difference between the protocols compromises the comparison between sequences produced by different projects.

This work studies the difference between alternative trimming methods, with the objective of creating a new set of procedures to improve the detection and removal of unwanted sequences.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 206–209, 2005.

2 EST Trimming

Removal of artifacts can be done in many different ways, which can be simple or more elaborated. The simplest solutions are usually fastest, an important factor when the data volume that must be processed is high.

For example, a simple strategy to trimming low quality regions is the execution of an algorithm to determine a subsequence of maximum sum as implemented by phred [4]. Each sequence processed by this algorithm has its low quality extremities removed thus the remaining sequence has the the minimum probability error.

Many projects perform the quality analysis through sliding windows. Usually, the sequence is covered base by base in both directions, from the extremities, with a window that has a fixed size. The sliding window method searches for regions that have at most a certain number of bases which qualities values lower than a minimum value.

Other example is the vector and adapter removal. A simple way of performing this task is using a software like cross_match [4] to mask unwanted regions. Regions that have good score alignment with vector or adapter sequences are masked with Xs. Thus, analyzing the X regions is possible to identify the vector and adapter artifacts. Telles and Silva [6] make use of this technique and, additionally, classify the vector neighborhood into seven different classes. More complex solutions, such as the implemented by LUCY, search in an adaptive way, guided by the quality of the analyzed region.

In this work we implemented two sets of procedures. The first set implemented was based on our interpretation of the procedure described in the work of Telles and Silva [6].

The second set, called *New Schema*, was also based in the set proposed by Telles and Silva, but it has differences in sequence treatment. The major difference is in the input sequence of each step. In their set of procedures, all identified artifacts are removed before the sequence is submitted to the next step. In the new proposed set, the complete sequence is analyzed in each phase. The initial idea is to simplify detection methods for further refining of the techniques that did not show good results.

The motivation for this strategy was the observation of sequences processed by the procedures described by Telles and Silva. We observed that an artifact could be detected or not influenced by the detection or not of other artifact in a previous step. For example, if a vector is not identified, the detection of a poly-A/T tail cannot occur because of the proximity criterion required by the method.

Another point that we observed was the omission of artifacts that overlap with other artifacts. For example, certain adapters have an extremity that overlaps the extremity of the vector where they are inserted. In this case, when a vector is identified and discarded, the remaining adapter sequence is not detected because its size becomes too small for the identification criterion.

The first step of the New Schema is ribosomal RNA detection and it is exactly the same as in the trimming set of Telles and Silva.

The second step is low quality trimming that is performed with a maximum subsequence algorithm similar to the one implemented in the software phred.

Vector detection is the next step. Detection is made with cross_match using 12 for minimum match and 20 for minimum score. Any subsequence identified is marked as vector. This strategy simplifies vector detection and obviates classification of the neighborhood of vector regions. In our method, the whole sequence is analyzed and this can result in more vector regions. This sequence fragmentation does not represent a problem because the last step of trimming preserves only the longest subsequence that does not contain an artifact.

After vector detection, the sequence is searched for adapters that were used in the sequence cloning process. Here, the criterion is to mark as adapter all regions that have an alignment with score greater than or equal to the size of the adapter minus four bases.

The following step is poly-A/T tail detection. The solution proposed is to perform an alignment of the target sequence with a long chain of 200 As or Ts using the software swat. All regions that have an alignment with a minimum score of 10 are marked as a poly-A or poly-T region.

The last step of the trimming procedure set implemented for the New Schema is the identification of all maximum subsequences that do not contain any artifact and have at least 100 bases. In addition, the subsequence must have at least 50 bases with quality greater than 20. Only the longest subsequence will be preserved. If there are several subsequences that meet the two criteria above, the method will choose the one with greater quality sum.

3 Comparative Analysis

The tests necessary to validate the method developed in this work were made with the sequences available in the Cattle EST Project site [2]. The sequences are extracted from placentas of *Bos taurus* individuals. A total of 12620 sequences, distributed in 174 96-well plates, were obtained. These sequences are the ones that had not been discarded by the trimming process implemented by the project.

To perform the ribosomal contamination detection we constructed a repository of ribosomal sequences of mammals. The choice for mammal sequences is appropriate because *Bos taurus* is a mammal and ribosomal RNA is highly conserved across the species.

Since all sequences had already been processed by the Cattle EST project, we expected that identification of rRNA sequences would not occur. However, the procedures implemented discarded 100 sequences.

Low quality trimming did not discard any sequence. This result was expected because of the observation that the trimming process performed by the Cattle EST project had discarded 24.5% of the sequences.

Both trimming procedure sets identified vector regions in 12461 sequences (99.5% of 12520 sequences preserved by the ribosomal detection phase).

The most dramatic difference between the two sets was shown by the adapter detection methods. The solution described by Telles and Silva found adapter regions in 91 sequences (0.7%), while our method detected them in 12311 sequences (98.3%). In their method, the vector is removed before the detection of the adapter, which, in this case, has a 6-base overlap with the vector sequence. Therefore, the remaining adapter could not be detected because its too short.

Our schema was also capable to detect more poly-A/T tails. It detected poly-A tails in 1957 sequences (15.5%), while Telles and Silva's method detected them in 658 sequences (5.3%). Our method found poly-T tails in 955 sequences (7.6%), while Telles and Silva's method found them in 718 sequences (5.7%).

4 Conclusion

Our study evidences that the possibility of improvement in the trimming procedures is real. The New Schema proposed shows that the strategy of performing the detection of the artifacts individually, without constructing relationships among the different types of artifacts, can produce good results.

The next steps of our work are to refine the methods developed and to create new methods for trimming of slipped sequences. We will also work with procedures of contamination detection.

To make better comparisons, we obtained the sequences of the Sugarcane EST project to use in the next validation tests. We intend to make clustering of the sequences trimmed by our set of procedures and compare with Telles and Silva's results.

Supplementary material to this work can be found in the technical report [1] available at http://www.ic.unicamp.br/ic-tr/.

References

1. C. Baudet and Z. Dias. New EST trimming strategy. Technical Report IC-05-09, Institute of Computing - University of Campinas, May 2005.
2. Cattle EST Project - The W. M. Keck Center for Comparative and Functional Genomics, University of Illinois at Urbana-Champaign, January 2005. http://titan.biotec.uiuc.edu/cattle/cattle_project.htm.
3. H. Chou and M. H. Holmes. DNA sequence quality trimming and vector removal. *Bioinformatics*, 17:1093–1104, 2001.
4. P. Green. Phrap Homepage: phred, phrap, consed, swat, cross_match and RepeatMasker Documentation, March 2004. http://www.phrap.org.
5. T. E. Scheetz, N. Trivedi, C. A. Roberts, T. Kucaba, B. Berger, N. L. Robinson, C. L. Birkett, A. J. Gavin, B. O'Leary, T. A. Braun, M. F. Bonaldo, H. P. Robinson, V. C. Sheffield, M. B. Soares, and T. L. Casavant. ESTprep: preprocessing cDNA sequence. *Bioinformatics*, 19(11):1318–1324, November 2003.
6. G. P. Telles and F. R. da Silva. Trimming and clustering sugarcane ESTs. *Genetics and Molecular Biology*, 24(1-4):17–23, December 2001.

A Modification of the Landau-Vishkin Algorithm Computing Longest Common Extensions via Suffix Arrays

Rodrigo de Castro Miranda[1] and Mauricio Ayala-Rincón[1,2]

[1] Programa de Mestrado em Informática
[2] Departamento de Matemática, Universidade de Brasília, Brasil
rodrigo.miranda@acm.org, ayala@mat.unb.br

Abstract. Landau and Vishkin developed an $O(kn)$ algorithm for the approximate string matching problem, where k is the maximum number of admissible errors and n the length of the text. This algorithm uses suffix trees for an $O(1)$ running time computation of the longest common extensions between strings. We present a variation of this algorithm which uses suffix arrays for computing the longest common extensions.

1 Introduction

Matching strings with errors is an important problem in Computer Science, with applications that range from word processing to text databases and biological sequence alignment. Landau and Vishkin [7] developed an $O(kn)$ algorithm for matching a pattern to a string of length n with at most k differences. The algorithm iterates through the diagonals of the table of the Smith-Waterman dynamic programming classical solution and uses a suffix tree for constant-time jumps along the diagonals, bypassing character-by-character matching. We present a variation of the Landau-Vishkin algorithm which instead of suffix trees uses suffix arrays enhanced with a table of longest common prefixes [8] for computing the $O(1)$ jumps along the diagonals of the dynamic programming table. The space usage of the proposed modification of the Landau-Vishkin algorithm is better, since suffix arrays use less space than suffix trees.

2 Problem Definition and the Landau-Vishkin Algorithm

Given two strings $T = t_1...t_n$ and $P = p_1...p_m$, $m \leq n$ over an alphabet Σ, we say that P is the i-th *suffix* of T, denoted by T_i, if $p_1 \dots p_m = t_i \dots t_n$ such that $1 \leq i \leq n$ and $m+i-1 = n$. We say that P is a *prefix* of T if $p_1 \dots p_m = t_1 \dots t_m$ such that $m \leq n$. The *longest common prefix* of P and T is the longest string that is a prefix of both P and T. The *longest common extension* (LCE) of P and T is the length of their longest common prefix. The *edit distance* between two strings is defined as the minimum number of rewriting steps of *elimination*, *insertion* or *substitution* of symbols needed to transform one string into the other. The

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 210–213, 2005.

approximate string matching problem with k differences between a pattern string P and a text string T is the problem of finding every pair of positions (i, j) in T such that the edit distance between P and $t_i...t_j$ is at most k.

The Landau-Vishkin Algorithm. A brief overview of the Landau-Vishkin algorithm is presented. For a detailed explanation we refer the reader to [3, 7].

The algorithm works by building paths in the dynamic programming table of the classical Smith-Waterman solution. A path is said to contain an error at cell (i, j) if $p_i \neq t_j$ or if the transition from cell (i, j) to cell $(i + 1, j)$ or to cell $(i, j + 1)$ is in the path. A path with d errors is called a *d-path*. The algorithm iterates through each diagonal i and finds the d-paths from the $(d - 1)$-paths that end at diagonals $i - 1$, i and $i + 1$. When $d = k$, the algorithm is finished and every k-path that reached row m is a match of P in T with at most k errors.

The algorithm iterates k times through $O(n)$ diagonals. It runs in time $O(kn)$ because the extension of a d-path from the three paths of the previous iteration is done in $O(1)$ by using an $O(1)$ LCE query which is based on an $O(1)$ *lowest-common-ancestor* (LCA) query over a suffix tree. Also it runs in space $O(kn)$ because it is unnecessary to represent the whole dynamic table: in fact, a d-path is represented in $O(d)$ space.

Suffix Trees and Their Use for Computing LCAs. A suffix tree $\mathcal{T}$ for a string $T = t_1...t_n$ over an alphabet Σ is a rooted tree that has interesting properties for string matching applications. We refer the reader to [9, 3] for further explanation. A suffix tree can be built using $O(n)$ running-time and space complexity.

Given a rooted tree $\mathcal{T}$, an ancestor of a node v is a node which is on the unique path from v to the root. The LCA of two nodes x and y is the deepest node in $\mathcal{T}$ that is an ancestor of both x and y.

In a suffix tree $\mathcal{T}$ for the string T, given any two leaves i and j of $\mathcal{T}$, corresponding to the suffixes T_i and T_j of T, the LCA of i and j gives us the longest common prefix of T_i and T_j.

A rooted tree with n nodes may be pre-processed in time and space $O(n)$ in order to allow $O(1)$ LCA queries, as described in [3, 1].

3 Modification of the Landau-Vishkin Algorithm

Our proposal is to substitute the use of suffix trees on Landau-Vishkin algorithm with the use of suffix arrays for computing the longest common prefixes.

3.1 Suffix Arrays

A suffix array *Pos* for a string T is an array which gives us a lexicographically ordered sequence of the suffixes of T [8]. It can be constructed from T in time $O(n)$ and uses $O(n)$ space as seen in [4, 5]. An *enhanced suffix array* is a suffix array augmented with a LCP table. The LCP table is the array *lcp* of n elements such that $lcp[i]$ is the LCE of the suffixes $Pos[i]$ and $Pos[i+1]$. The *lcp* array can be constructed in linear time from Pos.

3.2 Longest Common Extension Computation Using Suffix Arrays

Given an enhanced suffix array for the string $P\#T$[1], we can pre-process the *lcp* array in $O(n)$ time and answer LCE queries in $O(1)$ time.

The LCE between two suffixes S_a and S_b of S can be obtained from the *lcp* array in the following way: given i and j, $i < j$ such that $Pos[i] = a$ and $Pos[j] = b$, then the LCE of S_a and S_b is $lcp(i,j) = \min_{i \leq k < j} lcp[k]$. Thus LCE queries can be answered by a *range-minimum-query* (RMQ) over a range in *lcp*. As it turns out, it is possible to pre-process an array of integers (such as *lcp*) in $O(n)$ so that a RMQ in a given interval of the array is answered in $O(1)$. The idea presented below follows the algorithm based on Cartesian trees given in [2].

A *Cartesian tree* $\mathcal{C}$ for a sequence of real numbers $x_1 \ldots x_n$ is defined as a binary tree with nodes labeled by those numbers, such that the root is labeled by x_m where $x_m = \min\{x_i \mid 1 \leq i \leq n\}$, the left subtree is the Cartesian tree for $x_1 ... x_{m-1}$ and its right subtree is the Cartesian tree for $x_{m+1} \ldots x_n$.

Given a Cartesian tree $\mathcal{C}$ for the array x, a RMQ of the interval $x_i \ldots x_j$ can then be found by simply finding the LCA of nodes i and j of $\mathcal{C}$, which can be done in $O(1)$ after $O(n)$ pre-processing (see section 2). The Cartesian tree can be built in $O(n)$ using the algorithm given in [2].

Thus, in order to answer LCE in $O(1)$ with $O(n)$ pre-processing, we first build an enhanced suffix array in $O(n)$ time and space for the string $T\#P$. We then create a Cartesian tree $\mathcal{C}$ for the LCP table, and pre-process it in $O(n)$ so that we can query the LCA of any two nodes of $\mathcal{C}$ in $O(1)$. Given the suffixes i and j of $P\#T$, their LCE will be the result of a RMQ over $lcp_i \ldots lcp_j$, which is given by an $O(1)$ LCA query over $\mathcal{C}$.

3.3 Proposed Algorithm

The proposed algorithm is then the same Landau-Vishkin one substituting the suffix tree for an enhanced suffix array, and the LCA queries over the suffix tree for a RMQ over the LCP table:

1. Build the enhanced suffix array for the string $P\#T$.
2. Pre-process the *lcp* array so that we can answer RMQ in $O(1)$
3. For d from 1 to k iterate through every diagonal i of the dynamic programming table:
   - 3.1. Using the $O(1)$ RMQ extend the $(d-1)$-paths that end at diagonals $i-1$, i and $i+1$ to a d-path ending in diagonal i.
   - 3.2. Choose the d-path that ends at the cell that has the largest column index.

[1] '#' is a *sentinel* character — i.e., a character which is neither in P nor in T.

The modified algorithm runs in time and space $O(kn)$ as well. The construction and maintenance of suffix arrays is done in $O(n)$ time and space as well as the building and maintenance of Cartesian trees. Since after $O(n)$ pre-processing LCA queries are done in $O(1)$, the construction of the approximate matches is computed in $O(kn)$ running time and space.

Although the theoretical bounds coincide with the original algorithm, in practice we are able to use less space during pre-processing than with suffix trees. Supposing we are dealing with a good implementation of suffix trees where we use about 12 bytes per character (see [6]), constructed with $\frac{3}{2}n$ nodes. The space used for the LCA pre-processing is then $12n + A\frac{3}{2}n$ bytes, where A is the space used by the LCA pre-processing for each node of the suffix tree. The suffix array and the lcp array can be built with a total of $8n$ bytes. If one builds the labeled Cartesian tree $\mathcal{C}$ using $12n$ bytes such that the labels of its nodes are actual values of lcp, instead of indexes, we can discard the suffix array and the lcp array altogether. Since the $\mathcal{C}$ has exactly n nodes, the final amount of space used for the preprocessing is then $12n+An$ bytes, which is an economy of $\frac{A}{2}n$ bytes over the suffix tree-based version.

4 Concluding Remarks

We have shown that it is possible to change the Landau-Vishkin approximate string matching algorithm to use suffix arrays instead of suffix trees for its computation of longest common extensions between suffixes of the text and the pattern, while keeping the same running time and space complexity. Due to the use of suffix arrays and companion data structures actual space usage is likely to be better than the standard version.

References

1. M. Bender and M. Farach-Colton. The LCA Problem Revisited. In *LATIN 2000*, volume 1776 of *LNCS*, pages 88–94. Springer, 2000.
2. H. N. Gabow, J. L. Bentley, and R. E. Tarjan. Scaling and related techniques for geometry problems. In 16^{th} *ACM STOC*, pages 135–143, 1984.
3. D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.
4. J. Kärkkäinen and P. Sanders. Simpler linear work suffix array construction. In *Int. Colloquium on Automata, Languages and Programming*, volume 2719 of *LNCS*, pages 943–955. Springer, 2003.
5. P. Ko and S. Aluru. Space-Efficient Linear Time Construction of Suffix Arrays. *Journal of Discrete Algorithms*, to appear.
6. S. Kurtz. Reducing the space requirement of suffix trees. *Softw., Pract. Exper.*, 29(13):1149–1171, 1999.
7. G. Landau and U. Vishkin. Introducing Efficient Parallelism into Approximate String Matching and a new Serial Algorithm. In 18^{th} *ACM STOC*, pages 220–230, 1986.
8. U. Manber and G. Myers. Suffix arrays: A new method for on-line string searches. TR 89-14, University of Arizona, 1989. (SIAM J. Comput., 22(5):935–948, 1993.)
9. E. Ukkonen. On-line Construction of Suffix-Trees. *Algorithmica*, 14:249–260, 1995.

The BioPAUÁ Project: A Portal for Molecular Dynamics Using Grid Environment

Alan Wilter[1], Carla Osthoff[1], Cristiane Oliveira[1], Diego E.B. Gomes[2],
Eduardo Hill[2], Laurent E. Dardenne[1], Patrícia M. Barros[1],
Pedro A.A.G.L. Loureiro[2], Reynaldo Novaes[3], and Pedro G. Pascutti[2]

[1] LNCC, Laboratório Nacional de Computação Científica,
Petrópolis, Brazil
{alan, osthoff, cris, dardenne, patricia}@lncc.br
http://www.lncc.br/
[2] IBCCF, Instituto de Biofísica Carlos Chagas Filho,UFRJ,
Rio de Janeiro, Brazil
{diego, hill, ploureiro, pascutti}@biof.ufrj.br
http://www.biof.ufrj.br
[3] HP-Brasil, Hewllet Packart Brasil,
Porto Alegre, Brazil
reynaldo.novaes@hp.com

Abstract. This paper describes BioPAUÁ Project, a new portal for Molecular Dynamics (MD) simulations over a computational grid environment. It integrates MD simulations and analyses tools with grid technologies to provide support to biomolecular *in silico* experiments. The objective of BioPAUÁ Project is to offer a tool, as well the facility, for researches working in several important fields (*e.g.,* bioinformatics, structural biology, biochemistry, medicinal chemistry, biopharmacology). At first, the possible user does not need any special skill in MD simulations, however, advanced ones are also well contemplated. The project methodology is based on MYGRID middleware and uses GROMACS package in order to run simulations. This work is developed by LNCC/MCT, with IBCCF/UFRJ collaboration, and supported by HP Brazil R&D.

1 Introduction

There are several projects in bioinformatics [1,2] with similar aims to the ones proposed by BioPAUÁ, but no one, as far as we concern, poses the special features presented here. BioPAUÁ hosts Molecular Dynamics (MD) tools and it is able to perform relatively complex computer simulations of proteins in a computational grid environment, all that made available through a web portal at *http://www.biopaua.lncc.br*.

The development of this Portal places numerous computing challenges as well as bioinformatics modelling issues. Since it is the first release of a still beta version, it has several designed (and desired) missing options plus some unwilling occasional service failures. Nevertheless, it is operational, given that we were able to accomplish some results depicted in works [3,4,5] with the help of the Portal.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 214 – 217, 2005.

In this paper, we describe, not in deep, the BioPAUÁ portal, which unites MD tools and Grid technologies with the intention of that their combined effect may support experimental science. The goal of BioPAUÁ Project is to offer a tool, as well the facility, for researches working in several important fields like bioinformatics, structural biology, biochemistry, medicinal chemistry and biopharmacology.

In simple words, the Portal accepts jobs of MD, as defined by the user (novice or expert), and will spread them over a grid of computers.

2 BioPAUÁ Portal Description

The BioPAUÁ portal provides submission and execution in MD services, based on GROMACS package [6], through a grid computing platform. However, it is not possible to enclose all options of GROMACS utilisation, which is also able to do, for instance, normal mode analysis, free energy calculations and so on. Even in MD and optimisation protocols that are offered, we have to restrict the options based on our experience, in order to facilitate the introducing, for the user, in the MD techniques field. Nevertheless, the expert in GROMACS is able to use our grid facility via Portal, as described bellow. In this way, any sort of user is eligible to exploit BioPAUÁ.

The user will be able to access Portal services in the following manner:

1. By submitting any PDB file, containing amino acids residues, with or without a topology file for ligand (in ITP file GROMACS format). Any residue or molecule group not identified automatically by GROMACS and without its respective topology, provided by the user, will be neglected for simulation.
2. By submitting any PDB file, as long as it has its residues identification matching the ones recognised by GROMACS (like nucleotides bases, some saccharides and lipids), or with its respective topology file in ITP format.
3. By submitting just a TPR file, previously designed by the GROMACS expert user. In this mode, likely any GROMACS application is virtually feasible.

To overcome the lack of topologies, users can access Dundee PRODRG2 Server [7] (*http://davapc1.bioch.dundee.ac.uk/programs/prodrg/*), which builds particular topologies, whose parameters are not inserted in GROMACS topology database. Nevertheless, user must be aware about likely topologies issues.

Users can monitor their jobs by cancelling, removing, extending, and eventually, downloading results files, with logs, trajectories, energies and some common simple analyses. The sequence of GROMACS commands executed is also available to users, hoping that this information may help them to acquire some skill in GROMACS.

3 Technical Information

Behind the portal, the first front-end facility is a beowulf cluster based on common PCs supplied by HP Brazil, running Debian GNU/Linux with a kernel ready for OpenMosix (*http://www.openmosix.org*), not activated, but also operational. All internal networks are 100 Mbps Fast Ethernet. Then, such cluster is connected to others ones around Brazil, from the Northeast to the South states, via MYGRID/OURGRID [8] over Internet, using SSH protocol, the only security measure adopted by now. The grid community is known as PAUÁ Network.

The whole process runs as follow:

1. User accesses the BioPAUÁ Portal and submits a MD job to server, over an encrypted connection. The server hosts the web service, MYGRID and all files needs by system, where every user has a special folder in hard disk and his/her activities are monitored by MYSQL database. Once started, user can only wait and has to monitor job status at Portal.
2. The server pre-process the files and instructions supplied by user and, via MYGRID, sends the user job to an idle machine, which can be local or remote, on OURGRID community. The first step is to check which computer system is being accessing. By now only machine with GNU/Linux are acceptable. Second, it checks for GROMACS. If it is not already there, then a previously built binary package of GROMACS will be sent. This operation is done only once for each machine, during its first time access.
3. After job finalization, output files are sent back to the server, including graphical analyses files in XMGRACE format, and stored on the specific job folder. So, user will be able to download files.

OURGRID / MYGRID Project
The OURGRID Project (*http://www.ourgrid.org*) [8] is a collaborative effort involving Hewlett-Packard (HP) and Federal University of Campina Grande (UFCG) to research and develop solutions of usage and management of computational grids. MYGRID is the user broker when dealing with the grid. The OURGRID Community is responsible for assembling grid to be used by MYGRID instances.

GROMACS Software Package
GROMACS [6] (*http://www.gromacs.org*) is a programme to solve Newtonian equations of motion for systems up to millions of atoms, in a extremely high performance. It also comes with a large selection of flexible tools for trajectory analysis and it is ready for visualization with graphical tools. In addition, GROMACS is free software, available under the GNU General Public License.

PAUÁ Network Project
PAUÁ, which means "everything" in Tupi-Guarani, is an initiative created by HP Brazil R&D to build a countrywide Brazilian Grid. PAUÁ currently involves 11 different universities and research centres is a 250-node grid that supports the execution of Bag-of-Tasks (BoT) applications whose tasks are independent.

4 Final Remarks

The solution proposed here seems to show potential since it intends to facilitate the use of MD techniques employing frequent idle computers in a grid. In spite of this, there are some drawbacks. The most important problems are related to the computer time that simulations take to be executed, as well as the size of output files which must be transferred over Internet. Since simulations can last for days to weeks, the remote compute where job is being executed can present failures. To cope with that, MYGRID/OURGRID developers are working on a checkpoint tool to restore the last saved state of a job. About the problem of file transference, not much has been done, except that PAUÁ Network will have some links upgraded to Gigabit Ethernet system.

Despite the mentioned problems, we got some interesting results using the Portal facilities, even while in development; we can cite three here. *i*) Derived from some MD simulations carried out via Portal, the most important result that Batista et al. [3] could achieve was the decline in binding affinity for inhibitor relative to non-B subtypes when compared to subtype B, in accordance with some previous experimental results, which can, in due course, favour the emergence of drug resistance. *ii*) Studying falcipain-2 complexed with E64 and Z-LR-AMC by MD, at BioPAUÁ, Gomes et al. [4] have proposed some line of directions to structure-based design of inhibitors for this protease. *iii*) França et al. [5] used the Portal services in their work to perform some molecular dynamics simulations. MD results combined with docking studies were employed to propose structures and to study the dynamic behavior in the active site of nine potential lead compounds as selective inhibitors of *Plasmodium falciparum* Serine Hydroxymethyltransferase.

For future work we are improving the first version to offer more options of simulation and analyses. We are also investigating how to add new Molecular Biology and Pharmacology applications, like high-throughput drug screening.

References

1. Rojnuckarin, A., Livesay, D. R., Subramaniam, S.: Biomolecular Reaction Simulation Using Weighted Ensemble Brownian Dynamics and the University of Houston Brownian Dynamics Program. Biophysical Journal 79 (2000) 686-693
2. Keahey, K., Papka, M. E., Peng, Q., Schissel, D., Abla, G., Araki, T., Burruss, J., Feibush, E., Lane, P., Klasky, S., Leggett, T., McCune, D., Randerson, L.: Grids for Experimental Science: The Virtual Control Room, in proceedings of the Challenges of Large Applications in Distributed Environments (CLADE), Honolulu, HI (June 2004)
3. Batista, P., Wilter, A. Durham, E. H. A. B., Pascutti, P. G.: Molecular Dynamics Simulations Applied to the Study of Subtypes of HIV 1 Protease Common to Brazil, Africa and Asia. Journal Cell Biochemistry and Biophysics (2005) in press
4. Gomes, D. E. B., Rössle, S. C. S., Bisch, P. M., Pascutti, P. G.: Molecular Modeling and Dynamics of Falcipain-2 Protease Complexes, a Contribution for Drug Development against Malaria. Submitted to Biophysical Chemistry (2005)
5. França, T. C. C., Wilter, A., Ramalho, T. C., Pascutti, P. G., Figueroa-Villar, J. D.: Molecular Dynamics of the Interaction of Plasmodium falciparum and Human Serine Hydroxymethyltransferase with 5-Formyl-6-hydrofolic Acid Analogues: Design of New Potential Antimalarials. Submitted to Journal of Computer-Aided Molecular Design (2005)
6. van der Spoel, D., Lindahl, E., Hess, B., van Buuren, A. R., Apol, E., Meulenhoff, P. J., Tieleman, D. P., Sijbers, A. L. T. M., Feenstra, K. A., van Drunen, R., Berendsen, H. J. C.: Gromacs User Manual Version 3.2, www.gromacs.org (2004)
7. van Aalten, D.M., Bywater, R., Findlay, J.B., Hendlich, M., Hooft, R.W., Vriend, G.: PRODRG, a Program for Generating Molecular Topologies and Unique Molecular Descriptors from Coordinates of Small Molecules. J. Comput. Aided Mol. Des. 10 (1996) 255-262
8. Andrade, N., Costa, L., Germoglio, G., Cirne, W.: Peer-to-peer Grid Computing with the OurGrid Community. Proceedings of the SBRC (2005)

Analysis of Structure Prediction Tools in Mutated MeCP-2

Dino Franklin and Ivan da Silva Sendin

Universidade Federal de Goiás - Campus Avançado de Catalão
{dino, sendin}@catalao.ufg.br

Abstract. Methyl-CpG-binding protein 2 (MeCP2) belongs to the DNA-binding protein family that selectively binds to DNA methylated CpG-islands. MeCP2 acts like a transcriptional repressor, that contains a N-terminal methylated DNA-binding domain (MBD), and a C-terminal transcriptional repression domain (TRD). Mutations in MECP2 gene have been associated to Rett Syndrome - a neurological disorder linked to X-chromossome, and one of the most common causes of physical and intellectual dysfunction in females. The calculation of MeCP2 MDB had been solved, but the effects of the mutations on the protein's structure and, consequently, functions have not been analyzed. Databases, systems, tools, and, more recently, protein structure motifs databases available on Internet make it possible to predict ab initio protein structure quickly. This extended abstract looks at the the use of these tools to analyze the effects of MeCP2's mutations, which cause Rett syndrome, in the original protein structure.

1 Introduction

Rett syndrome is a neurological disorder first described by Andreas Rett in 1966 [1] in a article written in German. The pathology was only more widely known and understood when Hagberg et al. published a paper in English [2]. Since then, several researches have been performed and, particularly in brain, it was noticed selective fails in connections between neurons, and altered neurotransmitters quantities [3] . Later on an initial apparently normal development, generally up to 6 to 18 months old, the child with Rett syndrome starts a regression period that affects mainly the speech and hand usage [1, 2]. The pathology reaches one of 15,000 born girls [4]. In 1999, Amir et al. [5] mapped Rett syndrome locus to a gene localized in X-chromosome, particularly to Xq28 position, where is localized MeCP2 gene. The link between Rett syndrome and X-chromosome explains the reason why males are rarely affected, though when it happens the consequences are severe because, as only one X-chromosome is activated per cell, all his cells are mutation carriers. It also explains the symptoms variability to girls with the same mutation. MeCP2 gene contains three exons, and a long untranslated 3' region that may have a structural or functional role because it has well conserved homologues [6]. MeCP2 gene codifies a protein (MeCP2) that

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 218–221, 2005.

contains 486 amino acids. MeCP2 contains at least two functional domains: the methyl-cytosine-binding domain (MBD) that contains 85 amino acids; and the transcriptional repression domain (TRD) with 104 amino acids. The MeCP2 MBD were calculated and analyzed by Nuclear Magnetic Resonance (NMR) [7] and its function is to bind to 5-methyl-cytosine residues in CpG dinucleotides in gene promoter regions that are subject to repression after DNA-methylation [8]. The MeCP2 TRD, in its turn, interacts with the histone deacetylase, and with a transcription co-repressor, SIN3A. Together, they cause the transcription repression through the core histone deacetylation.

It has been shown that protein domains are the fundamental units for analysis of structure, function, and evolution protein [9]. The domains present distinct structural conformations and can be considered as building blocks of protein structure. Know protein structure and, consequently, its domains is essential to understand protein functions. This protein modular characteristic has the pro in offering stability to proteins. Besides this, the proteins evolution, by duplications, mutations, and natural selection, allows that large quantity of proteins - specially the eukaryotic extracellular proteins - has multiple domains with distinct functions, though cooperative function. A structural domain can be defined as a protein substructure, composed of an amino acid chain - containing from 40 up to 350 amino acids - able to folding in a compact and stable form. The region of a protein that is in charge for a specific biological function is called functional domain. The functional domain is identified by experiments where it is extracted residues from the amino acids chain up to the minimal polypeptidic chain and which does not present the original function. Although it is not necessary, in most of case the functional and structural domains comprehend the same residues [10].

2 Protein Structure Prediction

Obtaining the protein's three-dimensional structure demands hard work. The process consists of purifying; lyophilizing the protein, up to obtaining/calculating an image by the crystallography X-Ray, or Nuclear Magnetic Resonance (NMR). The protein domain databases available on Internet became essential for improving the prediction time. An efficient method to predict protein structure is necessary, particularly in this post-genomic era, when new proteins have been discovery on a daily basis. At this moment, biological research is restricted by a lack of insight that partly originates from our ability to efficiently manage biological databases, and by the lack of adequate software tools. Most of the systems and tools available are based on comparison - alignment algorithms - e.g., a just discovered protein is compared with proteins (or models of proteins) saved on protein databases. For some predictions these systems have a very good prediction accuracy [11], though the number of databases and different approaches can leave the beginner in this type of research lost.

There are three main alternative approaches for predicting mutated protein structure based on Internet available resources:

- **Ab initio Prediction.** The protein secondary, tertiary structure prediction or its domains, and boundaries domains prediction can be based only on the residues sequence, and its physic-chemical properties. This approach has been improved, and can be the alternative to calculating a new protein structure when one does not find a homologous in databases. The prediction reliability gets up to 70% [12].
- **Homology Modeling.** The protein structure prediction can also be based on comparison with beforehand calculated protein structures databases [10]. The homologous structure - generally saven on PDB format - can be edited and recalculated using an algorithm that tries to optimize the structure of its mutant using some constraints - e.g. strucutural energy.
- **A Combination of Both Approaches.** MeCP2, for example, contains more than one domain. Actually it contains two domains, a MBD and a TRD. The MBD already has been solved and one can use Homology Modeling to predict a structure of the MBD mutated. Since the TRD was not pre-calculated, one can use the first approach to obtain a predicted TRD structure.

3 Results

We evaluate the main tools that are available on Internet to analyze the effects of MeCP2 mutations - that cause Rett syndrome - in protein structure. Since the results of the first approach does not present new significant information, we focus on the two others appraches. The tools evaluation will be based on the taken time to solve the problem, on the accuracy of results, and labour. Finally, The results also can be used as references to analyze of the links between protein mutation, structure, function, and Rett syndrome phenotypes.

References

1. Rett, A.: Uber ein eigenartiges hirnatrophisches Syndrome bei hyperammonamie in Kindsalter (On a unusual brain atrophy syndrome in hyperammonemia in childhood). Wiener Medizische Wochenschrift. **116**(1966) 723–726.
2. Hagber, B., Aicardi, J., Dias, K., Ramos, O.: A progressive syndrome of autism, dementia, ataxia, and loss of purposeful hand use in girls Rett's syndrome: report of 35 cases. Ann. Neurol. **4** (1983) 471–479.
3. Kerr, A.M. et alii.: Mind and brain in Rett disorder. Brain & Development. **23**(2001)S44–S49.
4. Kerr, A.M., Stephenson, J.B.P.: Rett's syndrome in the west of Scotland. Br.Med.J. **291**(1985)579–582.
5. Amir, R.E., Van de Veyver, I.B., Wan, M., Tran, C.Q., Francke, U., Zoghby, H.Y.: Rett syndrome is caused by mutations in X-linked MECP2, encoding methyl-CpG-inding protein2. Nat.Genet. **23**(1999)184–188.
6. Coy, J.F., Sedlacek, Z., Bachner, D., Delhius, H., Poustka, A.: A complex pattern of evolutionary conservation and alternative polyadenylation within the long 3'-untranslated region of the methyl-CpG-binding protein 2 gene (MECP2) suggests a regulatory role in gene expression. Hum. Mol. Genet. **7**(1997)1253–1262.

7. Wakefield, R.I., Smith, B.O., Nan, X., Free, A., Soteriu, A., Uhrin, D. et al.: The solution structure os the domain from MeCp2 that binds to methylated DNA. J. Mol. Biol **291**(1999)1055–1065.
8. Nan, X.S., Meehan, R.R., Bird, A.: Dissection of the methyl-CpG-binding domain from the chrmossomal protein MeCP2. Nucleic Acids Res. **21**(1993)4886–4892.
9. Nelson, D., Cox, M.: Lehninger: Principles of Biochemistry. 4th edition (2004). W.H.Freeman Pub.
10. Kong, L., Ranganathan, S.: Delineation of modular proteins:Domain boundary prediction from sequence information. Briefings in Bioinformatics. **5**(2004)179–192.
11. Murvai, J. et alii.: Prediction of protein functional domains from sequences usin artificial neural networks. Genome Research. **11**(2001)1410–1417.
12. Bystroff, C., Shao, Y.: Fully automated ab initio protein strucuture prediction using I-SITES, HMMMSTR and ROSETA. Bioinformatics. **18**(2002)S54–S61.
13. Sali, A., Blundell, T.L.: Comparative protein modelling by satisfaction of spatial restraints. Protein Structure by Distance Analysis. Ed: H. Bohr and S. Brunak. IOS Press, Amsterdam.(1994), 64–86.
14. Prokop, M., Damborsky, J., Koca, J.: TRITON: in silico construction of protein mutants and prediction of their activities. Bioinformatics. **16**(2000)845–846.

Protein Loop Classification Using Artificial Neural Networks

Armando Vieira[1] and Baldomero Oliva[2]

[1] Physics Dept. ISEP, Rua S. Tome, 4200 Porto, Portugal
asv@isep.ipp.pt
[2] Structural Bioinformatics Laboratory (GRIB) IMIM/Universitat Pompeu Fabra,
C/ Doctor Aiguader, 80, Barcelona-08003, Catalonia, Spain
oliva@imim.es

Abstract. We used Artificial Neural Network for protein loop classification based on the amino acid sequence alone. A new algorithm recently proposed, the Hidden Layer Learning Vector Quantization (HLVQ) was used and its accuracy compared with traditional Multilayer Preceptrons (MLP). The HLVQ algorithm achieved superior accuracy correctly classifying most loops.

1 Introduction

Since the seminal work of Jones [1] on secondary structure prediction that Artificial Neural Networks (ANN) has become a essential tool for protein structure prediction. Neural Networks are connectionist machines that learn by example with very good generalization capabilities. They have been applied with success in protein structure prediction, protein interaction and protein classification [2].

However, one difficulty in applying ANN to bioinformatics is the large dimensionality of the search space. For instance, a sequence of 10 amino acids represents a search space of 20^{10} possibilities and requires a network with 200 inputs. Training such large networks is difficult, requires large datasets of known examples and the risk of overfitting considerable.

To alleviate these difficulties feature extraction techniques should be used. However, these techniques always discharge some information, while some problems are intrinsically high dimensional. In bioinformatics these cases abound, like gene identification or prediction of secondary structure of proteins.

Loop prediction is a good test-bed to the much harder protein-folding problem. Despite improvements in prediction of protein structure [3], modelling the conformation of loop remains a challenging problem [4]. Loops represent an important part of the protein structure often determining the functional specificity of the protein. The conformation of a polypeptide chain forming the loop has to be calculated from the sequence of the segment while flanking regions and the structure of the rest of the protein may also influence the loop conformation.

Hidden Layer Learning Vector Quantization is an algorithm developed to work with high dimensional datasets [5]. It has the advantage of being relatively simple while less prune to overfitting.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 222 – 225, 2005.

In this work we use HLVQ neural networks to classify protein loops based on amino acids sequence alone using a non-redundant database of proteins with less than 40% sequence identity was taken from SCOP and used to generate the ArchDB [5] database. This database was used to train and test the neural network.

2 The Hidden Layer Learning Vector Quantization (HLVQ)

The Hidden Layer Learning Vector Quantization (HLVQ) [5] is implemented in three steps. First, a multilayer perceptron is trained using back-propagation. Second, supervised Learning Vector Quantization is applied to the outputs of the last hidden layer to obtain the code-vectors $\vec{w}_{ci}$ corresponding to each class c_i in which data are to be classified. Each example, $\vec{x}_i$, is assigned to the class c_k having the smallest Euclidian distance to the respective code-vector:

$$k = \min_j \left\| \vec{w}_{c_j} - \vec{h}(\vec{x}) \right\| \tag{1}$$

where $\vec{h}$ is a vector containing the outputs of the hidden layer. In the third step the perceptron is retrained with two differences. First the error correction is not applied to the output layer but directly to the last hidden layer. The output layer is therefore ignored from now on. The second difference is in the error correction backpropagated to each hidden node:

$$E = \frac{1}{2} \sum_{i=1}^{N_h} \left(\vec{w}_{ck} - \vec{h}(\vec{x}_i) \right)^2 \tag{2}$$

After retraining the MLP a new set of code-vectors, and the process is repeated until convergence is achieved.

3 Application of HLVQ to Protein Loop Classification

Loop prediction can be seen as a mini protein-folding problem. The correct conformation of a given segment of a polypeptide chain has to be calculated from the sequence of the segment influenced by flanking regions that span the loop and by the structure of the rest of the protein that cradles the loop.

We used a classification database of structural motifs, ArchDB. This database contains 12665 clustered loops in 451 structural classes with information about ϕ - φ angles in the loops and 1492 structural subclasses with cover both the ϕ - φ angles and the relative locations of the bracing secondary structures [7].

Loops are classified in fives types according to the bracing secondary structure type: α–α loops, α–β loops, β–α loops and β–β loops that are further split into β–hairpins (which are those loop between two β strands with at least one hydrogen bond between both strands) and β–links (also named β–archs), the complementary set.

In this work we use the information contained in the amino acid sequence to classify the loop type. As a first step, only loops of length five were considered since this is the most representative category. Amino acids from the flanking regions we not considered. This can be considered a test bed to the general problem of classifying loops of any size.

We used the orthogonal coding for the amino acids and tested several networks with different hidden layers. The selected neural network has 100 inputs and 10 hidden nodes. The MLP networks were trained by backpropagation with a learning rate of 0.05 and a momentum of 0.4.

3.1 Prediction of β-β Loops

We first consider a sub-problem: discriminate a loop having a β-β link bracing secondary structure from a loop with β-β hairpins. This is an important problem since the β-β hairpin loop highly constrains the local topology of the protein. An accurate classification of this type of loop is therefore very helpful for super-secondary structure prediction codes.

Table 1 show the results obtained on a subset of 702 links and 2240 hairpins using five-fold cross validation. An accuracy 88% (79%) for the β-β link and 87% (84%) for β-β hairpins was found respectively for HLVQ and MLP. HLVQ is clearly superior to MLP and from now on we will only use it.

Table 1. Confusion matrix, in percentage, for β-β loops obtained by HLVQ and MLP - in parenthesis

	Predicted	
Real	β-β link	β-β hairpin
β-β link	88.4 (79.4)	11.6 (20.6)
β-β hairpin	12.5 (16.1)	87.5 (83.9)

3.2 Classification of All Loops

We now consider the whole sample containing 702 β-β links, 1015 α -β, 2240 β-β hairpins, 1739 β-α and 915 α-α. Results are presented in Table 2. The most accurate predictions are the β-β hairpins (96%), which is not surprising as these loops are the most common having a clearly identifiable characteristics. Worst prediction occurs for the β-β links with only 46% being correctly classified, 28% wrongly assigned to α -β and 20% to β – α loops. Comparing these results with Table 1 we conclude that β-β links are the most difficulty to classify, although they are clearly distinct from β-β hairpins. This low accuracy on β-β links is also due to a small representativity of this class in the database.

Table 2. Performance of HLVQ for all loop types

Predicted Real	β-β link	α -β	β-β hairpin	β-α	α-α
β-β link	45.9	28.5	3.7	19.8	2.1
α-β	8.8	67.4	1.2	18.0	4.6
β-β hairpin	0.4	0.9	96.1	2.1	0.5
β-α	4.4	6.2	2.4	79.5	7.6
α –α	4.0	15.7	1.3	20.3	58.6

4 Discussion and Conclusions

We show that HLVQ algorithm is a promising approach to classify high dimensional data. It is robust, relatively simple to implement and it can handle many features, even if they are irrelevant for the solution. These characteristics of HLVQ may be very useful for other applications in bioinformatics, like protein-protein interactions and secondary protein structure prediction.

Due to the high accuracy obtained for some loops types, like β-β hairpins, our results may be used to boost secondary structure prediction of proteins by correcting results obtained for the loop regions – the most difficult sections to predict.

In future we will use the HLVQ to classify loops of any size and to identify more details of loop by mapping the amino acids sequence to a three letter alphabet discretization of the Ramachandran plan.

References

1. Jones D. T.: Protein secondary structure prediction based on position-specific scoring matrices, J Mol Biol (1999) 292:195-202.
2. Weinert W R, Lopes H S, Neural networks for protein classification, App. Bioinformatics 3 (2004) 41-48.
3. Venclovas C, Zemla A, Fidelis K, Moult J. Assessment of progress over the CASP experiments. Proteins 2003;53 Suppl 6:585-595.
4. Kuang R, Leslie C R, Yang A, Protein backbone angle prediction with machine learning approaches, Bioinformatics (2004) **20,** 1612-1621.
5. Vieira A. and Barradas N. P. (2003) A training algorithm for classification of high dimensional data, Neurocomputing, 50C, 461-472.
6. http://gurion.imim.es/archdb/
7. Espadaler J, Fernandez-Fuentes N, Hermoso A, Querol E, Aviles FX, Sternberg M, Oliva B.: Arch-DB: Automated protein loop classification as a tool for Structural Genomics, Nucleic Acids Res, 32 (2004) 185-188.

VIZ - A Graphical Open-Source Architecture for Use in Structural Bioinformatics

Ricardo M. Czekster and Osmar Norberto de Souza

Laboratório de Bioinformática,
Modelagem e Simulação de Biossistemas - LABIO
PPGCC - FACIN, PUCRS, Av. Ipiranga, 6681. Prédio 16 - Sala 106,
90619-900 Porto Alegre, RS, Brasil
osmarns@inf.pucrs.br

Abstract. Protein structure visualization is crucial for understanding its function inside the cell. Each year, laboratories around the world deposit protein structures on a central database for further analysis and research. The result is a large amount of structures being deposited (approximately 31,000 in may 2005). Visualization is a very powerful tool to help in the analysis, aiding data understanding and interpretation. The present work suggests an architecture to help the rapid construction of visual biomolecular software, specifically designed to be simple, modular and scalable. The architecture, called *VIZ*, employs high quality open-source libraries offering simple data structures and customizable options. The architecture can be used to start a new visual software project to visualize and represent individual protein structures, as well as multiple conformations from molecular dynamics simulation trajectories.

1 Introduction

Large amounts of protein structures are being deposited on the Protein Data Bank (PDB) repository each year [1]. Searching relevant information within the data is not a trivial task, and research efforts are being directed for the development of interactive tools that possess the ability to highlight important regions on the data [2].

The present work defines an open-source architecture to help analysis of protein structure and dynamics visualization. We are proposing an easy to use architecture, called *VIZ*. The goal is to produce a solid base for future development of softwares for high quality visualization of experimental biomolecular structures, as well as ensembles of conformations obtained from molecular dynamics simulations.

2 Visualization and Open-Source Solutions

Visualization offers powerful tools to understand and gain insight from data, helping researchers to identify and quantify interesting regions. It is very important to convey relevant information, not only pretty images. This research

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 226–229, 2005.

has many applications such as in computer simulation analysis, business decision making, and Bioinformatics, just to name a few [3].

Open-source software is present in today's main applications such as operating systems, networking hardware, telephones, and many others, including Bioinformatics, due to the adoption of well-known file formats and freely available genomic data on the Internet [5].

Among the most commonly open-source visualization tools used nowadays, we highlight *VMD - Visual Molecular Dynamics* [9] and *PyMOL* [10]. Both tools also offers scripting and are very stable. For advanced analysis, scripting skills are mandatory.

3 The *VIZ* Architecture

The*VIZ* architecture is designed to be simple, modular and scalable. Another aspect worth of note is the use of the MVC (*Model-View-Controller*) pattern. In MVC, there must be a separation between the data structure (the *Model*), the user interface (the *View*) and the operations (the *Controller*) [4].

3.1 Software Libraries

This proposed architecture uses the following libraries and IDEs (Integrated Development Enviroments): *DevC++* (version 4.9.9.1), a free C/C++ IDE [11]; *OpenBabel* (version 1.100.2), a free open library designed to read/write file formats used in molecular modeling and chemistry; *OpenGL*, a graphical library having many visualization functions [7]; and *FLTK* (version 2.0) used for the GUI (*Graphical User Interface*) [6].

3.2 Modules and Class Diagram

VIZ uses some data structures already defined by *OpenBabel*. A class named *OBMol* contains a list of *OBResidues* and each *OBResidue* contains a list of *OBAtoms*. Each of this classes can attach an *OBGenericData*, used for customized data.

The *VIZ* architecture is divided in modules with their Class Diagrams (Figure 1):

1. Main Module: A main class, called *VIZ* is responsible for opening protein files and creating the graphic window (*VIZOpenGLWindow*). Contains the atom list, which all other modules access. *VIZGenericData* is inherited from *OBGenericData* and contains specific information for each *OBAtom* or *OBResidue* such as the bonding style to be used (lines or cylinders).
2. Rendering Module: has a *VIZOpenGLWindow*, responsible for redrawing the scene and rendering objects.
3. Interface Module: generates callbacks for mouse and keyboard events.
4. Coloring Module: color atoms (using the *VIZColorer* class), used to display aditional information about the inner protein structures.

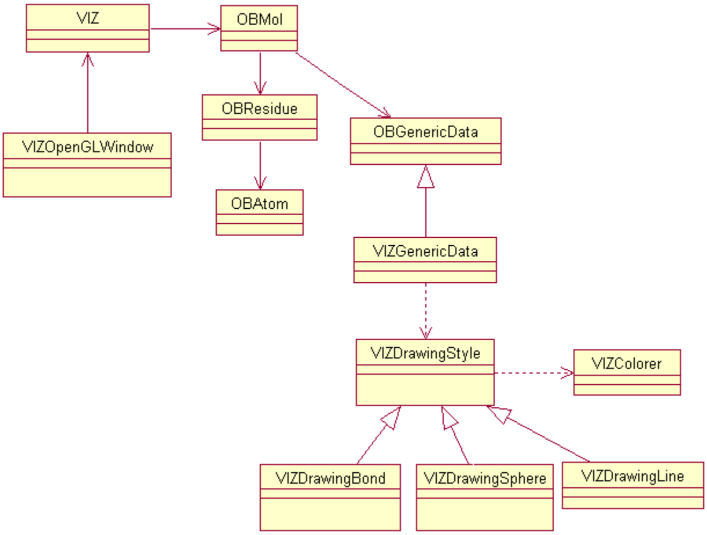


Fig. 1. The *VIZ* architecture represented as a simplified UML Class Diagram

4 Results

We used *VIZ* architecture to build a protein visualization tool, named *ProteinVIZ*, capable of opening a PDB file and having the following molecular representations: *Lines*, *Bonds*, *VDW* (Van der Waals) or *CPK* (atoms as spheres and bonds as cylinders). *ProteinVIZ* enables color modification of atoms/residues by the following types: Chain, Residue, Atom Type and Residue Type. It also allows atom radii alteration.

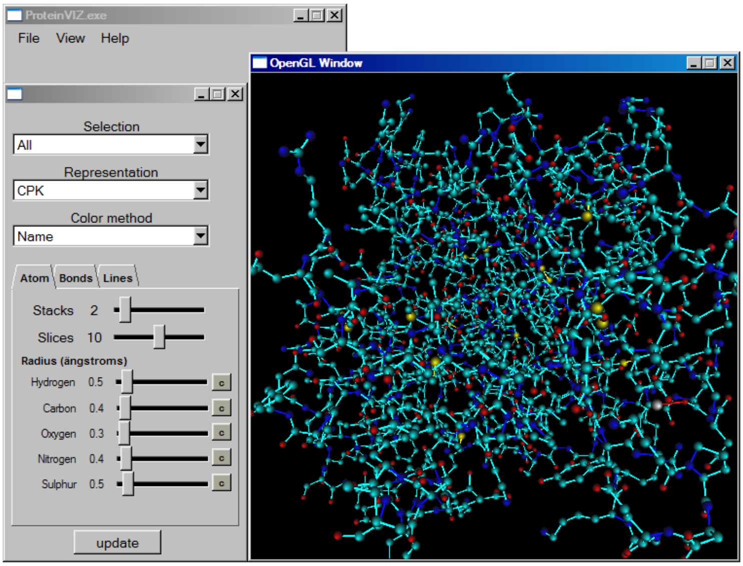
A simple CPK representation of a protein using *ProteinVIZ* is shown in Figure 2. The left side, together with an auxiliary window illustrates some modifications available to users.

5 Conclusions, Perspectives and Acknowledgements

There is a need in the structural biology community for high-quality, insightful visualization software. *VIZ* presently focuses on protein structure visualization. However, it will evolve to a visualization and analysis tool for multiple molecular dynamics simulation trajectories, featuring visual data mining approaches in order to represent simulation events inside the rendering scene.

The *VIZ* architecture is in the early stages of development, but seems very promising in terms of modularity capabilities and maintainability and will be distributed under the LGPL license. The software can also be used in education, teaching molecular visualization techniques.

We thank the referees for critical reading of this work. We also thank FAPERGS and CAPES for financial support.



Selection
All

Representation
CPK

Color method
Name

Stacks 2

Slices 10

Radius (ängstroms)

Hydrogen 0.5

Carbon 0.4

Oxygen 0.3

Nitrogen 0.4

Sulphur 0.5

Fig. 2. CPK representation on *ProteinVIZ*

References

1. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne: The Protein Data Bank. Nucleic Acids Research, 2000, vol. 28, pp. 235-242.
2. Aditya Vailaya, Peter Bluvas, Robert Kincaid, Allan Kuchinky, Michael Creech, Annette Adler: An Architecture for Biological Information Extraction and Representation. Bioinformatics, 2005, vol. 21, pp. 430-438.
3. Kwan-Liu Ma: Visualization - A Quickly Emerging Field. ACM SIGGRAPH Computer Graphics Quaterly, 2004, vol. 38, number 1, pp. 4-7.
4. E. Krasner and S. T. Pope: A cookbook for using the model-view-controller user interface paradigm in smalltalk-80. JOOP, 1988, Aug./Sept.
5. Matthew T. Stahl: Open-source software: not quite endsville. Drug Discovery Today, 2005, vol. 10. number 3, pp. 219-222.
6. Fast Light Toolkit (FLTK). Accessed on March, 2005. Available on http://www.fltk.org/
7. OpenGL - The Industry Standard for High Performance Graphics. Accessed on March, 2005. Available on http://www.opengl.org/
8. Open Babel: A Package to Decypher Computational Chemistry. Accessed on March, 2005. Available on http://openbabel.sourceforge.net/
9. Humphrey, W., Dalke, A. and Schulten, K.,: VMD - Visual Molecular Dynamics J. Molec. Graphics, 1996, vol. 14, pp. 33-38.
10. DeLano, W.L.: The PyMOL Molecular Graphics System. DeLano Scientific, San Carlos, CA, USA. 2002.
11. Bloodshed Software - Providing Free Software to the internet community. Accessed on March, 2005. Available on http://www.bloodshed.net/

Selection of Data Sets of Motifs as Attributes in the Process of Automating the Annotation of Proteins' Keywords

Ana L.C. Bazzan* and Cassia T. dos Santos

Instituto de Informática, Universidade Federal do Rio Grande do Sul,
Caixa Postal 15064, 91.501-970, Porto Alegre, RS, Brazil
{bazzan, ctsantos}@inf.ufrgs.br

1 Introduction and Related Work

Automatic annotation tools are becoming popular since the biologists and curators of databases cannot cope with the volume of sequences to be annotated manually. One way to automate the annotation is to use techniques of symbolic machine learning to derive rules to guide this annotation. However, the training instances tend to have too many attributes, turning the machine learning process difficult and time consuming.

The aim of this paper is to evaluate the information provided by those attributes, which can come from different data sets, regarding a simple task: classifying proteins according to a given set of keywords. Despite its simplicity, the task is very relevant because the Keyword field is an important one in the SWISS-PROT database and gives several hints to experts regarding proteins function and structure. Instead of using thousands of attributes during the machine learning process, we study which set of these attributes can potentially contribute more to the annotation process. Once those rules are generated, they are used to fill the Keyword field in the TrEMBL database (a computer-annotated supplement of SWISS-PROT).

The idea of automating the annotation is not new. Machine learning tecnhiques have been widely used in automated annotation process. An approach based on these techniques to generate rules based on already annotated keywords of the SWISS-PROT database is described by [3]. Such rules can then be applied to unannotated protein sequences in TrEMBL.

In [1] similar methods were employed to automate the annotation of Keyword for proteins appearing in the genome of organisms of the *Mycoplasmataceae* family. However, as said, one issue with this approach is that it still uses too many attributes (all motifs from InterPro and PROSITE cross-referenced in SWISS-PROT). We believe that the time consumed in the training task can be reduced if the correct set of attributes is used.

* Authors partially supported by CNPq; the project is partially supported by CNPq and by FAPERGS.

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 230–233, 2005.

2 Data and Methods

Here we use data about proteins from the model organism *Arabidopsis thaliana*, which is available in SWISS-PROT, to feed the Layer II of ATUCG, our agent-based environment for annotation [2]. SWISS-PROT[1] provides a high level of annotation of each protein, also including extensive cross-references to other databases of motifs, patterns, and profiles. We use some of these cross-references as attributes in the machine learning process. Specifically, we use the following. PROSITE[2] characterizes biologically significant sites in proteins. Pfam[3] is a database of alignments and HMMs covering many common protein domains. PRINTS[4] is a compendium of protein fingerprints. ProDom[5] families are built by an automated process based on a recursive use of PSI-BLAST. Finally, InterPro[6] uses a collection of profiles from PRINTS, Prosite, ProDom, Pfam and SWISS-PROT, which creates a unique, non-redundant characterization of a given protein family, domain or functional site.

The data used comes from a local version of the SWISS-PROT database (status of May, 2004), in which 2817 proteins relating to *A. thaliana* were found. Many keywords appeared in the data but we are focusing on those whose number of instances is higher than 100. The number of keywords satisfying this criterion is 27 (those that appear in Table 1). Since the aim here is to compare data sets of motifs we use all motifs which are cross-referenced in SWISS-PROT as attributes. The number of attributes, by data set, is: 1316 (Intepro), 907 (Pfam), 220 (Prodom), 589 (Prosite), 246 (Prints), thus 3278 in total. Also, we have imposed a constraint on the quality of the rules generated by C4.5: each rule must cover a minimum number of 25 instances, a number that is approximately 1% of the number of training instances. The quality of each rule generated by C4.5 was evaluated via 5-fold cross-validation (CV).

3 Results and Discussion

In Table 1, the first column is a list of the keywords which met the above mentioned criteria. The second column gives the global error. The third and fourth blocks of columns relate to the statistics for the positive and the negative classes respectively. In these two blocks, averages (due to the n-fold CV) of the number of instances, the absolute error, and the percentage of error are shown. Also, for the positive class only, the table shows confidence as defined in [3].

Due to lack of space, we omit the other tables, showing in Table 2 only the equivalent of the last line of Table 1 (average over all keywords). When the

[1] http://www.expasy.ch/sprot/
[2] http://www.expasy.ch/prosite
[3] http://www.sanger.ac.uk/Pfam/
[4] http://bioinf.mcc.ac.uk/dbbrowser/PRINTS/PRINTS.html
[5] http://protein.toulouse.inra.fr/prodom.html
[6] http://www.ebi.ac.uk/interpro/

Table 1. Evaluation Test (5-fold CV) - Attributes Used: Interpro

	Global	Class (Keyword)			Non-Class
Keyword	Error (%)	Instances	Error (%)	Conf.	Error (%)
ATP-binding	21.60 (3.80)	53.6	21.20 (39.55)	0.87	0.40 (0.08)
Alternative-splicing	27.20 (4.80)	27.2	27.20 (100.00)	0.00	0.00 (0.00)
Calcium	8.20 (1.40)	22.2	7.80 (35.14)	0.75	0.40 (0.07)
Cell-wall	8.80 (1.60)	24.2	7.80 (32.23)	0.74	1.00 (0.19)
Chloroplast	71.00 (12.60)	71	71.00 (100.00)	0.00	0.00 (0.00)
Coiled-coil	23.60 (4.20)	33	21.00 (63.64)	0.57	2.60 (0.49)
DNA-binding	33.00 (5.80)	47.4	33.00 (69.62)	0.79	0.00 (0.00)
Glycoprotein	31.80 (5.70)	49.2	29.80 (60.57)	0.72	2.00 (0.39)
Heme	4.00 (0.70)	32.6	3.80 (11.66)	0.87	0.20 (0.04)
Hydrolase	36.80 (6.50)	51.8	36.60 (70.66)	0.78	0.20 (0.04)
Iron	13.00 (2.30)	27.4	12.80 (46.72)	0.77	0.20 (0.04)
Metal-binding	23.20 (4.10)	38.6	23.00 (59.59)	0.78	0.20 (0.04)
Mitochondrion	44.20 (7.80)	44.2	44.20 (100.00)	0.00	0.00 (0.00)
Multigene-family	152.60 (27.10)	70	0.00 (0.00)	0.86	4.60 (1.33)
Nuclear-protein	55.40 (9.80)	76.8	55.40 (72.14)	0.85	0.00 (0.00)
Oxidoreductase	34.80 (6.20)	63.4	34.60 (54.57)	0.87	0.20 (0.04)
Phosphorylation	15.20 (2.70)	25.8	11.20 (43.41)	0.56	4.00 (0.74)
Plant-defense	12.00 (2.20)	23.4	12.00 (51.28)	0.75	0.00 (0.00)
Protein-transport	21.40 (3.80)	23.2	21.40 (92.24)	0.32	0.00 (0.00)
Repeat	42.60 (7.50)	62.4	42.60 (68.27)	0.84	0.00 (0.00)
Ribosomal-protein	34.00 (6.00)	34	34.00 (100.00)	0.00	0.00 (0.00)
Signal	60.20 (10.70)	98.8	59.20 (59.92)	0.87	1.00 (0.22)
Transcription-regulation	26.00 (4.60)	47.4	26.00 (54.85)	0.85	0.00 (0.00)
Transferase	43.20 (7.70)	57.4	43.00 (74.91)	0.77	0.20 (0.04)
Transit-peptide	69.00 (12.30)	69	69.00 (100.00)	0.00	0.00 (0.00)
Transmembrane	79.60 (14.10)	111.8	78.40 (70.13)	0.84	1.20 (0.27)
Transport	40.40 (7.20)	48.2	40.40 (83.82)	0.67	0.00 (0.00)
Average	38.25 (6.79)	49.41	32.09 (63.51)	0.62	0.68 (0.15)

classification is performed with attributes only from single databases in Table 2, in most cases the error in the non-class is low. However, looking at error rates regarding the positive class only (fourth column), some are unacceptable (e.g. 95.07% for ProDom). Similar conclusion can be drawn for confidence. If we consider attributes only from the InterPro database, we see that the error rate in the positive class is lower than it was the case when only ProDom was used. This is valid for all keywords (though not shown here).

For the other data sets, the trend is that global error is low (e.g. 7.75% for PRINTS) but the error rate for the positive class is high. Better confidences and error rates are achieved when using the following databases: InterPro, Pfam, and also for the combinations: InterPro+PROSITE, and InterPro+PROSITE+Pfam). However, in these last cases, the combination brought no increase: using attributes from InterPro alone is as good as using attributes from InterPro plus other data sets.

Table 2. Evaluation Test (5-fold CV) – Error, number of instances and confidence, for each data set of attributtes (average over all keywords)

	Global	Class (Keyword)			Non-Class
Database	Error (%)	Instances	Error (%)	Conf.	Error (%)
All	37.98 (6.74)	49.50	31.94 (63.31)	0.62	0.65 (0.14)
Interpro+Prosite+Pfam	37.98 (6.74)	49.51	31.90 (63.19)	0.62	0.70 (0.15)
Interpro+Prosite	37.96 (6.73)	49.51	31.88 (63.16)	0.62	0.70 (0.15)
Interpro	38.25 (6.79)	49.41	32.09 (63.51)	0.62	0.68 (0.15)
Pfam	39.52 (7.01)	49.21	33.13 (65.38)	0.60	0.68 (0.15)
Prosite	40.35 (7.15)	49.32	34.32 (68.92)	0.57	0.44 (0.09)
Prints	43.70 (7.75)	48.64	37.13 (73.55)	0.47	0.31 (0.06)
Prodom	52.54 (9.32)	50.36	47.77 (95.07)	0.20	0.23 (0.04)

Finally, a note on the still high level of error rate. This is due to two main factors: low level of annotation of Keyword in SWISS-PROT and the unbalance of the two classes. This issues were investigated somewhere else and are not the focus of the present paper, which aims at comparing the data sets.

4 Conclusions

Using all available data regarding motifs as attributes is prohibitive for symbolic machine learning methods. This paper discusses the use of several data sets in order to evaluate which one(s) is/are more valuable regarding the task of producing rules for annotation of the field Keyword in TrEMBL.

One sees that some data sets of attributes perform similarly. In particular, using all attributes (i.e. from all databases together) does not perform better than using only InterPro or only Pfam. Combinations of attributes (e.g. PROSITE+InterPro or PROSITE+InterPro+Pfam) do not perform much better than each of these data sets alone. ProDom or PRINTS should not be used alone as data set in the automated techniques, at least at this time when the data set is small. Since each of these databases has its particularities, the expert in the domain of annotation should decide which one to use. In the absence of this information, InterPro is a safe choice since it is based on the others.

References

1. A. L. C. Bazzan, S. C. da Silva, P. M. Engel, and L. F. Schroeder. Automatic annotation of keywords for proteins related to *Mycoplasmataceae* using machine learning techniques. *Bioinformatics*, 18(S2):S1–S9, October 2002.
2. A. L. C. Bazzan, R. Duarte, A. N. Pitinga, S. L. F., S. C. Silva, and F. A. Souto. ATUCG–an agent-based environment for automatic annotation of genomes. *International Journal of Cooperative Information Systems*, 12(2):241–273, June 2003.
3. E. Kretschmann, W. Fleischmann, and R. Apweiler. Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on SWISS-PROT. *Bioinformatics*, 17:920–926, 2001.

Bioinformatics Tools for HIV-1 Identification in Southern Brazilian States

Ardala Breda[1,2], Cláudia Lemelle Fernandes[1,2],
Sabrina Esteves de Matos Almeida[1], Heitor Moreira Franco[3],
Maria Lúcia Rosa Rossetti[1], Rosângela Rodrigues[3],
Luís Fernando Brígido[3,4], and Elizabeth Cortez-Herrera[1]

[1] Centro de Desenvolvimento Científico e Tecnológico,
Fundação Estadual de Produção e Pesquisa em Saúde,
Porto Alegre - RS, Brazil
[2] Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas,
Pontifícia Universidade Católica do Rio Grande do Sul,
Porto Alegre - RS, Brazil
[3] Instituto Adolfo Lutz,
São Paulo - SP, Brazil
[4] Programa Nacional de DST/AIDS,
Ministério da Saúde, Brazil
{abreda, cfernandes}@inf.pucrs.br

1 Introduction

HIV/AIDS pandemic affected 39.4 million people at the end of 2004, spreading at the rate of 15.000 new infections per day [1]. Although Brazil ranks in fourth in number of reported AIDS cases, limited information concerning the molecular diversity of HIV-1 circulating subtypes is known [3]. Southern Brazil has a particular HIV-1 epidemic, whereas subtype B dominates other regions of the country and subtype C reported cases are rare, in southern states the subtypes C and B have equivalent proportions, and the subtype C seems to be growing up since it was first described in Porto Alegre city, capital of Rio Grande do Sul (RS), at 90's.

The characterization of the particular population of Southern Brazil, where a large number of HIV-1 infected subjects are under antiretroviral (ARV) treatment, underscores its potential usefulness in clinical, treatment, and vaccine trials in Brazil [12]. This study focus mainly in the data analysis and characterization of HIV-1 subtypes circulating in Southern Brazil states - which account's for over 20% of the total Brazilian infections.

2 Materials and Methods

The sixty-eight samples sequences from the three states of Southern Brazil (Paraná - PR, Santa Catarina - SC and RS) for subtype determination were

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 234–237, 2005.

obtained from Laboratório Central do Estado do Rio Grande do Sul - LACEN-RS data bank, from patients chronically infected already under ARV treatment that are failing ARV therapy. Samples collected between the years 2002 and 2003 were chosen randomly. The protease (*pr*) and transcriptase (*rt*) genes from the *pol* region (1.302 base pairs, corresponding to nucleotides (*nt*) 2253-3555 in HIV-1 HXB2) were sequenced and analyzed for subtype determination.

Multiple alignments between the sample sequences and the sequences of reference isolates for each of the known subtypes included in the 2001 compendium of the Los Alamos National Laboratory (LANL) [7], both in FASTA format, were performed with ClustalX (1.83) program [14]. Gaps were manually removed with Bioedit program [4]. Phylogenetic inferences were performed by the neighbor-joining method using the HKY model of substitution [5] implemented in PAUP 4.0 beta [13] and its reliability was estimated by 1000 bootstrap replications.

Phylogenetic trees were generated with PAUP 4.0 beta [13] and visualized with the TreeView program [10]; a bootstrap value joining the query sequence with a particular subtype was considered to be significant if it exceeded 70%. The sequences that had an outlier behavior (probable recombinant form - mosaics) were selected to further analysis in the bootscanning package of the SIMPLOT software version 2.5 [8], to verify the evidence of recombination as well as the breakpoints map. A window size of 400bp was chosen. Recombinant authentication was performed with the likelihood ratio test [6] using Modeltest, version 3-06 [11] and PAUP 4.0 beta software [13].

3 Results and Discussion

Samples subtype were identified when they cluster with reference sequences from LANL [7] with bootstrap values above 70%. Samples with bootstrap values below 70% on phylogenetic trees were analyzed by bootscaning [8] for recombination breakpoints delimitation. Phylogenetic neighbor-joining trees of partial segments confirm the subtype assignments of segments obtained by bootscanning, bootstrap support segments clustering with reference subtypes. When necessary, posterior analyses with appropriate substitution model chosen by Modeltest [11] were performed for each of the mosaics in support of their subtype authentication.

Breakpoints of the eight mosaics are located at *nt* positions 2532 (sample 014), 3002 and 3202 (sample 028), 3010 (sample 033), 3212 (sample 036), 2632 (sample 049), 2602 (sample 051), 2616 and 3102 (sample 073) and 3002 and 3202 (sample 074) according to HXB2.

Table 1 and Table 2 summarizes the data of calculated proportions and region distribution of HIV-1 *pr* and *rt* gene subtypes for the sixty-eight samples.

As expected for subtype distribution of Brazilian subtypes, B strains was the main subtype observed (48.5%), but C strains has a higher number of infected subjects than F1 subtype (Table 1), probably because of the samples origin, since at Southern Brazil there is a higher prevalence of subtype C infections, diverse from the other geographical regions [3, 12].

Table 1. Estimate prevalence of HIV-1 *pol* subtypes at South Brazil

Southern Brazil	Subtype B	Subtype C	Subtype F1	Mosaics
n=68	48.5%	30.9%	8.82%	11.8%

Table 2. Proportion of HIV-1 *pol* subtypes by states of South Brazil

State	Subtype B	Subtype C	Subtype F1	B/F mosaic	B/C mosaic
PR (n=19)	12 (63%)	4 (21%)	1 (5.3%)	2 (10.5%)	–
SC (n=18)	8 (44%)	7 (39%)	2 (11%)	1 (5.5%)	–
RS (n=31)	13 (42%)	10 (32%)	3 (10%)	2 (6%)	3 (10%)

In PR state, the subtype B has a important prevalence of infected subjects (63%), more than in the other studied states, where there was an almost equivalent number of subtype B and C cases (44% B and 39% C for SC and 42% B and 32% C for RS) (Table 2). Mosaics B/C and B/F were found in areas of co-circulations of these subtypes, been B/C observed only at RS state. The B/F is the most common recombinat form in Brazil [2], since these two subtypes are the most frequent related in this country.

4 Conclusion

Bioinformatics tools allow us to identify the subtypes circulating in a population, been possible a reliable characterization of HIV-1 epidemics for each of the regions in a country with continental proportions as Brazil. The similar prevalence of subtypes B and C in this study makes Southern Brazil a perfect setting for clinical, treatment, and vaccine trials, where a control group both of subtype B and C infected individuals with similar ethnic characteristics are readily available.

The HIV-1 virus is characterized by it's high genetic variability, rapid evolution, and diversification, most because of genetic recombination within and between different subtypes [12]. The high incidence of mosaics (11.8%) is in agreement with the 2002 WHO estimation, that countries where multiple subtypes co-circulate could have a percentage of recombinants between 8% and 24% [9].

References

1. AIDS Epidemic Update 2004. http://www.unaids.org/bangcock2004/report.html, 2004.
2. Rodrigo M. Brindeiro, Ricardo S. Diaz, Ester C. Sabino, Mariza G. Morgado, Ivone L. Pires, Luís Brígido, Maria C. Dantas, Draurio Barreira, Paulo R. Teixeira, Amilcar Tanuri, and the Brazilian Network for Drug Resistance Surveillance. Brazilian Network for HIV Drug Resistance Survaillance (HIV-BResNet): a survey of chronically infected individuals. *AIDS*, 17(7):1063–1069, 2003.

3. Ana Maria Barral de Martínez, Edel Figueirêdo Barbosa, Paulo César Pelegrino Ferreira, Fabíola Adriene Cardoso, Jussara Silveira, Gabriela Sassi, Cláudio Moss da Silva, Vera Mendonça-Signorini, and Carlos Maurício de Figueiredo Antunes. Molecular Epidemiology of HIV-1 in Rio Grande, RS, Brazil. *Revista da Sociedade Brasileira de Medicina Tropical*, 35(5):471–476, 2002.
4. Tom A. Hall. BioEdit: a User-Friendly Biological sequence Alignment Editor and Analysis Program for Windows 95/98/NT. *Nucl. Acids. Symp. Ser.*, 41:95–98, 1999.
5. Masami Hasegawa, Hirohisa Kishino, and Takato Yano. Dating of the Human-Ape Splitting by a Molecular Clock of Mitochondrial DNA. *Journal of Molecular Evolution*, 21:160–174, 1985.
6. John P. Huelsenbeck and Keith A. Crandall. Phylogeny Estimation and Hipotesis Testing Using Maximum Likelihood. *Annual Review of Ecology and Systematics*, 28:437–466, 1997.
7. Los Alamos National Laboratory, New Mexico, USA. Theoretical Biology and Biophysics Group HIV Sequence database. http://hiv-med.lanl.gov.
8. J. K. Carr M. Salminem, Donald S. Burke, and F. E. McCutchan. Identification of Recombination Breakpoints in HIV-1 by Bootscanning. *AIDS Research and Human Retroviruses*, 11:1423–1425, 1995.
9. Saladin Osmanov, Claire Pattou, Neff Walker, Bernhard SchwardlSnder, Jose Esparza, the WHO-UNAIDS Network for HIV Isolation, and Characterization. Estimated Global Distribution and Regional Spread of HIV-1 Genetic Subtypes in the Year 2000. *Journal of Acquired Immune Deficiency Syndromes*, 29(2):184–190, 2002.
10. Roderic D. M. Page. TreeView: An Aplication to Display Phylogenetic Trees on Personal Computers. *Computer Applications in the Biosciences*, 12:1423–1425, 1996.
11. David Posada and Keith A. Crandall. MODELTEST: Testing the Model of DNA Substitution. *Bioinformatics*, 14(9):817–818, 1998.
12. Esmeralda A. J. M. Soares, Rodrigo P. Santos, José Augusto Pellegrini, Eduardo Sprinz, Amilcar Tanuri, and Marcelo A. Soares. Epidemiologic and Molecular Characterization of Human Immunodeficiency Virus Type 1 in Southern Brazil. *Journal of Acquired Immune Deficiency Syndromes*, 34(5):520–526, 2003.
13. David Swofford. PAUP 4.0: Phylogenetic Analysis Using Parsimony (and Other Methods). Technical report, Sunderland, MA:Sinauer Associates, 1999.
14. Julie D. Thompson, Toby J. Gibson, Frédéric Plewniak, Frantois Jeanmougin, and Desmond G. Higgins. The Clustal_X Windows Interface: Flexible Strategies for Multiple Sequence Alignment Aided by Quality Analysis Tools. *Nucleic Acids Research*, 25(24):4876–4882, 1997.

Fact and Task Oriented System for Genome Assembly and Annotation

Luciano A. Digiampietri, Julia M. Perdigueiro, Aloisio J. de Almeida Junior,
Daniel M. Faria, Eric H. Ostroski, Gustavo G.L. Costa, and Marcelo C. Perez

Instituto de Computação, Universidade Estadual de Campinas,
CP 6176, Campinas, SP 13084-971 BRAZIL
luciano@ic.unicamp.br

Abstract. We present a preliminary description and results of a system to help the curation of genome assembly and annotation. Standard tools are used for these tasks, and our methodology focuses on user guidance, data visualization and integration, and data browsing aspects.

1 Introduction

The usual concern of most of activities, tools and infrastructure related to genomic analyses is with computer systems functionality. Many systems are developed in an *ad hoc* way following only functional requirements. This development methodology pays little attention to characteristics like user interface and usability. We have developed a simple methodology to make the user-interaction part of genome assembly and annotation more user-friendly and therefore more effective. Based on this methodology we have implemented a web-based prototype. This prototype is being used as the main tool for the assembly and annotation of the *Xanthomonas axonopodis pv aurantifolii* strains *B* and *C* genomes at LBI [4] with the support of USP [1] and UNESP [5].

2 System Development Methodology

The system presented here was developed following a generic methodology specified by us at LBI. This methodology allows the development of any computational infrastructure which requires a flow of activities and that provides data mining and visualization mechanisms. This methodology has the following phases: (i) identification and description of tasks to be done; (ii) description of facts to be considered; (iii) development of fact analysis and visualization tools; (iv) development of examples or tutorials on how to execute each task; (v) development of tools for accomplishing the tasks. We have applied this methodology to improve a genome assembly and annotation system used at our laboratory.

Facts are characteristics observed in the set of available data. Facts are the basis for all the analysis and conclusions which will be made during assembly and annotation. *Tasks* are actions which must be executed (automatically or

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 238–241, 2005.

Table 1. Assembly tasks and facts

Task	**Facts**
Contigs management	set of reads, phrap and genscaff results
Links management	reads from the same insert found on different contigs
Contigs projection on the reference genome	alignment between the reference and the target genomes
Supercontigs management	contigs, links, gap closures and alignments to the reference genome
Management of inserts to be subcloned and sequenced	reads and links information

manually) with the objective of getting closer to the desired solution. For example, a set of facts can be observed in the result of the phrap [3] assembly and postprocessing by the genscaff program [6], such as a possible link between contig x and contig y. A task must obtain conclusions about the facts, for instance, to conclude whether contigs x and y are adjacent or not. For each kind of fact, data analyses and visualization tools were developed to ease the understanding and the making of a decision. Some examples of genome assembly tasks are: contigs management, links management, selection of clones to be subcloned and sequenced, comparison between the target and the reference genomes and *supercontig* management (supercontig is a set of linked contigs).

Figure 1 shows some of the graphical results of our tools (showing contig, supercontig, link and projection with reference genome information). All figures are automatically generated and have hyperlinks to allow easy data browsing.

One of the most complex tasks during genome assembly is to decide whether two contigs are linked or not. Our system used the following facts to help in decision making: (1) links between those contigs; (2) conservation of the order regarding the reference genome (based on alignment against a reference genome);

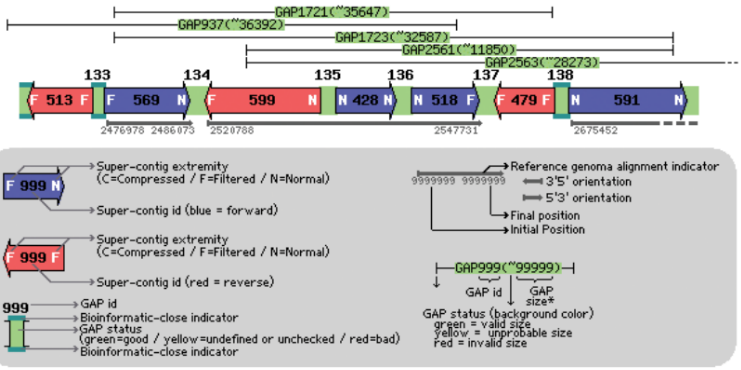


Fig. 1. Supercontig information: contigs, links, gaps and projection over the reference genome information

and (3) bionformatics gap closure (a sub-assembly using only reads in a particular region that successfully closes a gap). By integrating these facts, our system facilitates the curatorial part of the genome assembly process, decreasing the need for new sequencing.

3 Results, Conclusions and Future Work

Complex information systems that require intense user interaction deserve special care on user-related issues, such as usability and interface. Large-volume data processes, such as genome assembly and annotation, require special care on data presentation, through graphic visualizations, data summaries and data integration. We have briefly described a simple methodology that helped us create a web-based system that allowed us to achieve good results in a genome assembly process. The detailed description of each one of the tasks and facts, as well as the specification of tutorials or examples for each task, makes possible a more conscious, easy and systematic use of the system.

The system proposed is being used on the *Xanthomonas axonopodis pv aurantifolii* strains *B* and *C* genomes assembly and annotation. Before the work described in this paper, these two genomes were being assembled using a traditional system, which had no specific computational help for assembly curators. The use of our system showed quantitative and qualitative gains with respect to previous assembly results. The main gains were: (i) all data is integrated in a database management system (DBMS), making it possible to make efficient queries to every object involved in the project; (ii) low training cost of new assembly and annotation team, due to tutorials developed for the execution of each task; (iii) greater assembly efficiency through a better use of data. The most important practical conclusion of this case study was the reduction on the number of supercontigs without the need of new sequencing, causing greater genome coverage. Table 2 compares the results obtained by our system to the ones available before we put our system to use. This table shows that thorough our system we obtained better results on every analyzed characteristic, refining the assembly and being more efficient on the use of available data.

As a future step we intend to package tools (making them more generic and reusable), and extending the system for dealing also with comparative genomics.

Table 2. Comparison between previous results and results from our approach

Data	**Previous Results**	**Our results**	**Situation**
Number of supercontigs	45	35	Improved
Total number of contigs in the supercontigs	225	234	Improved
Average number of contigs by supercontig	5	6.69	Improved
Number of base pairs on supercontigs	4934046	5105624	Improved
Valid links on supercontigs	180	199	Improved
Number of new closed gaps	87	91	Improved

Another future work is the development and usage of ontology to publish data on the Web through XML [2], increasing interoperability.

More detailed descriptions and tools can be obtained through e-mail contact with LBI: lbi@ic.unicamp.br.

Acknowledgements. The work described in this paper was partially financed by CAPES, Fundecitrus and MCT-PRONEX SAI project. The Laboratory of Bioinformatics, in which most of this work was done, is partly funded by FAPESP. We acknowledge the others researchers that had worked in the sequencing and assembling of the genomes presented here.

References

1. Departamento de Bioquímica, Instituto de Química, University of Sao Paulo. http://www2.iq.usp.br/bioquimica/
2. Extensible Markup Language (XML) 1.0 (Third Edition) (2004). http://www.w3.org/TR/2004/REC-xml-20040204
3. Green P. Phrap: phragment assembly program. http://www.phrap.org.
4. Laboratory for Bioinformatics (LBI), Institute of Computing, University of Campinas. http://www.lbi.ic.unicamp.br
5. Laboratório de Bioquímica e Biologia Molecular (LBM), UNESP. http://www.lbm.fcav.unesp.br/
6. Setubal, J. and Werneck, R.: A program for building contig scaffolds in double-barreled shotgun genome sequencing. *Technical Report* IC-01-05, Institute of Computing, Unicamp, 2001.

A Clustering Strategy to Find Similarities in Mycoplasma Promoters

João Francisco Valiati and Paulo Martins Engel

Universidade Federal do Rio Grande do Sul, Instituto de Informática,
C. Box 15.064, 91.501-970 Porto Alegre, RS, Brazil
{jvaliati, engel}@inf.ufrgs.br

Abstract. This paper presents a neural network clustering strategy to identify regularities in a dataset of Mycoplasma promoter sequences. The traditional way that prokaryotic promoters are identified is proven inadequate to the Mycoplasma family. Our clustering approach tries to discover regularities in base pair compositions of the dataset sequences to give clues which indicate the presence or absence of promoters. Several experiments with leave-one-out strategy and a negative dataset revealed a best way to fit model parameters. Preliminary results are promising for creating a computational model able to find promoter regions in Mycoplasmas.

1 Introduction

The promoter recognition apparently is a dominated task to prokaryotic organisms; the early studies about transcription in *E. coli* revealed a great consensus in base pairs composition of its sequences. However the systematic used to *E. coli* data is not suitable to identify promoters in all prokaryotic organisms.

Although many experiments have been already carried out and some methodologies applied, the problem of promoter recognition is not yet completely solved. Some of the reasons for this is the lack of experimental data for evidence of the existence of the promoters due to high costs, bad determination of which non-promoter regions are adequate for learning, the lack of data of others organisms, the difficult characterization of the problem due to the lack of a more accurate region localization, and so on.

The investigation reported in this paper presents a clustering strategy to identify the promoter regions in Mycoplasma family that indicates the beginning of putative genes. The next section shows the biological concept of the traditional promoter region and the difficulties observed in Mycoplasmas. The third section exposes the experiment developed, describing the dataset, the clustering strategy, and the evaluation of experiments. The fourth section discusses the obtained results and the last section presents the conclusion and discussion.

2 Promoter Characterization

The promoter regions play an important role in protein synthesis because they are responsible for determining the bound between the DNA region that is transcribed in

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 242 – 245, 2005.

mRNA, to produce protein and the remaining information that is not transcribed. The promoters are found in portions that precede the beginning of genes, the so-called transcription start site (TSS). This place is the point where the RNA polymerase, enzyme responsible for the transcription process, contacts the promoter region [2].

The standard promoter definition is observed in *E. coli*. Figure 1 shows a prototypical pattern promoter. The sequence consists of a -35 hexamer separated by 17 bp of the -10 hexamer, located 7 bp upstream of the TSS [5].

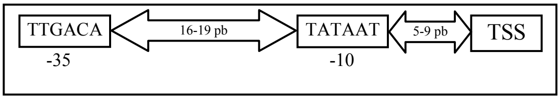


Fig. 1. The pattern promoter

2.1 Promoters in Mycoplasma

Experiments realized with *M. pneumoniae* promoter sequences revealed that there are several possible -10 region: TA(AGT)AAT, TAA(GT)AT, TACTAT and TATTAA and a weak consensus in the -35 region, which shows a short sequence, TTGA, to be relatively conserved. Previous studies demonstrated that the poor definition in -35 region occurs due to insufficient data to identify some conservation [7].

This may reflect a more complex process of transcriptional initiation than might be expected. The researchers concluded that features which appear infrequently in other bacterial species appear to be common in *M. pneumoniae* [4].

3 Method and Experiment

3.1 Clustering Strategy

The clustering strategy applied in our experiment follows the Adaptive Resonance Theory (ART) proposed by Carpenter and Grossberg. It comprises a set of neural networks which support competitive learning in such a way that a new cluster is formed whenever an input pattern is sufficiently different from any existing cluster prototype, according to a vigilance parameter. Clusters are represented by individual output units, as usual; but in an ART network the output units are uncommitted until they are needed [1].

ART is compounded by several models each one with a particularity. The model used is the ART1; like the other models it tries to solve the stability-plasticity dilemma [6]. This refers to the conflicting goals for the neural networks to remain stable in the condition to which they have converged (i.e. to retain their memories of what has been learned) while at the same time being receptive to new learning [3].

3.2 Experiment Description

The dataset utilized in the experiment was extracted from Nucleic Acids Research article: Transcription in *Mycoplasma pneumoniae*, and was organized by Weiner III, Herrmann and Browning in the Zentrum für Molekulare Biologie Heidelberg [7]. The reported experiment shows 32 promoter sequences obtained from experimental approach with *M. pneumoniae* M129.

These sequences were aligned by -10 region where there is a major consensus. Each sequence was compounded by 50 nucleotides because in this length are inserted the supposed two principal features of promoter region. After that, a 6 base pairs window positioned exactly over the -10 region, and another window over the supposed -35 region, enclosing ten base pairs, where a 6 base pair set corresponding to the promoter is supposed to be inserted, referring to the promoter were extracted from the original sequence. The junction of these two windows composes a characteristic promoter sample. Then this dataset of 32 samples of promoters was converted to the binary codification known as BIN4 [8].

The next step was to submit the preprocessed samples to a clustering method that uses a competitive learning to find groups with similar composition in nucleotide sequence. The clusters were obtained to each promoter region, i.e., the -10 region was applied to train one ART1 net and the -35 region was used to train another ART1 net. Each one of theses networks was responsible for creating its respective clusters.

Besides of the dataset for constructing the cluster, another dataset was applied as validation set. While the data for generating clusters were represented only for promoter sequences, the validation data were constructed from sequences that represent non-promoters. These sequences were obtained from two intergenic regions of *M. pneumoniae* located between one gene in forward strain and another gene in reverse strain; these regions contain no regulatory information about promoters. The non-promoter sample was generated by a sliding displacement of the two proposed windows with one-position increment along the start of the sequence until reaching the final length of each sequence. In this way, 2220 non-promoter samples were generated.

This validation set is very important because it serves to observe how to fit the vigilance parameter, once it searches for equilibrium between the clusterization of positive and negative datasets.

4 Obtained Results

The results evaluated the classification error obtained with the application of leave-one-out method to promoter samples and the classification error for the propagation of non-promoter samples by the clusters generated with all promoter samples.

The best results suggest values to the vigilance parameter to train each model setup in 0.7 for -35 region and 0.1 for -10 region, respectively. This configuration showed a classification error less than 10%. The propagation of all non-promoter samples for this combination resulted in 164/2220 misclassified samples.

5 Conclusions and Discussion

This paper reports an experiment that uses a clustering strategy based on Artificial Neural Networks to identify similarities in *M. pneumoniae* promoters. The dataset to obtain the clusters was extracted from experimental data that supplies specifications about nucleotides composition of some sequences.

The insufficient amount of data, only 32 samples, to represent promoters and the low data characterization that doesn't follow the classical description for prokaryotic promoters like the *E. coli*, were the main problems.

One of the goals of this experiment was to fit the vigilance parameters of the neural models based on similarities of promoter samples and its capacity to reject negative samples, for constructing a generalized model to identify similarities and to produce representative clusters.

The presented results are a preliminary contribution to the several realized experiments to investigate and to discovery regularities and patterns to elucidate the promoter recognition problem in the Mycoplasma family.

References

1. Carpenter, G. A., Grossberg, S. ART2: self-organization of stable category recognition codes for analog input patterns. Applied Optics, vol. 26, no. 23, (1987) 4919-4930
2. Cooper, G.M., Hausman, R.E.: The Cell: A Molecular Approach 3d. ed. Amer. Soc. Microbiol., Washington and Sinauer Assoc., Sunderland, MA (2003)
3. Fyfe, C. Artificial Neural Networks. Departmant of Computing and Information Systems. The University of Paisley, Scotland, UK (1996)
4. Herrmann, R. Molecular Biology of the Bacterium Mycoplasma pneumoniae. Report. Zentrum für Molekulare Biologie Heidelberg. Universität Heidelberg. Germany (2002)
5. Lewin, B. Genes VII. Oxford University Press: New York (1999)
6. Valiati, J. F.: Reconhecimento de Voz para Comando de Direcionamento por Meio de Redes Neurais. PPGC da UFRGS. Porto Alegre (2000)
7. Weiner3rd, J., Herrmann, R., Browning, G. F.: Transcription in Mycoplasma pneumoniae. Nucleic Acids Research, 28, (2000) 4488-4496
8. Wu, C. H., McLarty, J. M.: Neural Networks and Genome Informatics. Elsevier Science. New York (2000)

Gene Prediction by Syntenic Alignment

Said Sadique Adi and Carlos Eduardo Ferreira

Institute of Mathematics and Statistics (IME) - University of São Paulo (USP),
Rua do Matão 1010 – Cidade, Universitária 05508-900 – São Paulo (SP) – Brazil
{said, cef}@ime.usp.br

Abstract. Given the number of available genomic DNA, one now faces the task of identifying the functional parts of such raw sequence data, like the protein-coding regions. The gene prediction problem can be addressed in several ways. The most recently methods make use of the similarities between regions of two unannotated genomic sequences in order to find their genes. In this paper we present a new comparative-based heuristic to the gene prediction problem. It relies on a syntenic alignment of two genomic sequences. We have implemented the proposed heuristic in a computer program and confirmed its validity on a benchmark including 50 pairs of human and mouse genomic sequences.

1 Introduction

The gene prediction problem can be defined as the task of finding the genes encoded in a DNA sequence of interest. In other words, given an eukaryotic DNA sequence, we would like to correctly determine the beginning and end positions of its protein-coding regions. The genes of most eukaryotic organisms are separated by long stretches of intergenic DNA and their coding fragments, named *exons*, are interrupted by non-coding ones, the *introns*. A typical multi-exon eukaryotic gene has the structure shown in Figure 1:

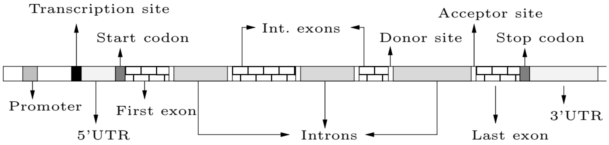


Fig. 1. Simplified structure of a multi-exon eukaryotic gene

Gene prediction methods can be roughly classified into two main categories, named *ab initio* or intrinsic methods and **similarity-based** or extrinsic methods. The first ones rely on statistical information that alone or in conjunction with some signals previously identified in the target sequence allow the identification of its coding, non-coding and intergenic regions. The similarity-based methods make use of the homology between the genomic sequence and a fully annotated transcript, like cDNAs or proteins, in order to accomplish the gene prediction task. Recently, with the huge amount of newly sequenced genomes,

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 246–250, 2005.

new similarity-based methods are being successfully applied in the task of gene prediction. In some way different from the traditional extrinsic methods, the so-called **comparative-based** methods, pioneered by Batzoglou *et al.*[2], rely on the similarities between regions of two unannotated genomic sequences in order to find the genes encoded in each of them.

In this work we present a new comparative-based heuristic to the gene prediction problem. It is based on a pairwise alignment that takes into account the existence of intermittent similarities between the compared sequences. This heuristic was implemented and evaluated on a well-known data set including a number of single and multi-exon sequences. The results in both nucleotide and exon levels look promising and compare with that presented by another comparative-based gene prediction tool.

2 Syntenic Alignment and Gene Prediction

The basic idea of our heuristic is to construct an alignment of the two genomic sequences taking into account the fact that they include intergenic, intronic and exonic regions. This type of alignment, where regions with different levels of conservation are considered, can be referred as *syntenic alignment*. To construct it, we make use of the ideas proposed by Almeida *et al.* in [1]. In these works, the authors present a dynamic programming alignment algorithm whose main idea is to heavily penalize mismatches and gaps inside conserved regions of the two sequences and to penalize in a slightly way its occurrences inside non-conserved regions. To this end, the score of a best syntenic alignment are stored in two different sets of matrices: one for the similar regions and another for the regions where differences are more probably to occur.

In the current work, taken two genomic sequences s and t as input, the best syntenic alignment between them is searched for by making use of seven matrices H, S_e, S_i, I_e, I_i, D_e, D_i, where H stores the score of a best alignment between s and t ending inside an intergenic region, $S_{e/i}$ stores the score of a best alignment between s and t ending with a residue of each one of them inside an exonic/intronic region $I_{e/i}$ stores the score of a best alignment between s and t ending with a insertion in one of them inside an exonic/intronic region and $D_{e/i}$ stores the score of a best alignment between s and t ending with a deletion in one of them inside an exonic/intronic region.

It is worthwhile to note (Figure 1) that eukaryotic genes, with meaningless exceptions, start and end with an exon. Moreover, the first exon of any eukaryotic gene begins with a start codon and the last one ends with a stop codon. It is also well known that the big majority of the internal exons are located between conserved splicing sites. From these ideas, the following recurrences can be used to compute the matrices H, S, I and D. In what follows, we refer to a substring of s ending at position i as s_i and to a substring of t ending at position j as t_j. T is a given threshold. About $P(s_i)$ and $P(t_j)$, they represent the probability of s_i and t_j be a true splicing site. These values were calculated by using the conditional probability matrices described by Salzberg in [5].

$$H[i][j] = \max \begin{cases} H[i-i][j], H[i][j-1], & \\ S_e[i-1][j]-d, D_e[i-1][j]-d, I_e[i-1][j]-d, & \text{if there exists} \\ S_e[i][j-1]-d, D_e[i][j-1]-d, I_e[i][j-1]-d. & \text{a stop codon} \end{cases}$$

$$S_e[i][j] = w_e + \max \begin{cases} S_e[i-1][j-1], D_e[i-1][j-1], I_e[i-1][j-1], \\ H[i-i][j-1], \text{ if there exists a start codon} \\ S_i[i-1][j-1] + P(s_i) + P(t_j), \text{ if } P(s_i) \text{ and } P(t_j) > T \end{cases}$$

$$I_e[i][j] = \max \begin{cases} S_e[i][j-1] - (h_e + g_e), D_e[i][j-1] - (h_e + g_e), I_e[i][j-1] - g_e, \\ H[i][j-1], \text{ if there exists a start codon} \\ S_i[i][j-1] - (h_e + g_e) + P(s_i) + P(t_j), \text{ if } P(s_i) \wedge P(t_j) > T \end{cases}$$

$$D_e[i][j] = \max \begin{cases} S_e[i-1][j] - (h_e + g_e), D_e[i-1][j] - g_e, I_e[i-1][j] - (h_e + g_e), \\ H[i-1][j], \text{ if there exists a start codon} \\ S_i[i-1][j] - (h_e + g_e) + P(s_i) + P(t_j), \text{ if } P(s_i) \wedge P(t_j) > T \end{cases}$$

$$S_i[i][j] = w_i + \max \begin{cases} S_i[i-1][j-1], D_i[i-1][j-1], I_i[i-1][j-1], \\ S_e[i-1][j-1] - k + P(s_i) + P(t_j), \text{ if } P(s_i) \wedge \wedge P(t_j) > T \end{cases}$$

$$I_i[i][j] = \max \begin{cases} S_i[i][j-1] - (h_i + g_i), D_i[i][j-1] - (h_i + g_i), I_i[i][j-1] - g_i, \\ S_e[i][j-1] - (k + h_i + g_i) + P(s_i) + P(t_j), \text{ if } P(s_i) \wedge P(t_j) > T \end{cases}$$

$$D_i[i][j] = \max \begin{cases} S_i[i-1][j] - (h_i + g_i), D_i[i-1][j] - g_i, I_i[i-1][j] - (h_i + g_i). \\ S_e[i-1][j] - (k + h_i + g_i) + P(s_i) + P(t_j), \text{ if } P(s_i) \wedge P(t_j) > T \end{cases}$$

In the above recurrences, d and k correspond to non-negative scalars used to penalize the beginning of an intergenic and intronic region respectively.

3 Tests

In order to evaluate our approach, we have implemented the above recurrences and tested the program on a benchmark including 50 pairs of single gene sequences from human and mouse. These sequences were compiled from the data set used by Jareborg *et al.* in the training and testing of the SGP-2 gene prediction program. About the parameters used, they were experimentally estimated. To access the accuracy of our program, we made use of the specificity (Sp) and sensitivity (Sn) measures introduced by Burset and Guigó in [4]. The average values of specificity and sensitivity achieved by our program, at both nucleotide (Sp_n, Sn_n) and exon levels (Sp_e, Sn_e), are shown in the first four columns of Table 1.

Table 1. average values of specificity and sensitivity

Our Program				Utopia			
Sp_n	Sn_n	Sp_e	Sn_e	Sp_n	Sn_n	Sp_e	Sn_e
0.87	0.94	0.43	0.45	0.86	0.98	0.38	0.52

Despite the good value of sensitivity at the nucleotide level, the specificity of our approach still needs some improvements. The lowest value of specificity at the nucleotide level is mainly due to the number of mispredicted bases at the limits of the annotated exons in the sequences, where the similarity rate is as high as that of the exonic regions. This problem becomes more evident when the first and last exons of the genes are considered. With respect to the behavior of our approach at the exon level, low values of both specificity and sensitivity were achieved in this case. One fact that contributes to the low level of specificity at the exon level is the number of false exons predicted by our approach. From the total of 384 predicted exons, 57 have no intersection with an annotated exon. The majority of these mispredicted exons are located outside the genes and a little bit far from their real limits. This is in accordance with a number of works in the literature, like that presented by The Mouse Genome Sequencing Consortium[6], attesting a high level of conservation between the human and mouse genomes. Finally, it is important to note that, despite the number of misleading bases, the results of our approach compare with that presented (last four columns of Table 1) by another comparative-based gene prediction tool named Utopia[3].

4 Discussion

In this work we presented a program where two evolutionary related sequences are compared in order to identify their genes. It is based on a syntenic alignment of the two analyzed sequences. To the construction of this alignment, the main idea is to heavily penalize mismatches and gaps inside the coding regions and to penalize in a slightly way its occurrences inside the non-coding regions of the two sequences. This approach was implemented and thus tested on a benchmark including 50 pairs of single gene sequences. Despite the low specificity and sensitivity at exon level, our program has achieved promising results at the nucleotide level. They compare with that presented by another comparative-based gene prediction tool.

The main drawback of our approach is related to the existence of well conserved regions outside the genes searched for. This leads to a number of mispredicted exons and additional bases identified as codings at the 5'-UTR and 3'-UTR regions of the predicted genes. One way to overcome this problem is by making use of another statistical information that can give us better insights about the real start and stop codon in the sequences. This work, jointly with a fine tuning of the parameters, is in progress in the hope that better values of specificity and sensitivity at both nucleotide and exon level can be achieved in the future.

References

1. Almeida, N.F., Setubal, J.C., Tompa, M.: On the use of don't care regions for protein sequence alignment. *IC Technical Reports 99-07* (1999)
2. Batzoglou, S., Pachter, L., Mesirov, J., Berger, B., Lander, E.S.: Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res.* **10** (2000) 950-958.

3. Blayo, P., Rouzé, P., Sagot, M.-F.: Orphan gene finding - An exon assembly approach. *Theoretical Computer Science* **290** (2003) 1407-1431.
4. Burset, M., Guigó, R.: Evaluation of gene structure prediction programs. *Genomics* **34** (1996) 353-357.
5. Salzberg, S.L.: A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Computer Applications in the Biosciences* **13(4)** (1997) 365-376.
6. The Mouse Genome Sequencing Consortium: Initial sequencing and comparative analysis of the mouse genome. *Nature.* **420(6915)** (2002) 520-562.

Real Time Immersive Visualization and Manipulation of the Visible Human Data Set

Ilana de Almeida Souza, Claudiney Sanches Junior, André Luiz Miranda da Rosa,
Patrícia Trautenmüller, Thiago Tognoli Lopes, and Marcelo Knörich Zuffo

Laboratório de Sistemas Integráveis (LSI),
Engenharia de Sistemas Eletrônicos da Escola Politécnica da USP,
Avenida Prof. Luciano Gualberto,
travessa 3 nº 380 - CEP - 05508-900 - São Paulo/SP, Brasil
{iasouza, sanches, amiranda, pmuller, tognoli, mkzuffo}@lsi.usp.br

Abstract. The aim of this paper is to present a real time immersive visualization and manipulation of the full color visible human dataset. A data glove and stereo shutter glasses were used to provide interactive and stereoscopic 3D visualization of this set of images in a multiprojection immersive environment.

1 Introduction

The main goal of this project is the interactive volumetric visualization of the Visible Human data set in a multiprojection immersive environment, where the user may visualize and manipulate the head and part of the neck of the virtual man. Volumetric hardware graphics boards named VolumePro were essential to render these images [1]. We used a data glove to realize some movements of the virtual head and a stereo shutter glasses to provide stereoscopic view in the multiprojection immersive environment.

2 System Modeling

This project was implemented in C and C++ language, using VLI VolumePro, OpenGL GLUT, imgall and Glass libraries. The VLI VolumePro Library is needed for the correct use of the VolumePro board, making it's use unique when working with the GLUT. The Glass library was used to synchronize the computer set (cluster), making the project become a distributed system.

The images are photographs taken from a frozen human body and the ice in the photos has a blue aspect, and we needed to develop a segmentation technique to remove the ice, making the background black.

In a pre-processing stage we had to create four new files for each raw image: three images with eight bits each - one for the R values, one for the G values and another for de B's; and a fourth file formed by the union of the other three, resulting on a

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, pp. 251 – 256, 2005.

unique 21 bits RGB image, providing a colorful 24 bits RGB image of 2048x1216 size for each slice of the Visible Man.

We converted the images to HSI type, working with only the images containing the Hue values. Applying geometric diffusion to eliminate some images details, the object's contours were traced using the outlierg function from imgall library and then dilation was applied twice to thicken the contour and to eliminate errors. We had to develop an algorithm to eliminate the connected components between an interval. Any connected component with total number of pixels outside this interval is eliminated from de image. After applying two erosions, we implemented a technique to find a contour with one pixel thickness. To complete the segmentation, every pixel inside this contour belongs to the segmented image.

The VolumePro 500 board supports only grayscale 8 (eight) and 12 (twelve) bits in .VOX formats and each 12 bits .VOX file consists of four bits sets separated by an identification spot. Every bit 1 (one) corresponds to the identification spot that separates the bits sets. Each four bits sets correspond to the R, G and B values, respectively, showing that each .VOX files has a total of 16 bits.

To render a colorful image, it was necessary to split the 24 bits RGB original set into three different grayscale volumes, one for R (R.VOX), one for G (G.VOX) and another one for B (B.VOX), each one with 8 bits. After that, it was created a color table and a scale to convert each grayscale value to the respective color value. Then the three images were gathered in a memory buffer to be used as an OpenGl texture, wich was applied to a cube that is drawn in the main window, as shown in figure 1.

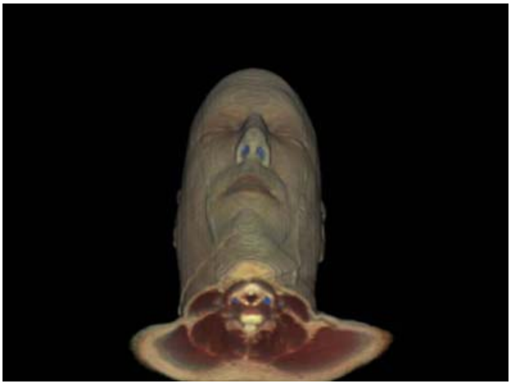


Fig. 1. Visible Man – main window

To obtain a correct stereoscopic vision, this process must be repeated for two points of view or cameras and two proceedings were developed, one that generates the right eye vision and other that works with the left eye vision. So the system creates six (6) grayscale images that are converted in two (2) RGB images and then presents them interpolating at the stereoscopic glass frequencies.

We used the data glove to manage the virtual man real time movements in the multiprojection immersive environment. With some a priori movements, the user may rotate the image at a maximum horizontal or vertical 180° angle, provide transparency (figure 2) or make an arbitrary cut in the volume (figure 3). Each action may discharge functions that move the volumetric data (.VOX) and it's necessary to generate new renderings to form a new stereoscopic image.

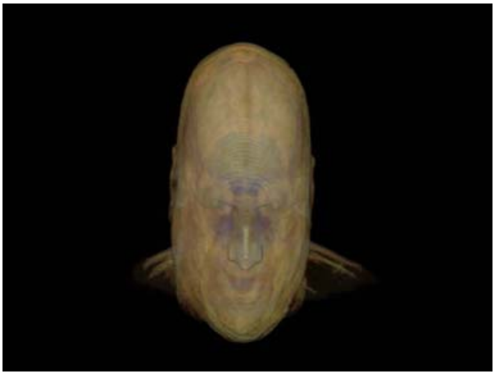


Fig. 2. Visible Man – active transparency

Gesture functions are finger open/close binary setups, where the thumb is an exception. There are 24 = 16 possible gesture combinations, and the number zero gesture is defined when all fingers are closed (except the thumb) and the 15th gesture is when all fingers are opened. Figures 4 and 19, respectively. For invalid gesture, the number is defined as –1. Table 1 shows every possible combinations and it's respective referenced figures [2].

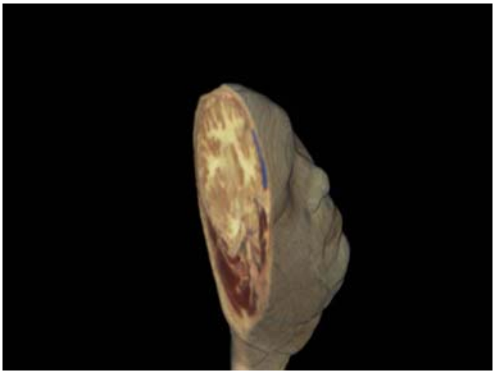


Fig. 3. Visible Man – volume cut

Table 1. Data glove sensors positions [2]

Gesture	**Sensors position**				**Gesture description**	**Fig**
	B	**C**	**D**	**E**		
0	0	0	0	0	Fist	4
1	0	0	0	1	Index finger point	5
2	0	0	1	0	Up yours	6
3	0	0	1	1	Two finger point	7
4	0	1	0	0	Ring finger point	8
5	0	1	0	1	Ring index point	9
6	0	1	1	0	Ring middle point	10
7	0	1	1	1	Three finger point	11
8	1	0	0	0	Little finger point	12
9	1	0	0	1	Howzit	13
10	1	0	1	0	Little middle point	14
11	1	0	1	1	Not ring finger point	15
12	1	1	0	0	Little ring point	16
13	1	1	0	1	Not up yours	17
14	1	1	1	0	Not index finger point	18
15	1	1	1	1	Flat hand	19

For movements we used the following gestures:

- Gesture 0: Rotate in y axis with an maximum 180° angle;
- Gesture 1: Apply the cut plane;
- Gesture 3: Rotate in x axis with an maximum 180° angle;
- Gesture 7: Rotate the cut plane in x and y axis with an maximum 180° angle;
- Gesture 8: Apply transparency;
- Gesture 9: Activate stereoscopic visualization;
- Gesture 14: Rotate the cut plane in z-axis with a maximum 180° angle.

As mentioned before, this three-dimensional virtual man was visualized on a cluster based 5-side CAVERNA digital. This implementation considers 5 high-end PCs with Graphics Accelerators and Volume Pro boards attached to them and high-speed synchronization provided by a Gigabit-Ethernet switch.

To support stereoscopy, the user has to use the StereoGraphics CrystalEyes shutter glasses with wireless infrared connection, with three shutter glasses.

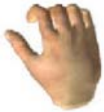


Fig. 4. Gesture 0

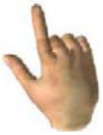


Fig. 5. Gesture 1

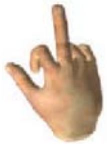


Fig. 6. Gesture 2

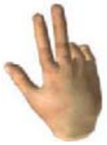


Fig. 7. Gesture 3

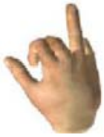


Fig. 8. Gesture 4

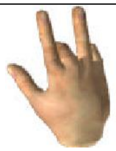


Fig. 9. Gesture 5

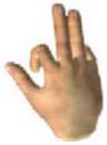


Fig. 10. Gesture 6

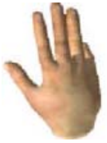


Fig. 11. Gesture 7

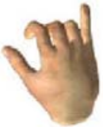


Fig. 12. Gesture 8

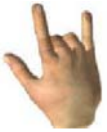


Fig. 13. Gesture 9

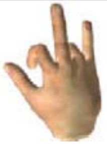


Fig. 14. Gesture 10

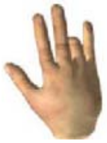


Fig. 15. Gesture 11

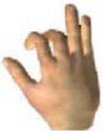


Fig. 16. Gesture 12

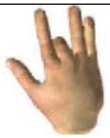


Fig. 17. Gesture 13

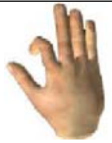


Fig. 18. Gesture 14

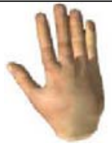


Fig. 19. Gesture 15

3 Conclusions and Future Work

This work presented a new proposal for segmenting Visible Human database for volumetric visualization on multiprojection immersive environment using the data glove and stereo shutter glasses. It consists with images segmentations and real time tri-dimensional rendering in Cave environment, using the VolumePro 500 board. The segmentation was made in C language and the visualization in C++ language, operational system independent. These languages were chosen due to their portability.

In a future work, another segmentation technique should be developed, this time to separate some parts of the human body, to generate a volume only from a region of interest. The user would be able to interact with some parts like the brain or the bones.

We should develop a freight distributor to improve the project performance too. As the Visible Human needs to be rendered by a maximum number of three windows simultaneously, we always have cluster nodes available that could be used to make a faster system. A tracker device must be added to provide better interaction with the data, tracking all the hand movements.

Acknowledgements. We thank the staff of CAVERNA digital, specially Leonardo Nomura and Breno Santos for their help and attention while developing this project.

References

1. Souza, I.A., Sanches-Jr, C., Binatto, M.B., Lopes, T.T., Zuffo, M.K. (2004). Direct Volume Rendering of the Visible Human Dataset on a Distributed Multiprojection Immersive Environment, Anais do VII Symposium on Virtual Reality (SVR'04), São Paulo, SP, Outubro, 183-194.
2. "Manual da Data Glove 5DT" disponível em http://www.5dt.com, acessed in March, 11-2005.

Author Index