BRAZILIAN SYMPOSIUM ON BIOINFORMATICS

August, 29 - 31, 2007
Angra dos Reis – Rio de Janeiro
Brasil

BSB 2007 Poster Proceedings

Conference Chair

Sérgio Lifschtiz
Department of Informatics - Pontificial Catholic University/Rio de Janeiro
Brazil

Volume Editors

Marie-France Sagot – INRIA – France
Maria Emília M. T. Walter – University of Brasília – Brazil

Promotion

Brazilian Computer Society – SBC

Brazilian Symposium on Bioinformatics (2007 : Angra dos Reis, RJ)
B827 BSB 2007 poster proceedings, August, 29-31, 2007, Angra dos Reis, Rio de Janeiro, Brasil / conference chair Sérgio Lifschtiz ; volume editors Marie-France Sagot, Maria Emília M. T. Walter. -- [Angra dos Reis] : Brazilian Computer Society, [2007].

1 CD-ROM.

ISBN 978-85-7669-123-5

1. Bioinformática – Congresso. I. Lifschtiz , Sérgio. II. Sagot, Marie-France. III. Walter, Maria Emília M. T. IV. Título.

CDD21 004
CDD21 570

Preface

The Brazilian Symposium on Bioinformatics (BSB 2007) was held in Angra dos Reis (Rio de Janeiro), Brazil, August 29-31, 2007, on the Portogalo Suite Hotel. It is an event promoted by the Brazilian Computer Society's (SBC) special committee for computational biology (CEBioComp). BSB 2007 was the second BSB symposium, though BSB is a new name for a predecessor event called Brazilian Workshop on Bioinformatics (WOB). This previous event had three consecutive editions in 2002 (Gramado, Rio Grande do Sul), 2003 (Macaé, Rio de Janeiro), and 2004 (Brasilia, Distrito Federal). The change from workshop to symposium reflects the increased quality and interest of the meeting. BSB 2007 was held co-located with the International Workshop on Genomic Databases (IWGD 2007).

For BSB 2007, we had 60 submissions: 36 full papers and 24 extended abstracts, submitted to two tracks, computational biology/bioinformatics and applications. The second track was created in order to receive and discuss research works with a biological approach, and so to reinforce the participation of the biologists on the event. From these, 13 full papers and 6 extended abstracts were selected to be published on the Lecture Notes on Bioinformatics (LNBI)/Lecture Notes on Computer Science (Springer-Verlag, Germany). These papers and abstracts were carefully refereed and selected by an international program committee of 48 members, with the help of some additional reviewers, all of whom are listed on the following pages. But, as we received many interesting works that could not be included on the LNBI, we decided to create another volume containing these other selected works. We believe that this poster proceedings represents a good contribution to research in bioinformatics and computational biology, as well as in molecular biology.

The editors would like to thank: the authors, for submitting their work to the symposium, and the invited speakers Roded Sharan (Tel-Aviv University-Israel), Alberto Martín Rivera Dávila (Fundação Oswaldo Cruz-Brazil) and João Paulo Kitajima (Allelyx Applied Genomics-Brazil); the program committee members and the other reviewers for their support in the review process; the general chair Sérgio Lifschitz and the local organizers Daniel Xavier de Sousa, Cristian Tristão and Paulo Roberto Gomes; the symposium sponsors (see list in this volume); and Nalvo Franco de Almeida Jr., João Carlos Setubal, José Carlos Mombach, Marcelo de Macedo Brígido, and again Sérgio Lifschitz, members of the CEBioComp.

August 2007

Marie-France Sagot
Maria Emilia M. T. Walter

Organization

BSB 2007 was organized by the Department of Informatics - Pontifical Catholic University of Rio de Janeiro/Brazil.

Executive Committee

Conference Chair:	Sérgio Lifschitz Pontifical Catholic University of Rio de Janeiro Brazil
Local Arrangements:	Daniel Xavier de Sousa Cristian Tristão Paulo Roberto Gomes Pontifical Catholic University of Rio de Janeiro Brazil

Scientific Program Committee

Program Chairs:	Marie-France Sagot INRIA, France *Computational biology and bioinformatics*
	Maria Emilia Machado Telles Walter University of Brasilia, Brazil *Applications*

Program Committee

Said S. Adi	(Federal University of Mato Grosso do Sul, Brazil)
Nalvo F. Almeida	(Federal University of Mato Grosso do Sul, Brazil)
Alberto Apostolico	(Accademia Nazionale dei Lincei and Georgia Tech)
Fernanda Baiao	(UNIRIO, Brazil)
Valmir C. Barbosa	(Federal University of Rio de Janeiro, Brazil)
Ana Lúcia Bazzan	(Federal University of Rio Grande do Sul, Brazil)
Marcelo M. Brígido	(University of Brasilia, Brazil)
Edson N. Cáceres	(Federal University of Mato Grosso do Sul, Brazil)
André P. L. F. de Carvalho	(University of São Paulo-São Carlos, Brazil)
Maria Cláudia Cavalcanti	(Military Institute of Engineering, Brazil)
Dominique Cellier	(University of Rouen, France)
Laurent Dardenne	(National Laboratory of Scientific Computation, Brazil)
Alberto M. R. Dávila	(Fiocruz, Brazil)
Zanoni Dias	(University of Campinas, Brazil)
Carlos E. Ferreira	(University of São Paulo, Brazil)
Ana T. Freitas	(Technical University of Lisbon, Portugal)
Richard Garrat	(University of São Paulo-São Carlos, Brazil)
Raffaele Giancarlo	(Universita degli Studi di Palermo, Italy)
Katia S. Guimarães	(NCBI, USA/ Federal University of Pernambuco, Brazil)
David Huson	(University of Tubingen, Germany)
João P. Kitajima	(Alellyx, Brazil)
Gunnar Klau	(Freie Universitt Berlin, Germany)
Gad Landau	(University of Haifa, Israel)
Melissa Lemos	(Pontifical Catholic University of Rio de Janeiro, Brazil)
Natalia Martins	(Embrapa/Biological Resources and Biotechnology, Brazil)
Wellington Martins	(Catholic University of Goias, Brazil)
Marta Mattoso	(Federal University of Rio de Janeiro, Brazil)
Alba C. M. A. Melo	(University of Brasilia, Brazil)
Antonio B. Miranda	(Fiocruz, Brazil)
Satoru Miyano	(The University of Tokyo, Japan)
José C. Mombach	(Federal University of Santa Maria, Brazil)
Nadia Pisanti	(University of Pisa, Italy)
Alexandre Plastino	(Federal Fluminense University, Brazil)
Leila Ribeiro	(Federal University of Rio Grande do Sul, Brazil)
David Sankoff	(University of Ottawa, Canada)
Luiz F. Seibel	(Pontifical Catholic University of Rio de Janeiro, Brazil)
João C. Setubal	(Virginia Bioinformatics Institute, USA)
Roded Sharan	(Tel-Aviv University, Israel)
David Sherman	(CNRS, France)
Siang W. Song	(University of São Paulo, Brazil)
Marcílio C. P. de Souto	(Federal University of Rio Grande do Norte, Brazil)
Osmar Norberto de Souza	(Pontifical Catholic University of Rio Grande do Sul, Brazil)
Guilherme P. Telles	(University of São Paulo-São Carlos, Brazil)
Cristina Vieira	(INRIA, France)
Sérgio Verjovski-Almeida	(University of São Paulo, Brazil)
Martin Vingron	(Max Planck Institute, Germany)
Michael Waterman	(University of Southern California, USA)
Fernando von Zuben	(University of Campinas, Brazil)

Additional Reviewers

Christian Baudet
Markus Bauer
Luciano Digiampietri
Alan Mitchell Durham
Cristina G. Fernandes
Ivan Gesteira Costa Filho
Alexandre Paulo Francisco
Ronaldo Fumio Hashimoto
Dennis Kostka
Alair Pereira do Lago
Helena Cristina Gama Leitão
Ana Carolina Lorena
Simone de Lima Martins
Mariá Cristina Vasconcelos Nascimento
Christian Rausch
Christine Steinhoff
Yoshiko Wakabayashi

Sponsoring Institutions

Brazilian Computer Society
CAPES

Special Committee for Computational Biology from the Brazilian Computer Society

Nalvo Franco de Almeida Jr.	Federal University of Mato Grosso do Sul - Brasil (coordinator)
João Carlos Setubal	Virginia Bioinformatics Institute - USA
José Carlos Mombach	Federal University of Santa Maria - Brasil
Marcelo de Macedo Brígido	University of Brasília - Brasil
Maria Emília M. T. Walter	University of Brasília - Brasil
Sérgio Lifschitz	Pontifical Catholic University of Rio de Janeiro - Brasil

Table of Contents

Selected works

Impaired Expression of NER Gene Network in Sporadic Solid Tumors

Mauro A. A. Castro(1), José C. M. Mombach(2), Rita M. C. de Almeida(3), and José C. F. Moreira(1)

(1)Departamento de Bioquímica, UFRGS. (2)Centro de Ciências Rurais, São Gabriel, UNIPAMPA/UFSM. (3)Instituto de Física, UFRGS. Rio Grande do Sul, Brazil

Abstract. Nucleotide repair genes are not generally altered in sporadic solid tumors. However, point mutations are found scattered throughout the genome of cancer cells indicating that the repair pathways are dysfunctional. We present here a statistical analysis comparing ten gene expression pathways in human normal and cancer cells using SAGE data. We find that in cancer cells nucleotide excision repair (NER) and apoptosis are the most impaired pathways and have a highly altered diversity of gene expression profile when compared to normal cells. We propose that genome point mutations in sporadic tumors can be explained by a structurally conserved NER with a functional disorder generated from its entanglement with the apoptosis gene network.

Keywords: Cancer, nucleotide excision repair, gene networks, SAGE

1 Introduction

Cancer cells have large and small abnormalities in their genetic material: additional or missing chromosomes, mutated genes and other types of alterations. The lost of genome stability pathways is associated with genetic deterioration of cancer cells and is one of the most important aspects of carcinogenesis. In fact, mutations in mismatch repair (MMR), nucleotide-excision repair (NER), base-excision repair (BER) and recombinational repair genes have been causally implicated in the acquisition of a genome instability phenotype [1].

Genome instability in solid tumors originates from either somatic mutations (observed in the majority of sporadic cancers) or germline mutations (associated to rare hereditary cancer syndromes). Considering the list of repair genes presented in Cancer Gene Census [2], germline mutations can be observed in NER, BER and MMR, while somatic mutations are described only in recombinational repair (homologous recombination and non-homologous end joining). On the other hand, mutations in apoptotic genes are recurrently observed in both types of solid tumors as listed in the census. The genotype signature of the mal-functioning of these stability gene networks is twofold: aneuploidy and/or random point mutations. The omnipresence of random point mutations in sporadic solid tumors and the recurrent absence of mutations in nucleotide repair

genes suggest a functional deficiency in these stability pathways without structural alterations in the related DNA sequence. In this work we present a comprehensive statistical analysis of ten gene expression pathways in normal and cancer cells using serial analysis of gene expression (SAGE) data [3] from the public gene expression resource available at Cancer Genome Anatomy Project (CGAP).

2 Methods

Human cancer and normal tissue SAGE libraries are retrieved using SAGE Library Finder tool at SAGE Genie website (http://cgap.nci.nih.gov/SAGE). In the SAGE database, a SAGE library corresponds to one tumor sample exam, which is made from mRNA extracts from different tissue preparations (bulk, short term culture, antibody purified, microscope dissected or cell line) and histology (cancer or normal). One such library gives the amount of every detected transcript in the sample, each one being labeled by a 10-letter tag, corresponding to 10 bases close to the poly-A tail, whose length is long enough to discriminate every possible transcript. Transcripts related to different gene networks may be grouped and used to quantify and characterize their expression activity. Here we analyze both the amount of transcripts production and its diversity in ten gene pathways, chosen due to either their recognized relation with genome stability (apoptosis, chromosome stability, mismatch repair, nucleotide-excision repair, base-excision repair and recombinational repair) or, as a control group, due to their essential life supporting activities (ribosome, ATP synthase, electron transport chain, and glycolysis). The tumor types were selected such that they present a library of normal cells, to be used as control.

To obtain a quantitative expression of sample distribution of SAGE tags, we have measured the information content of SAGE libraries using Shannon Information Theory defined as follows. Consider n as the number of all selected SAGE libraries of a given tumor type. Each library of this set is labeled by α ($\alpha = 1, ..., n$) and has N_α tags, among M_α possible ones, that is, possible transcripts. For a given SAGE library in this set, we can define $s(i, \alpha)$ as being the number of transcripts (tags) of a given type i, ($i = 1, , M_\alpha$), whose sum for a given α adds up to N_α. The probability $p(i, \alpha)$ that, among the N_α tags of the α-library, a randomly chosen transcript is of the type i is written as

$$p(i, \alpha) = s(i, \alpha)/N_\alpha \tag{1}$$

such that $\sum_i p(i, \alpha) = 1$.

The normalized entropy function H_α is defined as

$$H_\alpha = -\frac{1}{\ln M_\alpha} \sum_i^{M_\alpha} p(i, \alpha) \ln p(i, \alpha) \tag{2}$$

where we have divided all terms by the factor $\ln(M_\alpha)$ in order to normalize the quantities, guaranteeing that $0 \leq H_\alpha \leq 1$. The idea is to compare among samples

of different tissues that may present different numbers of M_α possibilities (e.g. different numbers of possible transcripts). While N_α reflects gene expression activity (the amount of tags in the α-th library), H_α reflects the spread of the distribution $s(i,\alpha)$, i.e., it measures the diversity that exists in the α-th library. Finally, in order to normalize the quantities by sets of tags, taking as reference normal tissue histology, we define the relative diversity h_α for any given set of genes as

$$h_\alpha = \frac{H_\alpha^c}{H_\alpha^c + H_\alpha^\gamma} \tag{3}$$

where H_α^c and H_α^γ are, respectively, the diversity of cancer and normal SAGE libraries. Observe that $0 \leq h_\alpha \leq 1$, and $h_\alpha < 0.5$ implies $H_\alpha^c < H_\alpha^\gamma$, that is, the transcript distribution in the α-th library is narrower in cancer cells than in the normal tissue, while $h_\alpha > 0.5$ represents the inverse case. In analogy, the relative gene expression activity n_α of the α library is defined as

$$n_\alpha = \frac{N_\alpha^c}{N_\alpha^c + N_\alpha^\gamma} \tag{4}$$

where N_α^c and N_α^γ are, respectively, the gene expression activity of cancer and normal tissue (i.e. number of SAGE tags). Again, $0 \leq n_\alpha \leq 1$, and $n_\alpha < 0.5$ implies $N_\alpha^c < N_\alpha^\gamma$, that is, in this library the cancer cells have lower gene activity, producing less transcripts than the normal case.

3 Conclusions

In order to consolidate these results and simultaneously compare all gene expression pathways, we present in Fig. 1A the average values of the relative activity n_α for each gene network. As we can observe, NER and apoptosis present the lowest amount of relative activity ($P < 0.001$), indicating an altered state of gene expression. In contrast, NER has the highest relative diversity ($P < 0.001$) (Fig. 1B), which shows that the low level of gene expression occurs together with changes on gene expression profile of this repair pathway. In cancer cells programmed cell death mechanism is in general structurally impaired, what is coherent with the observed gene expression profile of apoptosis transcripts. However, NER is in general structurally intact in sporadic solid tumors, since no somatic mutations in NER genes have been reported to be causally implicated in oncogenesis. The observed transcript profile then suggests that NER transactivation dependent functions are affected in cancer cells. As both apoptosis and NER networks are simultaneously affected, a causal correlation is plausible, considering that both networks are entangled. Concerning apoptosis and NER, $p53$ plays a key role, connecting both networks. In fact, there are many reports in the literature pointing $p53$ affecting both dependent and independent transactivation NER functions, as well as affecting apoptosis [4]. Also, it is reasonable to assume that damage in a specific gene function may affect its neighbors in the network, causing perturbations that may disrupt the whole network. NER malfunctioning could then account for random point mutations scattered throughout the cancer cell genome.

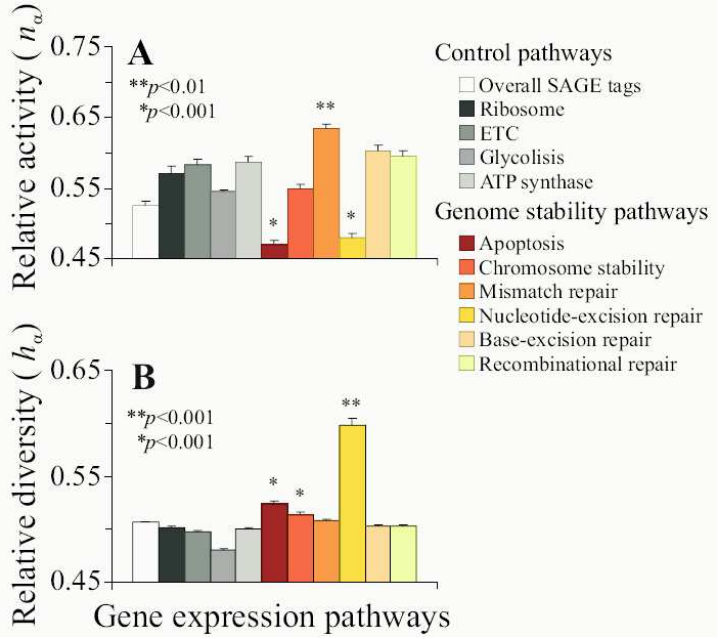


Fig. 1. Statistical comparisons among gene expression pathways according to diversity and number of SAGE tags. (A) Relative activity n_α as defined in Eq.(4). (B) Relative diversity h_α as defined in Eq.(3). The values are expressed as mean ± SEM (n=492). Statistical analyses were carried out using Kruskal-Wallis one-way analysis of variance test followed by Mann-Whitney test for comparisons. *Different from controls with $P < 0.001$; **different from others with $P < 0.001$. [3]

References

1. Hoeijmakers, J.H.J.: Genome maintenance mechanisms for preventing cancer. Nature **411** (2001) 366–374.
2. Futreal, P.A. *et al.*: A census of human cancer genes. Nat. Rev. Cancer **4** (2004) 177–183.
3. Castro, M.A.A., Mombach, J.C.M., de Almeida, R.M.C., Moreira, J.C.F.: Impaired Expression of NER Gene Network in Sporadic Solid Tumors. Published by Oxford University Press. All rights reserved. Nucleic Acids Res. **35** (2007) 1859–1867.
4. Sengupta, S., Harris, C. C.: *p*53 traffic cop at the crossroads of DNA repair and recombination. Nat. Rev. Mol. Cell Biol. **6** (2005) 44–55.

A tool for cluster number estimation in SOM-based gene expression pattern analysis

Elmer A. Fernández[1,2], Mónica Balzarini[1,3]

[1] CONICET, Argentina,
[2] Engineering College, Catholic University of Córdoba, Argentina
[3] Statistics&Biometry, Agricultural College, National University of Córdoba,, Argentina.
{elmerfer, mbalzari}@gmail.com

Abstract. Cluster methods are crucial to study genomic patterns of coexpressed genes. Neural network algorithms such as self organizing map (SOM) have been extensively used to cluster gene expression data. Result visualization techniques are important tools for cluster recognition in SOM. In this work a simple tool that implements an algorithm to identify and visualize clusters is proposed. It is based on two concepts (Relative Position and Q statistics) that can be applied to a SOM network. The Relative Position is a new SOM node-adaptive attribute defined from the node moving within a two dimensional space imitating the movement of the SOM codebook vectors in the input space. By means of the Q statistics the algorithm evaluates the SOM structure providing an estimate of the number of clusters underlying the data. The tool allows the visualization of the cluster node patterns facilitating cluster interpretation.

Keywords: Self Organizing Maps, data visualization.

1 Introduction

The amount of data generated in gene expression experiments with DNA microarray technology is huge and it should be analyzed under a knowledge discovery framework [1], which is an overall process of finding and interpreting patterns from data where data mining is one of the crucial steps. One of the main data mining tasks in the analysis of Gene Expression Patterns (GEP) is to find sets of similar genes (clusters) which are biologically useful and functionally meaningful. Even though cluster analysis has proved to be a powerful tool to investigate "natural" clusters of GEP, it has several drawbacks, for instance it presents difficulties in the estimation of an optimal number of clusters to describe the data structure.

Nevertheless, several clustering techniques have been increasingly applied in GEP, range from hierarchical clustering [2,3], clustering by simulated annealing [4] to neural network algorithms such as Self-Organizing Map or SOM [5,6,7,8]. Self Organizing Maps have coupled with different strategies to identify clusters on the net which span from the interpretation of each SOM-node as a cluster [6] to the use of specific SOM visualization tools such as the U-Matrix [9, 11] . The U-matrix method

builds a bi-dimensional color grid (usually in gray-scale) where on top of each node the value of the average distance between adjacent nodes is displayed. In this way light-colors stands for short distance (a valley) and dark-colors for larger distance (a hill). In this way the cluster are identified by visual inspection. However, there some nodes representing a mixed pattern between clusters already "learned" in the SOM structure. These are known as transition nodes (TN) and they could be misinterpreted as clusters.

Most of the visualization methods for SOM works on the codebook vectors which have the same dimensionality as the input space and they can only be applied once SOM training is finished. The visualization method implemented here works in a bi-dimensional virtual geometric space (VGS) spanned by means of new coordinates – the Relative Position (RP) - introduced in the SOM algorithm. The RP of the nodes mimics the movements of the codebook vectors in the input space over the reduced VGS as the SOM training process takes place. In this way, the final position of the nodes in the VGS resembles the movements and final position of the codebook vectors in the input space. After this process, the RP are displayed and the cluster is easily recognized. The tool implements two main concepts: 1) "The Relative Position " (RP) which is used to improve visualization of groups of SOM and 2) "The Q statistic" used for cluster number estimation handling transition nodes. These concepts were deeply tested with artificial and real data sets [12].

2 Material and Methods

2.1. Procedures

2.1.1. The classical SOM algorithm

Assuming a p-dimensional input space during an iterative SOM training process, the input sample $X_g = \{x_{g1}, x_{g2}, ..., x_{gp}\}$ is presented to the net (e.g. bidimensional array) and the Best Matching Node (BMN) or "winning" node $W_{ij} = \{w_{ij}^1, w_{ij}^2, ..., w_{ij}^p\}$ is found. This BMN and its neighbors are updated according to the following equation

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t) \cdot h(t, ij, i'j') \cdot \left\| X_g - W_{ij}(t) \right\| \quad (1)$$

where "α" is a decreasing gain function, "t" is the iterative step, "ij" is the node's position on the SOM, "$i'j'$" the BMN position, "h" is a decreasing neighborhood function centered at the BMN position. The values of the codebook vectors tend to be similar to the inputs in such a way that, at the end of the training process, similar cases remain close to each other on the map. As result of the SOM algorithm, the p-dimensional data is usually represented in a two-dimensional display. Elements of the input data that are close together in the input space will be arranged, after the learning process, close to each other in the SOM structure. This is the central idea supporting the SOM algorithm. Each node in the SOM array structure is usually characterized

both by its unique identifier (node index or row-column index) in the array (the array index position, AIP) and by its codebook vector.
Clusters are recognized as a group of nodes in the SOM array structure rather than considering each cluster as a node. However, the algorithm nature of the SOM net will set nodes in intermediate places between clusters, the transition nodes (TN) and the patterns stored in them could be a "mix" of the flanking cluster patterns. In the U-Matrix visualization method, these transition nodes appear to be black, or "hill" nodes, biasing interpretations.

2.1.2 The proposed RP-Q method

Imagine the array as a two-dimensional geometric space and add another cartesian attribute to the nodes. This new attribute in the VGS will be named relative position (RP). In Figure 1, each node has a unique identity, the row-column array position or array index position (AIP) (showed as the square boxes) and also a cartesian position within the VGS characterized by coordinates $\{x,y\}$. A node at AIP "*ij*" will be associated to a *p*-dimensional weight vector $\left(W_{ij} = \left\{w_{ij}^1, w_{ij}^2, ..., w_{ij}^P\right\}\right)$ and a coordinate pair $\left(RP_{ij} = \left\{x_{ij}, y_{ij}\right\}\right)$. The relative positions of the nodes are initially assigned in random locations all over the two-dimensional VGS (circles in Figure 1). During the training process they are allowed to migrate towards the array index position (AIP) of the BMN in the array. The RP of the BMN moves toward its own array index position (black squares in Figure 1). The RP of the neighboring nodes (circles in Figure 1) moves toward the updated RP of the BMN. The learning rule for the RPs is as follows:

$$RP(t+1)_{kl} = RP(t)_{kl} + k \cdot \alpha(t) \cdot h(t, kl, ij) \cdot \left|\{k,l\} - RP(t)_{kl}\right| \qquad (2)$$

where "*kl*" is the array index position of the BMN, $k<1$ is a smoothing adaptive factor.

$$RP(t+1)_{ij} = RP(t)_{ij} + k \cdot \alpha(t) \cdot h(t, kl, ij) \cdot \left|RP(t+1)_{kl} - RP(t)_{ij}\right| \qquad (3)$$

During the last part of the training period (convergence phase [10]) the $\{k,l\}$ term in eq. (1) is replaced by RP_{kl}. As a result, the RPs mimics the movements of the codebook vectors in the input space and after adaptation, the RPs of the BMN's neighbours tend to be closer to the RP of the BMN (Figure 1).

The nodes Relative Position coordinates are displayed as a kind of scatter plot. Each node is represented by a circle whose diameter is proportional to the activation frequency (winning frequency) during the learning phase. The nodes are also linked to each other if the distance between them in the Relative Position Space or in the weight space is less than or equal to a threshold value $\left(D_{ij-i'j'} \le Thr\right)$.

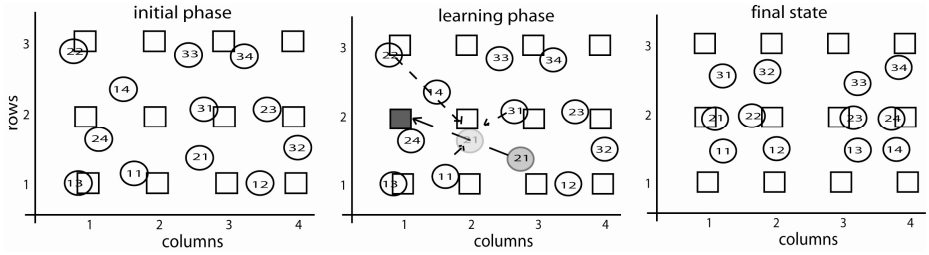


Figure 1: The RP concept and learning phase.

The nodes that are linked together are considered belonging to the same cluster. In this way, the nodes will be linked to each other (to form clusters) if the distance between them in the weight space is less than or equal to a specified threshold value. To estimate the number of clusters in a SOM with "*N*" nodes, we will set different threshold values, and evaluate the net structure at each threshold, via the weight vectors of the net, i.e. $\boldsymbol{W}_{ij}$ with "*ij*" being the node's position on the grid. In order to handle transition nodes (often present in SOM), thus, the qualified clusters are those with more than two nodes.

To estimate the number of qualified clusters, we obtain $d_{ij,i'j'}$, the distance (squared Euclidean distance) between the weight vectors "*ij*" and "*i´j´*". If the SOM has been partitioned in k clusters of nodes (C_1, C_2,…,C_k) at a particular threshold, being nr the number of nodes in cluster *r=1..k*, the sum of pair-wise distances for all nodes in cluster "*r*" is:

$$D_r = \sum_{ij,i'j' \in C_r} d_{ij,i'j'} \qquad (\mathbf{4})$$

and the pooled within cluster sum of squares around the cluster mean is:

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \qquad (\mathbf{5})$$

Hence, $\boldsymbol{W}_k$ is a measure of the within cluster "heterogeneity" of SOM node patterns. Additionally, the between cluster heterogeneity at the same particular threshold value defining *k* clusters is represented by

$$B_K = \sum_{r>r'} \left[d^*_{r,r'} \left(\frac{n(r) \cdot n(r')}{n(r) + n(r')} \right) \right] \qquad (\mathbf{6})$$

where $d^*_{r,r'}$ is the squared distance between the mean weight vectors of clusters C_r and $C_{r'}$ and n(.) the number of objects (data) in the cluster. In the RP-Q method, the resulting SOM structure at the threshold defining *k* clusters will be evaluated by means of the following statistic:

$$Q_k = \left(1 - \frac{nonQC}{k}\right)\left(\frac{B_k}{B_k + W_k}\right) \quad (7)$$

where *nonQC* and *k* are the number of non qualified clusters and the number of total clusters (*k*=*nonQC*+*QC*), respectively. The estimate number of qualified clusters is reached at the threshold value which maximizes Q_k.

The SOM-RP training system was implemented in C++ for Windows®, and the Q statistic and the RP visualization strategy were coded in Matlab® 5. The Matlab® tool allows opening a trained SOM-RP and visualizing the net structure based on different views (UMatrix, Relative Position, Array Index Cluster, Weight Plots). Main tool characteristics are shown in Figure 2: **RP**: Relative Space view, **TSL**: Threshold link slider tool, **CI**: Cluster Quality Information, **NI**: Node label information, **CP**: Cluster/Node pattern view, **TI**: Threshold link value information **CM**: Cluster Map view based on Array position. The Threshold link slider tool allows changes in the threshold link value to find out new clusters inside the first ones.

2.2 Data

The method was deeply tested in different artificial and GEP data sets in [12]. To illustrate the application of the presented tool, two well-known biological data sets were chosen:

1. The Leukemia Data Set (LDS) which consists of 38 bone marrow samples, where 27 belong to Acute Limphoblastic Leukemia (ALL) cells and 11 to Acute Myeloid Leukemia (AML) cells, obtained from acute leukemia patients during the diagnosis. [8]
2. The rat central nervous data set (RCNS) involving mRNA expression of 112 genes during rat central nervous system development, focused on the cervical spinal cord. The data provides a temporal gene expression ‘‘fingerprint’’ of spinal cord development based on major families of inter- and intracellular signaling genes. [7]

Both data sets (freely available) were preprocessed as in the original publications. For each data set different SOM net sizes were trained before to apply the RP-Q method, and the U-Matrix (http://www.cis.hut.fi/) was also employed as visualization tool for comparison purposes.

3 Results

For the LDS case, two nets were trained, one of them with 36 nodes (a 6x6 array with almost one node per sample) and other with 20 nodes (an array of 4x5). In Figure 2, the RP-Q tool shows the analysis of the smaller net. In the RP view panel two clusters could be identified, C_1 and C_2 representing the AML and ALL cancer types respectively.

In the NI panel it is possible to see that a node at array position {2,4} represent to B cells. The tool also let us to see the node or cluster pattern in panel CP. In Figure 2 the C_2 cluster pattern is shown.

The threshold link slider (TLS panel) allows us to change the value of the threshold link (cut-off between node distance values) to either find out new clusters inside the first ones or analyze the different relationships between nodes and clusters. For instance, in the 4x5 SOM net was possible to find a sub-cluster inside the cluster representing ALL cancer type. In this way the two cell types present in this cancer type emerged.

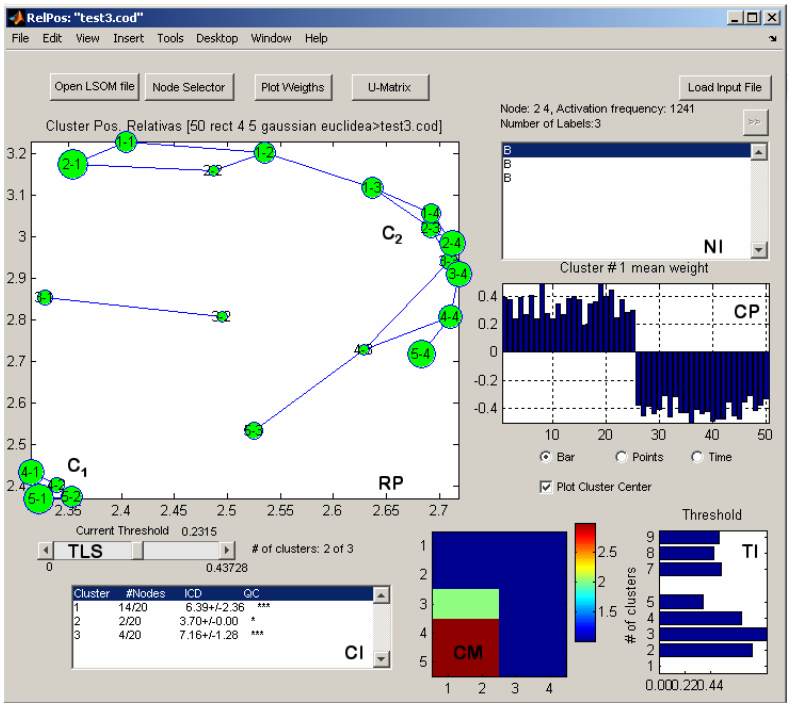


Figure 2: The tool to apply the RP-Q method.

In Figure 3 is presented the results from the bigger SOM where it is possible to compare the RP visualization method and the U-Matrix plot.

In this bigger SOM net the ALL cancer type was naturally split in their two cell types as clusters, C_2 and C_3 hold the B and T cell types. Cluster C_1 is the same as the one found in the 4x5 SOM net. In the RP plot, it was also identified two linked nodes as a non qualified cluster (non-QC). By means of the NI panel it was verified that no samples where associated to these nodes. The identification of clusters by means of the U-matrix is not as straightforward as it is from the RP-Q method.

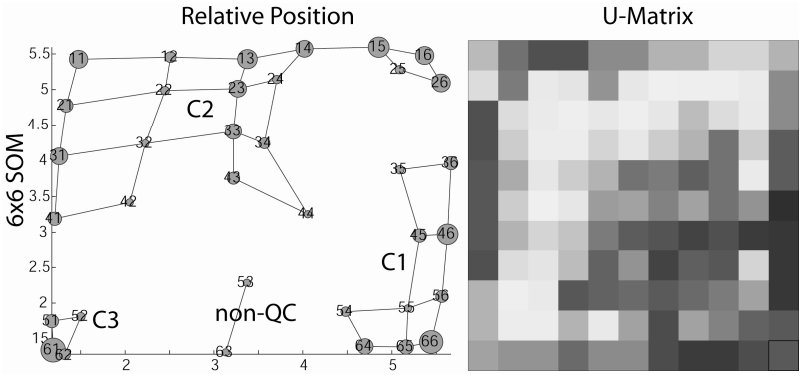


Figure 3: The RP visualization (left) and U-Matrix (right) for the two SOM nets trained with a leukemia data set.

In the second example we use the RNCS data set. It consists of 112 genes with relative expression over nine different time intervals; six clusters were originally reported and biologically justified [7]. Figure 4 compares the RP (left panels) and U-matrix (right panels) methods. In the RP plot the six clusters can be visualized. For the smaller net one of the reported clusters where identified as a non-QC cluster by means of the RP-Q method. These clusters present the smallest number of samples in the data set. By means of the U-matrix, the clusters are not completely recognized.

In Figure 5 the RP-Q tool is used to analyze the data set with a 6x6 SOM net, from which it is possible to identify all the clusters and interactively identify the genes belonging to each cluster with their corresponding gene expression pattern. In Figure 5 it is also possible to identify a transition node (*TN*). This is not possible from the U-matrix (Figure 4).

4 Discussion

The identification of clusters in GEP data is a very common task in the biological research [2-6]. The Self-Organized Map algorithm provides a way to map GE patterns [9].However, the analysis of the trained SOM demands visualization methods able to produce an estimate of the cluster number underlying the data [10,11]. In this work tool for better visualization and cluster number estimation in SOM is presented that implements the concepts of Relative Position for visualization and Q statistic for cluster number estimation [12]. The visualization of the SOM node distribution in the relative position space yields a more objective view of the SOM structure than the U-Matrix, one of the most commonly used methods to visualize SOM results [11].

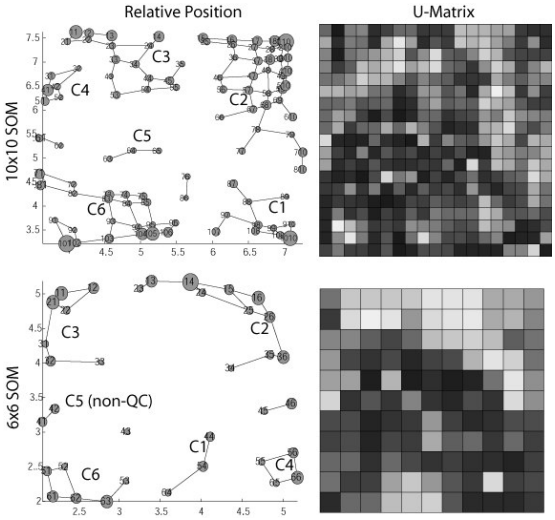


Figure 4: Clusters visualized by means of the RP-Q method (left) and U-matrix (right) for two SOM nets trained from a RCNS with six clusters.

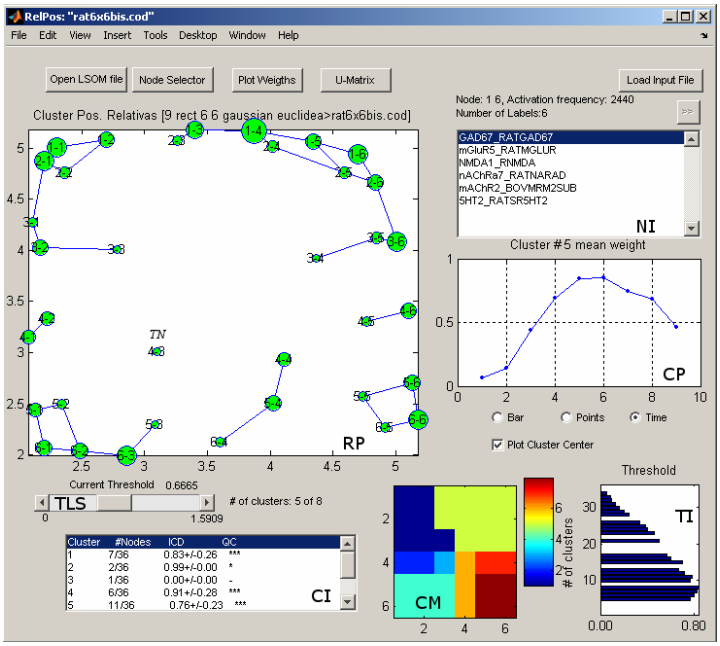


Figure 5: The RP-Q tool for the analysis of the Rat data ser with a 6x6 SOM net. *TN* is a transition node. The genes associated to node at array position {1,6} are shown in the NI panel and the expression pattern in CP for it's cluster.

The RP principle resembles the main idea of the Adaptive Coordinate method [14]. However, the RP-Q method has less computational overload because it does not need to store in memory the distance matrix between the codebook vectors and each input, nor does it need to calculate the relative change matrix. The RP-Q method also provides the "link between nodes" which helps to visualize the clusters and the relationship between nodes and clusters. This link (threshold value) is also useful to find sub-clusters and possible pathways of new input data. Moreover, the estimation of the number of clusters by means of the RP-Q method is invariant across runs once the net has been trained

The development of the Q statistic in the RP space was designed to settle a threshold value able to estimate the number of qualified clusters underlying the SOM structure. This is a good place to start the interactive analysis of the SOM structure. The random variable $\boldsymbol{W}_k$ involved in the calculation of the Q statistic is the kernel of the Gap statistic proposed by Tibshirani et al. [13] to estimate the cluster number after the application of a k-means algorithm to group p-dimensional objects. Here we use it over the weight vectors of the SOM and we show that it performs well under different SOM sizes and different data bases [12]. The basic relationship of within and between cluster sums of squares used by others´s procedures, was adjusted to take into account non-qualified clusters. Therefore, the Q statistic not only contrasts between vs. within cluster heterogeneity, but also takes into account the quality of the SOM structure emerging at each threshold.

References

1. Piatesky-Shapiro G. Tamayo P.: Microarray Data Mining: Facing the challenges. SIGKDD Explorations (2005) 5:2:1
2. Eisen MB. Spellman PT. Brown PO. Botstein D.: Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.* (1998), 14863-14868
3. Levenstien MA. Yang Y. Ott J.: Statistical significance for hierarchical clustering in genetic association and microarray expression studies. BMC Bioinformatics (2003)
4. Lukasin AV. Fuchs R.: Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics* (2001).17:5, 405-414
5. Töronen P. Kolehmainen M. Wong G. Castrén E.: Analysis of gene expression data using self-organizing maps. *FEBS Letters* (1999)451, 142-146
6. Tamayo P. Slonim D. Mesirov J. Zhu Q. Kitareewan S. Dmitrovsky E. Lander ES. Golub TR.: Interpresting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci.* (1999), 2907-2912.
7. Wen X. Fuhrman S. Michaelis GS. Carri DB. Smith S. Barker JL. Somogyi R. Large-scale temoral gene expression mapping of central nervous system development *Proc. Natl. Acad. Sci.* (1998), 334-339

8. Golub TR. Slonim DK. Tamayo P. Huard C. Gaasenbeek M. Mesirov JP. Coller H. Loh ML. Downing JR. Caligiuri MA. Bloomfield CD. Lander ES.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science* (1999)286, 531-537
9. Wang J. Delabie J. Aasheim HC. Smeland E. Myklebost O.: Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study. *BMC Bioinformatics* (2002)3:36
10. Kohonen T.: Self-Organizing Maps. 2nd Ed. *Springer* , (1997)Berlin
11. Vesanto J. : SOM-based data visualization methods. *IDA* (1999)3, 111-125
12. Fernández EA. Balzarini M.: *Improving cluster visualization in Self-Organizing Maps: Application in Gene Expression Data Analysis*. Computers in Biology and Medicine. In press. (2007)
13. Tibshirani R. Walther G. Hastie T.: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (2001) 63,2,441
14. Merkl D. Rauber A.: Alternative Ways for cluster visualization in Self-Organizing Maps. (1997) Workshop on SOM'97.

EGGview: Visualization of EGG Comparative Genome Data Using GBrowse*

Nalvo F. Almeida**, Marcel Y. Nakazaki, Andrey A. Tamura, Luciana Y. Hiratsuka, André C. Lima, Said S. Adi, Carlos J.M. Viana, and Leandro P. Brazil

Department of Computing and Statistics,
Federal University of Mato Grosso do Sul
CP 549, 79070-900, Campo Grande, MS, Brazil
nalvo@dct.ufms.br
http://egg.dct.ufms.br/projects

Abstract. Comparative genome analysis is very useful to help scientists in understanding functional and evolutionary issues of species. EGG is a tool for whole genome pairwise comparison that compares all-against-all predicted proteins of two genomes and finds, among others, pairs of homologous genes and specific genes of both genomes. We present EGGview, a simple and easily configurable approach to visualize comparative genome data from EGG output. EGGview runs on the top of the well-known web-based GBrowse framework.

Key words: genome comparison, gbrowse, comparative data visualization

1 Introduction

The increasing availability of genome data brings the need for tools to analyze and compare them, in order to gain clues about common functionalities and to get better understanding about changes in gene organization between related species. Whole genome comparison, specifically involving gene content and gene order conservation, is a powerful tool for studies of genomic evolution [8, 10].

A tool called EGG (Extended Genome-Genome comparison) [1] makes whole genome pairwise comparison by finding all pairs of orthologous genes using BLASTP program [2] (other versions of BLAST programs may be used). The comparison takes all the predicted proteins of both genomes into account, following all-against-all fashion. After that, a bipartite graph is built, where an edge represents a pair of orthologous genes. An edge of this graph is called a *match*. Formally, a match is a pair (g, h) of genes whose BLASTP e-values (both ways) are not greater than 10^{-5} and the alignments include at least 60% of each sequence. Note that a gene can be in several matches. When a gene h is the best

* This work was supported by CNPq and Fundect-MS.
** To whom correspondence should be addressed.

BLASTP hit found by g and vice versa, we have a *bi-directional best hit (BBH)*. Thus, a gene can participate at most of one BBH. When a gene g found no BLASTP hits on the other genome, we say that g is a *specific* gene.

After graph construction, EGG looks for organization structures in it, called *orthologous regions*. Basically, an orthologous region is a region in both genomes of closely matches [1]. Recent features of EGG package include the comparison of incomplete genomes, without any information about the structure of the genome. This program is called EGG-Lite, and assumes as input only multi-fasta files with the predicted proteins of both genomes.

EGG has been used successfully in several genome projects [3–6, 9, 13]. The main problem is its lacking of graphical visualization. The goal of this work is to present an approach to visualizing EGG with the use of the GBrowse framework [11], which is a combination of database and web-page tool for displaying and searching genomic annotations. By using this portable and flexible application, one can have detailed views of a genome. Tracks may be created and customized by the user. GBrowse has been widely adopted by genome projects, mainly because its support for third party annotation, made through GFF (General Feature Format) formats.

GBrowse is not suitable for comparative analysis, since does not support links between tracks. Thus, there is a need to develop independent tools for visualizing genome comparisons. In order to overcome this need and avoid effort duplication, we propose EGGview, a tool for visualizing comparative data from EGG, such as BBHs and specific genes. This is made by scripts that build GFF files to be used by GBrowse. EGGview inherits GBrowse functionalities and shows genome pairwise comparison and its functional annotations in a same environment.

2 Implementation

EGGview scripts, such as more examples of cross-species comparisons are available for download at `http://egg.dct.ufms.br/projects`. The following software environment must be installed before using EGGview: Perl 5.005 or higher (www.perl.org); BioPerl 1.5.2 or higher (www.bioperl.org); Apache Web Server 2.0 or higher (www.apache.org); and GBrowse 1.66 or higher (www.gmod.org).

EGGview is very simple to use, just by running one of the Perl scripts directly from EGG output files, `egg2gff3.pl` for EGG or `egg-lite2gff3.pl` for EGG-Lite. Optionally, both scripts can be run by adding corresponding options when EGG or EGG-Lite are called. The job to be done by the scripts is to build the GFF and the configuration files to be used by GBrowse. The configuration file is used to manage information to the entire system, and can be set by the user. All annotation data except EGG output ones are in the same format as required by GBrowse. Bio::DB::GFF schema should be used to store all data in a database.

3 Discussion

The main features of EGGview are to map genes of the query genome into real coordinates of the reference genome, based on the BBHs, and to show specific reference genome genes. Thus, at least the reference genome must be complete. EGG has been used in the *Xanthomonas* project [3], where *X.axonopodis* pv *citri* was compared to *X.campestris* pv *campestris*. Figure 1 illustrates the use of EGGview, choosing *X.campestris* as reference. An additional and useful feature was implemented, in which a pop-up window shows up when the user mouses over the BBH. In this window, both BLAST reciprocal alignments can be seen in HTML or in PDF format. Figure 2 illustrates this feature.

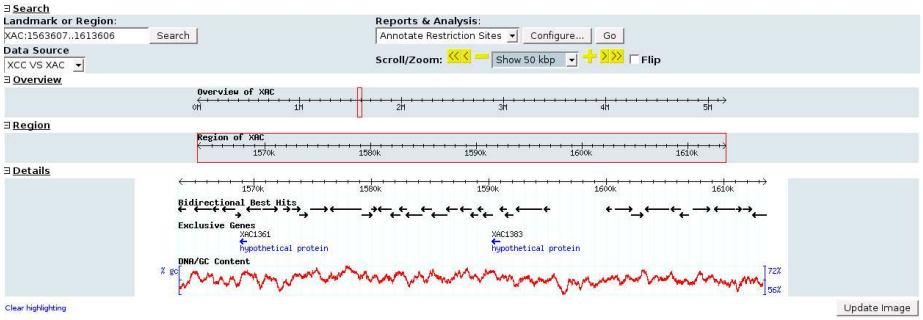


Fig. 1. Main features for the comparison of two *Xanthomonas*.

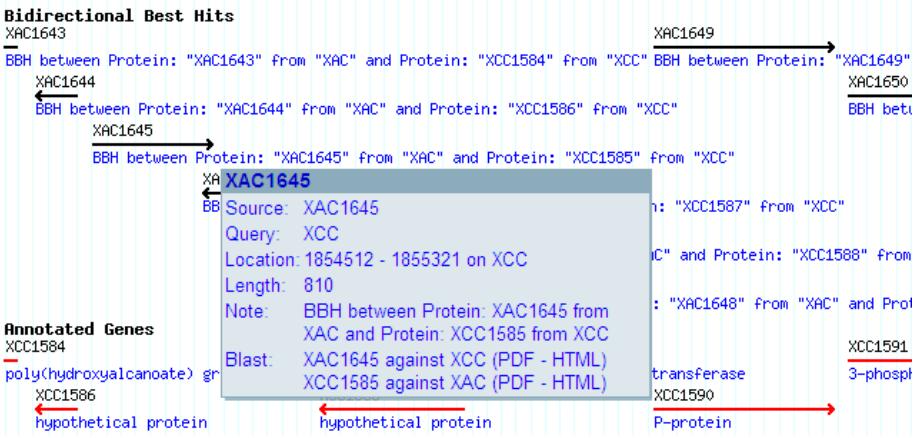


Fig. 2. Pop-up illustrating information about a BBH found between two *Xanthomonas*.

Other GBrowse-like visualization approaches that show synteny information have been published [7, 12]. Our goal here is not to compete with those works. Instead, we are interested in providing a graphical interface specifically to EGG package, that is intensively used by our group and by several genome projects [3–6, 9, 13]. EGGview is still being analyzed and much remains to be done. For example, the next step we are work on is to include the orthologous regions when two complete genomes has been compared.

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. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acid Research*, 25:3389–3402, 1997.
3. A.C. Rasera da Silva, J.C. Setubal, and N.F. Almeida *et al.* Comparison of the genomes of two *xanthomonas* pathogens with differing host specificities. *Nature*, 417(6887):459–463, 2002.
4. M.S.S. Felipe, M.E.M.T. Walter, M.M. Brgido, and N.F. Almeida *et al.* Transcriptome characterization of the dimorphic and pathogenic fungus paracoccidioides brasiliensis by est analysis. *Yeast*, 20(3):263–271, 2003.
5. C.B. Monteiro-Vitorello, L.E.A. Camargo, J.C. Setubal, and N.F. Almeida *et al.* The genome sequence of the gram-positive sugarcane pathogen leifsonia xyli subsp. xyli. *Molelular Plant-Microbe Interactions*, 17(8):827–836, 2004.
6. L.M. Moreira, R.F. de Souza, N.F. Almeida, J.C. Setubal, J.C.F. Oliveira, L.R. Furlan, J.A. Ferro, and A.C.R da Silva. Comparative genomics analyses of citrus-associated bacteria. *Annu. Rev. Phytopathology*, 42:163–184, 2004.
7. X. Pan, L. Stein, and V. Brendel. Synbrowse: a synteny browser for comparative sequence analysis. *Bioinformatics*, 21(17), 2005.
8. J.C. Setubal and N.F. Almeida. Detection of related genes in procaryotes using syntenic regions. In *DIMACS Workshop on Whole Genome Comparison*. DIMACS Center, Rutgers University, February 2001.
9. M.A. Van Sluys, J.C. Setubal, and N.F. Almeida *et al.* Comparative analyses of the complete genome sequences of pierce's disease and citrus variegated chlorosis strains of xylella fastidiosa. *J. Bacteriology*, 185(3):1018–1026, 2003.
10. B. Snel, G. Lehmann, P. Bork, and M.A. Huynen. STRING: a web-server to retrieve and display the repeatedly occuring neighbourhood of a gene. *Nucleic Acids Research*, 28:3442–3444, 2000.
11. L.D. Stein, C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J.E. Stajich, T.W. Harris, A. Arva, and S. Lewis. The generic genome browser: a building block for a model organism system database. *Genome Res.*, 12(10):1599–1610, Oct 2002.
12. H. Wang, Y. Su, A.J. Mackey, E.T. Kraemer, and J.C. Kissinger. Synview: a gbrowse-compatible approach to visualizing comparative genome data. *Bioinformatics*, 22(18):2308–2309, 2006.
13. D.W. Wood, J.C. Setubal, and N.F. Almeida *et al.* The genome of the natural genetic engineer *agrobacterium tumefaciens* c58. *Science*, 294:2317–2323, December 2001.

Topological Indices and Graph Theory: a Useful Tool for the Characterization of Peptides?

Alan Talevi[1,2], Carolina L. Bellera[1] and Luis E. Bruno-Blanch[1]

[1] Medicinal Chemistry, Department of Biological Sciences, Faculty of Exact Sciences, Universidad Nacional de La Plata, B1900AVV, La Plata, Buenos Aires, Argentina, lbb@biol.unlp.edu.ar

[2] Instituto de Investigaciones Fisicoquímicas Teóricas y Aplicadas (INIFTA), Department of Chemistry, Faculty of Exact Sciences, Universidad Nacional de La Plata, B1900AVV, La Plata, Buenos Aires, Argentina.

Abstract. Recently, interest has grown in the prediction of properties and function of peptides and proteins, arising the need to identify molecular descriptors suitable for establishing structure-property relationships. In order to prove if topological indices derived from Graph Theory could be applied in the characterization of peptides, we have generated two sets of peptides: one composed by the 120 possible pentapeptides obtained from permutations of five amino acids; the other composed by 30 octapeptides obtained from permutations of eight amino acids. Dragon software was applied in the calculation of 578 topological descriptors. The range of values assumed by each descriptor, its information content (quantified through Shannon Entropy) and degree of degeneracy are proposed as criteria for the selection of the descriptors which best characterize the peptides sets. Results indicate that topological indices may be valuable in the design of peptides with therapeutic applications and, in time, for the characterization of proteins.

Keywords: Topological Indices – Peptides – Drug Design

1 Introduction

The recent completion of the human genome and the fact that thousands of organisms are being sequenced at present has changed the face of Molecular Biology and Medicinal Chemistry. The genetic basis of disease are being identified, together with the potential ability to interfere with it. New potential therapeutic targets are being revealed as possible alternatives to the relatively few traditional targets. However, potential targets provided by genome projects are not usually accompanied by a meticulous understanding of their function. With this background, new ways of capturing structural information of peptides and proteins are needed, in order to establish relationships between their structure and their properties and functions. This study aims to determine if topological indices (TIs) may be suitable structural descriptors to characterize peptides and, in time, proteins.

The term TIs refers to molecular descriptors derived from molecules representations called graphs, in which atoms are represented as vertices and covalent chemical bonds are represented as edges. In graphs, geometrical features of molecules (such as bond length and angle) are not explicitly considered and connectivity of atoms is regarded as their fundamental characteristic. Using molecular graphs, the chemical structure of an organic compound may be expressed through graph matrices, polynomials, spectra, spectral moments, counts of paths and walks and, finally, TIs derived from algebraic operations on any of these elements. They offer a simple way of measuring molecular size, shape, branching and complexity, and molecular similarity [1,2]. The most common used TIs are derived from the adjacency matrix *A* (whose elements take values of one if the associated atoms are directly connected by a covalent chemical bond and zero otherwise) and the distance matrix *D* (with each element taking the value of the topological length of the shortest path between two atoms) [2-3]. The elements of *A(n x n)* can be expressed as follows, *n* being the number of atoms in the molecule:

$$A_{ij} = \begin{cases} 1 & \text{if atoms } i \text{ and } j \text{ are bonded by a covalent bond} \\ 0 & \text{if atoms } i \text{ and } j \text{ are not bonded by a covalent bond} \end{cases} \quad \textbf{(1)}$$

The elements of D(n x n) are defined as follows, np being the length of the shortest path between the vertices i and j:

$$D_{ij} = \begin{cases} 0 & \text{if } i = j \\ n_p & \text{if } i \neq j \end{cases} \quad (2)$$

For exemplification purposes, Fig. 1 shows the graph derived from the molecule of phenol and both A and D matrices.

TIs have been widely applied in drug discovery for the characterization of active compounds and the rational search of new therapeutic agents [4-6]. Our group has successfully applied them in the process known as virtual screening, identifying new anticonvulsants and antichagasic agents from large compound databases of 10^4 to 10^5 drug-like compounds [7-10].

In the last years, interest has grown on peptides as source of new therapeutic agents [11-13]. Besides, as mentioned, the completion of the human genome six years ago has highlighted the importance of techniques for the prediction of proteins structure and function. In Medicinal Chemistry, improvement of those techniques could serve to optimize the target selection and the drug design process [13, 14].

Could TIs be used to predict the biological activities of peptides, in a similar way as they are used to predict the activities of traditional drugs? Is it possible to apply descriptors derived from Graph Theory in the prediction of a protein function or structure, just from the knowledge of its amino acid sequence? The present paper is a first step towards answering the former of these issues. We have evaluated the potential ability of 578 TIs to characterize the 120 pentapeptides that can be obtained

from permutation of five different amino acids and 30 octapeptides obtained from permutations of eight different amino acids.

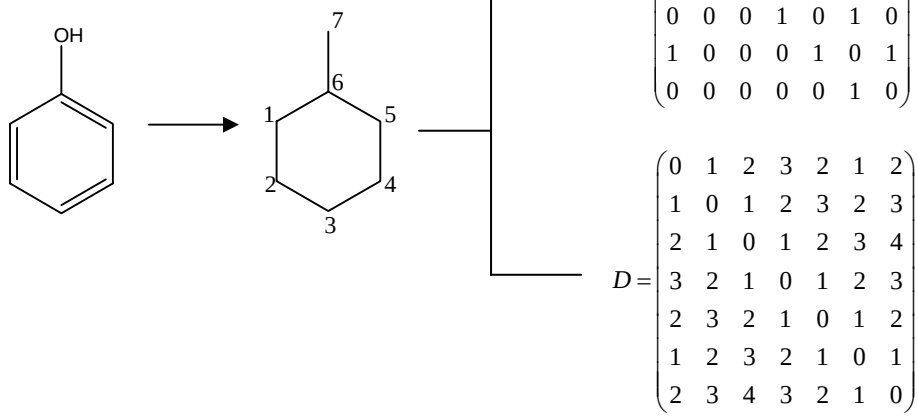


Fig. 1. Graph derived from the structure of phenol, and its associated *A* and *D* matrices. Although numeration of the graph is arbitrary, TIs are graph invariants and their value will remain the same no matter the proposed numeration.

2 Methodology

We intended to identify TIs that are sensitive to structural variations in peptides structures. A chemical descriptor whose value varies little within the dataset of molecules to be studied has little power to distinguish these compounds. This fact was observed by Bajorath et al. as the “bandwidth potential” of the descriptor [15]. For example, a numerical descriptor counting the number of rotatable bonds in a molecule has significantly higher intrinsic variability than a descriptor which detects the presence or absence of a particular structural motif and, therefore, assumes only binary values. In other words, a descriptor that, for a given set of molecules, assumes values that are well-distributed over a wide range of values has great information content and is proper for the characterization of that set of compounds through structure-activity and structure-property relationships. Moreover, and ideal descriptor would be able to assume different values for all the structures in the dataset, meaning that it is sensitive to the structural differences among them all.

Keeping this in mind, we have generated the structures of the 120 possible pentapeptides which can be obtained from all possible permutations of the amino acids Arg, Phe, Ser, Trp and Tyr and 30 of the possible octapeptides that can be obtained from permutations of the amino acids Asp, Cys, His, Ile, Leu, Thr, Tyr and Val. Note that Tyr differs from Phe only by the presence if an hydroxyl group, while Ile and Leu differ in the position of a $–CH_3$ group and Ile and Val only difference is an additional $–CH_2$ group in Ile lateral chain. Thus, if topological descriptors are able to distinguish between peptides whose only difference is the exchange of any of that pairs of similar amino acids in the amino acid sequence, that would be and indicator of high descriptive capability.

Dragon software was used for the computation of 578 TIs [16]. Descriptors with no variance and little variance were excluded from further analysis through the commands “exclude constant variables” and “exclude near constant variables” included in Dragon. For each set of peptides, the results were analyzed considering: range of values of each descriptor (normalized to the average value of the descriptor for the correspondent set of peptides); distribution of the descriptor values normalized to the average value of each descriptor through observation of comparative histograms; and degeneracy of each descriptor in each peptide set. We will consider the descriptor is degenerated if it assumes the same value for several of the peptidic structures. For further quantification of the intrinsic variability of the descriptors, we have performed Shannon Entropy (SE) analysis of the descriptors distribution histograms, as described by Bajorath et al [15, 17]. SE is defined as:

$$SE = -\sum pi \times \log_2 pi \tag{3}$$

In this formulation, p is the probability of observing a particular descriptor value in a given bin of the histogram. p is calculated from the number of compounds with a descriptor value that falls within a specific histogram bin, or count (c), for a specific data interval i. Thus, p is calculated as:

$$pi = ci / \sum ci \tag{4}$$

Equation 3 contains a logarithm to the base 2, which corresponds to a scale factor and permits the resulting SE to be considered as the number of binary bits necessary to capture the information contained within the descriptor variation. SE values for different data sets can be directly compared, provided a uniform binning scheme can be defined. This is the case when data sets are represented in histograms where the data range is divided into the same number of bins. In histogram representations, largest possible SE (maximum information content) is obtained when data points are evenly distributed over all data intervals, and maximum SE thus corresponds to $\log_2$ of the number of histogram bins.

3 Results

Table 1 shows the values of four descriptors for each of the 120 pentapeptides. We have kept Dragon´s nomenclature of the descriptors. MATS5v corresponds to Moran autocorrelation (lag 5) weighted by atomic Van der Waals volumes [18]; MATS6p symbolizes Moran autocorrelation (lag 6) weighted by atomic polarizabilities [18]; VRA1 corresponds to Randic-type eigenvector-based index from adjacency matrix [19]; ATS8m corresponds to Broto-Moreau autocorrelation of a topological structure (lag 8) weighted by atomic mass [20]. As illustrative examples, we have included in Table 1:

- Descriptors whose values fall within a wide range of values but that also present high degeneracy (same value for different peptidic structures) and, as a consequence, poor intrinsic variability (MATS5v and MATS6p). In the case of MATS5v and MATS6p, the descriptor assumes unique values for none of the 120 structures.
- Descriptors with low degeneracy, a wide range of values and high intrinsic variability (VAR1). VRA1 assumes unique values for 118 out of 120 pentapeptides (which represents more than 98% of the total pentapeptides analyzed). In other words, VAR1 is an example of an almost ideal descriptor for the characterization of the 120 pentapeptides. This descriptor is calculated through application of a Randic-type formula:

$$VRA1 = \sum (a_i a_j)^{-1/2} \tag{5}$$

  where a_i and a_j are local graph invariants obtained from the eigenvector corresponding to the largest negative eigenvalue of the adjacency matrix A [19]. This eigenvector varies in a consistent manner and induce reasonable intramolecular ordering of vertices: if the vertices corresponding to the longest chain of the molecule are numbered sequentially starting from one end and the side chains atoms are then numbered further beginning from those closest to atom 1, it is observed that lower values of the eigenvector correspond to vertices of lower degree, farther from the center of the molecule or from vertices of high degree such as tertiary or quaternary carbons.
  Note that VAR1 assumes different values for those pentapeptides in which Phe and Tyr exchange places in the sequence (for example, arg-phe-ser-trp-tyr and arg-tyr-ser-trp-phe), even when these amino acids only differ in the presence of a hydroxyl group in Tyr.
- Descriptors that, although presenting a narrow range of values, have a high intrinsic variability and moderate degeneracy for the considered compounds (ATS8m).

The previous observations are further supported by the comparative histograms (fig. 2) and the values of SE (table 1) for these four descriptors. SE value approaches the maximum ($\log_2 15 = 3.91$) in the case of ATS8m, indicating the high information

content of this descriptor. VRA1, which as explained has high variability among the 120 pentapeptides, also shows high SE value.

Table 1. Values of four TIs for the 120 pentapeptides. Maximum, minimum and average values of each descriptor are presented. SE is calculated from 15-bins histograms.

Peptide	MATS5v	MATS6p	VRA1	ATS8m
arg-phe-ser-trp-tyr	0.003	-0.003	9012.415	143.760
arg-phe-ser-tyr-trp	0.003	-0.004	17943.690	140.476
arg-phe-trp-ser-tyr	0.003	-0.003	4622.455	143.110
arg-phe-trp-tyr-hin	0.017	0.024	4072.852	155.227
arg-phe-tyr-ser-trp	0.003	-0.004	17390.830	139.827
arg-phe-tyr-trp-ser	0.017	0.024	7522.839	155.227
arg-ser-phe-trp-tyr	0.003	-0.003	8813.192	144.463
arg-ser-phe-tyr-trp	0.003	-0.004	16969.750	141.180
arg-ser-trp-phe-tyr	0.003	-0.003	4961.875	144.463
arg-ser-trp-tyr-phe	0.003	-0.003	4927.059	144.256
arg-ser-tyr-phe-trp	0.003	-0.004	16898.930	141.180
arg-ser-tyr-trp-phe	0.003	-0.003	8665.139	144.256
arg-trp-phe-ser-tyr	0.003	-0.003	4457.245	143.110
arg-trp-phe-tyr-ser	0.017	0.024	3175.552	155.227
arg-trp-ser-phe-tyr	0.003	-0.003	4750.589	143.759
arg-trp-ser-tyr-phe	0.003	-0.003	4647.545	143.553
arg-trp-tyr-phe-ser	0.017	0.024	3152.137	155.227
arg-trp-tyr-ser-phe	0.003	-0.003	4335.444	142.903
arg-tyr-phe-ser-trp	0.003	-0.004	17461.950	139.827
arg-tyr-phe-trp-ser	0.017	0.024	7523.069	155.227
arg-tyr-ser-phe-trp	0.003	-0.004	17938.610	140.476
arg-tyr-ser-trp-phe	0.003	-0.003	8858.917	143.553
arg-tyr-trp-phe-ser	0.017	0.024	4025.524	155.227
arg-tyr-trp-ser-phe	0.017	0.024	4025.524	155.227
phe-arg-ser-trp-tyr	-0.022	-0.028	6165.053	142.359
phe-arg-ser-tyr-trp	-0.022	-0.029	12262.54	139.076
phe-arg-trp-ser-tyr	-0.022	-0.028	3461.839	141.986
phe-arg-trp-tyr-ser	-0.008	-0.001	2905.176	154.103
phe-arg-tyr-ser-trp	-0.022	-0.029	12122.78	138.703
phe-arg-tyr-trp-ser	-0.008	-0.001	5177.146	154.103
phe-ser-arg-trp-tyr	-0.022	-0.028	4713.250	142.577
phe-ser-arg-tyr-trp	-0.022	-0.029	9175.335	139.294
phe-ser-trp-arg-tyr	-0.022	-0.028	3440.650	142.716
phe-ser-trp-tyr-arg	-0.015	-0.010	4304.127	147.891
phe-ser-tyr-arg-trp	-0.022	-0.029	8285.299	139.433
phe-ser-tyr-trp-arg	-0.015	-0.010	4391.405	147.891
phe-trp-arg-ser-tyr	-0.022	-0.028	4611.125	140.948
phe-trp-arg-tyr-ser	-0.008	-0.001	3155.295	153.341
phe-trp-ser-arg-tyr	-0.022	-0.028	5989.344	141.460
phe-trp-ser-tyr-arg	-0.015	-0.010	8016.141	146.911
phe-trp-tyr-arg-ser	-0.008	-0.001	3870.265	152.927
phe-trp-tyr-ser-arg	-0.015	-0.010	4635.760	145.985
phe-tyr-arg-ser-trp	-0.022	-0.029	10590.540	137.665
phe-tyr-arg-trp-ser	-0.008	-0.001	4413.631	153.341

phe-tyr-ser-arg-trp	-0.022	-0.029	9362.504	138.177
phe-tyr-ser-trp-arg	-0.015	-0.010	4732.556	146.911
phe-tyr-trp-arg-ser	-0.008	-0.001	2808.311	152.927
phe-tyr-trp-ser-arg	-0.015	-0.010	4635.760	145.985
ser-arg-phe-trp-tyr	0.010	-0.013	3765.677	152.917
ser-arg-phe-tyr-trp	0.01	-0.015	7011.294	149.633
ser-arg-trp-phe-tyr	0.010	-0.013	2703.796	152.917
ser-arg-trp-tyr-phe	0.010	-0.013	2666.760	152.710
ser-arg-tyr-phe-trp	0.010	-0.015	7000.156	149.633
ser-arg-tyr-trp-phe	0.010	-0.013	3709.054	152.710
ser-phe-arg-trp-tyr	0.010	-0.013	2977.855	152.431
ser-phe-arg-tyr-trp	0.010	-0.015	5590.461	149.148
ser-phe-trp-arg-tyr	0.010	-0.013	2773.499	152.293
ser-phe-trp-tyr-arg	0.016	0.005	3642.196	157.468
ser-phe-tyr-arg-trp	0.010	-0.015	4846.232	149.010
ser-phe-tyr-trp-arg	0.016	0.005	2889.362	157.468
ser-trp-arg-phe-tyr	0.010	-0.013	4281.429	152.431
ser-trp-arg-tyr-phe	0.010	-0.013	4179.860	152.224
ser-trp-phe-arg-tyr	0.010	-0.013	5063.771	152.293
ser-trp-phe-tyr-arg	0.016	0.005	6762.983	157.468
ser-trp-tyr-arg-phe	0.010	-0.013	4920.287	152.086
ser-trp-tyr-phe-arg	0.016	0.005	6764.449	157.468
ser-tyr-arg-phe-trp	0.010	-0.015	5695.135	149.148
ser-tyr-arg-trp-phe	0.010	-0.013	2976.336	152.224
ser-tyr-phe-arg-trp	0.010	-0.015	4920.224	149.010
ser-tyr-phe-trp-arg	0.016	0.005	2915.325	157.468
ser-tyr-trp-arg-phe	0.010	-0.013	2746.828	152.087
ser-tyr-trp-phe-arg	0.016	0.005	3686.691	157.468
trp-arg-phe-ser-tyr	-0.022	-0.034	8050.795	136.225
trp-arg-phe-tyr-ser	-0.008	-0.008	5216.586	148.341
trp-arg-ser-phe-tyr	-0.022	-0.034	8965.385	136.598
trp-arg-ser-tyr-phe	-0.022	-0.034	8738.074	136.391
trp-arg-tyr-phe-ser	-0.008	-0.008	5150.054	148.341
trp-arg-tyr-ser-phe	-0.022	-0.034	7770.940	136.018
trp-phe-arg-ser-tyr	-0.022	-0.034	8943.838	135.186
trp-phe-arg-tyr-ser	-0.008	-0.008	5991.577	147.580
trp-phe-ser-arg-tyr	-0.022	-0.034	11865.710	135.698
trp-phe-ser-tyr-arg	-0.015	-0.016	15670.140	141.150
trp-phe-tyr-arg-ser	-0.008	-0.008	7177.505	147.166
trp-phe-tyr-ser-arg	-0.015	-0.016	14748.700	140.224
trp-ser-arg-phe-tyr	-0.022	-0.034	10199.680	136.816
trp-ser-arg-tyr-phe	-0.022	-0.034	9962.492	136.609
trp-ser-phe-arg-tyr	-0.022	-0.034	11761.830	136.955
trp-ser-phe-tyr-arg	-0.015	-0.016	15250.400	142.130
trp-ser-tyr-arg-phe	-0.022	-0.034	11381.160	136.748
trp-ser-tyr-phe-arg	-0.015	-0.016	15194.360	142.130
trp-tyr-arg-phe-ser	-0.008	-0.008	5898.495	147.580
trp-tyr-arg-ser-phe	-0.022	-0.034	8637.037	134.980
trp-tyr-phe-arg-ser	-0.008	-0.008	7194.398	147.165
trp-tyr-phe-ser-arg	-0.015	-0.016	14813.980	140.224
trp-tyr-ser-arg-phe	-0.022	-0.034	11518.990	135.492
trp-tyr-ser-phe-arg	-0.015	-0.016	15686.230	141.150

tyr-arg-phe-ser-trp	-0.022	-0.029	12603.750	137.560
tyr-arg-phe-trp-ser	-0.008	-0.001	5357.407	152.960
tyr-arg-ser-phe-trp	-0.022	-0.029	12717.750	137.933
tyr-arg-ser-trp-phe	-0.022	-0.028	6274.696	141.009
tyr-arg-trp-phe-ser	-0.008	-0.001	2947.706	152.960
tyr-arg-trp-ser-phe	-0.022	-0.028	3471.454	140.636
tyr-phe-arg-ser-trp	-0.022	-0.029	10916.220	136.522
tyr-phe-arg-trp-ser	-0.008	-0.001	4547.415	152.198
tyr-phe-ser-arg-trp	-0.022	-0.029	9681.549	137.033
tyr-phe-ser-trp-arg	-0.015	-0.010	4869.483	145.768
tyr-phe-trp-arg-ser	-0.008	-0.001	2857.924	151.784
tyr-phe-trp-ser-arg	-0.015	-0.010	4683.258	144.842
tyr-ser-arg-phe-trp	-0.022	-0.029	9585.994	138.151
tyr-ser-arg-trp-phe	-0.022	-0.028	4831.456	141.228
tyr-ser-phe-arg-trp	-0.022	-0.029	8666.974	138.290
tyr-ser-phe-trp-arg	-0.015	-0.010	4549.593	146.748
tyr-ser-trp-arg-phe	-0.022	-0.028	3458.919	141.366
tyr-ser-trp-phe-arg	-0.015	-0.010	4391.895	146.748
tyr-trp-arg-phe-ser	-0.008	-0.001	3167.651	152.198
tyr-trp-arg-ser-phe	-0.022	-0.028	4535.119	139.598
tyr-trp-phe-arg-ser	-0.008	-0.001	3934.250	151.784
tyr-trp-phe-ser-arg	-0.015	-0.010	7981.302	144.842
tyr-trp-ser-arg-phe	-0.022	-0.028	5923.114	140.110
tyr-trp-ser-phe-arg	-0.015	-0.010	8159.672	145.768
Max	**0.017**	**0.024**	**17943.690**	**157.468**
Min	**-0.022**	**-0.034**	**2666.760**	**134.980**
Average	**-0.006**	**-0.012**	**7081.429**	**145.767**
Range (normalizaed to average)	**6.16**	**4.73**	**2.16**	**0.15**
SE	**2.50**	**3.00**	**3.36**	**3.69**

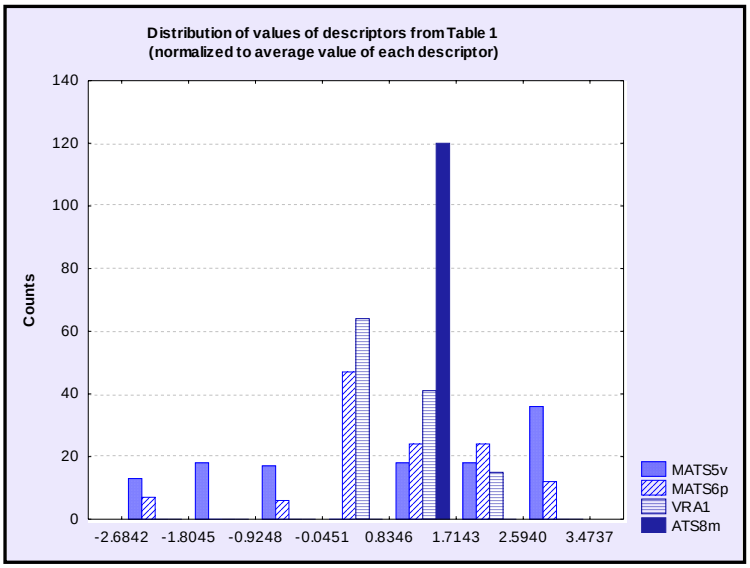


Fig. 2. Distribution of values for the four descriptors presented in table 1 (normalized to the average value of each descriptor).

Table 2 presents the results for the set of 30 octapeptides. Although the range of values that the four descriptors assume are smaller, in most of the cases, than that of the pentapeptides, both VRA1 and AST8m assume unique values for each of the 30 molecules (even in those cases in which the positions of Leu and Ile or Val and Ile are exchanged in the sequence of amino acids, see for example fig. 3). Although Moran autocorrelations still show some level of degeneracy this seems to be quite smaller than in the case of the pentapeptides. (30% and 43% of molecules with unique descriptor values for MATS5v and MATS6p, in that order, and better distribution of values)

Table 2. Descriptor values for the 30-octapeptides set. Maximum, minimum and average values and range of values of each descriptor (normalized to the average value of the descriptor) are presented.

Peptide	MATS5v	MATS6p	VRA1	ATS8m
asp-cys-ile-val-thr-his-tyr-leu	0.035	0.012	1104.386	202.681
asp-cys-thr-val-ile-his-tyr-leu	0.035	0.012	1195.555	200.178
asp-his-tyr-ile-leu-val-thr-cys	0.038	0.004	941.589	204.669
asp-ile-val-tyr-thr-his-leu-cys	0.038	0.004	1002.089	212.217
asp-thr-tyr-ile-val-his-cys-leu	0.035	0.012	1154.518	207.643
asp-tyr-val-cys-ile-thr-leu-his	0.035	0.006	969.893	198.305
cys-asp-thr-tyr-his-leu-val-ile	0.036	0.007	895.615	208.761
cys-ile-asp-val-tyr-leu-thr-his	0.037	-0.001	876.126	201.851
cys-ile-leu-thr-his-val-asp-tyr	0.035	-0.006	990.900	203.109
cys-ile-val-his-asp-leu-tyr-thr	0.042	0.013	854.790	207.817
his-cys-thr-ile-val-leu-tyr-asp	0.026	0.005	1141.841	195.016
his-leu-asp-thr-cys-tyr-ile-val	0.026	0.006	1125.267	204.060

his-tyr-leu-asp-val-cys-thr-ile	0.023	0.003	814.028	194.920
his-tyr-thr-val-ile-leu-cys-asp	0.026	0.005	1197.631	200.168
his-val-leu-tyr-thr-asp-cys-ile	0.023	0.003	976.508	204.197
ile-asp-val-his-thr-tyr-cys-leu	0.044	0.021	1061.916	209.094
ile-cys-thr-asp-val-tyr-his-leu	0.044	0.021	1009.207	201.405
ile-his-thr-cys-val-leu-asp-tyr	0.043	0.010	924.560	196.839
leu-cys-thr-asp-val-tyr-his-ile	0.022	0.029	1165.301	206.461
leu-his-thr-cys-val-ile-asp-tyr	0.021	0.017	986.798	197.223
leu-tyr-val-asp-his-cys-ile-thr	0.028	0.036	836.387	210.394
thr-asp-tyr-his-val-leu-cys-ile	0.051	0.014	974.107	207.558
thr-his-cys-ile-asp-val-leu-tyr	0.049	0.001	917.240	206.356
thr-tyr-val-leu-ile-asp-cys-his	0.051	0.006	1007.319	207.596
tyr-cys-ile-thr-val-leu-his-asp	0.021	0.000	1174.082	192.731
tyr-his-ile-val-thr-asp-leu-cys	0.020	-0.011	1172.958	203.306
tyr-his-leu-val-thr-asp-ile-cys	0.020	-0.011	994.805	201.850
tyr-thr-leu-val-asp-ile-his-cys	0.020	-0.011	1055.348	208.083
val-his-thr-cys-ile-leu-asp-tyr	0.055	0.002	918.506	200.999
val-ile-cys-thr-his-leu-tyr-asp	0.060	0.016	870.524	200.161
Max	**0.060**	0.036	**1197.631**	**212.217**
Min	**0.020**	-0.011	**814.028**	**192.731**
Average	**0.034**	0.0075	**1010.326**	**203.188**
Range (normalizaed to average)	**1.16**	**6.27**	**0.38**	**0.10**

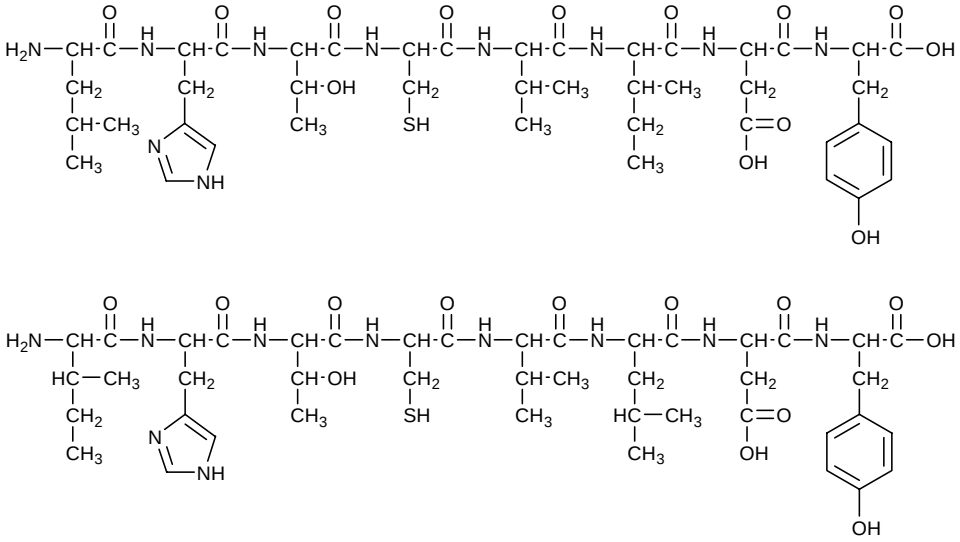


Fig. 3. Comparison of two octapeptides which differ only in the positions of Leu and Ile. The values of the four descriptors showed in table 3 for the peptide on the top (leu-his-thr-cys-val-ile-asp-tyr) are: MATS5v = 0.021; MATS6p = 0.017; VRA1 = 986.198; ATS8m = 197.223. The values for the peptide on the bottom (ile-his-thr-cys-val-leu-asp-tyr) are: MATS5v = 0.043; MATS6p = 0.010; VRA1 = 924.560; ATS8m = 196.839. Note how a slight change in the structure is reflected in significant changes in the values of the four considered descriptors.

4 Conclusions

In order to verify if TIs may be applicable in characterization of peptides, we have calculated 578 TIs for a set of 120 pentapetides and a set of 30 octapeptides; both sets were generated, respectively, by permutations of five and eight different amino acids. The degeneracy and the information content of the descriptors were used as criteria to determine if TIs may serve for the proposed application.

We have identified TIs with low degeneracy and good distribution of values in both sets. In the case of the pentapeptides set, high information content was verified through calculation of Shannon Entropy values. The results indicate that TIs may be useful in Quantitive Structure Activity and Quantitive Structure-Property Relationships (QSAR and QSPR) studies. Thus, TIs could be applied in rational search and design of peptides with therapeutic uses.

Acknowledgments. A. Talevi would like to thank CONICET for his Type 1 post-grade fellowship. L.E. Bruno-Blanch thanks the Facultad de Ciencias Exactas de la Universidad Nacional de La Plata (Incentivos UNLP) and the Agencia Nacional de Promoción Científica y Tecnológica (PICT BIB 1728/06-11985).

References

1. Balaban, A.T., Ivanciuc, O.: Historical Development of Topological Indices. In: Devillers, J., Balaban, A.T. (eds): Topological Indices and Related Descriptors in QSAR and QSPR. Gordon and Breach Science Publishers, Amsterdam (1999) 21-57
2. Ivanciuc, O., Ivanciuc, T., Cabrol-Bass, D.: QSAR for Dihydrofolate Reductase Inhibitors with Molecular Graph Structural Descriptors. J. Mol. Struct. (Theochem) 582 (2002) 39-51
3. Gallegos Saliner, A., Gironés, X.: Topological Quantum Similarity Measures: Applications in QSAR. J. Mol. Struct. (Theochem) 727 (2005) 97-106
4. Calabuig, C., Antón-Fos, G.M., Gálvez, J., García Domenech, R.: New Hypoglyacemic Agents Selected by Molecular Topology. Int. J. Pharm. 278 (2004) 111-118
5. Estrada, E., Uriarte, E., Montero, A., Teijeira, M., Santana, L., De Clercq, E.: A Novel Approach for the Virtual Screening and Rational Design of Anticancer Compounds. J. Med. Chem. 43 (2000), 1975-1985
6. Murcia-Soler, M., Pérez-Giménez, F., García-March, F.J., Salabert.Salvador, M.T., Díaz-Villanueva, W., Medina-Casamayor, P.: Discrimination and Selection of New Potencial Antibacterial Compounds Using Simple Topological Descriptors. J. Mol. Graph. Model. 21 (2003) 375-390
7. Bruno-Blanch, L.E., Gálvez, J., García-Domenech, R.: Topological Virtual Screening: a Way to Find New Anticonvulsant Drugs from Chemical Diversity. Bioorg. Med. Chem. Lett. 13 (2003) 2749-2754
8. Talevi, A., Bellera, C.L., Castro, E.A., Bruno-Blanch, L.E.: Application of Molecular Topology in Descriptor-based Virtual Screening for the Discovery of New Anticonvulsant Agents. Drug Future 31 (Suppl. A) (2006) 188
9. Talevi, A., Sella-Cravero, M., Castro, E.A., Bruno-Blanch, L.E.: Discovery of Anticonvulsant Activity of Abietic Acid through Application of Linear Discriminant Analysis. Bioorg. Med. Chem. Lett. (2007) doi: 10.1016/j.bmcl.2006.12.098

10. Prieto, J.J., Talevi, A., L.E. Bruno-Blanch.: Application of Linear Discriminant Analysis in the Virtual Screening of Antichagasic Drugs through Trypanothione Reductase Inhibition. Mol. Div. 10 (2006) 361-375
11. Gardiner, L., Coyle, B.J., Chan, W.C., Soultanas, P.: Discovery of Antagonist Peptides against Bacterial Helicase-Primase Interaction in B. stearothermophilus by Reverse Yeast Three-Hybrid. Chem. Biol. 12 (2005) 595-604
12. Real, E., Rain, J.C., Battaglia, V., Jallet, C., Pierrin, P., Tordo, N., Chrisment, P., D´Alayer, J., Legrain, P., Jacob, Y.: Antiviral Drug Discovery Strategy Using Combinatorial Libraries of Structurally Constrained Peptides. J. Virol. 78 (2004), 7410-7417
13. Ofran, Y., Punta, M., Schneider, R., Rost, B.: Beyond Annotation Transfer by Homology: Novel Protein-Function Prediction Methods to Assist Drug Discovery. Drug Discov. Today 10 (2005) 1475-1482
14. Floudas, C.A., Fung, H.K., McAllister, S.R., Mönnigmann, M., Rajgaria, R.: Advances in Protein Structure Prediction and De Novo Protein Design: a Review. Chem. Eng. Sci. 61 (2006) 966-988
15. Stahura, F.L., Godden, J.W., Xue, L., Bajorath, J.: Distinguishing between Natural Products and Synthetic Molecules by Descriptor Shannon Entropy Analysis and Binary QSAR Calculations. J. Chem. Inf. Comput. Sci. 40 (2000) 1245-1252
16. Talete srl DRAGON for Windows (Software for Molecular Descriptors Calculation) Version 4.0 (2003) http:www.talete.mi.it
17. Stahura, F.L., Godden, J.W., Bajorath, J.: Differential Shannon Entropy Analysis Identifies Molecular Property Descriptors that Predict Aqueous Solubility of Synthetic Compounds with High Accuracy in Binary QSAR Calculations. J. Chem. Inf. Comput. Sci. 42 (2002) 550-558
18. Moran, P.A.P.: Notes on Continuous Stochastic Phenomena. Biometrika 37 (1950) 17-23
19. Balaban, A.T., Ciubotariu, D., Medeleanu, M.: Topological Indices and Real Number Vertex Invariants Based on Graph Eigenvalues or Eigenvectors. J. Chem. Inf. Comput. Sci. 31 (1991) 517-523
20. Moreau, J., Broto, P., Fortin, M., Turpin, C. Réalisation sur Ordinateur d'un Screening de Structures Moléculaires de Substances Potentiellement Anxiolytiques à l'Aide de la Technique d'Autocorrélation. Eur. J. Med. Chem. 23 (1988) 275-281

Phthorimaea operculella granulovirus: sequence analysis of 5 genes from 16 geographical isolates

Marc Sporleder[1], Octavio Zegarra[1], Vilma Hualla[1], Reinhard Simon[1], and Jürgen Kroschel[1]

[1] International Potato Center (CIP) ,
Av. La Molina 1558, Lima-12, Peru
{corresponding author: Marc Sporleder, E-mail: msporleder@cgiar.org

Abstract. The granulovirus infecting the potato tuber moth, *Phthorimaea operculella* [Lepidoptera: Gelechiidae] (*Po*GV) is a naturally occurring, endemic entomopathogen that has been used efficiently as a selective biological insecticide. The virus has been isolated in various parts of the world. From 16 geographical isolates, five genes of *Po*GV that encode regulatory enzymes which manipulate the host's life cycle (*egt*), act on the apoptosis process (*p*49, *iap-op*1 and *iap-cp*5) or that are involved in establishing infection (*ie*-1), were PCR-amplified by using *Po*GV-specific primers. The DNA segments were sequenced and the gene protein alignments compared for genomic polymorphisms among the isolates. Comparison of the nucleotide sequences showed high similarity between all isolates, but analysis of protein alignments revealed low similarity of the isolate from Yemen (9% sequence identity) for the *iap-cp*5 gene, and of the isolates from Turkey (16%), Yemen (26%), and Kenya (37%) for the *egt* gene.

Keywords: potato tuber moth, microbial insecticides, baculovirus, genomic polymorphism, entomopathogens, geographic variants, DNA homology.

1 Introduction

The potato tuber moth, *Phthorimaea operculella* Zeller (Lepidoptera: Gelechiidae), is a major pest of potatoes in many tropical and subtropical regions of the world (4, 14). Larvae mine both leaves and tuber, in the field and in storerooms. In rustic farmers storage, where previously infested tubers brought into the store are the main source for subsequent pest propagation and rapidly increasing tuber infestations, tuber damage can be total within 2-4 months when left untreated (21). The granulovirus infecting *P. operculella* larvae (*Po*GV) is a prime microbial agent to control the pest in potato stores (7). Cottage-type mass production enterprises have been launched in Peru, Bolivia, Colombia, Egypt and Tunisia (12, 22). Granuloviruses (GV) belong to the highly host specific family of Baculoviridae, which are large viruses with double-strained, supercoiled, circular DNA of 88-156 kbp and are pathogenic for invertebrates (2). The potential of baculoviruses for pest control has been well

documented and they have proven to be effective and safe against many pests, especially of the economically important order of Lepidoptera (8, 9, 13).

*Po*GV has been isolated (1, 3, 10, 15, 23) and field tested (6, 11, 16) in various parts of the world, including South Africa, India, Australia, Tunisia, Peru, Kenya, Bolivia, and the Republic of Yemen. Laboratory studies directed towards evaluating the virulence of *Po*GV revealed high biological variation between *Po*GV isolated (18). Restriction endonuclease analysis (REN) of DNA from eight isolates of *Po*GV revealed three distinct but closely related genotypes (20). Recently, the complete DNA of a Tunisian isolate was sequenced providing a basis for genetic comparison of other *Po*GV isolates (5). 130 ORFs have been identified according to homology with other baculoviruses (5). Baculoviruses encode genes, which manipulate the biology of the host to enable them to have and effective infection. In this study, five viral genes (*egt*, *ie*-1, *p*49, *iap-op*1 and *iap-cp*5) that encode multifunctional regulatory enzymes during infection from 16 geographical *Po*GV isolates were compared biochemically as an initial research step to explain variation in biological activities of the virus isolates.

2 Materials and methods

Sixteen geographical isolates of *Po*GV from the collection at the International Potato Center (CIP), Lima, Peru, maintained at – 70°C, were used in this study. The place of origin of each isolates was Peru (4), Ecuador (2), Bolivia, Colombia, Chile, Australia, Indonesia, India, Turkey, Kenya, Yemen, and Tunisia. Exact dates of isolation are unknown. All of them were isolated from naturally diseased larvae collected from fields or storages and propagated in laboratory stocks of *P. operculella* maintained at CIP. Cloned genotype were obtained by three successive rounds of *in vivo* cloning using the egg-dip method (19). The virus purification process included ultra-centrifugation on a sucrose gradient and the DNA was extracted and purified using standard methods (17).

Five pairs of primers were design for PCR-amplification of the 5 genes, *egt*, *ie*-1, *p*49, *iap-op*1 and *iap-cp*5 (Table 1), by using the *Po*GV DNA sequence (Tunisian isolate) available from GenBank (http://www.ncbi.nlm.nih.gov) and the computer program ADNSTAR (Inc., Madison, Wis.). PCR denaturation temperature was 95ºC for 5 min followed by 30 thermal cycles of 92ºC (1 min, melting) and 56.4 °C (1 min, annealing) and a final cycle of 72°C (5 min, amplification). PCR products were separated by electrophoresis with ethidium bromide in a 1% agarose gel. Amplified fragments were successful purified using the kit provided by Promega (Clean-Up with Wizard® SV for Gel and PCR, Promega Corp., USA) (Figure 1a) and send for sequencing to Macrogen Company (http://www.macrogen.com/english/index.html). Nucleotides and proteins sequences were analyzed using the computer program Vector NTI 8.

Table 1. List of *Po*GV-specific primers disigned.

Genes	Forward primers	Reverse primers
*iap-op*1	AAAATTACGCAAACAATAATAAA	TATAAACGCGACAAAATTTACG
*iap-cp*5	TTTTTACAATAGATCCGAACCACAC	TCGCTATTTTCCACCAAAGCTA
*P*49	TTAGCATGTTGGGTTCGAGTC	AGAATGGCCACTATAGAAGAAAACA
ie-1	TGGTTCAAAACTCGTCTTCATAAC	GGCGCTGGGTAATCGATTAATT
egt	TATTTTGTTGGGTTCGATCA	CAAGAGTGGCAAAATTTATCGTA

3 Results and discussion

Nucleotide sequences of the *ie*-1, *iap-op*1, and *p*49 genes showed homology among all *Po*GV isolates. Nucleotide sequences of *iap-cp*5 revealed homology among most *Po*GV isolates, however, some mutations were detected in the isolates Tunisia and Colombia. Additionally, an insertion of two nucleotides in the gene of the isolate from Yemen changed the alignment of proteins, revealing 9% identity only compared to the Tunisian reference isolate (Fig.2A).

The *egt* gene was fount to be the most variable among the *Po*GV isolates (96 – 100% sequence identity). Nucleotide insertions and mutations were detected in the isolates Chile, Indonesia, Yemen, Ecuador (2), Tunisia, Kenya, Turkey and Yemen isolations. Analysis of protein alignments revealed low identity of the isolates Kenya (16%), Turkey (27%) and Yemen (37%) compared to the reference isolate (Fig 2B, Fig. 3). Other isolates show protein homology. Molecular weights of PCR-products from Turkey (1377 bp), Kenya (1484 bp), and Yemen (1376 bp) different compared to other isolates (1305 bp) due to additional nucleotides included with the regions flanking the gene (Fig. 1E). This allow to use this specific primer to re-identify virus after field releases, differentiating between the three isolates from Turkey, Yemen, and Kenya and the others *Po*GV isolates.

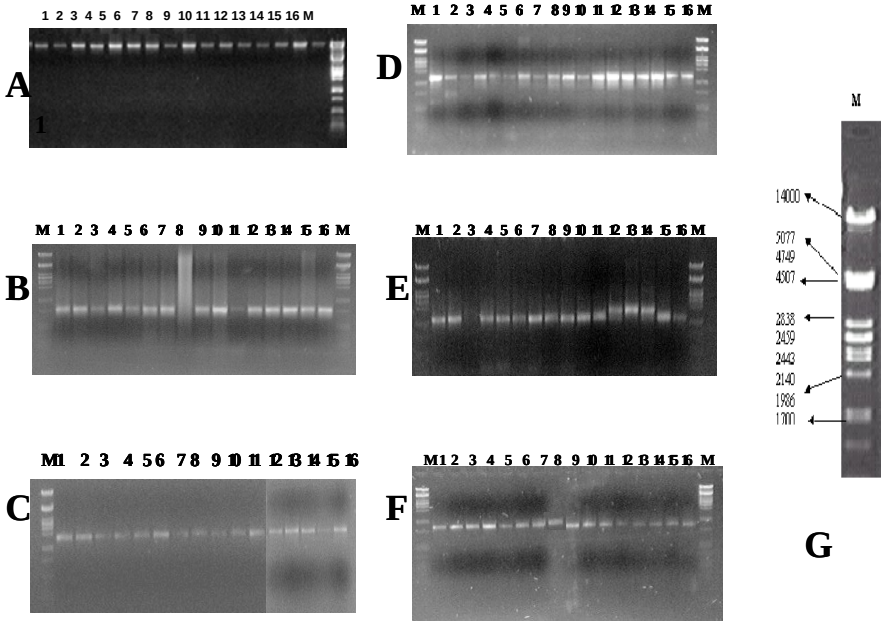


Fig. 1. DNA quality of the 16 *Po*GV isolation used (A), and molecular weights of PCR-products of the genes *iap-cp*5 (B), *iap-op*1gen (C), *p*49 (D), *egt* (E), *ie*-1 (F). DNA λ *Pst*1 Marker 1 is shown (G). Corresponding line numbers are: La Molina (1), Huaraz (2), Huancayo (3), Cusco (4), Ecuador-1 (5), Bolivia (6), Colombia (7), Chile (8), Australia (9), Indonesia (10), India (11), Turkey (12), Kenya (13), Yemen (14), Ecuador-2 (15) and Tunisia (16)

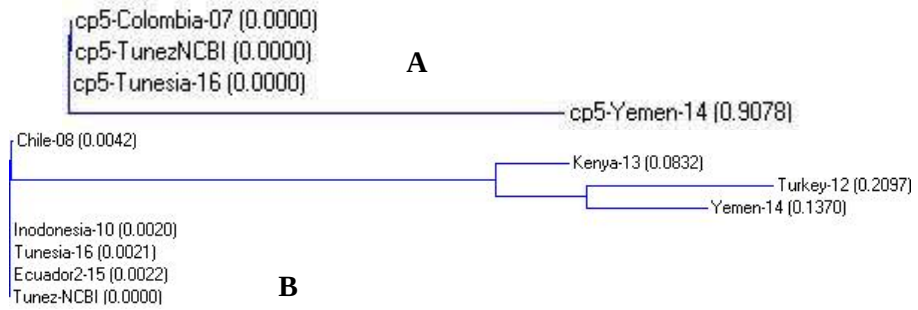


Fig. 2. Similarity of the protein expression of *iap-cp*5 (A) and *egt* (B) from *Po*GV isolates that presented genetic polymorphism.

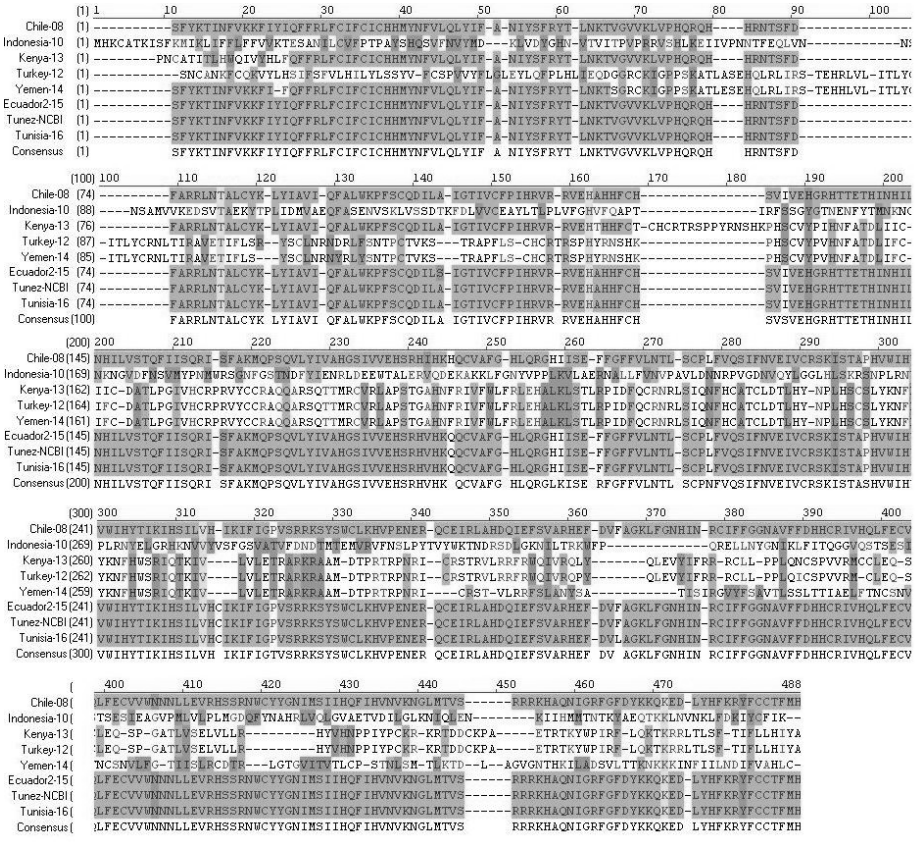


Fig. 3. Protein alignments of the *egt* mutant *Po*GV isolates.

Acknowledgments. This work was funded by the German Federal Ministry for Economic Cooperation and Development (BMZ), Germany and the government of Luxembourg.

References

1. Alcázar, J., Raman, K. V, Salas R. 1991. Un virus como agente de control de la polilla de la papa *Phthorimaea operculella*. Revista Peruana de Entomología 34:101-104.
2. Blissard, G. W., Rohrmann. G. F. 1990. Baculovirus diversity and molecular biology. Annual Review of Entomology 35:127-155.
3. Broodryk, S. W., Pretorius L. M. 1974. Occurrence in South Africa of a granulosis virus attacking tuber moth, *Phthorimaea operculella* (Zeller) (Lepidoptera: Gelechiidae). Journal of the Entomological Society of Southern Africa 37:125-128.

4. Cisneros, F., Gregory, P. 1994. Potato pest management. Aspects of Applied Biology 39:113-124.
5. Croizier, L. A., Taha, C. G., Lopez, F. 2002. Presented at the XX Congreso de la Asociacion Latinoamericana de la Papa, Lima, Peru.
6. Das, G. P. Lagnaoui, A., Salah, H. B., Souibgui, M. 1998. The control of the potato tuber moth in storage in Tunisia. Tropical Science 38:78-80.
7. Gelernter, W. D., Trumble, J. T. 1999. Factors in the success and failure of microbial insecticides in vegetable crops. Integrated Pest Management Reviews 4:301-306.
8. Gröner, A. 1986. Specificity and safety of baculoviruses. In R. R. Granados and B. A. Federici (ed.), The Biology of Baculoviruses, vol. 1. CRC Press, Boca Raton.
9. Huber, J. 1986. Use of baculoviruses in pest management programs. In R. R. Granados and B. A. Federici (ed.), The Biology of Baculoviruses, vol. 2. CRC Press, Boca Raton.
10. Kroschel, J. 1995. Integrated pest management in potato production in Yemen with special references to the interated biological control of the potato tuber moth (Phthorimaea operculella Zeller), vol. 8. Margraf Verlag, Weikersheim, Germany.
11. Kroschel, J., Kaack, H. J., Fritsch, E., Huber, J. 1996. Biological control of the potato tuber moth (*Phthorimaea operculella* Zeller) in the Republic of Yemen using granulosis virus: propagation and effectiveness of the virus in field trials. Biocontrol Science and Technology 6:217-226.
12. Lagnaoui, A. Salah, H. B., El Bedewy, R. 1996. Integrated management to control potato tuber moth in North Africa and the Middle East. CIP Circular 22:10-15.
13. Moscardi, F. 1999. Assessment of the application of baculoviruses for control of Lepidoptera. Annual Review of Entomology 44:257-289.
14. Radcliffe, E. B. 1982. Insect pests of potato. Annual Review of Entomology 27:173-204.
15. Reed, E. M. 1969. A granulosis virus of potato moth. The Australian Journal of Science 31:300-301.
16. Reed, E. M., Springett, B. P. 1971. Large-scale field testing of a granulosis virus for the control of the potato moth (*Phthorimaea operculella* (Zell.) (Lep., Gelechiidae)). Bullentin of Entomological Research 61:223-233.
17. Sambrock, J., Fritsch, E. F., Maniatis, T. 1989. Molecular cloning: a laboratory manual, second edition ed. Cold Spring Harbor Laboratory, New York.
18. Sporleder, M. 2003. The granulosis of the potato tuber moth *Phthorimaea operculella* (Zeller): Characterisation and prospects for effective mass production and pest control. Margraf Verlag, Weikersheim, Germany.
19. Sporleder, M., Kroschel, J., Huber, J., Lagnaoui, A. 2005. An improved method to determine the biological activity (LC_{50}) of the granulovirus *Po*GV in its host *Phthorimaea operculella*. Entomologia Experimentalis et Applicata 116:191-197.
20. Vickers, J. M., Cory, J. S., Entwistle, P. F. 1991. DNA characterization of eight geographic isolates of granulosis virus from the potato tuber moth (*Phthorimaea operculella*) (Lepidoptera, Gelechiidae). Journal of Invertebrate Pathology 57:334-342.
21. von Arx, R., Goueder, J., Cheikh, M., Temime, A. B. 1987. Integrated control of potato tubermoth *Phthorimaea operculella* (Zeller) in Tunisia. Insect Science and its Application 8:989-994.
22. Winters, P., Fano, H. 1997. Presented at the Working Paper Series - International Potato Center, Lima, Peru, 1997.
23. Zeddam, J. L., Pollet, A., Mangoendiharjo, S., Ramadhan, T. H., López Ferber, M. 1999. Occurrence and virulence of a granulosis virus in *Phthorimaea operculella* (Lep., Gelechiidae) populations in Indonesia. Journal of Invertebrate Pathology 74:48-54.

Predicting Physicochemical Properties for Drug Design Using Clustering and Neural Network Learning

Axel J. Soto[1,2], Ignacio Ponzoni[1,2], Gustavo E. Vazquez[1],

[1] Laboratorio de Investigación y Desarrollo en Computación Científica (LIDECC),
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur – Av. Alem 1253 – 8000 – Bahía Blanca - Argentina
{saj, ip, gev}@cs.uns.edu.ar

[2] Planta Piloto de Ingeniería Química (PLAPIQUI)
Universidad Nacional del Sur – CONICET
Complejo CRIBABB – La Carrindanga km.7 – CC 717 – Bahía Blanca - Argentina

Abstract. Prediction of physicochemical properties is of major concern for pharmaceutical research. In this context, machine learning methods are of great importance due to their contribution to the development of a plethora of models. Actually, many predictors exist but most of them do not correctly generalize when external data is presented. We present a novel framework for physicochemical property prediction, where training data is first clustered according to their structural similarity, and a classifier is trained for each conformed cluster. In this regard, the property prediction of a novel candidate drug is modelled by the classifier associated with the cluster that has more structurally-similar compounds with regard to the new putative drug. The generalization problem is not completely solved with the presented approach, but it allows to reduce the prediction error of the method. Artificial neural networks (NN) are used as classifiers and an analysis on logP (octanol-water distribution coefficient) is used to show the advantages of our proposal.

1 Introduction

Interest in Quantitative Structure Activity Relationship (QSAR) given by the scientific and industrial community has grown considerably in last decades. Nowadays, the need for property prediction is an important research issue in pharmaceutical science. Historically, when a new drug had to be developed, a 'serial' process started where drug potency (activity) and selectivity were examined first [1]. Many of the selected compounds failed at later stages due to ADMET (absorption, distribution, metabolism, excretion and toxicity) behaviour in the body. For example, a compound can be promising at first based on its molecular structure, but other factors such as aggregation, limited solubility or limited uptake in the human organism turn it useless as a drug.

Considering these pharmacokinetic screens at an early stage and in parallel with potency, allows drug development costs to get lower. In Di Masi [2], data was collected from a survey of 10 pharmaceutical companies and the estimated out-of-pocket

cost per new drug is of about US$ 400 million and compared with earlier studies, there is an increase in the annual rate of the costs (7.3%) above general price inflation. In ref [3] a description of drug development and research costs are detailed by the therapeutic category.

Currently, the failure rate of a potential drug before the market is still high. The main problem resides in the unknown nature of the rules that govern ADMET conduct. That is, the human body is a very complex system and it is not often evident how a drug interacts with the large number of macromolecules in the human organism. In this context, physicochemical properties are related to the interactions of the drug molecule with the surrounding aqueous environment.

So the trend consisted in the decomposition of this complex task in several stages in order to research ADME properties. For example, the first stage, absorption, was studied in [4] and a relationship was obtained from its logP (octanol-water partition coefficient) and other physicochemical properties. In this context, the emerging of combinatorial chemistry and HTS (high throughput screening) was of great progress because it allowed getting a vast amount of compounds in a short period and hence, constructing more robust models by the establishment of relationships.

In vitro systems allow the isolation of the desired investigation mechanism using human tissue preparations or cell lines. These systems are preferred to the understanding of a process based solely on *in vivo* data, since the observer of a complex *in vivo* system has no way to determine what interactions have influence in the behaviour of a drug. However, in some cases it can be distrusting to apply *in vitro* experiments due to the unrealistic scenarios where sometimes these assays are performed (e.g. different oxygen concentrations in human tissues, transformed human cells or incomplete cells).

For these reasons, in the last decade major research efforts were made to predict *in silico* (by computer means) ADME and physicochemical properties. These *in silico* methods, clearly cheaper than *in vitro* experiments, allow to examine thousands of molecules in shorter time and without the necessity of intensive laboratory work. Although *in silico* methods are not pretended to replace high-quality experiments at least in the short term, some computer methods have demonstrated to obtain as good accuracy as well-established experimental methods [5]. Moreover, one of the most important features of this approach is that a candidate drug (or a whole library) can be tested before being synthesized. Due to the gains in saved labour time, *in silico* predictions considerably help to reduce the large percentage of leads that fail in later stages of their development, and to avoid the amount of time and money invested in compounds that will not be successful.

QSAR (Quantitative structure-activity relationships) is a discipline used in chemoinformatics that comprises the methods by which chemical structure parameters are quantitatively correlated with a well defined process, such as biological activity or any other experiment. QSAR has evolved over a period of 30 years from simple regression models to different computational intelligence models that are now applied to a wide range of problems [6], [7], [8], [9].

1.1 Main objective

One important difficulty when QSAR is applied comes from the great diversity of chemical compounds and therefore it is not so simple to make generalizable-enough predictors that could be applied to any chemical. For this reason, it would be desirable to conform groups of compounds according to their similarity and so more specific predictors can be established. One way for achieving this objective is by means of clustering data [10], [11].

In this paper we analyze important deficiencies of existing methods and propose a QSAR approach that makes use of cluster analysis in the training process of a supervised machine learning method. Our hypothesis aims to diminish errors and also improves model understanding. This paper is organized as follows; section 2 details some aspects concerning property prediction and section 3 is particularly devoted to the issue of hydrophobicity prediction. The method and some theoretical aspects are presented in the fourth section; the results are illustrated in section 5. Finally section 6 outlines concluding remarks.

2 About property prediction

Prediction methods are more valuable when more seriously needed and more difficult to experimentally determine the property is. One of the most useful properties is hydrophobicity/hydrophilicity, since it considerably influences the behaviour of drugs in the body. This property corresponds to one of the first and most extensively modelled one. It is traditionally expressed in terms of the logarithm of the octanol-water partition coefficient (logP), mainly due to the result of the work of Hansch and Leo [6]. The value of logP can be used as rough early ADME screens to reject candidate drug developments as early as possible [7].

According to this growing interest, many computer models have been developed for predicting logP. These methods attempt to find a relationship between the molecular and structural properties of a compound and their logP value. There are two major approaches to encourage this task. Fragement-based methods were pioneered by one of the most widely used program CLOGP [6] which contains a database that has values for fragments as well as correction factors for fragment interactions. The method works breaking molecules into fragments, and logP value is estimated by summing up the corresponding fragment values and the correction interaction values. Other current prediction software that also works with this atom/fragment contribution group approach is reviewed in [12].

On the other hand, another approach for prediction consists in calculating various molecular properties assumed to be important for the value of the desired property. Each calculated value expresses a feature of the entire molecule, and they are used to construct a function for computing logP. Next section clarifies this topic by introducing the molecular descriptor concept.

This latter approach was greatly expanded when it was combined with machine learning techniques, especially neural networks (NN). A detailed review of this kind of methods can be found in [7], [13].

2.1 Data used for predictions in Chemoinformatics

Todeschini and Consonni [14] define a molecular descriptor as "the solution to a logical mathematical procedure that transforms chemical information encoded within a symbolic representation of a molecule into a useful number, or is the result of some standardized experiment to measure a molecular attribute". Descriptors can be calculated from a two- or three-dimensional molecular structure. Two basic categories of descriptors can be addressed: whole-molecular and substructure [15], [16]. Substructure descriptors declare the presence or absence of a particular group or interactions among groups. Clearly, they are preferred when fragment-based methods are utilized. Better understanding of models could be achieved with the use of these descriptors because it can be identified which group or interaction confer the target activity. Nonetheless, some drawbacks can be pointed, e.g. great dimensionality in the molecule representation and substructures that are not present in the training set.

The other type of descriptor corresponds to the property values calculated for capturing characteristics of the chemical compound (see preceding subsection). This kind of descriptors intends to be a more general way to understand molecular interactions. There are many types or families of whole-molecular descriptors. The simplest families are constitutional (1D, one dimension) and topological descriptors (2D), which describe the number of atoms, functional group or type and order of chemical bonds. Additionally, there are also other descriptors that represent or extract information of the 3D structure of the molecules, e.g. geometrical descriptors and a variety of electrostatic and quantum chemical descriptors.

In addition to the variety of descriptors, the availability of great quantity of chemical data is crucial in chemoinformatics. Unfortunately, the accessibility of experimental data needed for model development is restricted, since most of this chemical information resides in the private domain. Moreover, it is clear that good quality data is also as important as the amount of available information. Jónsdóttir et al. [8] presents an excellent review of small molecule databases relevant to overall drug discovery.

3 Issues in logP Prediction

The complexity of prediction methods and the vague understanding of the authentic relationship between structure and property, lead to some problems, many of them not completely solved. As formerly said, prediction methods use molecular descriptors to represent the structure of compounds. Many different descriptors and descriptor families are employed in logP prediction in the bibliography, for instance: constitutional and functional groups, electrostatic indices, etc. [12], [13]. With the exclusion of a few common descriptors, there is no consent agreement of the selection of which descriptors are relevant or influence the hydrophobic behaviour of a compound.

The number of variables used in a prediction method generally affects its performance. If a small amount of variables (less than required) is selected in the construction process (i.e. training, fitting or optimization) of the model, the result will be a poor model. On the other hand, when too many variables are used, overfitting [17] or

chance correlations [18] could occur and again the method will not behave as good as pretended with new examples.

Furthermore, generalization is also a problem when using redundant data, since it affects the bias of the method toward the over-represented subgroup. Jónsdóttir et al. [8] mention that chemoinformatics is 'still in its infancy' in this subject, because redundancy checks are not carried out during validation as it should.

Moreover, it would seem that we are still far from the universal logP predictor, i.e. a method that would reliably predict logP for any possible chemical [9]. Several predictor software has failed when being tested with external different data, since the available chemical space is extremely large and the company experimental data is not publicly released for the construction of models [8].

Novel proposals [15], [19] tackle this extrapolation problem with similarity measures applied to the predicted compounds in order to diminish errors or get a degree of the reliability of the computer calculation. These papers are related with ours in the sense that it also uses distance measures, not for predicting extrapolation errors but for improving interpolation accuracy.

4 Main Features of the Method

The idea of this article is to propose a general methodology for the usage of prediction methods specially oriented to be applied in QSAR. The core of this study is the general technique for physicochemical prediction model rather than our developed logP model itself.

In the previous section we anticipated that our method makes use of similarity measures in its prediction process. The main purpose of this approach is to get into our training examples and put similar compounds together so that detected differences and features in training set could be differently trained. As it is later showed, this proposal could help to improve the accuracy and achieve more simplicity of the constructed models.

The motivation of this paper is inspired in recent works in chemoinformatics [15], [19], [20] and computer science [21], [22]. In refs [15] and [19] better predictions are obtained when similarity is taken into account in the selection of the model. Wolpert and Gama show how the combination of classifiers in a stacked (or serial) way outperforms the performance of individually-applied classifiers.

These different ideas were taken to develop a model where combination of an unsupervised and a supervised machine learning method and where similarity measures were used. In this regard, the method uses an unsupervised approach to encourage a separation in the chemical space. This division is carried out by means of a clustering algorithm, and then within each cluster a classifier (supervised predictor) is applied.

Our developed model corresponds to an NN technique where we introduced a variation in training according to the recently described idea, but this same method could be applied to any machine learning prediction method that involves a learning process. Moreover, the method is also flexible with any kind of descriptors (whole-molecular or substructure) and it is not tight to a specified similarity criteria.

4.1 Similarity and Cluster Analysis

A never ending issue in statistics is how similarity can be calculated between two multivariate elements. A common representation for a set of chemical compounds represented by a set of *n* whole-molecular descriptors is by a point in the R_n space. Then, distance between compounds can be easily measured by applying the Euclidean, or more generally, Minkowski distance. Furthermore, many other distances can be applied, e.g. Gower, Canberra, Orloci's chord, standardized Euclidean or Cosine, which make use of scaled or standardized variables. In the case of substructure descriptors, any binary distance measure can be applied, e.g. Dice, Tanimoto or simple matching.

Cluster analysis aims at assembly groups of elements where intragroup distances are minimized and, on the other hand, intergroup distance becomes maximal. It is worth mentioning that cluster analysis differs from classification methods in the sense that the latter starts with predefined groups and the objective is the assignment of new elements to the pre-existent clusters.

There exist several clustering methods, and the first distinction is between hierarchical and partition methods. Hierarchical clustering is carried out by successive linkage (agglomerative) or separations (divisive). A critical aspect on hierarchical clustering is how the linkage is calculated, that is how the distance between a cluster and a single compound is considered. Here again, many options are available for linkage like: single, complete, average (weighted and not weighted), centred (weighted and not weighted), Hotteling, to take examples. Results of hierarchical clustering, either agglomerative or divisive, may be shown in a dendrogram where order and distance of linkages (divisions) may be obtained. This dendrogram in combination with a multivariate visualization method of the data, is useful to observe the manner of the clustering of data and to determine if that decomposition is satisfactory for the analyzer.

On the other hand, partition methods were conceived to group elements around k-kernel points, where k was *a priori* specified. This initialization of kernel points could be random or specifically defined. It also uses similarity measures, however it has no need to storage a similarity matrix for each pair of vertices, therefore it shows potential for working with a great amount of data. Nevertheless, it is difficult to find appropriate kernel points (randomly or defined) or its number in such great amount of variables of the data space. A complete explanation in this area can be found in ref [23].

4.3 Method

Given a learning set $D = (\vec{x}_i, y_i)$ with $i = 1,...,N$, where $\vec{x}_i = [x_1,...,x_m]$ is a chemical compound described by m descriptors and y_i is its target value. Cluster analysis is applied to the training set D, so that each compound is assigned to only one group, i.e. $\vec{x}_i \in \{Cl_1,...,Cl_k\}$ where k is the resulting number of clusters. Then, each Cl_i is trained by an ensemble of M neural networks (NNE) $NNE_i(Cl_i) = [NN_1(Cl_i),...,NN_M(Cl_i)]$. The k generated NNEs are models that map from the input space $\vec{X}$ to a vector with *M* target values.

At testing, when a new example $\vec{x}$ is presented, a function $\ell : (\vec{x}, Cl_1,...,Cl_k) \rightarrow [1...k]$ is applied in order to measure which is the nearest

cluster in relation to the new compound according to some pre-established similarity measure. Then, $NNE_{\vartheta}(\vec{x})$ is calculated and then averaged by their M components, where ϑ is the index of the nearest cluster of $\vec{x}$, that is $\vartheta = \ell(\vec{x}, Cl_1, ..., Cl_k)$.

Fig. 1 shows an outline of this process. As it is shown, our approach introduces some variations to the way that training and testing are traditionally carried out in chemoinformatics. In the first place, all compounds in the training set are clustered so that they stay together according to the selected similarity. Then, one ensemble of NNs is trained only with the data of each conformed cluster, not the entire training set.

Fig 1. Conceptual scheme of learning method

4.4 Experimentation

In order to describe our technique, we have developed a model for logP prediction by means of a back propagation feed-forward NN. We decided to work with a total set of first 4778 compounds (CAS-ordered) from the PHYSPROP database [24], 80% of them were used for training and the remainder was left for testing. In this case, we assume that k-fold cross validation is not mandatory given the great number of compounds included in the datasets. Functional groups descriptors [14] were used in the clustering process and constitutional descriptors [14] were used for prediction. All utilized descriptors correspond to the whole-molecular class.

For clustering as well as for NN input feeding, a pre-processing was carried out using principal components analysis (PCA) and its correlation matrix. PCA allows to reduce the number of descriptors, by transforming data into a new dimension space and discarding less influential descriptors that can be expressed by linear combination.

Cluster analysis was applied to the training set and seven groups were obtained. The distance used was cosine with a complete linkage. A cut off threshold for clustering was set up to the distance of 1.8. An ensemble of NNs was trained for each resulting cluster, where each ensemble was conformed by 10 NNs. Each cluster was trained until its early-stopping point [17]. At testing, the same distance as in clustering was used to detect the most similar group. All taken choices in the method, e.g. descriptors, distance metric, distance threshold for clustering, NN architecture and learning function, were decided after many trials and comparisons of results.

5. Result Analysis

In this section, results of the aforementioned method are showed in comparison with the analogue method but without clustering. Fig. 2 shows the dendrogram obtained by clustering the training set. In figure 3, a 3D graph of first 3 principal components with its clustering is showed.

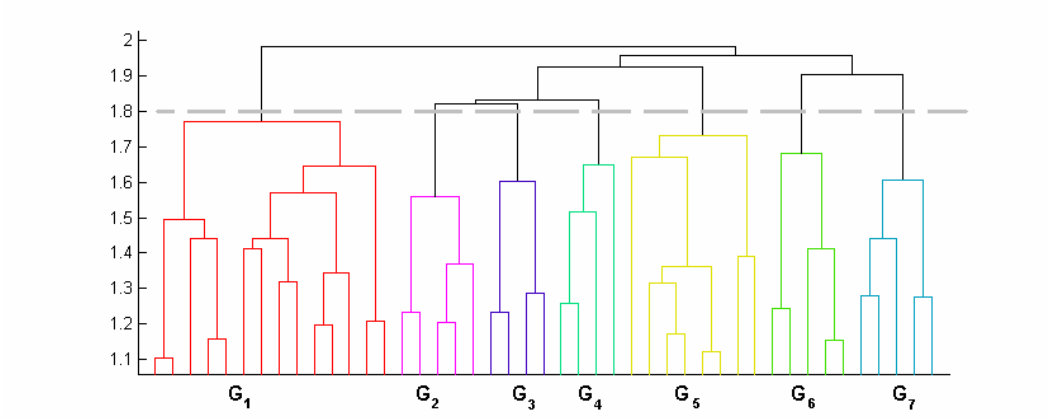


Fig. 2. Dendrogram for training set

Prediction results are showed and reported by its mean absolute error (MAE) and mean square error (MSE). Table 1 confirms the results of the comparison and figure 4 illustrates experimental versus predicted values were distance to the identity line shows the absolute error of prediction. It is important to note that although resulting errors are not too big, when no cluster was applied it was necessary to train and tune a much more complex NN to achieve an almost similar behaviour.

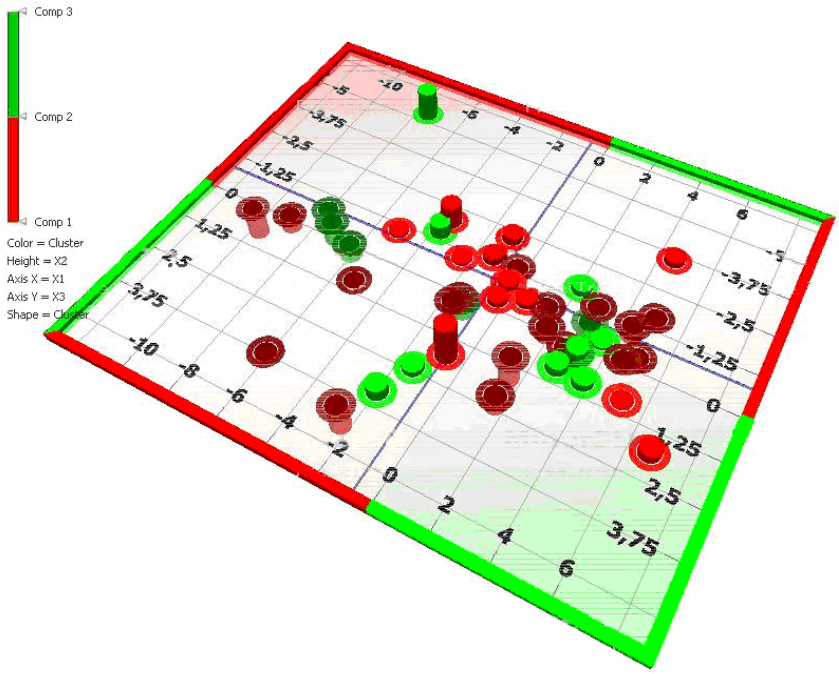
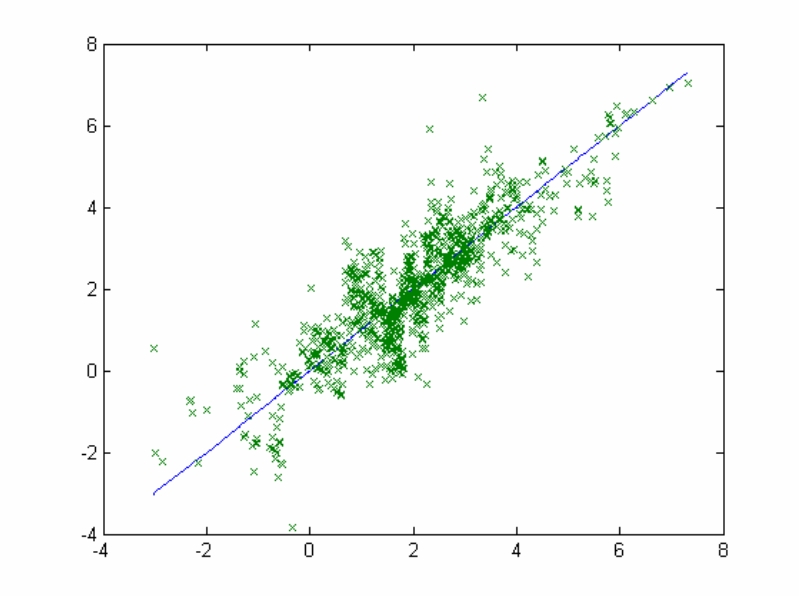


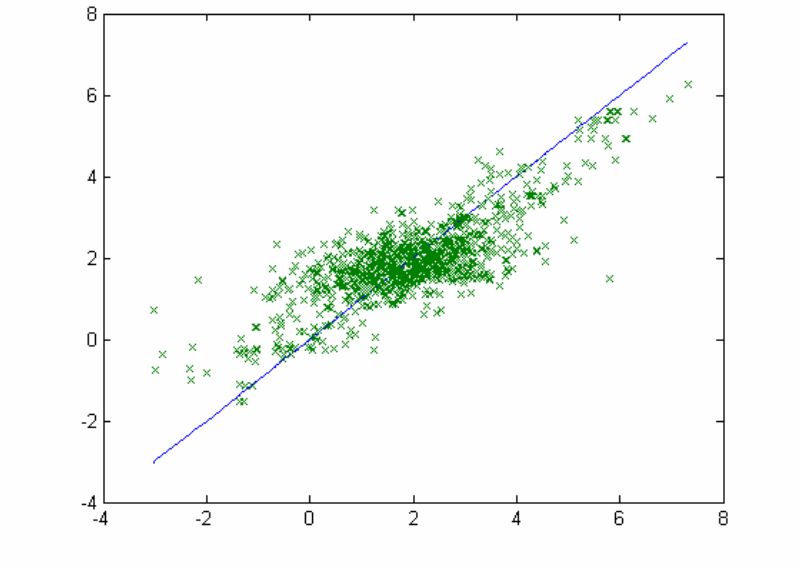
Fig. 3. Visualization of 15 compounds of three different clusters in their first three principal components.

Table 1. Comparison of results with an NN-predictor alone and an NN-predictor with cluster analysis. Results with clustering are averaged or aggregated from individuals models. (a) Mean absolute error (standard deviation in parentheses). (b) Mean square error. (c) Training Iterations. (d) Resulting variables of PCA preprocessing. (e) Number of adjustable weights

	MAE [a]	MSE [b]	Epochs [c]	Variables [d]	Weights [e]
NN	0.65 (±0.83)	0.72	3000	27	1275.00
Cluster + NN	0.61 (±0.66)	0.66	1785	24	371.85



(a) NN alone



(b) NN with cluster

Fig. 4: Predicted logP values versus experimental values

6. Conclusions

The combination of the clustering technique with a supervised learning method results useful due to the differentiation introduced in the training of each cluster. This differentiation allows getting several simpler models instead of a single complex; permitting a better understanding of models which is a highly-desirable feature of QSAR. In addition, training times are also lower and clustering introduces the possibility of parallel training.

Any clustering algorithm can be utilized, but it is important in the clustering stage a clear distinction among groups to be marked out. The quality and representativeness of the results are better when more significant the grouping of the model is. The comparison with other established logP methods was not carried out, because this work aims to propose a general methodology for property prediction rather than a model itself.

Previous cited works about classifier combination and obtained results encourage the exploration of other learning techniques that allow to improve the overall performance of our method. In this way, many more molecules are pretended to be taken into account and also other descriptors would be studied. Another idea to be considered is the descriptor selection by artificial intelligence methods. Finally, it would also be interesting to find a clustering of data not guided by geometrical distances, but by pharmacology classes. Other important step would be the interpretation of rules associated with the prediction.

Acknowledgments

Authors acknowledge the "Agencia Nacional de Promoción Científica y Tecnológica" from Argentina, for Grants Nº11-12778 and Cod. 917. They would also like to acknowledge SeCyT (UNS) for Grant PGI 24/N019.

References

1. Selick, H.E., Beresford, A.P, Tarbit, M.H.: The Emerging Importance of Predictive ADME Simulation in Drug Discovery. Drug Discov. Today 7, 2 (2002) 109-116
2. DiMasi, J.A, Hansen, R.W., Grabowski, H.G.: The Price of Innovation: New Estimates of Drug Development Costs. J. Health Econ. 22 (2003) 151-185
3. DiMasi, J.A., Grabowski, H.G., Vernon, J.:R&D Costs and Returns by Therapeutic Category. Drug Inf. J. 38, 3 (2004) 211-223
4. Lipinski, C.A., Lombardo, F., Dominy, B.W., Feeney P.J.: Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. Adv. Drug. Deliv. Rev. 23 (1997) 3-25
5. Agatonovic-Kustrin, S., Beresford, R.J.: Basic Concepts of Artificial Neural Network (ANN) Modeling and its Application in Pharmaceutical Research. J. Pharmaceut. Biomed. 22, 5 (2000) 717-727
6. Hansch, C., Leo, A.J.: Substituent Constants for Correlation Analysis in Chemistry and Biology. John Wiley, New York (1979)
7. Taskinen, J., Yliruusi, J.: Prediction of Physicochemical Properties Based on Beural Network Modeling. Adv. Drug Deliver. Rev. 55, 9 (2003), 1163-1183

8. Jónsdottir, S.Ó., Jørgensen, F.S., Brunak S.: Prediction Methods and Databases Within Chemoinformatics: Emphasis on Drugs and Drug Candidates. Bioinformatics. 21 (2005) 2145-2160
9. Tetko, I.V., Bruneau, P., Mewes, H.-W., Rohrer, D.C., Poda, G.I.: Can we estimate the accuracy of ADME-Tox predictions? Drug Discov. Today. 11 (2006) 700-707
10. Saitta, S., Raphael, B., Smith, I.F.C.: Combining Two Data Mining Methods for System Identification. Intelligent Computing in Engineering and Architecture. Lecture Notes in Artificial Intelligence. Vol. 4200. Springer-Verlag Berlin, Heidelberg Platz 3 (2006) 606-614
11. Hathaway, R.J., Bezdek, J.C., Huband, J.M.: Maximin Initialization for Cluster Analysis. Progress in Pattern Recognition, Image Analysis and Applications, Lecture Notes in Computer Science. Vol. 4225. Springer-Verlag Berlin, Heidelberg Platz 3 (2006) 14-26
12. Malík, I., Sedlárová, E., Csöllei, J., Andriamainty, F.,Kurfürst, P., Vančo, J.: Synthesis, Spectral Description, and Lipophilicity Parameters Determination of Phenylcarbamic Acid Derivatives with Integrated N-phenylpiperazine Moiety in the Structure. Chemical Papers. Versita, co-published with Springer-Verlag. 60, 1 (2005) 42-47.
13. Winkler, D.A.: Neural Networks in ADME and Toxicity Prediction, invited review, Drug Future, 2004.
14. Todeschini, R., Consonni V.: Handbook of Molecular Descriptors. Wiley-VCH, Weinheim Germany (2000).
15. Sheridan, R.P., Feuston, B.P., Maiorov, V.N., Kearsley, S.K.: Similarity to Molecules in the Training Set Is a Good Discriminator for Prediction Accuracy in QSAR. J. Chem. Inf. Comput. Sci. 44 (2004) 1912-1928
16. Livingston, D.J.: The Characterization of Chemical Structures Using Molecular Properties. A Survey. J. Chem. Inf. Comput. Sci. 40 (2000) 195-209.
17. Tetko, I.V., Livingstone, D.J., Luik, A.I: Neural Networks Studies. 1. Comparison of Overfitting and Overtraining. J. Chem. Inf. Comput. Sci. 35 (1995) 826-833.
18. Topliss, J.G., Edwards R.P: Chance Factors in Studies of Quantitative Structure-Activity Relationships. J. Med. Chem. 22, 10 (1979) 1238-1244.
19. Kühne R., Ebert R.-U., Schüürman G.: Model Selection Based on Structural Similarity-Method Descrption and Application to Water Solubility Prediction. J. Chem Inf. Model. 46 (2006) 636-641.
20. Martin, Y.C., Kofron, J.L.: Do structurally similar molecules have similar biological activity? J. Med. Chem. 45 (2002) 4350-4358.
21. Wolpert, D.: Stacked Generalization. Neural Networks. 5 (1992) 241-260.
22. Gama, J., Brazdil, P.: Cascade generalization. Mach. Learn. 41 (2000) 315-343.
23. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. 3rd edn, Prentice Hall, 1992.
24. The Physical Properties Database (PHYSPROP) is marketed by Syracuse Research Corporation (SRC), North Syracuse, USA at URL http://www.syrres.com/esc/.

Biota-RIO: a database of animal biodiversity in the State of Rio de Janeiro

Vinícius Schmitz Pereira Nunes[1], Alexandre Rossi Paschoal[1], Clarice Augusta Carvalho Cardoso[2], Ana Tereza Ribeiro Vasconcelos[1], Cláudia Augusta de Moraes Russo[2]

[1]Laboratório de Bioinformática, Laboratório Nacional de Computação Científica, LNCC/MCT, Petrópolis, RJ, Brazil. {Vinicius Schmitz, vsnunes@lncc.br}

[2]Departamento de Genética, Instituto de Biologia, Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro, RJ, Brazil. {Claudia Russo, claudia@biologia.ufrj.br}

Abstract. The Rio de Janeiro State presents one of richest neotropical biodiversities. Since much of the green covered area is within city limits, it is also one of the most threatened. Biota-RIO is an open access database that contains information on the animal species originally described for the State of Rio de Janeiro. The database is divided in two parts: classic systematics; and molecular systematics. The Biota-RIO will have, at the first release, information of the species of vertebrates such as: original description, photo, habitat, environment, endangered situation, articles and gene sequences that are available in GenBank. Following versions will include invertebrate species. The database will also present a tutorial, in which molecular biologist and classic systematics, willing to expand their scope, may learn the basics of modern (molecular) systematics.
Keywords: Database, species, biodiversity, Rio de Janeiro, systematics.

1 Introduction

How may we speed up the information flow without limiting the information gathering? Biodiversity issues are bound to be one of the highligths of this dilema at this point in our history. This is because world biodiversity declines at a pace much faster than we are able to uncover it. In this sense, we live at a turning point, in which society is fully aware on biodiversity crisis, but a clear picture on what should be done to minimize its effects has not yet emerged.

Databases are powerful tools, to organize and to manage vast amounts of information that were assembled in a (particular) user friendly way [1]. Thus, whenever large amounts of data are available, a database bounds to offer the user the best and the fastest information flow without limiting the information gathering [2]. The aim of our BIOTA-RIO database is to storage, retrieve and provide free-access to

valuable information on the animal biodiversity that were originally described for the state of Rio de Janeiro. Due to the large green covered area within city limits, most of the biodiversity in Rio is endangered and still largely unknown [3].

Many important species of Brazilian biomes, for instance, were described a long time ago, in local circulation journals. This reflects a need for XX or XIX century papers to complete our bibliography, hardly available even in average size libraries. Naturally, the original data on species description is crucial for the systematist and many times, its absence makes impracticable the systematic of the entire group. Apart from this difficulty, many original descriptions were carried out by foreign researchers, that deposited holotypes in museums all over the world, not leaving a single paratype in Brazilian collections. For most researchers in Brazil, access to these collections abroad is severely restricted, due to the high transportation costs and to the Brazilian legislation, that hinders the interchange of biological material with foreign countries.

In this project the proposal is the development of Biota-Rio, a database that will go to facilitate to the access the important biological data for the systematic of species that had been described for the State of Rio De Janeiro.

2 About the database

Biota Rio is divided in two parts: classic systematics and molecular systematics. The first part of the database provides the following information:

- The general taxonomic classification of the species, the original description of the species, the diagnose, and photos, if available. Also, other important references will also be included, for example, if the original description is too short, an additional and a more complete description will also be available in Biota-RIO.
- The current status of the species, that is, if it extinct, endangered, or threatened. In this case, the list of the IUCN will be used (www.iucn.org).
- The place of deposit of holotype and paratypes, including country, city, museum, and voucher number.
- Common or local name of the species. This will facilitate the communication between researcher and the local community, in the case of re-collections.
- Type locality, including the most detailed description possible, including photo and environment description, whenever available.

The second part of the database involves, molecular sequences that are available for the species. In this case, we will gather information from the GenBank such as: name of the gene, number of base pairs available, access number, marking of the individual (clone, variant, locality) when available, taxonomic rank, original article in which sequences were analysed, authors and address, nucleotide and protein (when applicable) sequence of the mitochondrial genes available.

An important aspect of the Biota-RIO will be the tutorial. It will provide a text in which molecular biologists will learn the basic aspects of systematics and classical systematists will learn molecular biology and sequence analysis basics. This tutorial will act as bridge in which zoologists and molecular biology will learn the

basics of modern molecular systematics, encouraging them in participating and collaborating in high scope projects in Brazil and abroad.

In the sequence analysis tutorial, for instance, the zoologist will be instructed in as the choice of a gene is much more important than the choice of the method of phylogenetic reconstruction [4]. Thus, the user will be able to select a gene with adequate variability for his/her particular phylogenetic problem, providing the first steps in the direction of a consistent phylogeny and a work in systematic of good level.

In order to retrieve the described biodiversity for the State of Rio de Janeiro, we will search in the Zoological Record, published by the Zoological Society of London and the BIOSIS, since 1864. It is considered the most complete database of zoological information, it covers today about 6000 scientific publications in the most diverse areas of zoology.

3 Implementation

For the development of the database BiotaRIO, we use the relational database MySQL (http://www.mysql.com) version 3.23.46 are used. The database is composed of twenty one tables that are responsible for storing all information. For system implementation, we will use the programming language CGI/PERL. The entire access system and information gathering in Biota Rio is focused on its web site (Fig. 1).

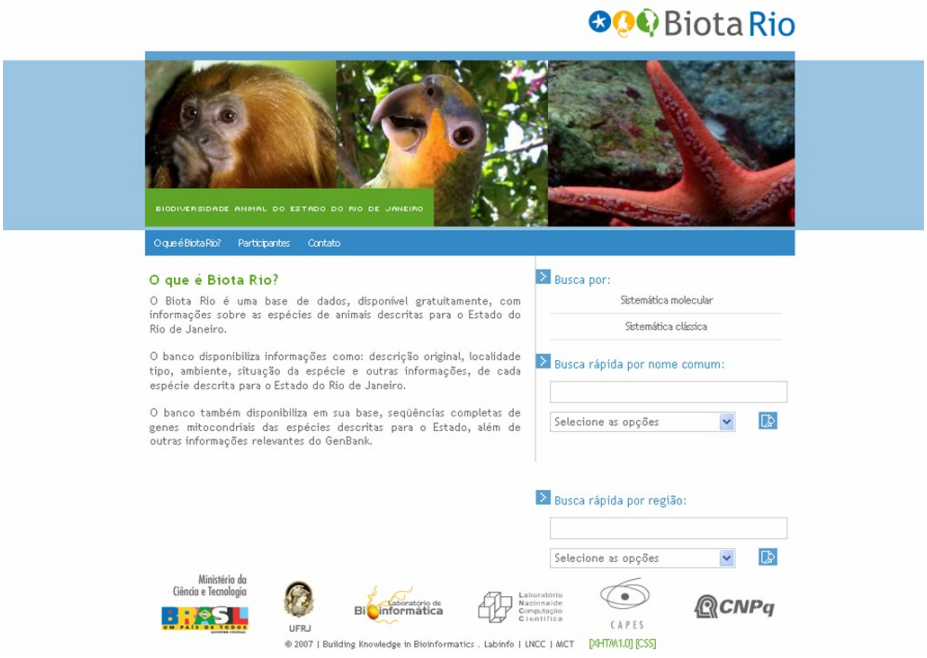


Fig1: Web site of Biota-Rio(www.biota-rio.lncc.br).

Acknowledgments. This work was supported by CNPq grants to CAMR and ATRV and by CAPES fellowships to VSPN and ARP.

References

1. Grumbling, G., Strelets, V., and The Flybase Consortium. Fkybase: anatomical data, images and queries. Nucleic Acids Research 34 (2006) D484-D488.
2. Li, H., Coghlam, A., Ruan, J., et al. Treefam: a curated database of philogenetic trees of animal gene families. Nucleic Acids Research 34 (2006) D572-D580.
3. Rodrigues, A.S.L., Pilgrim, J.D., Lamoreux, J. F. et al. The value of the IUCN Red List for conservation. Trends in Ecology and Evolution. 21(2006) 71-76.
4. Russo, C.A.M., Takezaki, N. and Nei, M. Efficiencies of different genes and different tree-building methods in recovering a known phylogeny. Molecular Biology and Evolution. 13 (1996) 526-539.

A Genetic Algorithm for Detection of Relevant Descriptors in ADMET Prediction

Rocío L. Cecchini[1,3], Axel J. Soto[1,2,3], Gustavo E. Vazquez[1], Ignacio Ponzoni[1,2]

[1] Laboratorio de Investigación y Desarrollo en Computación Científica (LIDECC),
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur – Av. Alem 1253 – 8000 – Bahía Blanca
Argentina
{rlc, saj, gev, ip}@cs.uns.edu.ar
[2] Planta Piloto de Ingeniería Química (PLAPIQUI)
Universidad Nacional del Sur – CONICET
Complejo CRIBABB – Camino La Carrindanga km.7 – CC 717 – Bahía Blanca
Argentina
[3] These authors contributed equally to this manuscript

Abstract. In this work, a novel approach for descriptor selection aimed to physicochemical property prediction is presented. The capacity of determining the most significant set of descriptors is of great importance due to their contribution for improving ADMET prediction models. The proposed methodology combines a genetic algorithm with decision trees. Experimental analysis was carried out for predicting the octanol-water partition coefficient (logP) using neural networks as prediction method. The performance results showed the good potential of this technique.

1 Introduction

The development of drugs is a very complex task since it is difficult to know the rules that govern ADMET (Absorption, Distribution, Metabolism, Excretion and Toxicity) behavior in the human body. ADMET properties are related to the way that a drug interacts with a large number of macromolecules and they correspond to the principal cause of failure in drug development [1].

In this regard, the emerging of *in silico* methods (by computer means) was very helpful given that they allowed to examine thousands of molecules in shorter time and without the necessity of intensive laboratory work. Although *in silico* methods are not pretended to replace high-quality (*in vivo* or *in vitro*) experiments at least in the short term, they have some advantages: e.g., reduce the percentage of leads that fail in later stages of their process, rule out a lead before its synthesis, diminish the time and money invested in compounds that will not be successful, etc. Furthermore, some computer methods have demonstrated to obtain as good accuracy as well-established experimental methods [1].

Quantitative structure-activity relationships (QSAR) and quantitative structure-property relationships (QSPR) are disciplines used in chemoinformatics that comprise the methods by which chemical structure parameters are quantitatively correlated with a well defined process or experiment. In this context, hydrophobicity is one of the most extensively modeled physicochemical property since the difficulty of experimentally determine its value, and also because it is directly related with ADMET properties. This property is traditionally expressed in terms of the logarithm of the octanol-water partition coefficient (logP). QSAR have evolved over a period of 30 years from simple regression models to diverse computational intelligence models that are now applied to a wide range of problems [2]. Nevertheless, the accuracy of the ADMET property estimations remain as a challenging problem [3].

One common way of expressing the structural composition of a molecule in a QSAR method is by way of the calculus of whole-molecular descriptors. Each descriptor defines a feature of the entire molecule, and its value could be obtained by experimental measures or numerical methods. According to the nature of descriptors, they are organized by families [4]. One major dilemma when logP is intended to be modeled by QSAR is that, with the exclusion of a few common descriptors, there is no general agreement of which descriptors are relevant or influence the hydrophobic behavior of a compound. This is an important fact, because overfitting and chance correlation could occur as a result of using more descriptors than necessary [5]. If less or different than influential descriptors are used, poor models come as a result. In this way, this work presents a novel systematized method for inferring most influential descriptors for any physicochemical property.

2 Genetic Algorithm for Descriptor Selection

In order to make the selection of most influential descriptors we implemented a genetic algorithm denominated DS-GA. The main objective of DS-GA is to find a descriptor selection that results suitable to describe logP behavior.

Binary strings are used to represent the individuals, each string of length M standing for a feasible descriptors selection, where M is the number of considered descriptors. A nonzero value in *i*th bit position means that the *i*th descriptor is selected (Fig. 1). For this work, we have constrained to a model where only p bits could be set active for each individual at the same time. In other words, the purpose of the DS-GA is to find the p most relevant descriptors.

A one-point crossover is used for the recombination. Non feasible individual could take place after crossover, because the number of nonzero bits may be different than p. This problem is solved by randomly setting or resetting bit locations as needed, to be up to p active bits. Since the crossover scheme inherently incorporates bit-flip mutation, we abstained to use a scheme of additional mutation. In the same way, the initial population is randomly generated imposing the same restriction of exactly p descriptors for each individual. Selection method of the DS-GA is tournament.

Fitness function is implemented making use of decision trees for evaluating predictive capacity of individual. In this work, decision trees are used as regression methods

over a fixed training set where only the descriptors indicated by the individual are used to regress against logP. Final fitness value is not regression error but prediction error over an independent fixed test set.

3 Experimentation

For this paper, we look for a set of ten descriptors in order to minimize the prediction error when they are used as input of a predictor method. In other words, we propose to find the ten most influential descriptors for logP from a given descriptor family.

In order to measure the performance of the aforementioned proposal, we used the output of the DS-GA as input for a neural network (NN) ensemble. This NN was specially designed for logP prediction. Our goal is to establish whether using the descriptor set obtained by DS-GA involves a significant improvement in the prediction accuracy in relation to other selection criteria. We decided to work with a total set of first 1200 compounds (CAS-ordered) from the PHYSPROP database [6], 50% of them were used for training, 16% for validation (Set 1) and the remainder was left for testing (Sets 2 and 3).

For the DS-GA runs we use typical parameter values: population size=45; crossover probability=0.8; tournament size=3. A phenotypic stopping criterion is used; the DS-GA stops when highest fitness of the population does not improve during ten generations. With respect to the NN ensemble, each ensemble consists of five NNs, and each one has a three-level architecture with five hidden nodes.

Table 1. Mean absolute errors (MAE) for logP prediction, using three different selection methods: DS-GA, random initialization and considering all constitutional descriptors [4]. Average result corresponds to the arithmetic mean of 15 different selections using the same method. Best result corresponds to the selection with best prediction capacity from the 15 different selections of the method.

Comparison	# Descriptors	Set 1	Set 2	Set 3
Average DS-GA	10	1.4193	1.3687	1.0751
Average Random	10	1.5166	1.4318	1.1571
Abs. Difference	-	*0.0973*	*0.0631*	*0.082*
LSD p-value	-	*0.0208*	*0.0005*	*0.0004*
Best DS-GA	10	1.2885	1.2221	1.0408
Best Random	10	1.3617	1.2860	1.0788
Abs. Difference	-	*0.0732*	*0.0639*	*0.038*
LSD p-value	-	*0.0034*	*0.0019*	*0.0781*
Best DS-GA	10	1.2885	1.2221	1.0408
All Constitutional	47	1.4431	1.3037	1.0415
Abs. Difference	-	*0.1546*	*0.0816*	*0.0007*
LSD p-value	-	*0.0003*	*0.0024*	*0.9877*

Table 1 shows that most differences in prediction errors of the established comparisons are significant at least to the 95% confidence level. Best DS-GA outperforms the Random and the all constitutional selection. Some DS-GA selections are not so good as expected but on average DS-GA performs quite well.

4 Conclusions and Future Work

The present work proposes a novel and systematized methodology for improving the understanding of structure-property relationships. This approach allows to detect which descriptors are the most influential to the molecule hydrophobicity. It is clear that our proposal is not restricted to logP, because this method could also be applied to any physicochemical property. In addition, to know which are the descriptors that best encodes a specific property leads to the reduction of the prediction errors, independently of the type of applied method.

The combination of several machine learning methods often outperforms the capacity of single individually-applied classifiers [7]. One of the key contributions of our proposal is the use of decision trees in the fitness function. This feature allows a fast evaluation of whether the descriptors possessed by an individual are able to obtain good prediction capacity. Neural networks could also be used as a nonlinear predictor for the fitness function, as in the GA proposed by So *et al.* [8] for estimating drug activity, however, decision trees were preferred in our paper due to time performance.

As future work, it would be interesting to experiment this proposal with other descriptor families. Moreover, DS-GA could also detect the most adequate number of descriptors to be taken into account for a predictor method, instead of fixing to a specific number. At this moment, we are also considering to use other AI methods as feature selection technique.

Acknowledgments

Authors acknowledge the "Agencia Nacional de Promoción Científica y Tecnológica" from Argentina, for Grants Nº11-12778 and Cod. 917. They would also like to acknowledge SeCyT (UNS) for Grant PGI 24/N019.

References

1. Selick, H.E., Beresford, A.P, Tarbit, M.H.: The Emerging Importance of Predictive ADME Simulation in Drug Discovery. Drug Discov. Today 7, 2 (2002) 109-116
2. Jónsdottir, S.Ó., Jørgensen, F.S., Brunak S.: Prediction Methods and Databases Within Chemoinformatics: Emphasis on Drugs and Drug Candidates. Bioinformatics. 21 (2005) 2145-2160
3. Tetko, I.V., Bruneau, P., Mewes, H.-W., Rohrer, D.C., Poda, G.I.: Can we estimate the accuracy of ADME-Tox predictions? Drug Discov. Today. 11 (2006) 700-707
4. Todeschini, R., Consonni V.: Handbook of Molecular Descriptors. Wiley-VCH, Weinheim Germany (2000)
5. Tetko, I.V., Livingstone, D.J., Luik, A.I: Neural Networks Studies. 1. Comparison of Overfitting and Overtraining. J. Chem. Inf. Comput. Sci. 35 (1995) 826-833
6. The Physical Properties Database (PHYSPROP) is marketed by Syracuse Research Corporation (SRC), North Syracuse, USA at URL http://www.syrres.com/esc/
7. Wolpert, D.: Stacked Generalization. Neural Networks. 5 (1992) 241-260
8. So, S-S., Karplus, M.: Evolutionary Optimization in Quantitative Structure-Activity Relationship: An Application of Genetic Neural Networks. J. Med. Chem. 39 (1996) 1521-1530

A Distributed Algorithm for Phylogenetics Inference

Felipe Fernandes Albrecht[1], Jomi Fred Hübner[2], and Alberto M. R. Dávila[3]

[1] Instituto Militar de Engenharia, Seção de Engenharia de Computação
[2] FURB, Departamento de Sistemas e Computação
[3] Fundação Oswaldo Cruz, Instituto Oswaldo Cruz, Departamento de Bioquímica e Biologia Molecular

Abstract. The phylogenetics analysis that uses numerical taxonomy has a distance matrix, with the distances between the taxons. One of the numerical taxonomy techniques is the least squares. The least squares phylogenetics technique has an objective function that represents the inferred tree quality. This paper proposes a distribution of the method defined by Felsenstein, called: an alternating least squares method approach to inferring phylogenies from pairwise distances. This distribution aim the reduction of the execution time. The method proposed by Felsenstein has a execution time delayed when the set of taxons is very big. The proposal distributes the generated trees in the work processes and eliminates low quality trees. With this distribution, it is obtained a gain in the 50% in the execution time for a set containing 80 taxons, however, resulting a little reduction in the quality of the inferred trees.

1 Introduction

Molecular phylogenetics is the study of the evolutionary relations between different taxons, being they, the DNA, RNA, or proteic sequences. One technique used in molecular phylogenetics is the numerical taxonomy, where a distance matrix with the distances between the taxons is used. The techniques of the molecular phylogenetic inference employing numeric taxonomy are diverse: [2–5]. One of them, the least squares technique [4], has an objective function that represents the inferred tree quality. This function, shown in the equation 1, calculates the difference between the tree taxons distances and the input matrix taxons distances.

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(D_{ij} - d_{ij})^2 \quad (1)$$

To compute the tree branch lengths, Cavalli-Sforza[4] and Felsenstein[6] describes a method that uses a set of linear equations, but Felsenstein[1] says that this method needs a tree with a specific topology and the resolution of the linear equations can be a complicated and an expensive task.

Felsenstein [1] shows an alternative least squares method. This method infers the tree by steps, each one adding a new taxon in the tree, searching among all possibles positions and choosing one that has the lower least squares. After choosing the best position and calculating the branch lengths, some optimizations are executed to decrease the tree least squares. A new set of trees is generated using the optimized tree of the previous step, and again it is chosen the tree with the lower least squares to perform the optimization. The iterations are repeated until all taxons of the input matrix are in the tree.

Analyzing this method, it is clear that it works like a search algorithm [7]. Where a data set is initially generated and is after a search among this set is performed to find the best data. In the case of the least squares method, the data set is all possible trees and the best data is the tree with the lower least square. The Figure 1 shows that at each algorithm step, a set of trees is created and the tree with lower last square is chosen. This selected tree is therefore used as the starting point for the trees created at the next step.

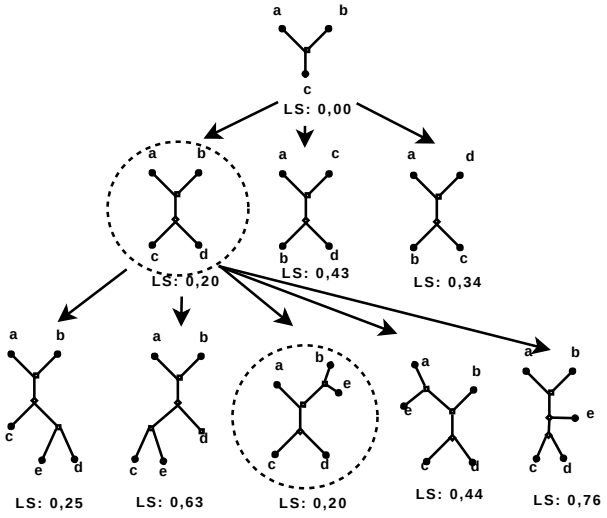


Fig. 1. Searching for the tree with lower least squares.

This algorithm can be distributed among several processes, where each process would be responsible for a tree set. Each process add new taxon in each tree of its set and in this way, creates new trees. Also, each process analysis its generated trees and eliminates those that has the worse least squares.

2 The Proposed Algorithm

The proposed algorithm is divided into two kinds of process: the scheduler and the workers. The scheduler creates the initial trees set and assigns a disjoint sub-

set of the set for each worker process. For this assignment, the scheduler creates all possible taxons triples and calculates the distances between the taxons that form it. Hence the scheduler gets the triples that has the lowers total distances and allocates a set of them for the workers. The size of the allocate is an user parameter.

For each algorithm iteration, the worker processes generate a new tree for each position of each tree where it is possible to insert a new taxon. After the generation of each tree, the worker calculates its least squares. When all the new trees was generated and its least squares calculated, the worker calculates the average $\bar{x}$ and standard deviation σ of these least squares values. Each worker processes use these values to eliminate trees that have least square greater than the threshold defined as "$\imath.\bar{x} + \jmath.\sigma$", where $\imath$ and $\jmath$ values are users parameters. This way, at each iteration a tree set is generated by each worker process and some trees are eliminated. The reason for this elimination is that the successors of these trees unlikely will obtain the better final least squares.

After the new trees have been generated and the worst trees of each process removed, the worker processes sends to the scheduler a descriptor containing the trees identification and theirs least-squares. Hence, the scheduler calculates the average and standard deviation of all remained trees and sends a message to the process that created the tree informing that the tree has not a good least squares value accordingly to all trees and must be removed. The elimination process is done thus in two stages to not allow the exaggerated grow of the number of the trees and to avoid the exponential growth in the number of trees. These trees will be the base for the creation of new trees in the following iteration.

3 Implementation and Results

The implemented software, called dleastsquares[4], was written in C language and for inter-process communication, the MPI standard with its implementation LAM [8] was used. To test the implementation performance, eight distances matrix with hypothetical distances was created. Dleastsquares and kitch, from PHYLIP [9] package, were evaluated and their executions times measured. The kitch software was executed at an Intel Pentium 4 3Ghz with 1 Gigabyte and the dleastsquares at a cluster containing five of these computers. The dleastsquares was executed with the following options: 4 initial triples by process and the minimum of 20 and maximum of 40 taxon by iteration.

The dleastsquares and the kitch was executed 8 times and the matrix sizes varies from 10 taxons to 80 taxons. It is shown in figure 3 the execution time. Note that for a matrix with less than 50 taxons, the kitch performance is better. Otherwise, with matrices with more than 50 taxons, the dleastquares performance outperforms the kitch. With a matrix with 80 taxons, the dleastsquares shows a time gaim of 50%.

This distribution approach obtained a gain of 50% for a matrix with 80 taxons. However, a reduction in the quality of the inferred trees was produced

[4] this software is freely avaliable in http://sourceforge.net/projects/distphylo

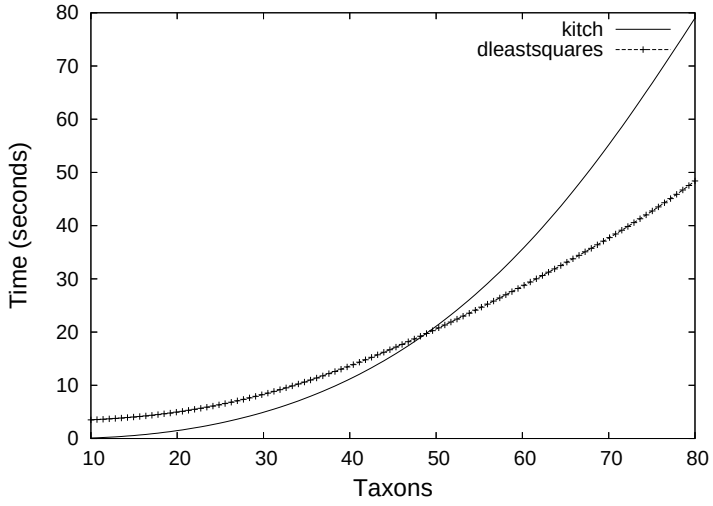


Fig. 2. Searching for the tree with lower least-squares.

because it is not done an exhausting search in all possible trees and all possible optimizations.

By this algorithm, it is possible to infer phylogenetic trees specifying parameters, like the analysed trees quantity at each iteration and the threshold for the tree elimination. Even not returning the best tree, the topology of the infered tree shows similarity with trees infered by others software.

References

[1] Felsenstein, J: *An Alternating Least Squares Approach to Inferring Phylogenies from Pairwise Distances.* Systematic Biology, **46**: 101–111. 1997.

[2] Sokal, R. R and P. H. A. Sneath: *Numerical Taxonomy.* W. H. Freeman, San Francisco, 1963.

[3] Saitou, N. and M. Nei: *The neighbor-joining method: A new method for reconstructing phylogenetic trees.* Molecular Biology and Evolution **4**: 406–425. 1987.

[4] Cavalli-Sforza LL, Edwards AWF: *Phylogenetic analysis: Models and estimation procedures.* Am J Hum Genet **19**: 233–257. 1967.

[5] Fitch, W. M and E. Margolia: *Construction of phylogenetic trees.* Science **155**: 279–284. 1967.

[6] Felsenstein, J: *Inferring Phylogenies.* Sinauer, Washington, 2004.

[7] Thomas H. Cormen and Charles E. Leiserson and Ronald L. Rivest and Clifford Stein: *Introduction to Algorithms.* The MIT Press, Cambridge, MA, 2001.

[8] Burns, Greg and Daoud, Raja and Vaigl James. Lam: an open cluster environment for MPI. *In: SUPERCOMPUTING SYMPOSIUM'94.* 42–386. Toronto: University of Toronto 1994.

[9] Felsenstein, J: **PHYLIP (phylogeny inference package)**, version 3.6. Washington, 2005. http://evolution.genetics.washington.edu/phylip/getme.html.

PHYLOGENETIC ANALYSIS OF THE *WRKY* TRANSCRIPTION FACTORS GENE SUPERFAMILY IN COFFEE PLANTS

RAMIRO Daniel[1,2], PETITOT Anne-Sophie[1], MALUF Miriam[3], FERNANDEZ Diana[1]

[1] IRD (Institut de Recherche pour le Développement), UMR1097 IRD/CIRAD/INRA, BP64501, 34032 Montpellier Cedex 5, France, [2] IAC (Instituto Agronômico de Campinas), Centro de Café 'Alcides Carvalho' - CP 28, 13.001-970 Campinas, SP - Brasil. [3]Embrapa Café, Centro de Café 'Alcides Carvalho' - CP 28, 13.001-970 Campinas, SP - Brasil.

Summary

WRKY family proteins are transcription factors involved in the regulation of development and plant defense response pathways. The *Arabidopsis thaliana* WRKY superfamily is made of 75 members. Common to these proteins is a DNA-binding region of approximately 60 amino acids in length which comprises the absolutely conserved sequence motif WRKY adjacent to a novel zinc-finger motif. A comparative phylogenetic analysis of the *WRKY* gene family in coffee and *A. thaliana* was conducted to assess the diversity of this family in coffee and to identify homologous coffee genes with putative function in defense responses to pathogens. Bioinformatic analysis of around 200 000 coffee Expressed Sequence Tags (ESTs) identified 313 ESTs with BLAST homologies to WRKY proteins. Almost 30 different putative WRKY genes were obtained, but only 25 unigenes encoding a protein with a WRKY domain were identified. Alignement of the WRKY domain sequences of the 25 coffee unigenes together with those of 72 *A. thaliana WRKY* genes showed a high conservation of the WRKY motif and the zinc-finger motif in the coffee WRKY domain. The 25 coffee WRKY members were distributed among the 3 main *A. thaliana* WRKY subgroups, with group I members displaying two WRKY domains, as expected. Conservation of the intron position within the WRKY domain sequence was evidenced when cloning the genomic sequence of one *WRKY* coffee gene (*CaWRKY1*). Clustering of the coffee *WRKY* genes based on the EST distribution in cDNA libraries made from tissues under several physiological conditions allowed to identify genes associated with development or with plant defense responses. To assess the involvement of *WRKY* genes in the coffee defense response pathways, gene expression patterns are being tested in coffee plants under several defense-related conditions.

Introduction

WRKY proteins are plant transcription factors encoded by a multigene family comprising over 74 genes in *Arabidopsis thaliana* (Eulgem *et al.*, 2000 ; Dong *et al.*, 2003) and more than 80 in rice (*Oriza sativa*) (Xie *et al.*, 2005). WRKY proteins are characterized by the presence of one or two DNA - binding domains which comprise the conserved WRKYGQK core motif (Eulgem *et al.*, 2000). Transcriptional regulation of a number of genes involved in several physiological processes may be driven by WRKY transcription factors (Eulgem *et al.*, 1999 ; Ülker and Somssich, 2004). So far, recent studies have shown that WRKY proteins probably have regulatory functions in seed development, sugar signalisation and plant defence responses to pathogens (for review Ülker and Somssich, 2004). Indeed, pathogen infection, wounding or treatment with salicylic acid (SA) have been shown to induce rapid expression of several *WRKY* genes from a number of plants (Dong *et al.*, 2003 ; Ryu *et al.*, 2006). In coffee (*Coffea arabica*), the *CaWRKY1* gene displayed altered expression patterns in response to biotic and abiotic treatments (Fernandez *et al.*, 2004 ; Ganesh *et al.*, 2006). Identification of regulatory genes involved in several physiological mechanisms such as disease resistance or

seed development would offer new tools for improving coffee (*C. arabica*) varieties for important agronomic traits. The aim of this study was to identify *WRKY* genes in the coffee genome by data mining large sets of Expressed Sequence Tags (ESTs) and to predict their involvement in different physiological processes based on their expression patterns.

Results

Identification of coffee WRKY genes

Coffee *WRKY* genes were retrieved from ESTs databases by keyword searches of annotated unigenes as well as by multiple BLAST searches using the WRKY domain sequence. The databases searched included (i) the Brazilian Coffee Genome Project ESTs database (http://www.lge.ibi.unicamp.br) which comprises more than 30 000 unigenes isolated from 27 cDNA libraries made from coffee (mostly *C. arabica*) tissues under several physiological conditions (Vieira *et al.*, 2006), (ii) the *C. canephora* ESTs database developed from 5 cDNA libraries made from coffee leaves and seeds at a range of developmental stages (http://www.sgn.cornell.edu) and comprising more than 13 000 unigenes (Lin *et al.*, 2005) and (iii) the IRD *C. arabica* EST database made of 1900 unigenes from defence-specific subtractive cDNA libraries (Fernandez *et al.*, 2004 ; Lecouls *et al.*, 2006).

Table 1. List of coffee unigenes encoding a putative WRKY transcription factor.

Coffee clone	Origin	*AtWRKY* best BlastX	*A. thaliana* group	Expression group
CaWRKY-C5	*C. arabica*	33	I	C
CaWRKY-FR2-5E8	*C. arabica*	33	I	A
CcWRKY-126831	*C. canephora*	33	I	pericarp
CaWRKY-C10	*C. arabica*	33	I	A
CaWRKY-23-A03	*C. arabica*	44	I	rust-induced
CcWRKY-119460	*C. canephora*	40	IIa	early-stage cherry
CaWRKY-C14	*C. arabica*	40	IIa	C
CcWRKY-130063	*C. arabica*	40	IIa	early-stage cherry
CaWRKY-C23	*C. arabica*	40	IIa	A
CaWRKY1	*C. arabica*	6	IIb	rust-induced
CaWRKY-C2	*C. arabica*	31	IIb	C
CaWRKY-C4	*C. arabica*	57	IIc	B
CaWRKY-C18	*C. arabica*	75	IIc	C
CaWRKY-C22	*C. arabica*	21	IId	C
CaWRKY-FR2-82A10	*C. arabica*	74	IId	A
CcWRKY-130733	*C. canephora*	21	IId	early-stage cherry
CcWRKY-125957	*C. canephora*	15	IId	pericarp
CaWRKY-C25	*C. arabica*	7	IId	C
CaWRKY-CB1-73G5	*C. arabica*	11	IId	B
CaWRKY-C24	*C. arabica*	27	IIe	A
CaWRKY-EA1-7B7	*C. arabica*	14	IIe	A
CcWRKY-125811	*C. canephora*	69	IIe	leaf
CaWRKY-C12	*C. arabica*	53	III	A
CaWRKY-C13	*C. arabica*	53	III	A
CaWRKY-C21	*C. arabica*	70	III	B
CaWRKY-C28	*C. arabica*	54	III	B

We identified 313 ESTs with BLAST homologies to WRKY proteins. Search for the specific DNA-binding protein domain (WRKYGQK sequence followed by a C2H2- or C2HC-type of zinc finger motif) (Eulgem *et al.*, 2000) was manually performed on the coffee unigene

sequences. Almost 30 different putative *WRKY* genes were obtained, but only 25 unigenes encoding a protein with one or two WRKY domains were identified (Table 1). The remaining unigene sequences either did not cover the WRKY domain or ended within the domain, thus impairing further analyses.

Classification of WRKY genes on the basis of the WRKY domain sequences

BLAST homology to *A. thaliana* WRKY sequences were searched in GenBank database. The C-terminal WRKY domain sequences (68 amino acid residues) of 72 *A. thaliana WRKY* genes and the 25 coffee unigenes were aligned and a phylogenetic tree was constructed using the Lasergene software package (DNAStar, Inc., USA). Coffee genes were classified into the 3 main *A. thaliana WRKY* genes groups (Eulgem *et al.*, 2000) (Fig.1 and Table 1). A high conservation of the WRKY motif and the zinc-finger motif was observed between the two plants. Group 3 *WRKY* coffee genes had a C2HC-type zinc-finger motif ($C\text{-}(X)_7\text{-}C\text{-}(X)_{23}\text{-}H\text{-}X\text{-}C$) whereas all other coffee WRKY genes had a C2H2-type ($C\text{-}(X)_n\text{-}C\text{-}(X)_p\text{-}H\text{-}X\text{-}H$).

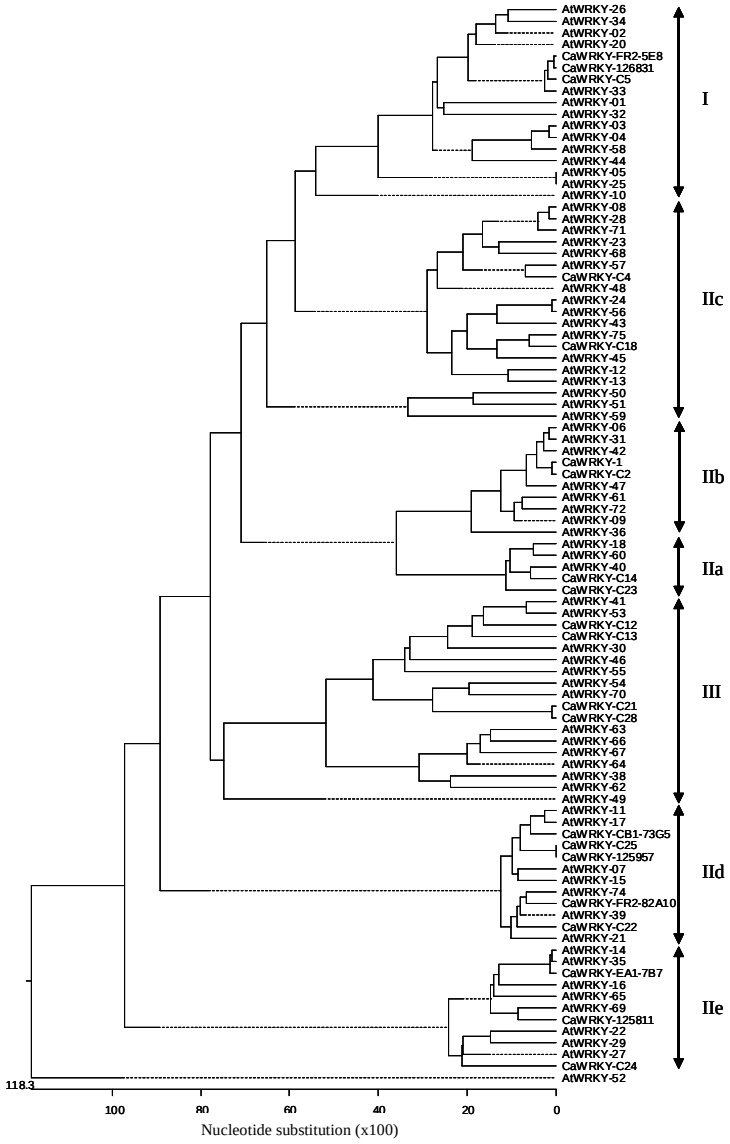


Fig.1. Dendrogram showing phylogenetic relationships between coffee and *A. thaliana* WRKY domains. Numbers on the right are the phylogenetic groups assigned to *A. thaliana* WRKY proteins (Eulgem *et al.*, 2000).

Alignement of the CaWRKY1 genomic and cDNA sequences (Petitot et al., 2006) showed the presence of an intron within the WRKY domain. The intron position (after the first Q residue of the zinc-finger domain) was highly conserved with that of A. thaliana WRKY genes (Eulgem et al., 2000).

Hierarchical classification of ESTs into expression groups

To identify coffee *WRKY* genes putatively associated with important physiological mechanisms such as development or plant defense responses, we analyzed the distribution of 17 *C. arabica WRKY* unigenes into the 27 cDNA libraries of the Brazilian coffee genes database. The presence/absence of WRKY ESTs in each cDNA library was recorded as a (0;1) matrix and used to construct a distance matrix (Simple-matching index) and a dendrogram with the UPGMA algorithm (Sneath and Sokal, 1973) contained in the software package TREECON, version 1.3b (Van de Peer and De Wachter, 1994). Coffee unigenes could be separated into 3 main groups based on their library distribution (Fig. 2). The first cluster (expression group A) grouped unigenes only present in cDNA libraries involved in plant development (different fruit stages, embryogenic calli and lines), the second cluster (expression group B) contained ESTs from a cDNA library made from acibenzolar-S-methyl and brassinosteroide-induced tissues. The remaining unigenes (expression group C) were each largely distributed over 4-10 cDNA libraries and could not be assigned to a particular physiological trait. Future work will aim at identifying coffee *WRKY* genes involvement in agronomically important traits.

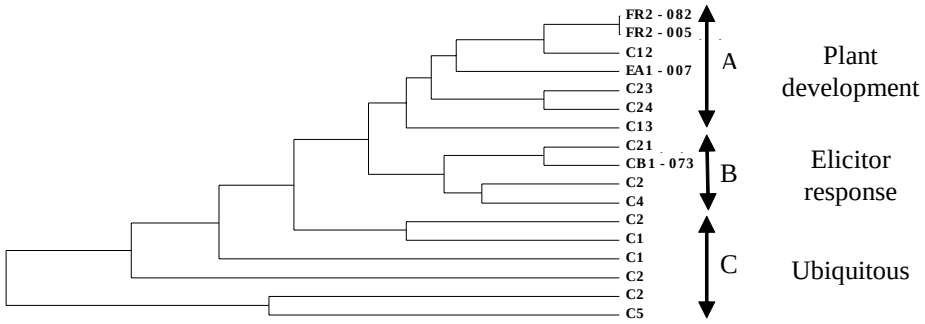


Fig. 2. Dendrogram showing relationships among *C. arabica WRKY* unigenes based on their expression data.

References

Dong J., Chen C. and Chen Z.. 2003. Plant Mol. Biol. 51:21-37.
Eulgem T., Rushton P.J., Robatzek S. and Somssich I.E. 2000. Trends Plant Sci. 5:199-206.
Fernandez D., Santos P., Agostini C. *et al.* 2004. Mol Plant Pathol., 5, 527-536.
Ganesh D., Petitot A.-S., Silva M. *et al.* 2006. Plant Science, 170:1045-1051.
Lecouls A.-C., Petitot A.-S. and Fernandez D. 2006. Proc. XXI Scientific Colloquium on Coffee, ASIC, Montpellier.
Lin C., Mueller L.A., Mc Carthy J. *et al.* 2005. Theor. Appl. Genet. 112:114-30.
Petitot A.-S., Lecouls A.-C. and Fernandez D. 2006. Phylogenetic origins and expression analysis of a duplicated *WRKY* gene in the polyploid species *Coffea arabica*. Proc. XXI Scientific Colloquium on Coffee, ASIC, Montpellier.
Ryu H.S., Han M., Lee S.K. *et al.* 2006. Plant Cell Rep. DOI10.1007/s00299-006-0138-1.
Ülker B. and Somssich I.E. 2004. Current Opinion in Plant Biology, 7, 491– 498.
Vieira L. G. E., Andrade A. C., Colombo C.A. *et al.* 2006. Brazilian Journal of Plant Physiology, 18, 95-108.
Van de Peer Y. and De Wachter R. 1994. Comput. Applic. Biosciences 10, 569-570.
Xie Z., Zhang Z.L., Zou X. *et al.* 2005. Plant Physiol, 137 :176-189.

Polynomial-sized ILP Models for Rearrangement Distance Problems

Zanoni Dias and Cid Carvalho de Souza

University of Campinas, Institute of Computing,
P.O.Box 6167, 13084-971,
Campinas, Brazil
{zanoni,cid}@ic.unicamp.br

Abstract. Genome Rearrangements research appeared in the last years to deal with problems such as, for instance, to find the minimum number of rearrangement events needed to transform one genome into another. In this work we present the first known polynomial-sized Integer Linear Programming (ILP) models for rearrangement distance problems. Specifically, we present ILP models for distance problems where events are restricted to reversals, or to transpositions or when both these events are allowed.

1 Introduction

Sequence comparison is one of the most studied problems in Computer Science. Usually we are interested in finding the minimum number of local operations, such as insertions, deletions, and substitutions that transform a given sequence into another given sequence. This is the edit distance problem, described in many Computational Biology textbooks [19]. However, several studies have shown that global operations such as reversals and transpositions (also called rearrangement events) are more appropriate when we wish to compare the genomes of two species [18].

A new research area called Genome Rearrangements appeared in the last years to deal with problems such as, for instance, to find the minimum number of rearrangement events needed to transform one genome into another. In the context of Genome Rearrangements, a genome is represented by an n-tuple of genes (or gene clusters). When there are no repeated genes, this n-tuple is a permutation. We proceed with a brief overview of the literature related to the present work.

The best studied rearrangement event is the reversal. A reversal inverts a block of any size in a genome. Caprara [5] proved that finding the minimum number of reversals needed to transform one genome into another is an NP-hard problem. Bafna and Pevzner [2] have presented a simple algorithm with approximation factor 2 for this problem. In 2002, Berman, Hannenhalli and Karpinski [4] gave the best known algorithm for the problem, with factor $\frac{11}{8}$.

Hannenhalli and Pevzner [11] have studied the reversal distance problem when the orientation of genes is known. In this case they proved that there

is a polynomial algorithm for the problem. This algorithm has been refined successively until 2004, when Tannier and Sagot [20] proposed a sub-quadratic algorithm. When just the distance is needed, a faster, linear algorithm due to Bader, Moret, and Yan [1] can be used. Meidanis, Walter e Dias [14] have shown that all the reversal theory developed for linear genomes can be easily adapted to circular genomes.

The rearrangement event called transposition has the property of exchanging two adjacent blocks of any size in a genome. The transposition distance problem, that is, the problem of finding the minimum number of transpositions necessary to transform one genome into another, has been studied by Bafna and Pevzner [3], who presented the first approximation algorithm for the problem, with factor $\frac{3}{2}$. Recently, Elias and Hartman [9] proposed a new $\frac{11}{8}-$approximation algorithm.

Transposition distance problem is still open: we do not know any NP-hardness proof, and there are no evidences that an exact polynomial algorithm exists.

Walter, Dias and Meidanis [15, 21] and Lin and Xue [13] studied the problem of finding the minimum number of transpositions and reversals necessary to transform one genome into another.

Greenberg, Hart and Lancia [10] and Meneses, Oliveira and Pardalos [16] gave an overview of Integer Linear Programming (ILP) models employed for solving problems in genomics and proteomics. Caprara, Lancia and Ng [6, 7] developed practical solutions for the reversal distance problem. They present an approach based on the use of Linear Programming (LP). In particular, they deal with LP relaxation of an ILP model with an exponential number of variables and constraints, using a branch-and-price column generation scheme.

To the best of our knowledge, this work proposes the first polynomial-sized ILP models for rearrangement distance problems. Specifically, we derive compact formulations for distance problems using reversals only, transpositions only and, finally, both events.

The paper is divided as follows. Section 2 provides the important concepts and definitions used throughout text. Section 3 presents the ILP models and Section 4 discusses some computational tests. Finally, Section 5 exhibits our conclusions and suggestions for future work.

2 Definitions

We now introduce a number of basic concepts used in Genome Rearrangements. Notice, however, that some definitions, for instance that of transposition and reversal, are different from the definition used in other areas. We start with some definitions which apply to rearrangement problems in general.

Definition 1. *An arbitrary genome formed by n genes will be represented as a permutation $\pi = [\pi[1]\ \pi[2]\ \ldots\ \pi[n]]$ where each element of π represents a gene. The identity genome ι_n is defined as $\iota_n = [1\ 2\ \ldots\ n]$.*

Definition 2. *A transposition $\rho(x, y, z)$, where $1 \leq x < y < z \leq n + 1$, is a rearrangement event that transforms π into the genome $\rho\pi = [\pi[1] \ \ldots \ \pi[x-1] \ \pi[y] \ \ldots \ \pi[z-1] \ \pi[x] \ \ldots \ \pi[y-1] \ \pi[z] \ \ldots \ \pi[n]]$.*

Definition 3. *A reversal $\rho(x, y)$, where $1 \leq x < y \leq n$, is a rearrangement event that transforms π into the genome $\rho\pi = [\pi[1] \ \ldots \ \pi[x-1] \ \pi[y] \ \pi[y-1] \ldots \ \pi[x+1] \ \pi[x] \ \pi[y+1] \ \ldots \ \pi[n]]$.*

Definition 4. *Given two genomes π and σ we define the transposition distance $d_t(\pi, \sigma)$ between these two genomes as being the least number of transpositions needed to transform π into σ, that is, the smallest r such that there are transpositions $\rho_1, \rho_2, \ldots \rho_r$ with $\rho_r \ldots \rho_2\rho_1\pi = \sigma$. We call sorting distance by transpositions, $d_t(\pi)$, the transposition distance between the genomes π and ι_n, that is, $d_t(\pi) = d_t(\pi, \iota_n)$.*

Definition 5. *Given two genomes π and σ we define the reversal distance between these two genomes, $d_r(\pi, \sigma)$, as being the least number of reversals needed to transform π into σ, that is, the smallest r such that there are reversal ρ_1, ρ_2, $\ldots \rho_r$ with $\rho_r \ldots \rho_2\rho_1\pi = \sigma$. We call sorting distance by reversals, $d_r(\pi)$, the reversal distance between genomes π and ι_n, that is, $d_r(\pi) = d_r(\pi, \iota_n)$.*

Definition 6. *Given two genomes π and σ we define the reversal and transposition distance $d_{rt}(\pi, \sigma)$ between these two genomes as being the least number of reversals or transpositions needed to transform π into σ, that is, the smallest r such that there are reversal or transpositions $\rho_1, \rho_2, \ldots \rho_r$ with $\rho_r \ldots \rho_2\rho_1\pi = \sigma$. We call sorting distance by reversals and transpositions, $d_{rt}(\pi)$, the reversal and transposition distance between genomes π and ι_n, that is, $d_{rt}(\pi) = d_{rt}(\pi, \iota_n)$.*

Definitions for the transposition distance problem. We now focus on the transposition case where we usually extend permutation π by adding $\pi_0 = 0$ and $\pi_{n+1} = n + 1$. This extended permutation will still be denoted by π.

A *transposition breakpoint* of a permutation π is a pair $x = (\pi_i, \pi_{i+1})$ such that x is not of the form (σ_j, σ_{j+1}) for some j such that $0 \leq j \leq n$. Therefore, to reach σ from π, we must have at least one operation "separating" π_i and π_{i+1}. Breakpoints are indicated by a bullet between π_i and π_{i+1} (see Figure 1). We denote by $b_t(\pi, \sigma)$ the number of transposition breakpoints of π with respect to σ. Breakpoints divide a permutation into *strips*.

0 • 5 • 1 2 • 4 • 6 7 • 3 • 9 • 8 • 10

Fig. 1. Strips and breakpoints of a permutation π = (0 5 1 2 4 6 7 3 9 8 10) with respect to σ = (0 1 2 3 4 5 6 7 8 9 10). Strips are the sequences between two consecutive breakpoints. In this case, $b_t(\pi, \sigma) = 8$.

Lemma 21 *For any distinct permutations π and σ, $3 \leq b_t(\pi, \sigma) \leq n + 1$.*

Lemma 22 *For any permutations π and σ and any transposition ρ we have that $\Delta b_t(\rho, \pi, \sigma) = b_t(\rho\pi, \sigma) - b_t(\pi, \sigma)$ and $\Delta b_t(\rho, \pi, \sigma) \in \{-3, -2, -1, 0, 1, 2, 3\}$.*

Theorem 21 *For any permutations π and σ, $\frac{b_t(\pi,\sigma)}{3} \leq d(\pi, \sigma) \leq b_t(\pi, \sigma)$.*

Theorem 22 *(Christie [8]) For any permutations π and σ there is a series of transpositions $\rho_1, \rho_2, \ldots, \rho_{d_t(\pi,\sigma)}$ such that $\rho_{d_t(\pi,\sigma)} \ldots \rho_2\rho_1\pi = \sigma$, which satisfies $\Delta b_t(\rho_k, \rho_{k-1} \ldots \rho_1\pi, \sigma) \geq 0$, for all $k \in [1..d_t(\pi, \sigma)]$.*

A powerful tool for studying the transposition distance is the *cycle graph* (also called *reality and desire diagram*) of two permutations. Suppose we want to compute $d_t(\pi, \sigma)$. We construct this diagram writing the origin permutation π in the following way. Replace each integer i by a pair of points $-i$ and $+i$, in this order, and add two extra points, one called $+0$ at the beginning of the sequence, and one called $-(n+1)$ at the end of the sequence. Now draw oriented black (*reality*) edges from $-\pi_1$ to $+0$, from $-\pi_{i+1}$ to $+\pi_i$, and from $-(n+1)$ to $+\pi_n$. Finally, draw oriented gray (*desire*) edges from $+0$ to $-\sigma_1$, from $+\sigma_i$ to $-\sigma_{i+1}$, and from $+\sigma_n$ to $-(n+1)$.

The cycle graph has exactly $n + 1$ black edges and the same number of gray edges. Figure 2 shows the cycle graph corresponding to a pair of permutations.

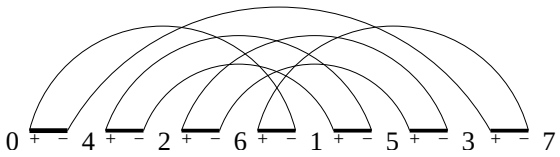


Fig. 2. Cycle graph for two permutations, π = [4 2 6 1 5 3] and σ = [1 2 3 4 5 6], as showed. In this figure, black edges are represented by horizontal lines and gray edges by arcs.

The cycle graph is composed of a number of cycles, with each cycle alternating between black and gray edges. The *size* of a cycle is the number of black edges in it (which is the same as the number of gray edges in it). Denote by $c_{odd}(\pi, \sigma)$ the number of cycles of odd size in the cycle graph of π and σ. In figure 2, we have $c_{odd}(\pi, \sigma) = 1$.

Lemma 23 *For any distinct permutations π and σ, $1 \leq c_{odd}(\pi, \sigma) \leq n + 1$.*

Lemma 24 *For any permutations π and σ and any transposition ρ we have that $\Delta c_{odd}(\rho, \pi, \sigma) = c_{odd}(\rho\pi, \sigma) - c_{odd}(\pi, \sigma)$ and $\Delta c_{odd} \in \{-2, 0, 2\}$.*

Theorem 23 *(Bafna and Pevzner [3]) For any permutations π and σ we have:*

$$\frac{(n+1) - c_{odd}(\pi, \sigma)}{2} \leq d(\pi, \sigma) \leq 3\frac{(n+1) - c_{odd}(\pi, \sigma)}{4}$$

3 ILP models for genome rearrangement problems

In this section we present mathematical formulations for the problems of rearrangement distances related to the events of transposition, reversal and also to the variant where both operations are allowed.

Common variables and constraints. We start by introducing the variables and constraints that are common to all distance models and whose role is to ensure that we only deal with valid permutations.

Generating valid permutations at each step. The binary variable B_{ijk} indicates if the i-th position of π, after the k-th operation has been completed has value j, for all $1 \leq i, j \leq n$ and all $0 \leq k < n$, that is:

$$B_{ijk} = \begin{cases} 1, \text{ if } \pi[i] = j \text{ after the } k\text{-th operation} \\ 0, \text{ otherwise} \end{cases}$$

Given these variables, the following constraints guarantee that the initial and final permutations are correct:

$$B_{i,\pi[i],0} = 1, \text{ for all } 1 \leq i \leq n \quad (1)$$

$$B_{i,\sigma[i],n-1} = 1, \text{ for all } 1 \leq i \leq n \quad (2)$$

It is easy to see that constraints (1) are satisfied if π represents the initial permutation. On the other hand, constraints (2) are fulfilled only if, after all operations have been done, the elements in the final permutation π match those in the target permutation σ.

As for the intermediate permutations, constraints (3) establish that any position of a permutation has exactly one value associated to it while constraints (4) enforce that every value is assigned to one position of each permutation.

$$\sum_{j=1}^{n} B_{ijk} = 1, \text{ for all } 1 \leq i \leq n, 0 \leq k < n \quad (3)$$

$$\sum_{i=1}^{n} B_{ijk} = 1, \text{ for all } 1 \leq j \leq n, 0 \leq k < n \quad (4)$$

Transposition distance. We now focus on the transposition operator. For all $1 \leq a < b < c \leq n+1$ and all $1 \leq k < n$, the binary variable t_{abck} indicates whether the t-th transposition switches the places of blocks $\pi[a \dots b-1]$ and $\pi[b \dots c-1]$ of π. So,

$$t_{abck} = \begin{cases} 1, \text{ if } \rho_k = \rho(a, b, c) \\ 0, \text{ otherwise} \end{cases}$$

Now, for all $1 \leq k < n$, the binary variable t_k is used to decide if the k-th transposition operation has modified the permutation, that is:

$$t_k = \begin{cases} 1, \text{ if } \rho_k = \rho(x, y, z) \text{ and } \rho_k \rho_{k-1} \dots \rho_1 \pi \neq \rho_{k-1} \dots \rho_1 \pi \\ 0, \text{ otherwise} \end{cases}$$

Besides the constraints given above, the next two sets of constraints are necessary to identify the transpositions actually chosen to be part of the solution:

$$t_k \leq t_{k-1}, \text{ for all } 1 \leq k < n \tag{5}$$

$$\sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \sum_{c=b+1}^{n+1} t_{abck} \leq t_k, \text{ for all } 1 \leq k < n \tag{6}$$

Constraints (5) ensure that if a transposition does not modify the permutation, the subsequent ones cannot modify it either. As for constraints (6), they impose that at most one transposition is done at each step.

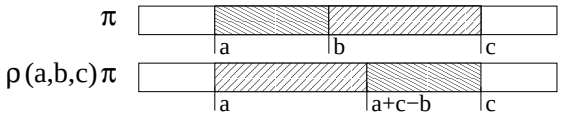


Fig. 3. Transposition $\rho(a, b, c)$ applied to a permutation π. The blocks $\pi[a, b-1]$ and $\pi[b, c-1]$ switch their positions.

Next, we discuss the constraints that reflect the modifications in the permutation caused by the transposition executed at each step. The analysis of the meaning of these inequalities is divided into three cases. To this end, we inspect each position i of the permutation to verify its value after the transposition $\rho(a, b, c)$ has been completed.

1. $i < a$ or $i \geq c$: the values in these positions remain unchanged.

$$\sum_{a=i+1}^{n-1} \sum_{b=a+1}^{n} \sum_{c=b+1}^{n+1} t_{abck} + \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \sum_{c=b+1}^{i} t_{abck} + \\ +(1 - t_k) + B_{i,j,k-1} - B_{ijk} \leq 1, \\ \text{for all } 1 \leq i, j \leq n, \text{ and all } 1 \leq k < n \tag{7}$$

Notice that the constraints (6) force the summation of the three first terms in (7) to be less or equal than one. If the sum is null, this means that the k-th transposition caused some modification in the permutation ($t_k = 1$) and will alter the value of position i. In this case, the inequality is trivially seen to be true for whatever the values assigned to position i before and after the k-th transposition. On the contrary, if the summation of the three first terms is one, two cases have to be considered:

- $t_k = 0$ (the two first terms are null): if $B_{i,j,k-1} = 1$, the constraint forces the variable $B_{i,j,k}$ to take a value of one. This is correct since no movement was done. On the other hand, if $B_{i,j,k-1} = 0$ the constraint becomes redundant.

- $t_k = 1$ (the third term us null): one of the two first terms has value one. But, the corresponding movement has not altered the element in position i. Therefore, when $B_{i,j,k-1} = 1$, $B_{i,j,k}$ is also forced to one, as expected. Again, if $B_{i,j,k-1} = 0$ the inequality is redundant.

2. $a \leq i < a + c - b$: after the transposition, these positions will be filled with the elements that were in positions from b to $c - 1$.

$$t_{abck} + B_{b-a+i,j,k-1} - B_{ijk} \leq 1, \qquad (8)$$
$$1 \leq a < b < c \leq n+1, a \leq i < a + c - b, 1 \leq j \leq n, 1 \leq k < n$$

This inequality is redundant unless the two first terms are both set to one. In this case, we have that $B_{b-a+i,j,k-1} = 1$, implying that element j that was stored in position $b - a + i$ before the transposition will be moved to position i, i.e., $B_{ijk} = 1$.

3. $a + c - b \leq i < c$: after the transposition, these positions will be occupied with the elements that were in positions from a to $b - 1$.

$$t_{abck} + B_{b-c+i,j,k-1} - B_{ijk} \leq 1, \qquad (9)$$
$$1 \leq a < b < c \leq n+1, a + c - b \leq i < c, 1 \leq j \leq n, 1 \leq k < n$$

Similar to what was observed in the previous case, inequality 9 is redundant unless the two first terms on the left-hand side are both equal to one. This means that the k-th transposition moves $B^{k-1}[a..b-1]$ to the positions preceding position c. From its definition, i represents one of the positions that will receive an element of this sub-vector. Thus, we have that $B^k[i] = B^{k-1}[b - c + i]$, for all $i \in [a + c - b..c - 1]$ and the last two terms cancel.

Using theorem 23, we can also obtain additional constraints to impose the upper and lower bounds as defined by Bafna and Pevzner [3].

$$t_k * n + k - 1 \geq LB(\pi, \sigma), \text{ for all } 1 \leq k \leq n \qquad (10)$$
$$t_k * k \leq UB(\pi, \sigma), \text{ for all } 1 \leq k \leq n \qquad (11)$$

where $LB(\pi, \sigma)$ and $UB(\pi, \sigma)$ are, respectively, the lower and the upper odd cycle bounds, which are easily computed from the permutations π ans σ.

Reversal Distance. To deal with reversals, we first define the following set of variables. For all $1 \leq a < b \leq n$ and all $1 \leq k < n$, the binary variable r_{abk} indicates if the k-th reversal affects the block $\pi[a \ldots b]$ of π. Thus,

$$r_{abk} = \begin{cases} 1, \text{ if } \rho_k = \rho(a, b) \\ 0, \text{ otherwise} \end{cases}$$

The binary variable r_k indicates if the k-th operation was a reversal that modified the permutation. Thus, for all $1 \leq k < n$, we have that:

$$r_k = \begin{cases} 1, \text{ if } \rho_k = \rho(x, y) \text{ and } \rho_k \rho_{k-1} \ldots \rho_1 \pi \neq \rho_{k-1} \ldots \rho_1 \pi \\ 0, \text{ otherwise} \end{cases}$$

Two sets of constraints are needed for the proper identification of the reversals chosen to be part of a solution. They are given below.

$$r_k \leq r_{k-1}, \text{ for all } 1 \leq k < n \tag{12}$$

$$\sum_{a=1}^{n-1} \sum_{b=a+1}^{n} r_{abk} \leq r_k, \text{ for all } 1 \leq k < n \tag{13}$$

Constraints (12) ensure that if the k-th reversal does not alter the permutation, none of the subsequent reversals will do so. As for constraints (13), they impose that at most one reversal is done at each step.

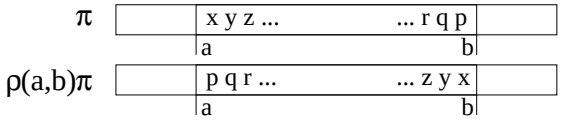


Fig. 4. Reversal $\rho(a, b)$ applied to a permutation π. The block $\pi[a, b]$ is completely reversed.

The next constraints deal with the changes in the permutation caused by reversals. The analysis is divided into two cases by inspecting what happens to each position i after the application of the reversal $\rho(a, b)$.

1. $i < a$ or $i > b$: these positions remain unchanged, and this is imposed by the constraints below.

$$\sum_{a=i+1}^{n-1} \sum_{b=a+1}^{n} r_{abk} + \sum_{a=1}^{n-1} \sum_{b=a+1}^{i-1} r_{abk} + \\ + B_{i,j,k-1} + (1 - r_k) - B_{ijk} \leq 1, \\ 1 \leq i, j \leq n, 1 \leq k < n \tag{14}$$

   Constraints (14) are the analogous counterparts of constraints (7) for the operations not affecting position i in the transposition distance problem.
2. $a \leq i \leq b$: the reversal changes the elements stored in these positions. This situation is tackled by the following constraints.

$$r_{abk} + B_{b+a-i,j,k-1} - B_{ijk} \leq 1, \\ 1 \leq a < b \leq n, a \leq i \leq b, 1 \leq j \leq n, 1 \leq k < n \tag{15}$$

   To be non-redundant, this inequality requires that the two first terms are set to one. In this case, $B_{i,j,k} = 1$, implying that the element j that was stored in position $b + a - i$ before the reversal is moved to position i.

Reversal and transposition distances. To formulate the reversal and transposition distance problem, we make use of all the variables defined earlier. Besides, we define the binary variable z_k to denote whether the k-th operation, which could be a reversal or a transposition, actually caused alterations to the permutation. Thus, for all $1 \leq k < n$, we have that

$$z_k = \begin{cases} 1, \text{ if } \rho_k = \rho(x,y) \text{ or } \rho_k = \rho(x,y,z), \text{ and } \rho_k\rho_{k-1}\ldots\rho_1\pi \neq \rho_{k-1}\ldots\rho_1\pi \\ 0, \text{ otherwise} \end{cases}$$

In what concerns the constraints in the model, we use all the inequalities introduced earlier with the exception of constraints (5), (12), (10) and (11). The former two inequalities are replaced by the following ones:

$$z_k \leq z_{k-1}, \text{ for all } 1 \leq k < n \tag{16}$$

$$r_k + t_k = z_k, \text{ for all } 1 \leq k < n \tag{17}$$

Constraints (16) ensure that if no modification took place in a given step then no operation is done in the next steps. Constraints (17) guarantee that at most one operation (a reversal or a transposition) is executed at each step.

3.1 Objective functions

Considering the variables and constraints defined earlier for each of the three distance problems, the objective functions for the transposition distance problem (ω_t), the reversal distance problem (ω_r) and for the reversal and transposition distance problem (ω_{rt}) can be written as: $\omega_t = \min \sum_{k=1}^{n-1} t_k$, $\omega_r = \min \sum_{k=1}^{n-1} r_k$, $\omega_{rt} = \min \sum_{k=1}^{n-1} z_k$.

3.2 Model sizes

It is easy to see that the models presented here are polynomial in the size of the permutation given at the input. Table 1 displays the model sizes for the three models with respect to the parameter n, i.e., the permutation size.

Table 1. Model sizes with respect to n.

Rearrangements Distance Models	*Variables*	*Constraints*
Tran position Distance	$O(n^4)$	$O(n^6)$
Reversal Distance	$O(n^3)$	$O(n^5)$
Reversal and Transposition Distance	$O(n^4)$	$O(n^6)$

4 Computational experiments

All the integer programming formulations were implemented in `Mosel` [17], an algebraic language especially designed to describe mathematical programming models. `Mosel` is part of the `Xpress` package which comprises also an optimization algorithm we use to validate our models. We carried out all our tests on microcomputer equipped with an Intel processor with a 3GHz clock, 1GB of RAM and running under a (Ubuntu) Linux operating system (kernel 2.6.15).

Table 2 summarizes our results. The `cpu` times reported refer to an average of 100 instances where the pair of permutations at the input were randomly generated. These times are given in seconds and as one can see, they grow very rapidly as the instance size increases. This behavior is due mainly to the model sizes. Though we have seen that they are polynomial in n, the order of the polynomials are too large and soon the computation of the resulting models become prohibitive in practice.

Table 2. Average times (in seconds) for the computation of rearrangement distances between two random permutations of a given size.

Size	*Transpositions*	*Reversals*	*Reversals and Transpositions*
2	0.047	0.048	0.049
3	0.050	0.050	0.053
4	0.071	0.083	0.160
5	0.141	0.386	1.244
6	2.785	2.649	21.025
7	49.967	42.897	49.906

Besides the standard model for the transposition distance problem described in the paper, we also implemented an alternative model that includes further constraints to avoid transpositions generating additional *transposition breakpoints*. According to [8], no optimal solution for the problem has transpositions of that sort. We tested the two models on two datasets. The κ_n and τ_n sets are well-known permutation families with n + 1 breakpoints, besides having simple formulae for their exact transposition distance. The results are reported in Table 3. The times showed in the table refer to an average of five runs to compute $d_t(\kappa_n)$ and $d_t(\tau_n)$ and are given in seconds. Columns `Break` and `Default` correspond, respectively, to the times achieved by the models with and without the additional constraints that eliminate transpositions that add breakpoints to the permutation. Columns `LB` and `UB` correspond, respectively, to the odd cycles lower and upper bounds. Finally, column `D` indicates the optimal transposition distance for each instance. A *timeout* is printed whenever a model could not find a solution within a time limit of 10 hours.

One can see that, in these experiments, the model with additional constraints (*Break*) was slower than the standard model (*Default*) in all cases but for instance

Table 3. Results for alternative models for the transposition distance problem.

	$\kappa_n = [n\ n-2\ \ldots\ n-3\ n-1]$					$\tau_n = [n\ n-1\ \ldots\ 2\ 1]$				
n	*Break*	*Default*	*LB*	*UB*	*D*	*Break*	*Default*	*LB*	*UB*	*D*
2	0.064	0.044	1	1	1	0.041	0.043	1	1	2
3	0.047	0.045	2	2	2	0.047	0.046	2	3	2
4	0.057	0.055	2	3	2	0.248	0.153	2	3	3
5	0.209	0.140	3	3	3	1.073	0.632	3	3	3
6	2.893	0.193	3	4	3	21.413	13.329	3	4	4
7	21.306	12.551	4	5	4	170.295	5851.061	4	6	4
8	850.048	296.840	4	6	4	timeout	timeout	4	6	5
9	154.565	143.479	5	6	5	timeout	timeout	5	6	5
10	timeout	timeout	5	7	5	timeout	timeout	5	7	6

τ_7. This means that the increase in the model size due to the addition of further constraints does not seem to pay off in general. However, we believe that the *Break* model may give better results in larger instances, mainly in those where the gap between the odd cycles upper and lower bounds are greater.

5 Conclusions and future work

In this paper we introduced the first polynomial-sized integer programming formulations for some problems in rearrangement distance. Though the modeling is an interesting achievement from the theoretical point of view, computational results with the formulations showed that the running times are still to big to be practical. We believe that there is room for some improvement in the transposition distance problem if we incorporate to our models the new bounds described in [9, 12]. Besides, another approach we are considering is the development of heuristics based upon these models. Similar strategies have been used successfully in other works [6, 7] that could justify the choice for this research direction.

Acknowledgments. The second author is supported by grants 03/09925-5 from FAPESP and 307773/2004-3 from CNPq.

References

1. D. A. Bader, B. M. E. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8(5):483–491, 2001.
2. V. Bafna and P. A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.
3. V. Bafna and P. A. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, May 1998.
4. P. Berman, S. Hannenhalli, and M. Karpinski. 1.375-approximation algorithm for sorting by reversals. In *Proceedings of the 10th European Symposium on Algorithms (ESA'2002)*, Lecture Notes in Computer Science, pages 200 – 210, Rome, Italy, September 2002. Springer.

5. A. Caprara. Sorting by reversals is difficult. In *Proceedings of the First International Conference on Computational Molecular Biology (RECOMB'97)*, pages 75–83, New York, USA, January 1997. ACM Press.
6. A. Caprara, G. Lancia, and S.-K. Ng. Fast practical solution of sorting by reversals. In *Proceedings of the 11th ACM-SIAM Annual Symposium on Discrete Algorithms (SODA'00)*, pages 12–21, San Francisco, USA, 2000. ACM Press.
7. A. Caprara, G. Lancia, and S.-K. Ng. Sorting permutations by reversals through branch-and-price. *INFORMS Journal on Computing*, 13(3):224–244, 2001.
8. D. A. Christie. *Genome Rearrangement Problems*. PhD thesis, Glasgow University, 1998.
9. I. Elias and T. Hartman. A 1.375-approximation algorithm for sorting by transpositions. *Transactions on Computational Biology and Bioinformatics*, 3(4):369–379, 2006.
10. H. J. Greenberg, W. E. Hart, and G. Lancia. Opportunities for combinatorial optimization in computational biology. *INFORMS Journal on Computing*, 16(3):211–231, 2004.
11. S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, January 1999.
12. A. Labarre. New bounds and tractable instances for the transposition distance. *Transactions on Computational Biology and Bioinformatics*, 3(4):380–394, 2006.
13. G.-H. Lin and G. Xue. Signed genome rearrangement by reversals and transpositions: Models and approximations. In *Proceedings of the Fifth Annual International Computing and Combinatorics Conference (COCOON'99)*, volume 1627 of *Lecture Notes in Computer Science*, pages 71–80. Springer, 1999.
14. J. Meidanis, M. E. M. T. Walter, and Z. Dias. Reversal distance of signed circular chromosomes. Technical Report IC-00-23, Institute of Computing - University of Campinas, December 2000.
15. J. Meidanis, M. E. M. T. Walter, and Z. Dias. A lower bound on the reversal and transposition diameter. *Journal of Computational Biology*, 9(5), 2002.
16. C. N. Meneses, C. A. S. Oliveira, and P. M. Pardalos. *Data Mining in Biomedicine*, chapter Mathematical Programming Formulations for Problems in Genomics and Proteomics. Springer, 2005.
17. Xpress Mosel, March 2007. http://www.dashoptimization.com/.
18. J. D. Palmer and L. A. Herbon. Plant mitochondrial dna evolves rapidly in structure, but slowly in sequence. *Journal of Molecular Evolution*, 27:87–97, 1988.
19. J. C. Setubal and J. Meidanis. *Introduction to Computional Molecular Biology*. PWS Publishing Company, 1997.
20. E. Tannier and M.-F. Sagot. Sorting by reversals in subquadratic time. In S. C. Sahinalp, S. Muthukrishnan, and U. Dogrusoz, editors, *Proceedings of the Fifteenth Annual Symposium on Combinatorial Pattern Matching (CPM'2004)*, volume 3109 of *Lecture Notes in Computer Science*, pages 1–13, Istanbul, Turkey, July 2004. Springer.
21. M. E. M. T. Walter, Z. Dias, and J. Meidanis. Reversal and transposition distance of linear chromosomes. In *Proceedings of the String Processing and Information Retrieval (SPIRE'98)*, 1998.

Using Gene Expression Analysis to Relate Disease, Genes, and Therapeutics

Mario R. Guarracino[1], Davide Feminiano[1], Francesca del Vecchio Blanco[2], and Salvatore Cuciniello[1]

[1] High Performance Computing and Networking Institute
{mario.guarracino,davide.feminiano,salvatore.cuciniello}@na.icar.cnr.it
National Research Council, Italy
[2] Department of General Pathology
Second University of Naples
francesca.delvecchioblanco@unina2.it

Abstract. To systematically pursue the discovery of the connection among diseases, expression of genes and treatments, we have devised machine learning methods that can be applied to collections of gene expressions profiles and drugs. In this study we report the advances in supervised learning methods that have been devised to analyze biological data and their application to select sets of genes to determine compounds that can be used in the treatment of a specific disease. As mining of data produced by medical equipments is becoming an increasingly challenging task, due to the size of the databases and the gradient of their update, new methods need to provide classification models that can handle the complexity of the problems. Then, a systematic mining of existing databases provides further insight in existing information. The results indicate the feasibility of the approach and suggest the value of large scale efforts in this direction.

1 Introduction

A fundamental challenge arising in bioinformatics is the development of methods and tools to connect diseases, genetic processes, and the action of treatments. Our goal is to provide methodology to discover existing relations among the expression of selected genes, diseases and drugs. With this methodology, a researcher can select drug candidates starting from genes, thus discovering unexpected relations.

Gene expression profiling has historically been applied to the mechanisms in biological pathways, such as the differentiation of breast cancers taking the BRCA1 mutations versus the BRCA2 and sporadic mutations [12]. Another standard application has been the discrimination of different but similar diseases, as in the case of classification of Acute Myeloid Leukemia versus Acute Lymphoblastic Leukemia [8]. Finally, gene expression has been used to predict cancer prognosis, in situations like the prediction of patient outcome after prostatectomy prostate cancer [22], malignant gliomas survival [19], clinical outcome

of breast cancer [25], and recurrence of hepatocellaur carcinoma after curative resection [13].

The traditional methods for analyzing gene expression profiles belong to *unsupervised learning*. Those algorithms start with a collection of multivariate data and produce groupings of samples, or combination of variables, based upon information inherent in the data, without additional outside information. Among the unsupervised methods, Eisen et al. [5] successfully applied hierarchical clustering to genomic problems, Bradley et al. [1] proposed local maximum clustering for gene expression data analysis, Kohonen et al. [16] proposed self-organizing maps (SOM) and Tavazoie et al. [24] applied K-means clustering and its variations to genetic network architectures. In recent studies, more sophisticated techniques belonging to *supervised methods*, that use a priori knowledge, have come into play. Among the supervised methods there are K-nearest neighbors (KNN) described in [4], support vector machines (SVM) [7], Fisher discriminant analysis (FDA) [4], regularized generalized eigenvalue classifier (ReGEC) [9, 26], and neural network analysis [20].

Here we envisage the use of supervised learning techniques in selecting candidate treatments of determined diseases. This idea is not entirely new. Some database are available to connect genes and drugs. Examples are Drugbank [27] and PharmGKB [14]. It is indeed possible to search Genbank for a specific gene and PharmGKB will provide information on how variation in human genetics leads to variation in response to drugs, and active bibliography on the selected subject. Another example is Drugbank, that combines chemical drug data with comprehensive drug target protein and gene expression information.

However, databases suffer from practical limitations. First, if the researcher is looking for data to confirm its speculation about a gene signature for a certain disease, it is very difficult, if not impossible, to derive an analysis for his results, using the data available in the database. Furthermore, it is not easy to determine a set of candidate treatments for a specific disease, from those available. Finally, the knowledge provided is static, in the sense that connection made available to users exists in the database and cannot be extrapolated from existing data. In other words, no machine learning and data mining tools are provided to mine the database.

Here we demonstrate on a simple but meaningful example that the application of a supervised learning algorithms in conjunction with tools to federate databases can provide an enhanced method for the relation of disease, gene expression profiles and treatments.

In this work we start describing *Incremental Learning and Decremented Characterization via Regularized Eigenvalue Classification* (ILDC-ReGEC), a supervised learning algorithm that uses a small number of samples and features for the classification of data. Then we show how, starting from the gene expression profiles, it is possible to collect information from databases and derive knowledge that can be used to select a set of candidate compounds for the treatment of a specific disease.

The notation used in the paper is as follows. A scalar will be denoted by y, while a vector by $\mathbf{x}$. Different vectors are identified by $\mathbf{x}_i$, where i is the index of a vectors set.

A unit vector will be denoted by $\mathbf{e}$ and 2-norm of $\mathbf{x}$ will be denoted by $\|\mathbf{x}\|$. The transpose of a matrix C is C^T.

The present work is organized as follows: in Section 2 we describe the Regularized Generalized Eigenvalue Classifier (ReGEC) method. In Section 3 we discuss the advantages of incremental methods. In Section 4 we detail ILD-ReGEC. In Section 5 a case study is analyzed. Finally, in Section 6 conclusions are drawn and future research directions are highlighted.

2 Regularized Generalized Eigenvalue Classifier

There are also efficient algorithms that exploit the special structure of a slightly different optimization problem, such as Generalized Proximal SVMs (GEPSVM) [17], in which the binary classification problem can be formulated as a generalized eigenvalue problem. This formulation differs from SVMs since, instead of finding one hyperplane that separates the two classes, it finds two hyperplanes that approximate the two classes. The prior study requires the solution of two different eigenvalue problems, while a classifier that uses a new regularization technique, known as Regularized General Eigenvalue Classifier (ReGEC) requires the solution of a single eigenvalue problem to find both hyperplanes [9].

Consider two matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{k\times m}$, that represent the two classes, each row being a point in the feature space.

Mangasarian et al. [17] proposes to classify these two sets of points A and B using two hyperplanes in the feature space, each closest to one set of points, and furthest from the other.

In Figure 1 it is shown that the first hyperplane is the closest to the set of points in A and the furthest from those in B and the second hyperplane is closest set of points in B and the furthest from those in A. Furthermore, Figure 1 shows that GEPSVM can linearly dicriminate sets that cannot be separated by a single hyperplane.

In order to satisfy the previous condition for the points in A, the two hyperplanes

$$K(\mathbf{x}, C)\mathbf{u}_1 - \gamma_1 = 0, \quad K(\mathbf{x}, C)\mathbf{u}_2 - \gamma_2 = 0 \tag{1}$$

can be obtained by solving the following two regularized optimization problems:

$$\min_{u,\gamma\neq 0} \frac{\|K(A, C)\mathbf{u} - \mathbf{e}\gamma\|^2 + \delta\| \left[\begin{smallmatrix}\mathbf{u}\\ \gamma\end{smallmatrix}\right] \|^2}{\|K(B, C)\mathbf{u} - \mathbf{e}\gamma\|^2} \tag{2}$$

and

$$\min_{u,\gamma\neq 0} \frac{\|K(B, C)\mathbf{u} - \mathbf{e}\gamma\|^2 + \delta\| \left[\begin{smallmatrix}\mathbf{u}\\ \gamma\end{smallmatrix}\right] \|^2}{\|K(A, C)\mathbf{u} - \mathbf{e}\gamma\|^2}, \tag{3}$$

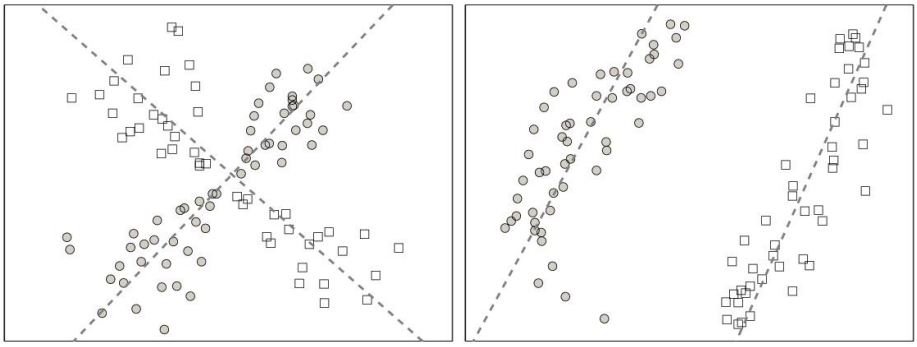


Fig. 1. Separation obtained with generalized eigenvectors.

where $C^T = \begin{bmatrix} A^T & B^T \end{bmatrix}$ and δ is the regularization parameter.

The number of eigenvalue problems can be reduced from two to one, using the new regularization method *ReGEC*, proposed by Guarracino et al. [9], by solving the following generalized eigenvalue problem:

$$\min_{u,\gamma \neq 0} \frac{\|K(A,C)\mathbf{u} - \mathbf{e}\gamma\|^2 + \delta\|\tilde{K}_B\mathbf{u} - \mathbf{e}\gamma\|^2}{\|K(B,C)\mathbf{u} - \mathbf{e}\gamma\|^2 + \delta\|\tilde{K}_A\mathbf{u} - \mathbf{e}\gamma\|^2}. \tag{4}$$

Here $\tilde{K_A}$ and $\tilde{K_B}$ are diagonal matrices with the diagonal entries from the matrices $K(A,C)$ and $K(B,C)$. The new regularization leads to a regularized problem which provides accuracy results comparable to the ones obtained by solving equations (2) and (3).

The eigenvectors related to minimum and maximum eigenvalues obtained from the solution of (4) provide the proximal planes P_i, $i = 1, 2$ to classify the new points. The distance of a point $\mathbf{x}$ from hyperplane P_i is:

$$dist(\mathbf{x}, P_i) = \frac{|K(\mathbf{x},C)\mathbf{u} - \gamma|}{\|\mathbf{u}\|}, \tag{5}$$

and the class of a point x is determined as

$$class(\mathbf{x}) = argmin_{i=-1,1}\{dist(\mathbf{x}, P_i)\}. \tag{6}$$

3 Incremental Methods

Incremental subset selection consists in constructing a small set of points that retains the information of the entire training set, providing comparable accuracy results. A kernel built from a smaller subset is computationally more efficient in predicting new elements, compared to the one that uses the entire training set. Furthermore, a smaller set of points reduces the probability of over-fitting the data. Finally, as new points become, the cost to retrain the algorithm decreases if

the influence of those new points on classification is only evaluated with respect to that subset, rather than to the whole training set.

The algorithm takes an initial set of points $C \supset C_0 = A_0 \cup B_0$ and the entire training set C as input, where A_0 and B_0 are sets of points in C_0 that belong to the two classes A and B. We refer to C_0 as the *incremental subset*. Let $\Gamma_0 = C \backslash C_0$ be the initial set of points that can be included in the incremental subset. ReGEC classifies all of the points in the training set C using the kernel from C_0. Let P_{A_0} and P_{B_0} be the hyperplanes found by ReGEC, R_0 be the classification accuracy and M_0 be the points that are misclassified. Then, among the points in $\Gamma_0 \cap M_0$ the point that is farthest from its respective hyperplane is selected, i.e.

$$\mathbf{x}_i = \mathbf{x} : \max_{\mathbf{x} \in \{\Gamma_0 \cap M_0\}} \left\{ dist(\mathbf{x}, P_{class(\mathbf{x})}) \right\}, \tag{7}$$

where $class(\mathbf{x})$ returns A or B depending on the class of $\mathbf{x}$. This point is the candidate point to be included in the incremental subset. This choice is based on the idea that a point very far from its plane either is needed in the incremental subset to improve accuracy, or it is an outlier. We update the incremental set as $C_1 = C_0 \cup \{\mathbf{x}_1\}$. Then, we classify the entire training set C using the points in C_1 to build the kernel. Let the classification accuracy be R_1. If $R_1 > R_0$ then we keep the new subset; otherwise we reject the new point, that is $C_1 = C_0$. In both cases $\Gamma_1 = \Gamma_0 \setminus \{\mathbf{x}_1\}$. The algorithm repeats until $|\Gamma_k| = 0$ at some k^{th} iteration. The s initial points are the training points closest the centroids determined by a simple *k-means* algorithm applied to each class. In [3] it has been shown that the k-means based selection criteria gives the best performance in term of stability and accuracy, with respect to random selection of initial points.

4 ILDC-ReGEC Algorithm

Features reduction methods find a *features subset* F highly correlated with a class [11]. Feature reduction is useful for many reasons: it improves performances of the training procedure avoiding many problems related to the curse of dimensionality. Indeed, in [10] it is shown that the growth of the number of features for a fixed number of samples degrades the classifier performances. Furthermore, data may be affected by noise and redundancy due to instruments and experimental conditions. The features subset F should contain only features needed to have a good accuracy of the training procedure. Features fall into one of three categories: *strongly relevant*, *weakly relevant* and *irrelevant*. *Strongly relevant* features are those necessary to obtain a good classification model; while *weakly relevant* are not always useful and irrelevant features are not important [15]. Feature reduction techniques can be divided in two classes: *feature transformation* and *feature selection*. Feature selection addresses the problem of searching a minimal set of features that maximizes the discrimination among classes. If there are n features, there are 2^n possible features subsets. When n in very large, it is impossible to find the optimal features subset, therefore a suboptimal solution needs to be found. Feature transformation methods obtain a new set of features

as linear combination of the original ones, that is, they linearly project points in a space of lower dimension. For each gene j the means μ_j^+ and μ_j^- are calculated considering each class separately. In the same way the standard deviations σ_j^+ and σ_j^- are calculated. These values are used to evaluate the discriminant between the two classes:

$$F(x_j) = |\frac{\mu_j^+ - \mu_j^-}{\sigma_j^+ + \sigma_j^-}| \tag{8}$$

The best features x_j are those with greater value of $F(x_j)$.
The function *FeatureSelection*, shown in Algorithm 1, calculates the $F(x_j)$ value for each gene and it sorts the features for decreasing the values of $F(x_j)$.

The algorithm sums $F(x_j)$ and it returns C^0, that represents the samples with genes those sum of the $F(x_j)$ is equal to a fixed percentage α of the total sum. In other words, the algorithm returns the genes x_{j_k} such that:

$$\sum_{k=1}^{m_\alpha} F(x_{j_k}) = \alpha \sum_{j=1}^{m} F(x_j), \tag{9}$$

with $0 \leq \alpha \leq 1$ and $m_\alpha \leq m$.

After selecting a subset of features ReGEC is applied in order to define the incremental subset. Solving ReGEC with a few features is very fast and therefore overall computational is improved.

As stated in section 2, at each step ReGEC section tries to add new points to the incremental subset. When a new point is added, ILDC-ReGEC checks whether it is possible to delete some features.

Let R_k be the classification accuracy with all considered features and with $\bar{R}_K$ the one obtained omitting one feature. If $\bar{R}_K >= R_k$ then it deletes last feature and the process is repeated. The procedure is restarted until further accuracy improvement is obtained.

The algorithm is depicted in Algorithm 1.

5 Results

ILDC-ReGEC has been implemented with Matlab 6.5. Results are calculated using an Intel Xeon CPU 3.20GHz, 6GB RAM running Red Hat Enterprise Linux WS release 3. Matlab function *eig* for the solution of the generalized eigenvalue problem is used as computational kernel of ReGEC.
Tests have been performed training the algorithms on a random sample of 90% of the dataset and validating the accuracy on the remaining 10%. Tests have been repeated 100 times. ILDC-ReGEC parameters have been obtained with a grid search on a sample of approximately 10% of the data set. First, δ has been determined between $[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$ for a fixed value of σ in the interval $[10^2 \quad 10^7]$. Then, σ is chosen to maximize accuracy. The number of initial points in each class has been chosen between 2 and 3. Finally, a decreasing value of α has been tested between 0.1 and 0.01, in order to have the minimum number of

Algorithm 1 ILDC-ReGEC(C_0, C,α)

$C^0 = FeatureSelection(C, \alpha)$
$\Gamma_0 = C^0 \setminus C_0^0$
$\{R_0, M_0\} = Classify(C^0, C_0^0)$
$k = 1$
$j = 0$
while $|\Gamma_k| > 0$ **do**
  $\mathbf{x}_k = \mathbf{x} : \max_{\mathbf{x} \in \{M_k \cap \Gamma_{k-1}\}} \left\{ dist(\mathbf{x}, P_{class(\mathbf{x})}) \right\}$
  $\{R_k^j, M_k^j\} = Classify(C^j, \{C_{k-1}^j \cup \{x_k\}\})$
  if $R_k^j >= R_{k-1}^j$ **then**
    $C_k^j = C_{k-1}^j \cup \{x_k\}$
    $FLAG = False$
    repeat
      $C_k^{j+1} = LastFeature(C_k^j)$;
      $\{\bar{R}_k^j, M_k^j\} = Classify(C_k^j, C_k^{j+1})$;
      if $\bar{R}_k^j > R_k^j$ **then**
        $R_k^j = \bar{R}_k^j$
        $j = j + 1$
      else
        $FLAG = True$
      end if
    until FLAG
  end if
  $\Gamma_k = \Gamma_{k-1} \setminus \{\mathbf{x}_k\}$
  $k = k + 1$
end while

features, fixed the accuracy. Our classification error on Golub dataset is 0.0286% and the mean number of selected genes by ILDC-ReGEC is 66.99. Then, the algorithm is trained with the complete dataset, yielding a single missclassified experiment and selecting 66 genes.

Table 1. Number of genes selected with different algorithms and percentage of agreement with ILDC-ReGEC

	ILDC-ReGEC	**Golub**	**Yeoh**	**Ross**	**Cheok**	**Naoe**	**Felix**	**Sjblom**
Number of Genes	67	50	606	477	166	3	2	122
Agreement	100%	46%	2%	3%	1%	33%	50%	1%

Various studies have been done to determine the gene signature of different forms of leukemia disease. In [8] the discrimination is dome between AML and ALL using SOMs, in [28] and [21] pediatric ALL is analyzed, in [2] and in [18] AML is studied, in [6], in [23] human breast and colorectal cancer are considered. In Table 2 and 1 we report the results obtained federating the datasets of those studies. In Table 1 we show the percentage of genes that have been selected by ILDC-ReGEC and referenced in the other studies. We note that 23 genes are present in Golub and in the present study. 8 genes selected by ILDC-ReGEC have been found in 2 more papers but not in Golub. Other 7 genes selected by ILDC-ReGEC have been found in at least one other papaer. The MPO myeloperoxidase gene is present in half of the studies, but not Golub. TOP2B, is as well present in 4 studies and it functions as the target for several anticancer agents.

Some of the genes that were selected by ILDC-ReGEC algorithm do not show up in any of the selected studies. Nevertheless they are supposed to have an important role in leukemia, as it has been proved in more recent studies, where their role in connection with tumor growth has been shown. Here we bescribe their role, for further reference, the reader can refer to the GenBank at www.ncbi.nlm.nih.gov/Genbank/.

MGST 1 is a member of MAPEG (Membrane Associated Proteins in Eicosanoid and Glutathione metabolism) family. It consists of six human proteins, two of which are involved in the production of leukotrienes and prostaglandin E, important mediators of inflammation. Other family members, demonstrating glutathione S-transferase and peroxidase activities, are involved in cellular defense against toxic, carcinogenic, and pharmacologically active electrophilic compounds. Expression of the protein produced by CST3 in vascular wall smooth muscle cells is severely reduced in both atherosclerotic and aneurysmal aortic lesions, establishing its role in vascular disease. POU2AF1 is a B-cell-specific transcriptional coactivator and was observed to be differentially expressed in the cells of patients with chronic lymphocytic leukaemia. VPREB (VpreB) is a 126 aa-long polypeptide with apparent MW of 16-18 kDa. It is expressed selectively at the early stages of B cell development PreBCR transduces signals for: 1) cellular proliferation, differentiation from the proB cell to preB cell stage, 2) allelic exclusion at the Ig heavy chain gene locus, and 3) promotion of Ig light

Table 2. Comparison among different gene selection techniques

	ILDC-ReGEC	Golub	Yeoh	Ross	Cheok	Naoe	Felix	Sjblom		ILDC-ReGEC	Golub	Yeoh	Ross	Cheok	Naoe	Felix	Sjblom
CFD	•	•							**M28170**	•							
MGST1	•								**CD19**	•		•	•				
CST3	•	•							**PYGL**	•							
CD33	•	•							**SMARCA4**	•	•	•					
TCF3	•	•	•						**CEBPD**	•		•					
ZYX	•	•	•						**CLU**	•							
CSTA	•								**SERPINB1**	•		•					•
POU2AF1	•		•	•					**MYB**	•	•						
M11722	•								**S100A13**	•		•					
APLP2	•								**SNTB2**	•							
CTSD	•	•							**CTSA**	•							
CCND3	•	•							**FAH**	•	•						
BLK	•		•	•					**SPI1**	•							
CD79B	•			•					**FCER1G**	•							
RAG1	•		•	•					**PLEK**	•							
TCL1A	•		•	•					**CD302**	•							
MPO	•		•		•	•			**MAD1L1**	•							
SPTAN1	•							•	**ANXA1**	•							
VPREB1	•								**LAMP2**	•			•				
MYL6B	•	•							**CYFIP2**	•	•						
HG1612	•								**MLLT11**	•							
RHOG	•								**CD24**	•		•	•				
TOP2B	•	•	•				•		**RNASE2**	•							
CD79A	•		•	•					**ELA2**	•				•			
GRN	•								**STMN1**	•	•		•				
PSMA6	•	•							**C18orf1**	•							
LYN	•	•							**LYZ**	•	•						
AZU1	•	•			•				**VIL2**	•							
CD63	•								**IL18**	•							
CD63	•								**NUP88**	•	•						
SRGN	•	•							**CDRP2**	•							
NPY	•		•						**LRPAP1**	•		•					
ZNF22	•	•							**IL7R**	•	•						

chain gene rearrangements. Thus, preBCR functions as a checkpoint in early B cell development to monitor the production of Ig mu heavy chain through a functional rearrangement of Ig heavy chain gene as well as the potency of Ig mu heavy chain to associate with Ig light chain. RHOG ARHG is a member of the RAS superfamily of genes, which encode GTP-binding proteins that act in the pathway of signal transduction and play a key role in the regulation of cellular functions. The GRN genes produce proteins regulating cell growth. However, different members of the granulin protein family may act as inhibitors, stimulators, or have dual actions on cell growth. Granulin family members are important in normal development, wound healing, and tumorigenesis. The CD63 proteins mediate signal transduction events that play a role in the regulation of cell development, activation, growth and motility. This encoded protein is a cell surface glycoprotein that is known to complex with integrins. It may function as a blood platelet activation marker. Also this gene has been associated with tumor progression. The SPI1 gene encodes an ETS-domain transcription factor that activates gene expression during myeloid and B-lymphoid cell development. The MLLT11 gene variously symbolized ALL1, HRX, or MLL located on 11q23 has been demonstrated to be fused with a number of translocation partners in cases of leukemia. t(1;11)(q21;q23) translocations that fused the MLL gene to a gene on chromosomal band 1q21 in 2 infants with acute myelomonocytic leukemia have been demonstrated. The N-terminal portion of the MLL gene is critical for leukemogenesis in translocations involving band 11q23. This gene encodes 90 amino acids. It was found to be highly expressed in the thymus but not in peripheral lymphoid tissues. In contrast to its restricted distribution in normal hematopoietic tissue, this gene was expressed in all leukemic cell lines tested. MD1L1 may play a role in cell cycle control and tumor suppression, as it has been shown recently. The protein encoded by NUP88 gene belongs to the nucleoporin family and is associated with the oncogenic nucleoporin CAN/Nup214 in a dynamic subcomplex. This protein is also overexpressed in a large number of malignant neoplasms and precancerous dysplasias.

We conclude that all genes selected by ILDC-ReGEC play a central role in biological mechanism related to leukemia. From those genes is now possible to search databases to select a set of treatments.

6 Conclusions and Future Work

In the present work we report the advances in supervised learning techniques that make it possible *i)* to select a subset of patients to characterize two different situations, *ii)* to determine a subset of genes that discriminate two similar but different diseases, *iii)* to use the obtained subset of genes to select treatments.

In the future, it is interesting to investigate how the knowledge that can be acquired by expert in therapeutics can be used in the classification models. Some attempts have already been done by [**?**]. In that case, knowledge is taken into account by the model in terms of constrains on the underlying optimization problem, but no clue is given on how to identify the regions where such knowledge

provides more insight. In future, we will investigate how to determine regions in which further knowledge is required to improve accuracy of classification models.

7 Acknowledgement

This work has been partially funded by Italian National Research Council and Centro Ricerche *Enrico Fermi*.

References

1. P. Bradley, O. Mangasarian, and W. N. Street. The local maximun clustering method and its application in microarray gene expression data analysis. *EURASIP J. Appl. Signal Proc.*, 1(51–61), 2004.
2. MH Cheok. Treatment-specific changes in gene expression discriminate in vivo drug response in human leukemia cells. *Nat. Genet.*, 2003.
3. C. Cifarelli, M. R. Guarracino, O. Seref, S. Cuciniello, and P. M. Pardalos. Incremental classification with generalized eigenvalues. *Journal of Classification*, 2006.
4. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.
5. M. Eisen and et al. Cluster analysis and display of genome-wide expression pattern. *Proc. Natl Acad. Sci. USA*, pages 14683–14688, 1998.
6. CA Felix. Secondary leukemias induced by topoisomerase-targeted drugs. *Biochim Biophys Acta*, 1998.
7. T. Furey and et.al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bulletin of the American Mathematical Society*, 16:906–914, 2003.
8. 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 monitoring. *Science*, 286:531–537, 1999.
9. M. R. Guarracino, C. Cifarelli, O. Seref, and P. M. Pardalos. A classification algorithm based on generalized eigenvalue problems. *Optimization Methods and Software, in print*, 2007.
10. I. Guyon and A. Elisseff. An introduction to variable and feature selection. *JMLR*, (3):1157–1182, 2003.
11. M. Hall. Correlation-based feature selection for machine learning. Master's thesis, Departement of Computer Science, Waikato University,Waikato,NZ., 1999.
12. I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, M. Raffeld, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Kononen, L. Bubendorf, W. Fehrle, S. Pttalunga, S. Gruvberger, N. Loman, O. Johannsson, H. Olsson, B. Wilfond, G. Sauter, O. P. Kallioniemi, A. Borg, and J. Trent. Gene-expression profiles in hereditary breast cancer. *The New England Journal of Medicine*, 344:539–548, 2001.
13. N. Iizuka, M. Oka, H. Yamada Okabe, M. Nishida, Y. Maeda, N. Mori, T. Takao, T. Tamesa, A. Tangoku, H. Tabuchi, K. Hamada, H. Nakayama, H. Ishitsuka, T. Miyamoto, A. Hirabayashi, S. Uchimura, and Hamamoto. Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocelllaur carcinoma after curative resection. *The Lancet*, 361:923–929, 2003.

14. T.E. Klein, J.T. Chang, M.K. Cho, K.L. Easton, R. Fergerson, M. Hewett, Z. Lin, Y. Liu, S. Liu, D.E. Oliver, D.L. Rubin, F. Shafa, J.M. Stuart, and R.B. Altman. Integrating genotype and phenotype information: An overview of the pharmgkb project. *The Pharmacogenomics Journal*, (1):167–170, 2001.
15. R. Kohavi and G. H. John. Wrappers for feature subset selection. *AIJ special issue on relevance*, 1996.
16. T. Kohonen. The self-organizing maps. *Proceedings of the IEEE*, 78(9):1464–1480, September 1990.
17. O. L. Mangasarian and E. W. Wild. Multisurface proximal support vector classification via generalized eigenvalues. Technical Report 04-03, Data Mining Institute, September 2004.
18. T Naoe. Prognostic significance of the null genotype of glutathione s-transferase-t1 in patients with acute myeloid leukemia: Increased early death after chemotherapy. *Leukemia*, 2002.
19. C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Phl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, P. M. Black, A. von Deimling, S. L. Pomeroy, T. R. Golub, and D. N. Louis. Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocelllaur carcinoma after curative resection. *The Lancet*, 63(7):1602–1607, 2003.
20. M. O'Neill and L. Song. Neural network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect. *Optimization Methods and Softwares*, 4:28–41, 2003.
21. ME Ross. Classification of pediatric acute lymphoblastic leukemia by gene expression profiling. *Blood*, 2003.
22. D. Singh, P.G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.
23. T Sjblom. The consensus coding sequences of human breast and colorectal cancer. *Science*, 2006.
24. S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22(3):281 – 285, 1999.
25. L. J. van't Veer, H. Dai, M. J. Van De Vijver, T. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. Van Der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:536–536, 2002.
26. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
27. D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res*, 34(Database issue), January 2006.
28. EJ Yeoh. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 2002.

22. Stekel, D. J., et al.: The Comparison of Gene Expression from Multiple cDNA Libraries. Gen. Res. 10 (2000) 2055-2061
23. Mudado, M. A., Barbosa-Silva, A., Torres, J. A., Paula-Pinto, S., et al.: K-EST: KOG Expression Sampling Tool. (submitted for publication).

Finding Clusters in Tridimensional Gene Expression Datasets

Tiago J. S. Lopes* and Guilherme P. Telles**

Instituto de Ciencias Matematicas e de Computacao, Universidade de Sao Paulo
Sao Carlos–SP, Brazil
tiagojab@yahoo.com.br, gpt@icmc.usp.br

Abstract. With the popularization of commercial microarray chips and public gene expression databases, it is affordable to use a tridimensional organization to verify the behavior of a large set of genes. Such datasets hold genes' expression under different experimental conditions at different moments in time. To extract information from this structure, we developed a new clustering algorithm to find groups of genes with similar expression levels along the conditions and timeslices. Using validity measures and both synthetic and real datasets, we compared our solution to an algorithm named TriCluster, what allows us to conclude that our algorithm is a good alternative to address the problem. More tests on real data are yet to be performed for a real evaluation of the algorithm scalability, sensitivity and performance.

1 Introduction

Assessment of transcription level from thousands of genes has been achieved with DNA microarrays [7]. After the hybridization of DNA chips with labeled probes, we obtain an image with spots, where each spot is a different gene and its color intensity represents the activity of the gene under given conditions. From an image segmentation process a real-valued array is created, where each row is a gene and each column is an experimental condition (referred here as *sample*).

Eisen et al. [2] used a clustering technique to find groups of genes which had similar behavior along the samples. His idea was that if the genes of a group have coherent expression levels, they should participate in the same cellular process. Since then, many clustering algorithms have been developed, but they share the same problem: as the number of samples increases their ability to find groups of genes with similar behavior along all conditions of the dataset decreases [5]. An attempt to solve this problem is known as subspace clustering, which aims at finding groups composed by subsets of rows and columns [8].

Zhao and Zhaki [10] extended the traditional dataset composed by genes × samples to a tridimensional matrix genes × samples × time. The addition of time provides a new way to understand the behavior of groups of genes. In this

* Corresponding author. Author acknowledges financial support of CNPq.
** Author acknowledges financial support of Fapesp.

structure, their algorithm named TriCluster was able to find overlapping clusters in subsets of rows, columns and timeslices. Another approach for clustering tridimensional data was described by Jiang et al. [6].

In this study we developed a new algorithm to find groups of genes in tridimensional gene datasets. It is greedy and differs from those described previously. Our algorithm may be seen as an extension of CLIQUE [1].

2 Algorithm

The algorithm input is a tridimensional matrix $M = G \times S \times T$ where $G = \{g_1, \ldots, g_m\}$ is the set of genes, $S = \{s_1, \ldots, s_n\}$ is the set of samples and $T = \{t_1, \ldots, t_l\}$ is the set of moments in time. Input parameters are δ, min_g, min_s and min_t. Our algorithm is divided in three steps. The first step is **to find dense units**. In this step, the algorithm finds groups of at least min_g genes with similar expression for each attribute. An *attribute* A_{jk} contains the expression levels of all genes in sample j and timeslice k sorted in nondecreasing order. We denote the minimum and maximum values of attribute A_{jk} as l_{jk} and h_{jk} respectively. For each attribute A_{jk} the algorithm calculates its density by $d(A_{jk}) = m/(h_{jk} - l_{jk})$ or $d(A_{jk}) = m$ if $h_{jk} = l_{jk}$. A unit is an interval $U \subseteq A_{jk}$ and its minimum and maximum values are denoted l_U and h_U. The density of unit U is $d(U) = |U|/(h_U - l_U)$ or $d(U) = |U|$ if $h_U = l_U$, and it is *dense* if $d(U) \geq \delta d(A_{jk})$, where $\delta \in \Re$, and $1 < \delta \leq \infty$.

When a dense unit is found, it is expanded until it becomes maximal. Additionally, if it is not the first to be found, we verify if the union to its adjacent dense unit is also dense. At the end of the attribute we verify if the remaining genes could be merged with the nearest dense unit to form a larger dense unit.

The second step is **to combine dense units from attributes** of the same timeslice. For all attributes in timeslice k, the algorithm enumerates all its dense units and creates a matrix with dimensions $m \times n$, where each cell contains the number of the dense unit for which the gene belongs. If the gene doesn't belong to any dense unit in that condition, the cell has value 0. The algorithm compares each line of this table to all others, column-by-column, and if they share at least min_s columns, then the algorithm establishes a two-dimensional cluster. The algorithm tests the existence of a larger cluster where these genes can be contained before creating a new cluster. In this step we have the creation of overlapping clusters.

The last step is **to combine two-dimensional clusters from different timeslices**, creating clusters $C = X \times Y \times Z$, where $X \subseteq G$, $Y \subseteq S$, $Z \subseteq T$. This step consists of selecting each two-dimensional cluster from a timeslice and finding its intersection with every other two-dimensional cluster from different timeslices. To be valid a cluster must have at least min_g genes, min_s samples and exist in at least min_t timeslices. Similarly to the second step, before creating a new cluster we verify the existence of a bigger cluster enclosing the elements of the recent discovery one. A not so tight bound for the algorithm's time complexity is $o(l^4 m^5 n^4 + l^4 m^4 n^5)$.

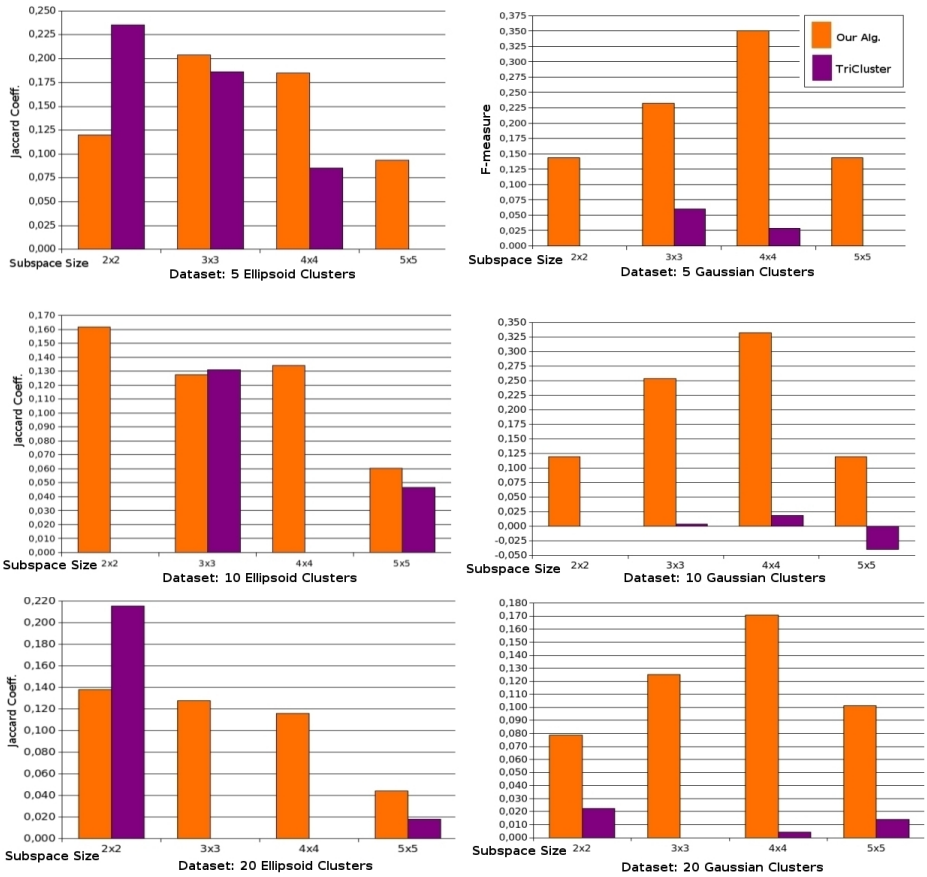


Fig. 1. Comparison of the average Jaccard Coefficient and F-Measure for synthetic datasets. An empty bar means that the algorithm produced no results.

3 Experiments

We used three external (Jaccard coefficient, purity and completeness) and two internal (compactness and connectedness) validity measures to assess the quality of results. Purity and completeness were combined through F-Measure [4]. We created six synthetic datasets with varying size and number of Gaussian and Ellipsoidal clusters. We executed our algorithm and TriCluster (binaries provided by the authors) for these datasets more than 200.000 times with a wide variation in their parameters [3], evaluating F-Measure and Jaccard coefficient. TriCluster has 7 parameters. We found the Pareto Front for F-Measure against Jaccard coefficient, which holds the best solutions of each algorithm. Using different initial subspace sizes we calculated the average of external indexes from the solutions of the front (Figure 1). We can see that our algorithm outperforms

TriCluster on most cases. Both algorithms produce lots of invalid results, which are those that found no clusters or where quality measures tends to zero. We applied the algorithms to a real yeast dataset with 14 timeslices, 13 samples and 7679 genes [9]. Using a similar approach, concerning the internal validity measure, we obtained similar results to synthetic datasets (data not shown).

4 Concluding remarks

We introduced an algorithm to cluster gene expression data in three dimensions that (i) finds groups by density and does not favor any particular cluster form, (ii) is capable of discarding genes from clusters if they have expression levels too high or low or are not coherent along the samples, (iii) performs subspace clustering and (iv) finds overlapping clusters, what is suitable for gene analysis.

The analysis on synthetic and real data allows us to conclude that the algorithm is a good alternative to clustering tridimensional gene expression data. Execution time for our algorithm is usually less than the time spent by TriCluster. TriCluster includes a worst case exponential time step. Other analysis on real data are yet to be performed and properly analyzed.

Future work on the matter may include the application of the algorithm to other biological datasets or to any dataset with a time component. The algorithm is simple and easy to parallelize. We believe that the algorithm is prone to be further extended to more than three dimensions.

References

1. R. Agrawal et al. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. of ACM SIGMOD*, pages 94–105, 1998.
2. M. Eisen et al. Cluster analysis and display of genome-wide expression patterns. *Proceeding of National Academy of Science*, 95(25):14863–14868, 1998.
3. M. Halkidi. On clustering validation techniques. *J. Intell. Inform. Syst*, 17:107–145, 2001.
4. J. Handl, J. Knowles, and D.B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21:3201–3212, 2005.
5. A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31:264–323, 1999.
6. D. Jiang, J. Pei, et al. Mining coherent gene clusters from gene-sample-time microarray data. In *Proc. of the 10th ACM KDD*, Seattle, USA, 1994.
7. S. Lucchini, A. Thompson, and J. C. D. Hinton. Microarrays for microbiologists. *Microbiology*, 147:1403–1414, 2001.
8. S.C. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Trans. on Comp. Bio. and Bioinf.*, 1:24–45, 2004.
9. P.T. Spellman, G. Sherlock, et al. Comprehensive identification of cellcycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Bio. of the Cell*, pages 3273–3297, 1998.
10. L. Zhao and M. Zaki. Tricluster: An effective algorithm for mining coherent clusters in 3D microarray data. In *Proc. of ACM SIGMOD*, pages 694–705, 2005.

Experiencing GARSA as a scientific workflow on grids

Sérgio Manuel Serra da Cruz [1], Fábio José Coutinho da Silva [1],
Alberto M. R. Dávila [3], Maria Luiza Machado Campos [2], Marta Mattoso [1]

[1] Computer Science Department – COPPE - Federal University of Rio de Janeiro, Brazil
[2] Computer Science Department - IM- Federal University of Rio de Janeiro, Brazil
[3] DBBM, Fiocruz- Oswaldo Cruz Institute, Brazil

{serra, fabio, marta}@cos.ufrj.br; davila@fiocruz.br, mluiza@nce.ufrj.br

Abstract. Bioinformatics experiments are typically composed of programs in pipelines manipulating an enormous quantity of data. Managing those experiments brings a set of challenges, such as: how to provide interoperability among tools to achieve better and faster experimental results, in addition to providing data and semantics to experiments. The best way of managing those experiments is through workflow management systems (WfMS). In fact, several such systems are found as products, open-source software and prototypes. Even though BPEL is becoming a de facto standard as a workflow definition language with its execution engine, most WfMS for scientific computing provide its own workflow language. Due to the lack of standards, the e-scientist is faced with the challenge of building its own WfMS or choosing between all available independent systems. In this work we discuss several features found in WfMS for the grid and present some relevant issues for bioinformatics experiments. We present our evaluation on modeling the GARSA genomic workflow and running it on two of those WfMS engines, i.e. Kepler and Taverna. Our contribution is to set technical and practical guidelines to help choosing an adequate WfMS environment for bioinformatics in grids.

Keywords: workflows, grid computing

1 Introduction

Bioinformatics experiments are typically composed of programs in pipelines manipulating an enormous quantity of data. Scientific workflows represent an attractive alternative to describe bioinformatics experiments. These experiments are usually built by manually composing third-party programs with their input and output data in an execution flow. Output data is analyzed and according to the experiment result, parameters are tuned, workflow is re-executed, programs are replaced on the workflow and partial re-executions are made. Ideally, scientists should be able to configure their own bioinformatics workflows by dynamically combining programs provided by different teams, finding alternative programs to choose from, tuning workflow programs, and running partial executions of the workflow. Moreover, once the workflow is defined, the scientist should not be concerned with program changes,

nor manage transformations from data output to data input along program chains. Finally, all these workflow executions should be registered, and made available for queries and for reuse.

Workflow Management Systems (*WfMS*) are automated coordination engines that control workflow specification, instantiation, execution, auditing and evolution. Initially, these WfMS were built to address traditional business workflows. With the popularity of Web services, it became simpler to manage workflows made of Web services due to its inherent interoperability issues. Thus several business process tools became Web service oriented. The idea of using Web services and WfMS also became attractive to model and manage scientific workflows, especially bioinformatics [1]. Several bioinformatics workflows are modeled and executed through Perl scripts which are hard coded and difficult to manage.

In the business scenario, the BPEL language [2,11] emerged as a *de facto* standard to define and execute workflows typically based on Web services. Several companies including IBM, Microsoft, Oracle provide efficient BPEL execution engines, and many open source BPEL engines are also available. However, the language alone with its execution engine is not enough to replace Perl scripts of bioinformatics workflows. A WfMS on top of a workflow language is necessary to add value to a scientific experiment to help composing, tuning, monitoring, steering and reusing bioinformatics workflows. Many of these services are not found in business WfMS. The idea of building all these services on top of BPEL is very attractive to focus on the services and keep the independence of the execution engine.

However, bioinformatics workflows often require high performance computing with grid computing environments requiring BPEL execution engines to be compatible to grid middleware execution engines. These high performance issues together with high volumes of data manipulation led the scientific community to build their own WfMS, each one with its own workflow language and execution engine. Thus, we are faced with a very large number of such systems to manage scientific workflows [3], where some are focused on specific domains such as Bioinformatics. Those systems are not interoperable between them and choosing the right system involves a serious and detailed analysis that includes technical and practical issues.

Scientific WfMS aim at helping users in having flexibility to define, execute and manage their experiments through workflow management tools. Particularly, in bioinformatics, it is important to: (i) design the workflow through a user-friendly interface, taking advantage of components reuse; (ii) execute the workflow in an efficient and yet flexible way through monitoring, steering and user interference; (iii) track provenance of data and services to add semantics to workflow execution; and (iv) access, store and manage data using DBMS, flexible data modeling and ontology support. Complementing those four items, issues such as compatibility with a grid middleware, open source code, perspectives of a large community of users and long term software support are also important points.

To help choosing between a large number of available independent WfMS, we discuss several features found in WfMS for the grid and present some relevant issues for bioinformatics experiments. We show our evaluation on modeling the GARSA genomic workflow and running it on two of those WfMS engines, i.e. Kepler [4,5,6] and Taverna [7]. Those systems were chosen due to their popularity and adherence to

most of the four elicited items. Our contribution is to set some technical and practical guidelines to help choosing an adequate WfMS environment for bioinformatics.

This paper is organized as follows. In section 2 we present concepts on scientific workflows and discuss execution issues on grid environments to define bioinformatics requirements. We provide a brief overview of the GARSA system in section 3, showing why the requirements are relevant to GARSA. Then, we present our experimental results while using the WfMS Kepler and Taverna to support GARSA. Section 5 discusses the pros and cons of using those WfMS engines to construct GARSA's workflow prototype. Finally, we conclude in section 6.

2 Workflow Management Systems in Grids

A workflow denotes a controlled execution of multiple tasks in a given environment. Scientific workflows are usually related with the automation of scientific processes in which scientific programs are structured, based on data and control dependencies. Workflows of scientific applications in Grid environments use multiple computing nodes to accomplish computations that would be time-consuming, or impossible to achieve on a sole node. They need to execute a large number of jobs; need to monitor and control workflow execution including ad-hoc changes; need to execute in an environment where resources are not known a priori.

According to Krauter et al.[8] and Zanikolas and Sakellariou [9], a scientific WfMS in Grids offers advantages, such as: (i) ability to build dynamic applications which can orchestrate distributed resources, (ii) utilization of resources that are placed in a particular site to increase throughput or decrease execution costs, (iii) execution spanning through numerous sites in order to obtain specific processing capabilities; (iv) integration of working groups involved with management of different parts of a given workflow, promoting cross-organizational collaborations. So, scientific WfMS in Grids is a good alternative to face bioinformatics challenges [6,10,11].

Zanikolas and Sakellariou [9], Yu and Buyya [3] and Venugopal, Buyya, Ramamohanarao [12] propose a set of complementary taxonomies that characterize and classify various approaches for building, executing and monitoring workflows on Grids. A taxonomy of grid monitoring systems is proposed in [9] to classify frameworks based on the components of the Global Grid Forum's Grid Monitoring Architecture. In [3] a taxonomy that classifies various approaches of scientific WfMS on Grids is presented with a detailed survey of existing WfMS for Grids. In [12] data grid architectural aspects are discussed, like data transportation, replication, resource allocation, and scheduling.

The categories defined at Yu and Buyya's taxonomy [3], contain a set of requirements which we found very useful to help e-scientists to choose a scientific WfMS for grid computing. This taxonomy is based on four categories: (i) design, (ii) scheduling, (iii) fault tolerance, and (iv) intermediate data movement. The requirements of design's category are: *workflow structure*, *workflow model* and *workflow composition language*. The first refers to workflows patterns [13], indicating the temporal relationship between the tasks. In DAG-based workflows, the structures can be classified as sequence, parallelism and choice, whereas in non-

DAG-based workflows there are all previous patterns and also iteration structures. The second defines the tasks and structure of a given workflow. A workflow model is classified as abstract (a workflow is specified without mapping to Grid resources) and concrete (it binds workflows tasks to specific resources). The latter leads users' ability to express workflows using a given composition language, such as BPEL.

The main requirements of scheduling category are *architecture* and *decision making*. The way tasks are scheduled on grid is very important for scalability, autonomy, quality and system's performance. Fault tolerance is associated with workflow execution failures. Workflow's failures are divided into two levels: task-level and workflow-level. The former can mask the effects of the execution failure of a given task in the workflow. The latter manipulates workflow structure in order to cope with erroneous conditions.

The data movement category aims at describing how the WfMS moves large amounts of data. For instance, input files need to be staged at remote site before processing tasks and output files may be required by child tasks processed on other resources. It is important to highlight that centralized and mediated automatic data movement are suitable for bioinformatics workflows, because monitoring and browsing intermediate results is a common task developed by e-scientists.

Considering the issues discussed in this section, we believe that choosing the ideal system involves a hard and detailed analysis that includes technical and practical aspects. At the introduction we listed four important issues for bioinformatics that represent detailed features within two categories of the taxonomy from [8], i.e. the design and data movement. In the next sub-sections we present Kepler and Taverna with respect to the categories defined by Yu and Buyya [8]. In section 4 we analyze Kepler and Taverna according to their support to the four items chosen to evaluate GARSA genomic workflow.

2.1 Kepler

The Kepler workflow system supports scientists in areas such as biology, ecology and astronomy to compose and execute scientific workflows. Such workflows range from low-level workflows of interest of grid e-Scientists to analytical knowledge discovery workflows for traditional scientists [5]. Kepler is an active open source Java cross-project, cross-institution collaboration that runs on top of the Ptolemy II system [4]. Although Ptolemy II was not originally intended for scientific workflows, it provides a reputable platform for building and executing workflows, and supports several models of computation.

Kepler is provided as a user-friendly workflow visual editor and enactor engine; concrete workflows are modeled in MoML (an XML dialect), allowing the specification of processing units (tasks), data transfer/transformation and execution. Kepler can handle local applications, web and grid services as well. It inherits the actor-oriented modeling system from PtolemyII, which enables the modeling and design of scientific workflows in a friendly way, allowing the creation of workflows for distributed applications just by dragging&droping built-in components, known as Actors. Kepler uses an Actor Component Library that encompasses different kind of actors, such as, local applications, database connectors, web services, grid-enable

actors. It also is possible to use domain-specific actors for case studies, such as phylogenetics actors or create composed actors (which behaves like sub-workflows). A workflow execution models can be defined by a *Director*, it imposes the execution orders and communications schemas used by workflow's actors.

According to Buyya's taxonomy, Kepler also supports Non-DAG based workflows, it can produce abstract and concrete workflows based on a XML composition language, MoML (Modeling Markup Language). Kepler´s scheduling strategy is based on a centralized enactment environment and the decision making is user-defined. Kepler also supports fault tolerance at task and workflow levels. Finally, it also supports centralized automatic data movement. Such approach, despite being easier to implement has some drawbacks like the high transmission time when handling huge amounts of data.

2.2 Taverna

MyGrid project aims at developing a grid middleware infrastructure specifically to support exploratory, data-intensive in-silico experiments in e-science [10]. It aims to support the e-scientist developing a toolkit of core components for designing, executing, managing and sharing experiments modeled as workflows. It is highly focused on bioinformatics.

Taverna is the WfMS of myGrid. It is an open source platform-independent system to compose, adapt and run workflows. In Taverna, workflows can execute as remote or local web services. Legacy applications can be incorporated using Soaplab-Gowlab wrapper tools [14]. Taverna as well as Kepler, is a cross-domain application, although its original focus has been on Biology. Thus, Taverna's services can be from other disciplines, such as chemistry, physics and medical informatics are emerging.

Taverna Workbench includes the Freefluo [14] component, which is responsible for running e-scientist's workflow. Freefluo enactment engine supports the Scufl (Simple Conceptual Unified Flow Language) workflow language that enables users to create and edit workflows through user-friendly tool. These workflows can be loaded in Taverna environment and materialized in XScufl (XML Scufl) format. A drawback is that the SCUFL language does not complies to the defacto standard for Web Services Workflow BPEL[2,11].

In Taverna, a workflow is modeled as a network of processors (nodes) connected by data links. Extensibility is provided through processor types. Processor types include: a single web service operation described in WSDL, a nested workflow, and a 'conceptual level' operation provided by an implementation protocol of lower-level operations [14]. In the next section we discuss the instantiation process of GARSA workflow in workflows management systems.

Considering Buyya's taxonomy, Taverna supports DAG-based workflows, it can produce abstract and concrete workflows based on a XML language named Scufl. Taverna's scheduling strategy is based on a centralized enactment environment and the decision making is user-defined, it means that an e-scientist can choose local or global scheduling according to the nature of its workflow application. Taverna also supports fault tolerance at task level, it can handle different kinds of failures and users can specify an alternate or alternates explicitly for any given Scufl processor.

Moreover, standard fault tolerance techniques such as retry and exponential back out of retry times are available. Finally, data movement in Taverna is centralized but automatic.

3 GARSA Workflow

GARSA workflow (Genomic Analysis Resources for Sequence Annotation) is a user-friendly web-based system, which is freely available under GPL license and has been used in the BioWebDB project [15]. It has been conceived to facilitate the tasks of integrating, analyzing and presenting genomic information derived from several bioinformatics tools and genomic databases, in a flexible way. Up to date, the following projects are using GARSA: *Trypanosoma vivax* (GSS and EST), Trypanosoma rangeli (GSS, EST and ORESTES), Phytomonas serpens (GSS), Bothrops jararaca (EST), Piaractus mesopotamicus (EST), Taenia solium (ORESTES), Crassostrea rizhophorae (EST), Piaractus mesopotamicus (EST) and Lutzomyia longipalpis (EST) [15]. GARSA accepts a) chromatograms, b) downloads from GenBank, c) Fasta files stored locally or a combination of all three and aims to analyze genomic data, including several tasks since cleaning of chromatograms up to phylogeny.

The latest stable version of the GARSA workflow is 1.5 that comprises a pipeline composed of 21 selected bioinformatics software packages. Its underlying platform includes Perl, Apache and MySQL, as well as several Linux-based packages which integrates (i) gene finders, (ii) phylogeny software, (iii) multi-project environment and (iv) user-based authenticated access. GARSA's conceptual data model can store inputs and analyzed data. Moreover, GARSA constitutes a multi-project system that uses five databases to manipulate: GO (Gene Ontology), ECC codes, Taxonomy, Vector and contaminant data. GARSA version 1.5 is being used since 2005 and has been successfully exploited by the BioWebDB consortium. In spite of, our development team is working on version 2.0, which should be launched soon. The new version contains various improvements and extensions, such as: gene finders for eukaryotes; orthologs identification; comparison of library results in intra/inter projects; more phylogeny analysis, including new programs (weighborg, model generator and phyml); and analysis of distant homologs through the HMMER and PsiBlast programs. Although next GARSA version brings many benefits, the GARSA workflow system does not cope with Web services technology nor attends some requirements of scientific workflows [15]. Currently, GARSA lacks flexibility in workflow design, there is not abstract/concrete definition and the composition language used is Perl. The scheduling strategy is based on a centralized view imposed by itself environment and the decision making is user-defined. GARSA offers execution monitoring but it is poor in data provenance support.

GARSA's main advantages lie in ontology support and data management tools. When choosing a generic WfMS to support GARSA it must keep its access to public databases as well as the project's own data modeling including workflow semantics. To help interoperability between different experiments data some conceptual and logical bioinformatics data modeling have been proposed, i.e. CHADO and the

Genomic Unified Schema (GUS) [16]. They present generic classes and relations so that workflow data can be shared, queried and exchanged. In GARSA, GUS schema has been chosen to represent all its workflow related data. Therefore, one of the most important requirements of a WfMS to support GARSA is its flexibility in data representation. It should be able to store and query our bioinformatics workflow data through the GUS schema.

4 Experiencing GARSA on Workflow Management Systems

In the next subsections we present our experiments and describe some implementation details. To evaluate a simpler GARSA workflow, we implemented in sequence: *Phred*, *Cap3* and *Blast*, having these three programs wrapped as Web services. Our evaluation with Kepler and Taverna addresses the four items elicited for bioinformatics workflows, especially GARSA, i.e.: (i) Workflow design; (ii) Workflow execution; (iii) Data provenance and semantic support; and (iv) Data management.

4.1 Experiencing GARSA on the Kepler System

We have installed Kepler both on Windows/Linux platforms and found it very easy to use. However, to run GARSA workflow on a grid-services environment we have to address some Grids complexities like: management of credentials, interaction with schedulers, and particularly, installation and deployment of scientific software. To help the experiment, we wrapped those bioinformatics legacy programs as Web services to expose them to Kepler. Despite the built-in support for SOAPlab on Kepler, the version 1, has some limitations, e.g., SOAPLab uses CORBA on the server side for finding, starting, controlling, and using applications. It also does not have a web-service-based notification system that can accept CORBA events and propagate them to clients.

On representing GARSA as a scientific workflow on Kepler (Figure 1) we could explore the cycle of an e-Scientist interaction with the environment and evaluate its support to GARSA according to our four items.

Workflow design. The design of an abstract workflow on Kepler is quite simple; e-Scientist establishes the workflow's model of computation by dragging director and then drags a set of actors which binds to the wrapped Web services through their WSDL files (Figure 1A). After that, the scientist should link actors through data/control ports. That portion of the task was quite hard to be accomplished, because all the data transformations were manually implemented unspecific Kepler's actors. Up to now, Kepler presents solely few generic transformations actors. So, much of the work spent was on parsing messages through Kepler XML parsing actors.

Workflow execution. The execution of GARSA workflow was simple, e-Scientist can use a control panel to steer a workflow instance (Figure 1A and 1B) while running a concrete workflow, whose tasks or sub-workflows are scheduled by a sole central scheduler. Kepler provides a visual programming interface that enables

composition, execution and harvesting distributed processes. Kepler's version (beta1) used during the tests did not present all requirements related to fault tolerance requirements. For instance, if an e-scientist needs to investigate a workflow running, maintenance actors should be inserted inside the flow, such approach seems to be a disadvantage for two reasons, first, it deviates his attention to an administrative question, second, it makes the workflow harder to understand and maintain.

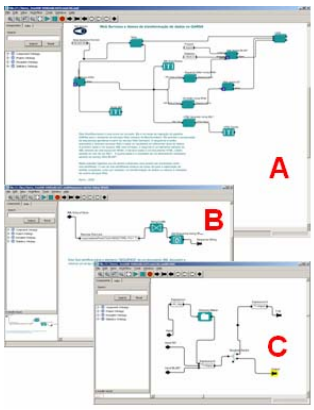


Fig. 1. GARSA on Kepler

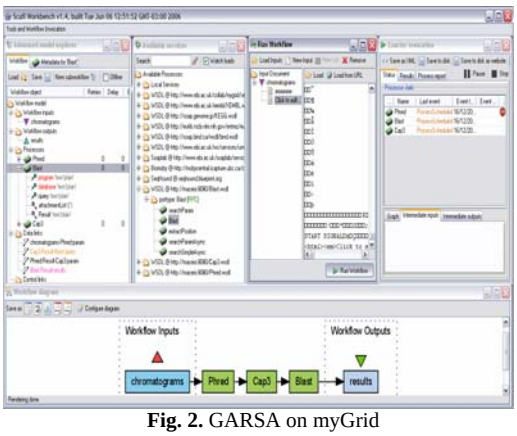


Fig. 2. GARSA on myGrid

Provenance and semantic support. Up to now, Kepler's semantic support through provenance is quite poor. According to Altintas et al. [17] and Bowers et al. [18] it is possible to extend Kepler to register provenance, but such frameworks are not yet freely available. Kepler does not cope with intermediate data movement requirements, it transfers data automatically but offers limited possibilities to annotate information about data provenance (intermediate and end results including files and database references), process provenance (data about workflow definition with data and parameters used in the run); execution provenance, (error and execution logs) and design provenance (information and decision took during workflow design phase). To address the lack of automatic provenance recording mechanisms, we developed a sub-workflow (composite actor) to capture and register such data on log files (figure 1C).

Data management. Kepler provides flexibility in data modeling. E-scientists can easily access several DBMS through Data Access actors. The same happens in using Grid FTP. It can manage and transfer huge amounts of distributed data automatically without user intervention. It has few facilities to manage bio-ontologies and cannot deal with LSIDs as universal identifiers. However, actors can be used to store semantic data with the user's chosen schema.

4.2 Experiencing GARSA on myGrid-Taverna

Taverna is an open source system developed in Java and available freely under the terms of the LGPL. In this work we use Taverna version 1.4 running on Windows XP. The installation process occurred easily and without problems. Before creating GARSA workflow in Taverna it was helpful to wrap the bioinformatics programs

(phred, cap3 and blast) in Web services. We used SoapLab [17], a SOAP-based analysis Web service to wrap legacy command-line applications into Web services.

Workflow design. The first step for the construction of a workflow in the Taverna can be done through service discovery that comprises the pipeline. In GARSA case, three services were wrapped by Soaplab, one for each program present in workflow. These services are added to Taverna environment through of inputting service's WSDL, creating a new WSDL Scavenger. From now on, all the operations described in WSDL (processors) can be added to the workflow model. That process is done for each web service presents in workflow. In this point, the relevant processors are added to model conforms presented in figure 2 (Advanced model explorer window). Workflow input and output created in our model are labeled as chromatograms and results, respectively. Finally, input and output ports are connected, chaining the workflow as depicted in figure 2 (Workflow diagram window). Another way of obtaining processors is through keyword search over more than 3000 services available to the Taverna Workbench. Taverna services can be local Java services, standard WSDL Web Services, or specialist processors created for use in myGrid (as with BioMart and BioMoby for example). BioMoby supplies a subset of bioinformatics services and pre-formatted objects ready for use.

Workflow execution. Run Workflow window (see figure 2) enables to load the GARSA workflow input. Then, a chromatograms file is loaded in run workflow. During execution, the user can steer the workflow execution through the pause/resume facility. Another way to pause the workflow execution is putting breakpoints in workflow steps. The e-Scientist has the possibility of observing the events returned for each processed entity. These events demonstrate the status of each workflow object while it is processing and also correspond to state transitions of the service component. Several messages are emitted such as *ProcessScheduled*, *ProcessComplete*, *serviceFailure*, etc. For each executed object is registered some information such as *name*, *last event*, *event timestamp*, *event detail* and *breakpoint*. It is also exhibited a graph representing the execution flow and the intermediate inputs/outputs.

Provenance and semantic support. Taverna offers the possibility to investigate the results obtained after the workflow execution. However, this relevant functionality is still very limited in this current version of Taverna. In Taverna Workbench there is the Provenance Browser window where the scientist can retrieve a log of the workflow execution and obtain a workflow version history. For each workflow execution is returned the workflow id, execution date and author name. Also it is possible to open the old versions for re-executing them. Semantic support is a positive issue provided by Taverna Workbench, e-scientists can add descriptive information to the workflow inputs through 'Metadata Editor' window. Moreover, Taverna offers support to ontologies and is fully integrated with Gene Ontology (GO), this allows the annotation of data, workflows and services and allows their classification by e-scientists. rapidly have their experiments related with GO. The new directions point to using new semantic patterns, such as: RDF, LSIDs.

Data management. Taverna with myGrid has very important and advanced tools for data management. It supports a suite of ontologies for each data category found in a bioinformatics experiment. DBMS access is also available. It works with LSID as a universal identifier, which helps data export. However, Taverna has its own data

representation through its specific schema definition. All data managed by the workflow is stored according to myGrid representation tables that can be queried and provide data provenance. All this rich semantic support prevents the use of any other logical schema different from myGrid's. This limitation allows experiment data exchange only among myGrid managed workflows.

5 Comparisons

Our research group is involved in the BiowebDB Consortium, which aims at supporting genomic workflows to provide interoperability among different analyses tools and more sensitive algorithms for distant homology detection. As far as we are concerned Kepler is more suitable than Taverna to accommodate GARSA and other BiowebDB workflows. We found Kepler more flexible, stable and grid-enable than Taverna although both present some limitations when compared with the user requirements defined on Section 2. The reasons that guided us to choose Kepler will be discussed as follows.

Kepler is not a bioinformatic-specific tool like Taverna, its interface is simple to use and present a feature that enables the steering of GARSA workflow execution. The reuse of single actors on Kepler is feasible, but composed actors (sub-workflows) are harder to reuse, besides up to now, concrete workflows are stored as single personal experiment, it can not be shared or updated in a collaborative way on a given repository.

Unlike Kepler, Taverna was designed with the purpose to support bioinformatics projects it support the use of LSIDs, annotations and present a quick access to Gene Ontology. It employs a hybrid architecture which includes Web services amongst other components, like Java, BioMoby and so on. Despite those facilities and myGrid initial motivation to grid computing, it is quite hard to use Taverna on a distributed grid environment. In addition, all this rich its semantic support is coupled to a specific data representation schema, which makes it almost impossible to use others data models like GUS schema.

Taverna is a plug-in based platform and presents user-friendly interfaces for designing and executing phases. However, design process has a poor diagrammatic tool to build workflow and chain inputs and outputs. The positive aspect in design process is the ease of finding services as the user can do a keyword search that covers bioinformatics service databases in Biomart and Biomoby platforms. During execution process the e-Scientist is limited to a low-level interaction with its workflow. Taverna supplies steering facilities, but putting away relevant capabilities, such as the possibility of user interfering in execution for redefining parameters, redesign part of the workflow model.

Despite Kepler's poor typed data, there is wide support to complex data transformations, like the ones required by GARSA's Web services. Up to now, we noticed that there is an increasing support to grid services and the incorporation of semantic types that can provide a link between the structural type and concept expressions from user's ontology. Considering workflow scheduling there is no built-in support to performance prediction, execution monitoring, annotations and built in

logging strategies on Kepler. However, considering that some of these workflow tasks will run on the grid, the workflow will be scheduled by grid tools which have efficient schedulers. Despite being a time-consuming and error prone features, it should also be drawn by the e-Scientist on the abstract workflow. Kepler offers support and facilities to connect and query many kind of standard relational database, its even possible to connect to functional genomics databases, such as GUS[16].

6 Conclusion and Future directions

As the usability and computation capacity of the Grid increases, there will be growing demand to integrate scientific applications and databases through scientific workflow. There are a number of important aspects to facilitate the development of the grid workflows using service oriented architecture, such as: 1) applications may be offered as a service; 2) services may be registered on a common repository or be discovered using search engines; 3) Web service based workflow composition tools may orchestrate different kinds of service as long as the services can exchange messages effectively. Integrating scientific workflow based on distributed grid services environments promises to be a chief advantage over local-based alternatives.

Our projects are driven by open source tools and standard proposals for Web services workflows, such as Java, OPAL, Tomcat-Axis, GT4 and GUS. Choosing an adequate WfMS can be a difficult decision. We addressed important WfMS support to assist potential e-scientists to estimate and plan the use of workflow engines within their bioinformatics projects through WfMS for Grids. In this paper, we shared our experience and insights gained reviewing common bioinformatics workflows requirements, using and evaluating Kepler and myGrid on distributed environments. In addition, a genuine workflow example, the GARSA workflow, was used to validate our evaluation. Real world GARSA users have different requirements for filtering, storing and retrieving data flowing through this workflow, we could migrate it the to scientific workflow engines on grid environments.

References

1. Stein, L. Creating a Bioinformatics Nation. Nature 417 (2002) 119-120.
2. Dumas, M., ter Hofstede, A.H.M. Russell, N., Verbeek H.M.W. and P. Wohed “Life After BPEL?” BPM Center Report BPM-05-23, BPMcenter.org, (2005).
3. Yu, J. Buyya, R., A taxonomy of scientific workflow systems for grid computing. ACM SIGMOD. Vol. 34 ,(2005) 44 - 49 .
4. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludäscher, B., Mock, S., “Kepler: An Extensible System for Design and Execution of Scientific Workflows”, 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM'04), June 2004.
5. Altintas, I., Birnbaum, A., Baldridge, K., Sudholt, W., Miller, M., Amoreira, C., Potier, Y. B. Ludaescher, “A Framework for the Design and Reuse of Grid Workflows”, Intl. Workshop on Scientific Applications on Grid Computing (SAG'04), LNCS 3458, Springer, (2005).. 120-133.

6. Ludäscher, B., Altintas, et al. “Scientific workflow management and the Kepler system” Concurrency and Computation: Practice and Experience. John Wiley & Sons, Ltd. v. 18, Issue 10, (2006). 1039-1065.
7. Taverna Workbench. Available from http://taverna.sourceforge.net/index.php
8. Krauter, K., Buyya, R. and Maheswaran M.. “A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing” . Software: Practice and Experience, John Wiley & Sons, Inc, NJ, USA, February 2002. pp. 32(2):135-164.
9. Zanikolas, S. Sakellariou R. “A taxonomy of grid monitoring systems” Future Generation Computer Systems 21 (2005) pp. 163–188.
10. myGrid: personalised bioinformatics on the information grid. V.19(1) 2003, pp. i302–i304.
11. Akram, A., Meredith, D., Allan, R. “Evaluation of BPEL to Scientific Workflows”. Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on. May 2006. Vol. 1, pp. 269- 274
12. Venugopal, S. Buyya, R. and Ramamohanarao. K. “A Taxonomy of Data Grids for Distributed Data Sharing, Management, and Processing”. ACM Computing Surveys, Vol. 38, March 2006.
13. W.M.P. van der Aalst. “Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management”. Lecture Notes in Computer Science, vol Springer-Verlag, Berlin, (2004). 3098 pp. 1-65.
14. Oinn, T., et al. “Taverna: Lessons in creating a workflow environment for the life sciences”. Concurrency and Computation Practice and Experience Grid Workflow Special Issue, 2005.
15. Dávila A. M. R. et al. “GARSA: genomic analysis resources for sequence annotation” Bioinformatics Volume 21 , Issue 23, 2005, pp. 4302-4303.
16. GUS – “The Genomics Unified Schema”. Available at: http://www.gusdb.org/about.php
17. Altintas, I. Barney, O. Jaeger-Frank E., “Provenance Collection Support in Kepler Scientific Workflow System”. Int. Provenance and Annotation Workshop 2006, pp. 1-15.
18. Bowers, S., Ludaescher, B., et al. “A Model for User-Oriented Data Provenance in Pipelined Scientific Workflows”. Int. Provenance and Annotation Workshop, 2006.

IGRAFU, a User-Friendly Tool based on Clusters of PCs for Reconstructing Phylogenetic Trees

Martha Torres, Cristianno Vieira, Glauber Gonçalves and Zilton Junior

Departamento de Ciências Exatas e Tecnológicas da Universidade Estadual de Santa Cruz, Ilhéus, Bahia, Brazil

Abstract. This paper presents IGRAFU, a user-friendly tool for phylogenetic trees reconstruction. This tool includes DiGrafu, a solution based on the distance method that integrates FastME, Weihgbor, BIONJ and NJ programs. IGRAFU also includes the Phyml program and a parallel version for Phyml bootstrap. This paper presents a performance evaluation and validation of proposed solutions.

1 INTRODUCTION

There are many phylogenetic tree reconstruction programs [1] [2] [3] [4] [5]. It is therefore difficult for the user to choose which program is the best to cover his (or her) needs. In order to alleviate this problem, many researchers have made studies to evaluate the developed programs [6] [7] [8] [9].

This paper presents IGRAFU as a user-friendly tool that is intended to help the user in choosing the most appropriate programs. Besides, molecular phylogenetic analysis requires a large amount of computation, then it is important to develop high performance solutions such as Fastdnaml [10] (an example of parallel solution). Also, IGRAFU includes a parallel implementation.

The remainder of this paper is organized as follows: In section 2.1 we describe IGRAFU. In section 2.2 we explain DiGrafu, which is based on the distance method. In section 2.3 we report the inclusion of Phyml [6], which is based on the maximum likelihood method, in IGRAFU. In section 2.4 we describe a parallel version of Phyml and its performance evaluation. Finally, in section 3 we present our conclusions and a description of the future work.

2 IGRAFU

IGRAFU is a user-friendly tool. It includes DiGrafu, which is a phylogenetic tree reconstruction program based on the distance method. It also includes Phyml, a phylogenetic tree reconstruction program based on the maximum likelihood method and a parallel version of Phyml boostrap. IGRAFU recognizes the following input formats: nexus, phylip or nafta. It includes the Hypertree [11] program for tree visualization and a text editor for visualization of the output data.

IGRAFU is a tool developed to run on PCs clusters. It can run in sequential mode in any one of the cluster nodes or in parallel mode using any number of cluster nodes. It has been developed in Java (version 1.5.0_01) and presently runs on linux platform.

IGRAFU is a self-contained graphic interface package, i. e., the installation package has all the programs and necessary modules for its functioning. The Figure 1 presents two screens of IGRAFU. The first screen shows the user option

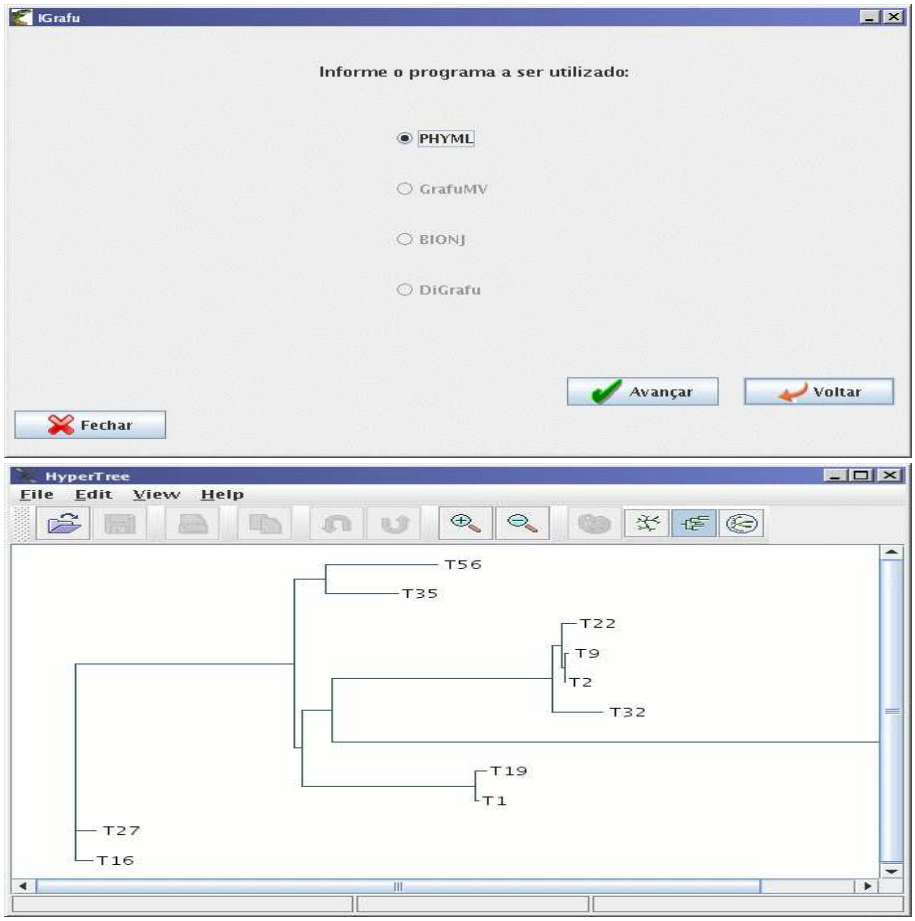


Fig. 1. Screens of IGRAFU

for phylogenetic tree reconstruction methods and the second screen illustrates the tree visualization through the Hypertree program.

2.1 DiGrafu

There are many programs based on the distance method. We have studied and selected five main programs: NJ [28], BIONJ [1], UPGMA, Weighbor [12] and FastME [29]. Then, we developed a tool that allows exploring the best characteristics of these programs. Each program uses as input a distance matrix which is calculated through the Dnadist 3.6 [13] program. This program is also integrated in DiGrafu.

Through Digrafu, the user does not worry about choosing the best suited program as our program selects the best option. Digrafu´s choice is based on the input data and an additional option provided by the user to set the priority on execution time, accuracy or a mix of both.

We used synthetic data sets in order to evaluate the distance-based methods under consideration. In a simulated environment, the true tree is artificially generated and forms the basis for comparison among the phylogenetic methods. We used the 5.000 trees generated by Guindom and Gascuel in [6], each tree having 40 *taxa*. These phylogenies have a broad variety of deviations from the molecular clock and various evolutionary rates. The mean branch lenght is equal to 0.06 substitutions/site and the average ratio of the lengths of the longest and the shortest lineages is equal to 3.4. According to [6], these values come from an analysis of substitution rates in various organisms and of numerous published phylogenies. Then, we generated sequences of 500 base pairs (bp) in length from these phylogenies using Seq-Gen [14]. This program generates sequences data sets based on a model tree and an evolutionary model and it is commonly used for generating synthetic data sets [6] [27] [8]. We generated sequences under the Kimura 2-parameter (K2P) model [15], with a transition/transversion ratio of 2.0 and the Juckes Cantor (JC69) model.

These sequences are passed to the phylogenetic reconstruction method, which infers a tree based on the given sequences. The inferred trees are then compared against the model tree for topological accuracy. We use the Robinson-Foulds distance [19] to measure the discrepancies between trees. This distance corresponds to the number of internal branches that are found in one tree and not in the other one. The Robinson-Foulds distance is calculated using the TreeDist program from Phylip [13]. This value is not normalized, if it is 0.0 then both topologies are identical, meaning that when the distances are smaller, the trees are more similar.

We plotted the Robinson-Foulds distance against the maximum pairwise divergence (MD) in the sintetic data sets. The (uncorrected) divergence between two sequences is the proportion of sites where both sequences differ. The figure 2 shows the results for K2P and JC69 models.

The FastME solution has four versions: the GME+FASTNNI solution is the combination of greedy ordinary least-squares minimum evolution tree construction algorithm (GME) and the FASTNNI tree swapping algorithm. The GME+BNNI solution is the combination of GME with the BNNI tree swapping algorithm based in the balanced minimum evolution framework. The BME + FASTNNI solution is the combination of balanced minimum evolution tree

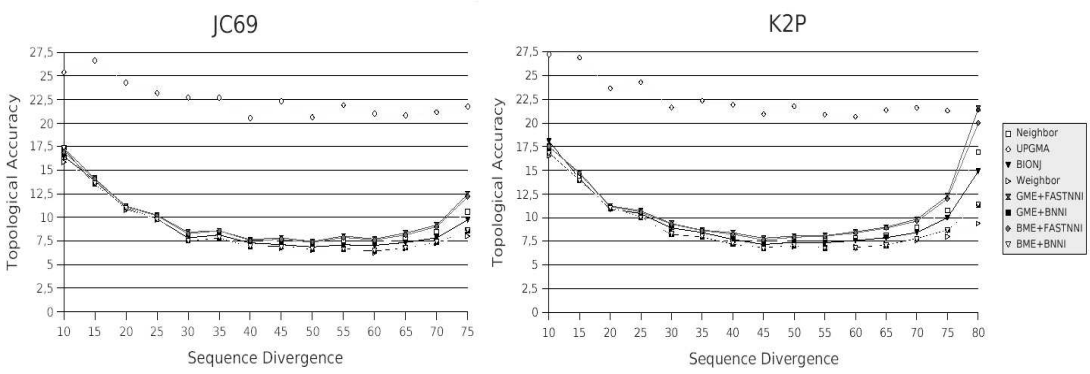


Fig. 2. Topological accuracy of Neighbor, UPGMA, BIONJ, Weighbor, GME+FASTNNI, GME+BNNI, BME+FASTNNI and BME+BNNI as a function of the divergence between sequences.

construction algorithm (BME) and the FASTNNI algorithm. The BME+BNNI solution is the combination of BME with BNNI.

The results are in accordance with expectations and with previously published simulations [6] [7] [8]. When the MD is low, phylogeny reconstruction is difficult because there is not enough information in the data to estimate the short internal branches. With a high MD, saturation corrupts the phylogenetic signal and reconstruction is again problematic. This explains why all methods perform better with medium divergences rates. Figure 2 indicates that UPGMA presented the worse values (higher values). This method is old and it is not good for sequences data sets. It can still be inferred from these figures that the Weighbor, GME+BNNI and BME+BNNI methods had been most accurate.

Also, we plotted the execution time against the sequence divergence. This execution time was measured using the times library of PERL and it includes the execution time of Dnadist program. Figure 3 shows these results. It indicates that all the methods had a similar behavior, except Weighbor, which increases the computational requirement in considerable way for using likelihood calculations to improve the exactness. BIONJ, was always the fastest method, except in the point of MD 80% of K2P model, where the FastME versions presented better performances.

Thus, DiGrafu is implemented taking into consideration the results shown above, where UPGMA is not taken in account by its poor performance. DiGrafu chooses the best method in each value of the sequence divergence and for each model. When the user selects the execution time as priority, DiGrafu executes BIONJ, which is the most efficient in almost all cases. When the user selects the execution time and accuracy, DiGrafu executes the method that provides to greater exactness, except Weighbor (its execution time is very high compared to the other methods). In these cases the FastME method (in the versions BME+BNNI and GME+BNNI) was the most used. When the user selects accu-

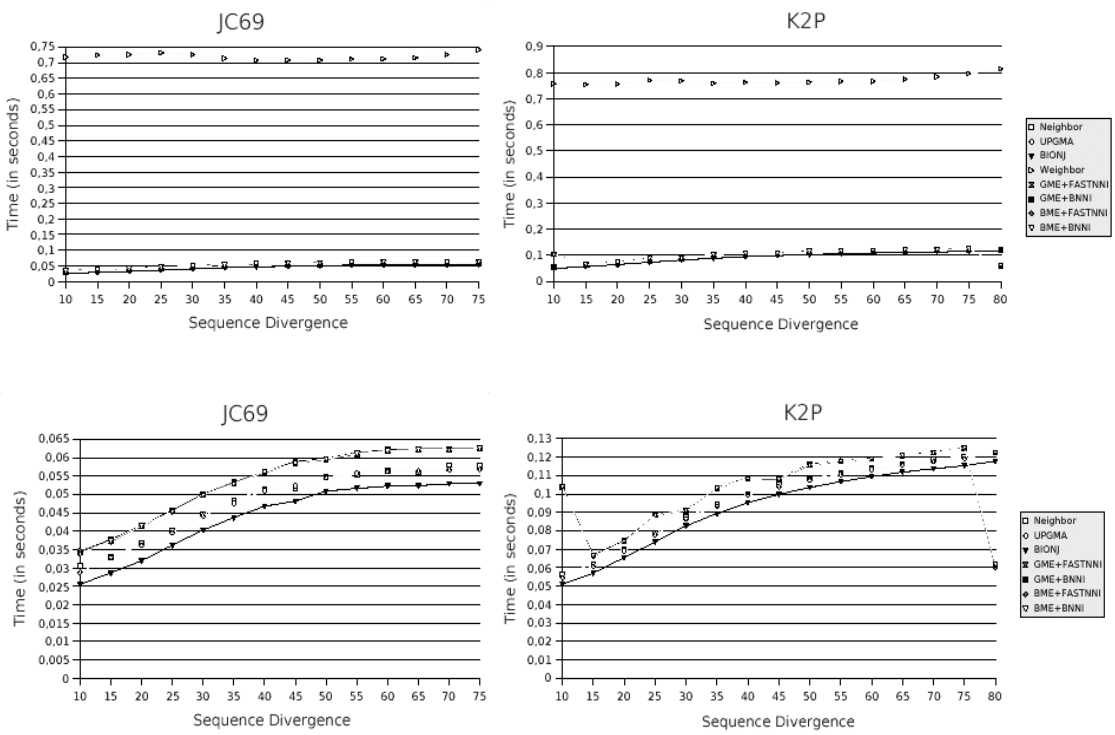


Fig. 3. Execution time of Neighbor, UPGMA, BIONJ, Weighbor, GME+FASTNNI, GME+BNNI, BME+FASTNNI and BME+BNNI as a function of the divergence between sequences.

racy, DiGrafu chooses the method with higher exactness. Weighbor is then the most used method and FastME, in the versions BME+BNNI and GME+BNNI, is chosen a few times.

In order to validate DiGrafu, it was submitted to the same previous performance tests, whose results are shown in figure 4. The results are in accordance with expectations. DiGrafu[ta] (execution time and accuracy) is always more accurate than or equal to DiGrafu[t] (execution time). DiGrafu[a] (accuracy) was the most accurate of the three. Figure 5 shows the execution time. This figure shows that DiGrafu[t] was the fastest and DiGrafu[a] was most demanding in processing time.

Digrafu in IGRAFU Figure 6 shows two screens of Digrafu in IGRAFU. At the first screen, the user must specify the input file (nexus, phylip or nafta format) and the output file. The user also chooses the execution time (execução in Portuguese) and/or accuracy (exatidão in Portuguese), and he must define if

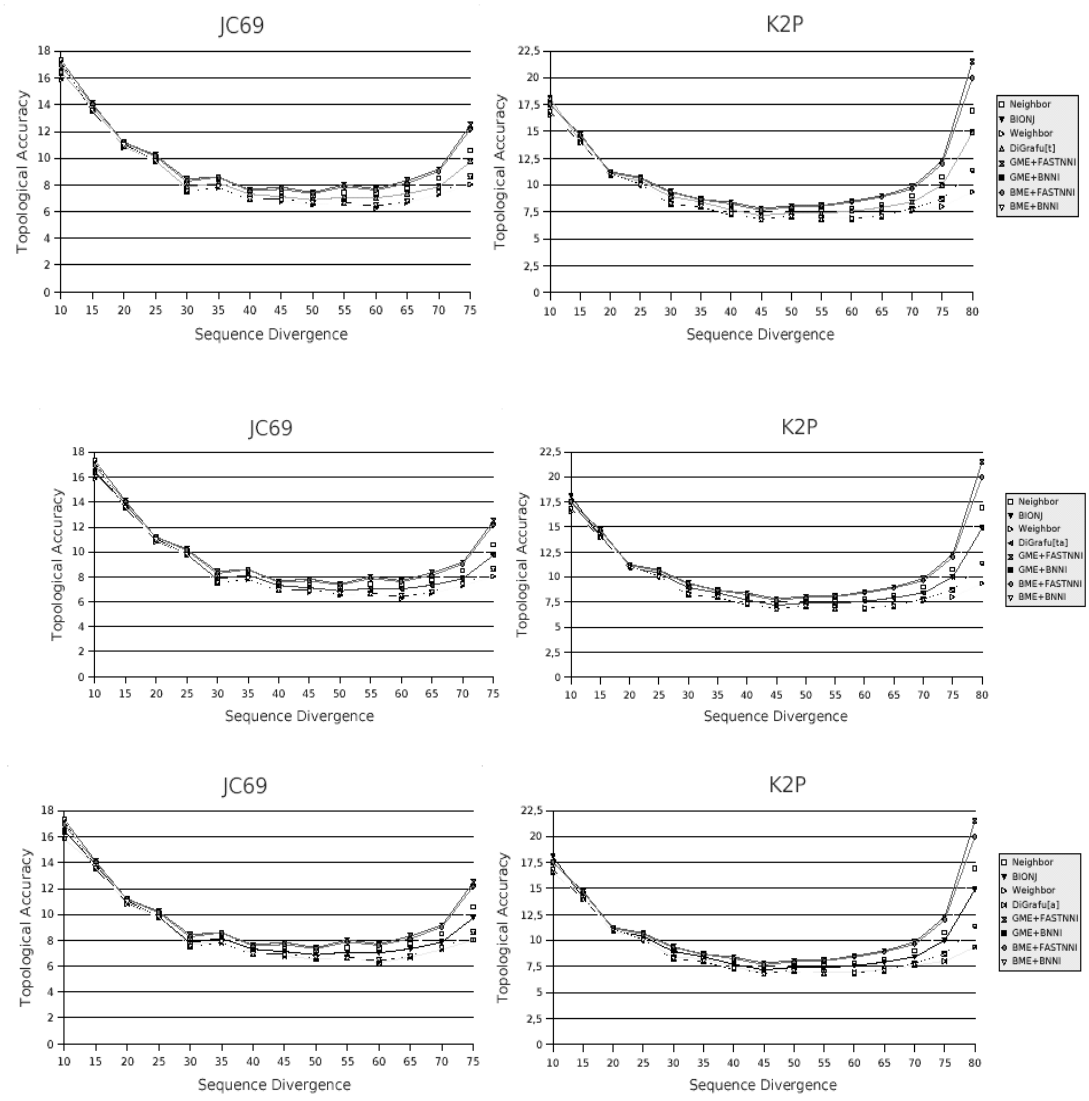


Fig. 4. Topological accuracy of DiGrafu[t](execution time), DiGrafu[ta] (execution time and accuracy) and DiGrafu[a] (accuracy) respectively for JC69 and K2P models as a function of the divergence between sequences.

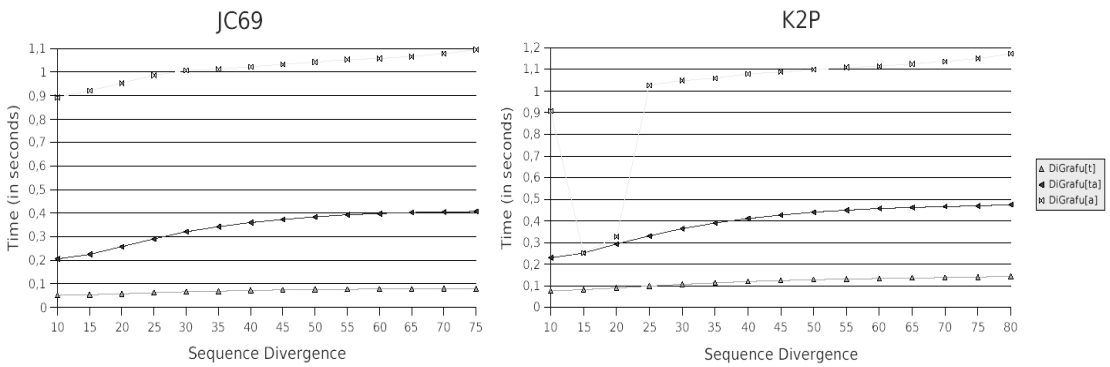


Fig. 5. Execution time of DiGrafu[t] (execution time), DiGrafu[a] (accuracy) and DiGrafu[ta] (execution time and accuracy) for JC69 and K2P models as a function of the divergence between sequences.

the data correspond to DNA or protein. At the second screen, the user defines the evolution model based on the parameters of Dnadist program.

2.2 Phyml

There are many maximum-likelihood based programs for phylogenetic reconstruction [18] [10] [2]. We selected Phyml because it is of public domain and mainly by its efficiency and accuracy [6] [17]. The current version implements several models of nucleotide sequence evolution: JC69, F81 [16], HKY [20] , TN93 [21] and GTR. The Dayhoff [22] and JTT [23] models for proteins are also available. A discrete gamma distribution [3] can be used to account for variable substitution rates among sites. The parameters of these models can be either user defined or fitted to the data by likelihood maximization. Phyml can also be used to refine a user-supplied tree [6].

Phyml program in IGRAFU The Phyml version included in Igrafu is 2.4.5. The figure 7 shows Phyml in Igrafu.

The first screen shows some options of Phyml, such as the input file, whether the data is DNA or Protein, whether the format is interleaved (intercalado in portuguese) or sequential (sequencial in portuguese) and if the bootstrap option is selected. The second screen illustrates the evolution models implemented in Phyml for DNA data. Also, it has the tree ("arvore" in portuguese) option, where it is possible to select the optimization options and the BIONJ program or an user supplied tree.

2.3 Parallel Bootstrap of Phyml

Phyml has two forms for executing bootstrap: internal or external. In the internal form, the bootstrap data is generated by Phyml using non parametric bootstrap

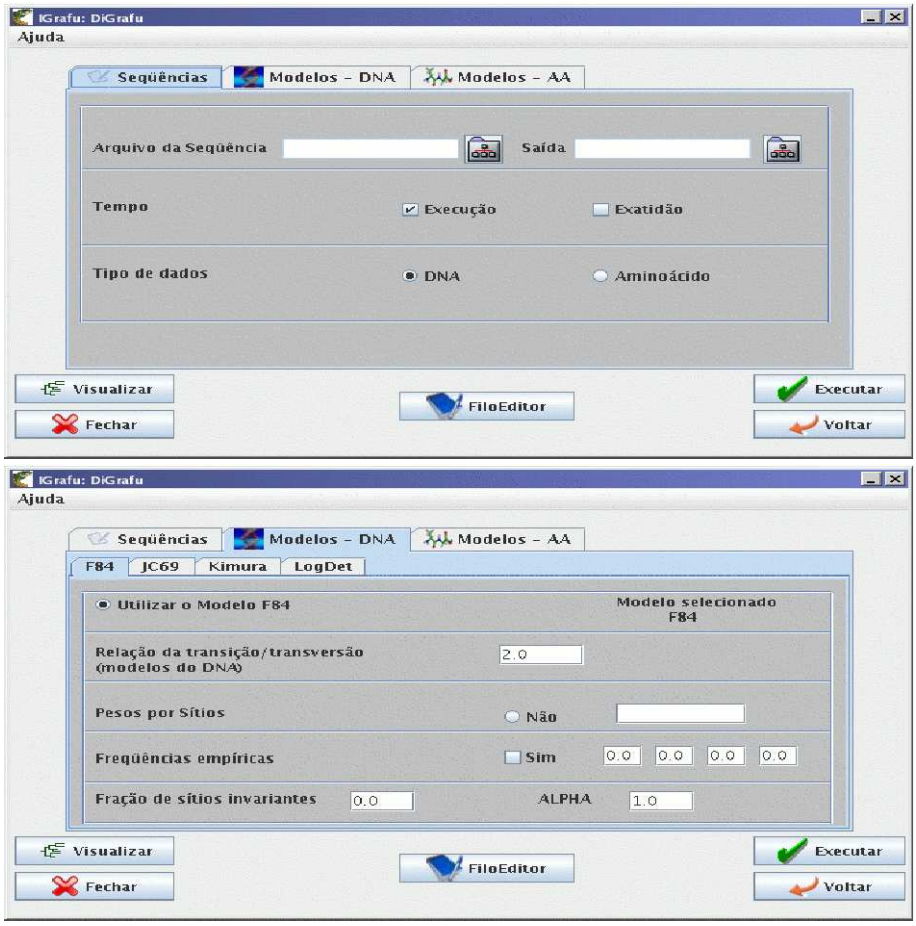


Fig. 6. Screens of DiGrafu in IGRAFU

analysis. It generates a single bootstrap tree in nexus format. In the external form, the bootstrap data can be generated for other program, such as Seqboot (Phylip [13]), for example. Phyml then generates an output file with all trees.

We implemented a parallel version in MPI [24] for internal and external parallel bootstrap in Phyml. Our parallel version for external form consists in dividing the data set for each processor, what is done parallelizing a loop. Each processor generates an output file with the generated trees.

In the parallel version for the internal form, there is a central processor; it generates all data for bootstrap in a matrix. This matrix is divided in parts for each processor using the MPI_Scatterv primitive of the MPI standard, and then a loop is parallelized in order that each processor executes its data sets and generates its statistics. Last, the partial score of each processor is sent to

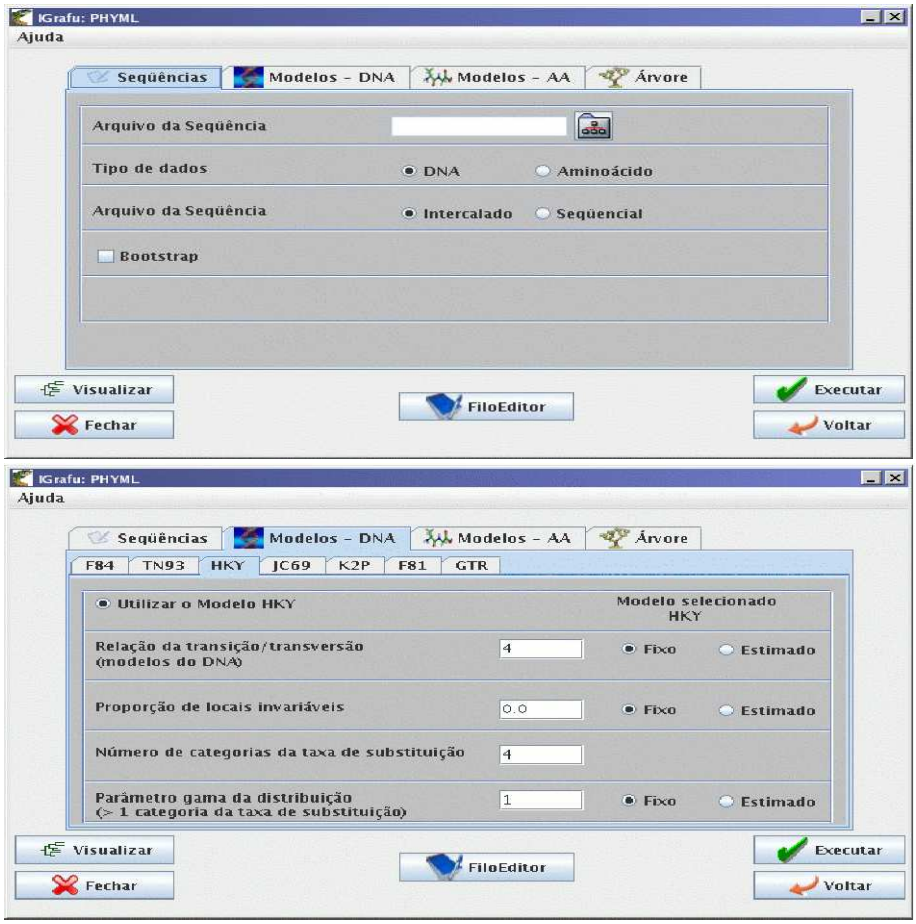


Fig. 7. Screens of Phyml in IGRAFU

the central processor by the MPI_Reduce primitive and the central processor generates an output file with the bootstrap tree.

Visualization in IGRAFU The figure 8 illustrates two screens for the choice of bootstrap. The first screen shows the choice of sequential mode and the choice of internal or external form. Also, it shows the option for which processing node to execute the program. The second screen shows the choice of parallel mode, how many nodes to use and the bootstrap number.

Results analysis We used our parallel cluster composed by five Xeon 3.2 Ghz 64 bits, 2 GB RAM and 160 GB HD computers interconnected by gigabit ethernet switch (http://labbi.uesc.br/cluster).

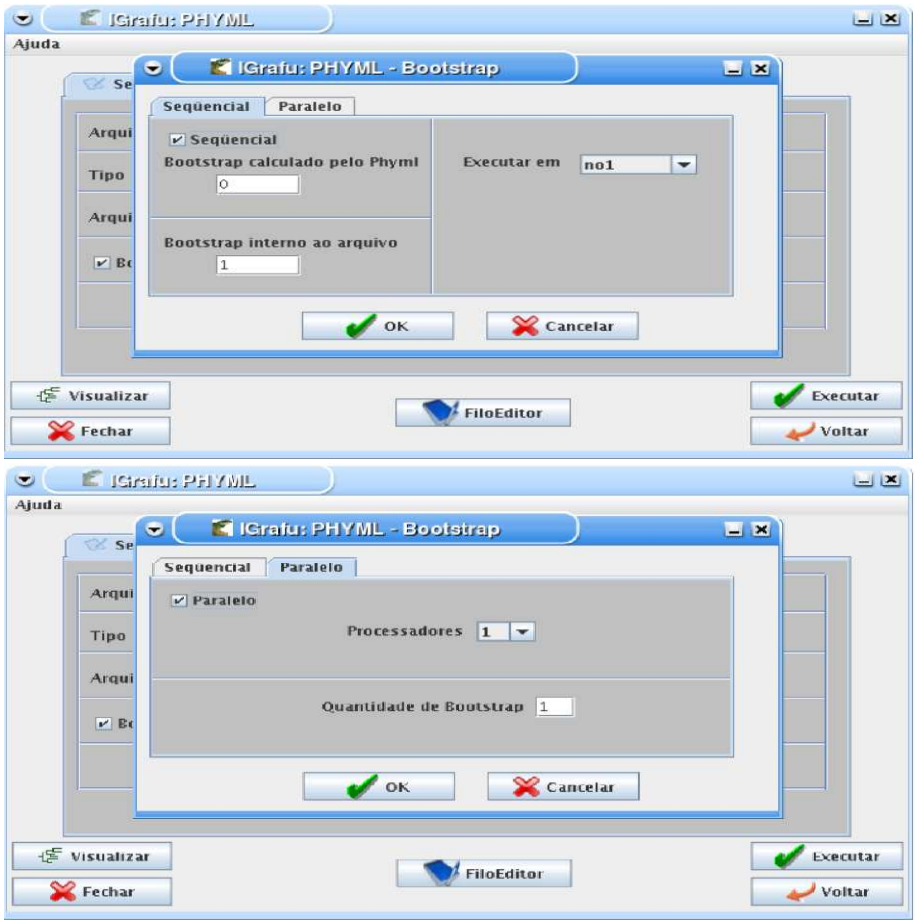


Fig. 8. Screens of the bootstrap choice in sequencial and parallel form

In order to validate the parallel version, we used the data sets of springtails [25] which are composed by 59 species and 589 sites. We ran a bootstrap for 100 data sets on sequential and parallel versions.

The output data for external form is the same on parallel and sequential versions because we used the results generated by Seqboot program for the two versions. The output data for internal form is not the same because the data is generated using random numbers, but it is similar in the two versions. Table 1 shows the execution time and speedup of the parallel version using internal form for springtails data. The execution time is reduced when the number of processors increases as there is communication overhead the speedup is not linear. Table 2 shows the execution time and speedup of the parallel version using external form for springtails. The execution time is again reduced when the number of

Table 1. Performance evaluation of internal parallel bootstrap of Phyml

Processors Number	*Execution Time (seconds)*	*Speedup*
1	623	1
2	419	1.49
3	266	2.34
4	210	2.97
5	177	3.52

Table 2. Performance evaluation of external parallel bootstrap of Phyml

Processors Number	*Execution Time (seconds)*	*Speedup*
1	581	1
2	300	1.64
3	197	2.94
4	160	3.63
5	120	4.84

processors increases, but, as there is no communication overhead, the speedup is quasi-linear.

3 Conclusions

We developed, implemented and tested DiGrafu. Our solution includes four popular distance-based programs; it selects which program to use analyzing input data set and efficiency and/or accuracy criteria. We developed, implemented and tested a parallel version of bootstrap for Phyml program. We also developed IGRAFU. It integrates DiGrafu, Phyml and the parallel version for Phyml. IGRAFU is a user-friendly tool for cluster machines. In future works, we intend to include others phylogeny reconstruction programs and parallel versions. Also, we intend to include a program similar to MODELTEST [26] in order to help the user to choose the evolution model and its parameters.

Acknowledgment We would like to thank FAPESB and UESC by the grants, as well as FAPESB and LABBI for the infrastructure.

References

1. Gascuel, O.: BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. **14** (1997) 685–695
2. Schmidt, H., Petzold,E., Vingron, M.,von Haeseler, A.: TREE-PUZZLE: Maximum likelihood phylogenetic analysis using quartets and parallel computing. Bioinformatics **18** (2002) 502–504
3. Yang, Z.: PAML: a program package for phylogenetic analysis by maximum likelihood. Comput. Appl. Biosci. **13** (1997) 555–556
4. Lemmon, A., Ovitch, M.: The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation. Proc. Natl. Acad. Sci. USA **99** (2002) 10516–10521
5. Gladstein D., Wheeler C.: POY Program and documentation. available at http://reseach.amnh.org/scicomp/projects/poy.php. (2003)
6. Guindon, S., Gascuel, O.: A Simple, Fast, and Accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. 52 **5** (2003) 696–704
7. Kuhner, M., Felsenstein, J.: A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol. 11 **3** (1994) 459–468
8. Ranwez, V., Gascue, O.: Quarted-based phylogenetic inference: Improvements and limits. Mol. Biol. Evol. **18** (2001) 1103–1116
9. Williams, T., Moret, B.: An Investigation of Phylogenetic Likelihood Methods. Proceedings of 3rd IEEE Symposium on Bioinformatics and Bioengineering (2003)
10. Olsen, G., Matsuda, H.,Hagstrom, R.,Overbeek, R.: FastDNAml: A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. Comput. Appl. Biosci. **10** (1994) 41–48
11. Bingham, J., Sudarsanam, S.: Visualizing large hierarchical clusters in hyperbolic space. Bioinformatics. 16 **7**(2000) 660–661
12. Bruno, W., Socci, N., Halpern, A.: Weighted neighbor joining: A likelihood-based approach to distance-based phylogeny reconstruction. Mol. Biol. Evol. **17**(2000) 189–197
13. Felsenstein, J.: PHYLIP (phylogeny inference package). Version 3.6a2. University of Whashington, Seatle. (1993)
14. Rambaut, A., Grassly, N.: Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. **13**(1997) 235–238
15. Kimura, M.: A simple method for estimating evolutionary rates of base substitution through comparative studies of nucleotide sequences. J. Mol. Evol. **16**(1980) 111–120
16. Felsenstein, J.: Evolutionay trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. **17**(1981) 368–376
17. Vinh, L., von Haeseler, A.: IQPNNI: Moving Fast Through Tree Space and Stopping in Time. Mol. Biol. Evol. 21 **8**(2004) 1565–1571
18. Swofford, D.: PAUP*: Phylogenetic analysis using parsimony (*and other methods).Sinauer, Sunderland, Massachusetts (1999)
19. Robinson, D., Foulds, L.: Comparison of phylogenetic Trees. Mathematical Biosciences **53**(1981) 131–147
20. Hasegawa, M., Kishino, H.,Yano, T.: Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol **22**(1985) 160–174

21. Tamura, K., Nei, M.: Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. Mol. Biol. Evol. **10**(1993) 512–526
22. Dayhoff, M.: Atlas of Protein Sequence and Structure. National Biomedical Research Foundation, Washington, D.C. Volume 5, Supplement 3(1978)
23. Jones, D., Taylor, W., Thornton, J.: The rapid generation of mutation data matrices from protein sequences. Comput. Appl. Biosci. **8**(1992) 275–282
24. Walker, D., Dongarra, J.: MPI: A Standard Message Passing Interface. Supercomputing 12 **1**(1996)
25. Stevens, M., Greenslade, P., Hogg, I.,Sunnucks, P.: Southern Hemisphere Springtails: Could Any Have Survived Glaciation of Antarctica?. Mol. Biol. Evol. 23 **5**(2006) 874–882
26. Posada, D., Crandall, K.: MODELTEST: Testing the model of DNA substitution. Bioinformatics Applications Note
27. Barker, D.: LVB: Parsimony and simulated annealing in the search for phylogenetic trees. Bioinformatics. **20**(2004) 274–275
28. Saito, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**(1987) 406–425
29. Desper, R., Gascuel, O.: Fast and accurate phylogeny reconstruction algorithms based on the Minimum-Evolution principle. Comput. Evol. 9 **5**(2002) 687–705

Towards a Conceptual Modeling Language for Biological Domains

José Antônio Fernandes de Macêdo[1], Sérgio Lifschitz[2],
Fabio Porto[1], Philippe Picouet[3],
Antonio Basilio de Miranda[4], and
Thomas Dan Otto[4]

[1] Ecole Polytechnique Fédérale de Lausanne (EPFL), School of Computer and Communication Sciences - Database Laboratory, Lausanne, Switzerland
{jmacedo.macedo,fabio.porto}@epfl.ch
[2] Pontificia Universidade Catolica do Rio de Janeiro (PUC-Rio), Informatics Department, Rio de Janeiro - RJ, Brazil
sergio@inf.puc-rio.br
[3] Ecole Nationale Supérieure des Télécommunications de Bretagne (ENSTB), LUSSI Group, Brest, France
philippe.picouet@enst-bretagne.fr
[4] Fundação Oswaldo Cruz (FIOCRUZ), DBBM, Rio de Janeiro - RJ, Brazil
{antonio,otto}@fiocruz.br

Abstract. In-silico experiments have shortened the path to new discoveries reducing the need of expensive and time consuming in-vitro experiments. The corresponding biological knowledge must be represented in terms of data. Traditional conceptual modeling languages have been used to build conceptual data schemas. However, they are not adequate to represent non-monotonic inheritance, complex relationships and multi-representation, to name a few. In this paper we present a set of constructs and features that should be considered in a conceptual modeling language in order to enforce most of the biological domain requirements. In addition the proposed data model is formalized using first order logic, which permits to verify the consistence of a generated data schema.

1 Introduction

Conceptual data languages are quite important in computer science as they are the key for translating the real world objects into computer executable systems. An important feature of conceptual data modeling languages is the facility of specifying *concepts* of the Universe of Discourse (UoD) using some kind of abstraction. Representation is the outcome of an abstraction process, followed by a classification process (e.g. clustering all entities that are considered similar). Besides, conceptual data modeling languages are tools that aim at facilitating the communication among end-users and designers by focusing only on "what" must be represented and not "how" data is implemented in computer systems.

The design of a conceptual schema from scratch is a hard and time consuming task [1]. The level of difficult of this task is dependent on the expressive power

of the constructs provided by the conceptual language used in performing this task. For example, in some specific application domains, such as the geographical and multimedia, a conceptual schema can be very hard to be achieved. The main reason is because conceptual languages, which were used to design the conceptual schema, are not equipped with adequate constructs to represent those domains. Thus, new conceptual languages must be developed in order to facilitate the representation of specific application domains. Some proposed spatio-temporal conceptual languages serve as good examples of those efforts in facilitating data representation for geographical information system domain.

At the best of our knowledge, there is no conceptual data modeling proposal specific for the biological domain. The modeling process of biological systems requires either extensions to foundational constructs used in traditional modeling languages (e.g. EER[2]) or the provision of new constructs to support specific domain aspects. Indeed, we could adopt a combination of ORM [3] or UML constructs.

Therefore, we claim that there is a need for a conceptual data modeling approach to support the representation of biological data. In this work we first present a possible solution for this problem, defining some conceptual constructs that are more adequate to represent biological concepts. We then apply some of our ideas to a practical case study in the molecular biology domain. Finally we conclude and comment on contributions and future work.

2 BioConceptual Language

We have raised a list of requirements that should be fulfilled by a language in order to conceptually represent objects and relationships. Current (generic) conceptual modeling languages do not or only partially satisfy these biological domain requirements. We will briefly mention below some requirements that justify an ad-hoc modeling approach. The reader should refer to [4,5] for further details and a complete discussion.

- domain complexity: e.g. one biological concept may be represented in different levels of abstraction; also, there are many similar but not identical concepts that are challenging to model, such as amino acids. Besides ambiguous definitions, even for basic concepts as a gene.
- special relationships: e.g. the specification of orders and patterns, due to DNA sequence orders and motifs, that are related to subsequences representing promoters, stop codons, and others; Also, there is a need to enrich semantics, in order to capture all different classes of the part-whole and part-of relationships.
- *is-a* links: inherited relationships in biology are typically non-monotonic and traditional conceptual modeling languages are limited to represent only those generalization relationships where properties from subclasses cannot redefine the inheritance from superclasses properties. Usual modeling tricks either under or over specificate the real world semantics.

– pot-pourri: constraints and the specification of automatic classification of instances; constraints over hierarchies; integration with ontology; large and overlapping hierarchies; the distinction of functions and structures; spatial and temporal information.

Enhancing Relationships Semantics

In biological domain relationships among concepts plays a fundamental role. Since new discoveries in biology are usually made by comparing data in hand against existing knowledge, it is very important that existing relationships among concepts have a clear semantics in order to facilitate new discoveries.

Aggregation relationships is an example of the semantic expressive limitation of current approaches. Although aggregation relationship is used to represent the configuration of real-world things they do not express how this configuration is assembled. For instance, to represent a transcribed region as a composition of introns and exons we need to specify in what order this is achieved. The representation of biological components configuration is very important because they permit life scientists to identify important patterns into DNA sequences (also known as motifs[6]).

We claim that a modeling language should incorporate a special constructor, called *configuration constraint*, to aid the designer to specify a configuration using aggregation relationships. This constructor can be used to specify a regular expression that may express the configuration of a specific aggregation relationship. The reader may refer to [5] for some examples of this and other constructors.

Implementing non-monotonic IS-A relationships

Sometimes a biology application wants to differentiate associations in hierarchy levels (i.e. non-monotonic inheritance). Figure 1 refers to the same example presented before but now using a *is-a* link between relationship types. It illustrates the case where the application wants to represent that every biomolecule may have multiple structural components but proteins have none or at most one alpha helix, which is a structural component. This figure shows a refinement of a role consisting in attaching the role to a subtype of the object type that is attached to the inherited role. The cardinalities of both roles are inherited by AlphaHelix with modifications. The design ensures that proteins have only one alpha helix structure. Indeed, the cardinality (0,N) in StructuralComponent role has been restricted to (0,1) in AlphaHelix role.

Multiple representations

Each application has its own perception of the real world, leading to specific requirements, both in terms of what information is to be kept and in terms of how

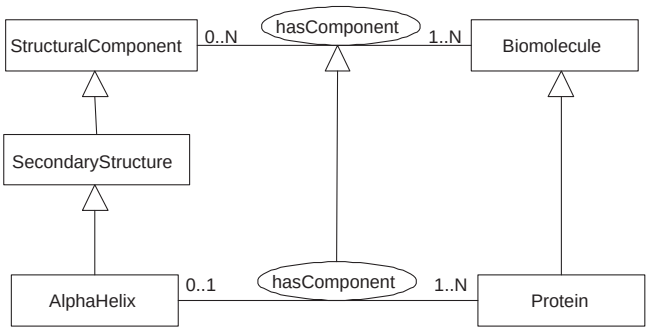


Fig. 1. Non-Monotonic Inheritance

the information is to be represented. Different applications, which have overlapping concerns about real-word phenomena, normally require different representations of the same phenomena. Differences may arise in all facets that make up a representation.

We propose a mechanism called *perception* that aims to specify perceptions of different applications. In order to illustrate this problem, we use an annotation application example that supports the research of different research groups for the same genome project. Both laboratories responsible for these projects use a similar range of biological elements (e.g. genes, proteins, etc), but utilize different representations of their data. Each laboratory has an analysis group which studies their regions, make annotations and suggest the associated functions. In order to succeed with the integration of both annotation systems a range of issues has to be tackle (e.g. same annotation vocabulary). Using the notion of perception, where each laboratory represents a separate perception, makes it possible to achieve this challenge in a flexible and logical manner.

To illustrate the possibility to add perception specification in a conceptual data model, let us assume now that three research laboratories use the concept Gene in different ways. The Figure 2 illustrates the representation of Gene concept and the specification of three perceptions (i.e. LaboratoryA, LaboratoryB and LaboratoryC). Perception information are organized hierarchically where each internal node represents a different user defined perception and each leaf node represents a data schema element, such as: object data type, relationship type or attribute type. The perception hierarchy is based on generalization relationships. Thus, structure and behavior can be inherited. In addition, overloading and overriding mechanisms can be applied to perceptions. The root perception node is always Context, which contains some basic generic structure as well as behavior that is shared by all inheritors, unless overridden.

The general idea is to associate object types, attributes or relationship types with perceptions. Associations between data schema elements and perception hierarchy may be implicit or explicit. Implicit association is made by using same names for perception and data schema elements. A dot notation is used to de-

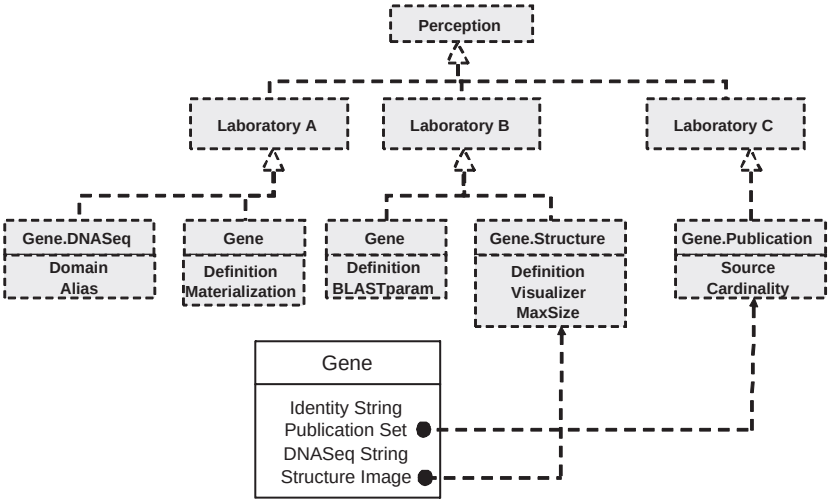


Fig. 2. Contexts and different representations to the Gene object data type.

scribe attribute names of object data types or relationship types. For example, the Gene object type is implicit associated with two perceptions: LaboratoryA and LaboratoryB throughout two perceptions named Gene as illustrated in Figure 2. Conversely, the explicit association is made by using dotted oriented lines as represented in Figure 2 between Publication attribute and Gene.Publication perception. An explicit association indicates that a data schema element (Publication attribute) only exists in the associated perception (Gene.Publication perception) whereas an implicit association, when exists, denotes a new perception of the data schema element in the associated perception.

3 Formalization Approach

In this section we show the formalization of our language. We have use the hybrid approach by Berardi et al [7] as our formalization framework. This approach aims to associate formal semantics for the graphical constructs permitting to take advantage of methodologies developed in software engineering and to use all framework developed for logic theories when necessary. Logical theory may help to permit the understanding formalization, verification, correction, automated reasoning and model inference. In this way we also permit that conceptual diagrams could be formally verified and manipulated by machines.

We use as descriptive language the first order logic because it presents a precise semantics with an expressiveness adequate to represent complex constraints of biology domain. An important aspect of this approach is to obtain a well-formed logical theory from conceptual diagrams. In this manner, we may explore the characteristics of this logical theory to simplify inferences, reaching decidability and adequate computational complexity for reasoning procedures.

The meta-model represents the alphabet that will be used to define the data schema. Each schema element is represented as a logical predicate, which associates an element of the meta-model (intension) with their instances (extension). For example, the Gene concept which is an object type may be syntactically defined by $Gene(x)$, where x stands for all possible instances of this concept.

Similarly, an attribute is also represented as a logic predicate related with object type that it belongs to. Thus, attributes are binary predicates complemented with assertions to represent data type and multiplicity. For example, the A attribute of T type from C class and multiplicity $i..j$ can be mapped to the binary predicate $A(x, y)$ additionally to the following assertions:

- Assertion for attribute data type: $\forall$ x, y.(C(x) $\wedge$ A(x,y) $\supset$ T(y)
- Assertion for attribute multiplicity:$\forall$ x.(C(x)$\supset$ (i $\leq$ #{y | a(x,y)}$\leq$ j) [5]

For example, the object type *Gene* that has the following attributes : *id*, *creation*, *authority* and *publication*. The *id* attribute is a number, creation is represented by a *date*, *authority* stores a string represent the author *name* and a set of *publications*. These definitions will be transformed into a set of the following formulas:

Attributes Axioms

$\forall$ x, y.(Gene(x) $\wedge$ id(x,y)) $\supset$ Number(y)
$\forall$ x.(C(x)$\supset$ (i $\leq$ #{y | id(x,y) $\leq$ 1)

$\forall$ x, y.(Gene(x) $\wedge$ creation(x,y)) $\supset$ Date(y)
$\forall$ x.(C(x)$\supset$ (i $\leq$ #{y | creation(x,y) $\leq$ 1)

$\forall$ x, y.(Gene(x) $\wedge$ authority(x,y)) $\supset$ String(y)
$\forall$ x.(C(x)$\supset$ (i $\leq$ #{y | authority(x,y)} $\leq$ 1)

$\forall$ x, y.(Gene(x) $\wedge$ publication(x,y)) $\supset$ Set(y)
$\forall$ x.(C(x)$\supset$ (i $\leq$ #{y | publication(x,y) $\leq$ 100)

Relationships are modelled as relationship types that model properties, which are not local properties because they involve another classes. Then, a relationship between two classes is a property in both classes. We may represent a binary relationship between C_1 and C_2 classes using a R binary predicate

$$\forall\ x_1,x_2.\mathbf{R}(x_1,x_2) \supset C_1(x_1) \wedge C_2(x_2)$$

Figure 3 illustrates a relationship type between Gene and Protein object types. According to our approach this relationship may be defined as follows:

[5] This is a reduced representation for the logical formula to represent the possible values of y

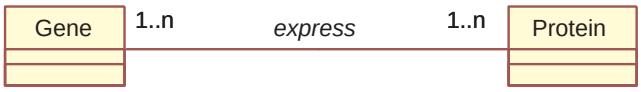


Fig. 3. A diagram showing a relationship type linking two object types

$$\forall\, x_1,x_2.\text{express}(x_1,x_2) \wedge \text{Gene}(x_1) \supset \text{Protein}(x_2)$$

The multiplicity of relationship types must also be expressed as predicates. We show how to represent multiplicity between C_1 and C_2 classes:

$$\forall\, x_1.C_1(x_1) \supset (min_1 \leq \#\{x_2 \mid \mathbf{R}(x_1,x_2)\} \leq max_1)$$
$$\forall\, x_2.C_2(x_2) \supset (min_2 \leq \#\{x_1 \mid \mathbf{R}(x_1,x_2)\} \leq max_2)$$

The complete logical representation of our example illustrated in Figure 3 is showed below:

Axioms
$\forall\, x_1,x_2.\text{express}(x_1,x_2) \wedge \text{Gene}(x_1) \supset \text{Protein}(x_2)$
$\forall\, x_1.\text{Gene}(x_1) \wedge (1 \leq \#\{x_1 \mid \text{express}(x_1,x_2)\}\ )$
$\forall\, x_2.\text{Protein}(x_2) \wedge (1 \leq \#\{x_1 \mid \text{express}(x_1,x_2)\}\ )$

Figure 4 illustrates the use of "IS-A" link indicating that there are twos types of genes: mouse and human. The "IS-A" link is used to denote the semantics of subsumption between two classes. It is also possible to include some constraints over involved classes in "IS-A" relationship. The regular use of these restrictions are:

- Disjoint: different subclasses do not have common instances;
- Complete: for all instance that belong to superclass they must belong to at least one subclass;

The definition of "IS-A" link and their features may be formally specified by the following formulas:

- "IS-A" link: $\forall x.C_i(x) \supset C(x)$ para $i = 1, ..., n$
- Disjoint Constraint: $\forall x.C_i(x) \supset C_j(x)$ para $i \neq j$
- Complete Constraint: $\forall x.C(x) \supset \bigvee_{i=1}^{n} C_i(x)$

Applying this formalization over the example illustrated in Figure 4 we became to the following axioms:

Axioms for "IS-A" links
$\forall x.Human(x) \supset Gene(x)$
$\forall x.Mouse(x) \supset Gene(x)$
$\forall x.Human(x) \supset \neg Mouse(x)$

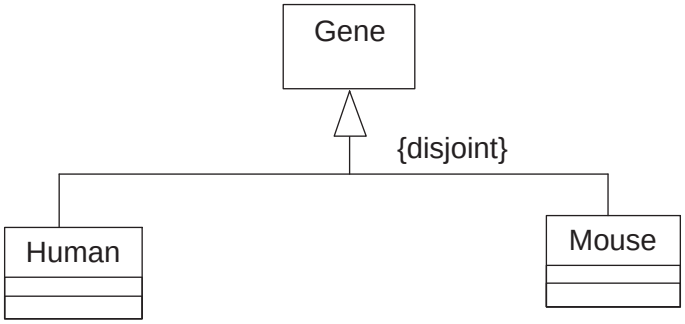


Fig. 4. Classifying Gene type into two subtypes using IS-A link

Our approach emphasizes the Constraint constructor that may be applied to another constructor (e.g. object type, relationship type, etc). Thus, constraints may be formalized through logical formulas directly written by modeler. For example, whether modeler want to establish that exist a relationship among genes but these relationship must only include homologue gene(i.e. genes that have 80% of similarity), he may associate a constraint to the respective relationship type declaring:

Axiom for Constraint
$\forall x, y.(x \neq y \wedge Gene(x) \wedge Gene(y) \wedge similarity(x, y, 80\%)) \supset homologue(x, y)$

The fact that we have an approach to map a conceptual diagram to a set of logical formulas permit us to verify formally relevant properties of the generated conceptual schema. For instance, some inferences may be executed over the schema such as:

- Object type consistence
- Schema consistence
- Subsumption verification
- Object type equivalence

An object type is consistent whether the data schema accepts an instance of this object type. Intuitively, an object type can be populated without violating any constraint imposed by the schema. The object type inconsistence indicates an design error. However, the detection of one error may enable the data modeler to redefine specifications. An object type is consistent if and only if there is a model that satisfies the set of all assertions derived from a specific BioConceptual schema. More formally, Let Γ model a set of assertions and $C(x)$ the predicate associated to the data type C. Then C is consistent if and only if:

$$\Gamma \models \forall x.C(x) \supset true$$

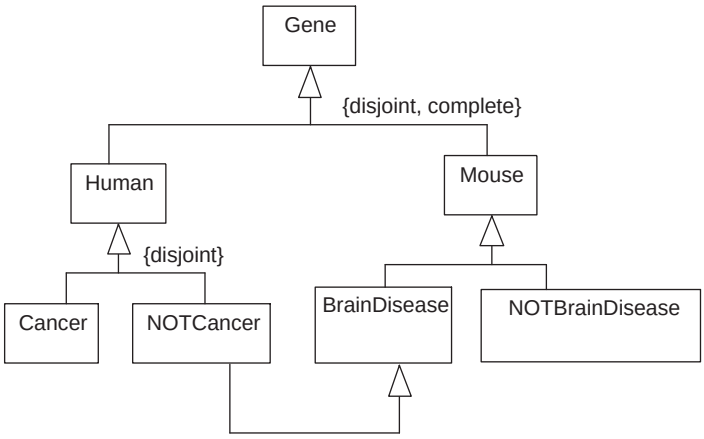


Fig. 5. An inconsistent conceptual schema

Figure 5 shows an example where an inconsistent schema.

Performing inferences over this schema we obtain the following logical consequences:

$$\Gamma \models \forall x.NOTCancer(x) \supset false$$
$$\Gamma \models \forall x.Human(x) \supset Cancer(x)$$
$$\Gamma \models \forall x.Cancer(x) \equiv Human(x)$$

The first logical consequence demonstrates that the $NOTCancer$ object type can have instances. The second logical consequence determines that all Human instances are also Cancer instances. Finally, we may deduce from the third logical consequence that Cancer and Human object types are the same. In this case, the resulting conceptual model must merge these two concepts into one.

4 Conclusions and Future Work

In this paper we have proposed some conceptual modeling language constructs and mechanisms for the molecular biology domain. Our proposal is established on object-oriented framework using an UML-like notation as graphical language. We have defined and formalized an initial set of constructs and features that meet requirements in terms of data modeling in the biological domain.

We have elicited biologists data modeling requirements towards the specification of a tailored data representation language. These requirements are related to fundamental conceptual modeling constructs and mechanisms that should serve as the basis for *ad-hoc* languages.

Our current and future research work focus with further studies of requirements, case studies and the specification of an actual data modeling language.

We claim that this would be a very important tool tailored that could help with database design, data integration and query definition, among other applications and user needs.

References

1. Chen, I.M.A., Markowitz, V.M.: An Overview of the Object-Protocol Model (OPM) and OPM Data Management Tools. Information Systems **20**(5) (1995) 393–418
2. Gogolla, M., Hohenstein, U.: Towards a semantic view of an extended entity-relationship model. ACM Transactions Database System **16**(3) (1991) 369–416
3. Halpin, T.: Object-Role Modeling. http://www.orm.net (2006)
4. de Macedo, J.A.F., Lifschitz, S., Porto, F., Picouet, P.: Dealing with some conceptual data model requirements for biological domains. In: IEEE Symposium on Bioinformatics and Life Science Computing (BLSC07). 2007 pages 651–656.
5. de Macedo, J.A.F., Lifschitz, S., Porto, F., Picouet, P.: A conceptual data model language for the molecular biology domain. In: IEEE International Symposium on Computer-based Medical Systems (CBMS07). 2007 pages 231–236.
6. Crochemore, M., Sagot, M.F.: Motifs in sequences: localization and extraction. In: Handbook of Computational Chemistry. Marcel Dekker Inc. (2005)
7. Berardi, D., Calì, A., Calvanese, D., De Giacomo, G.: Reasoning on uml class diagrams. (Technical report)

About a preference for stop-resistant codons in eukaryotic protein-coding genes

Francisco Prosdocimi[1], J. Miguel Ortega[1,*]
[1] Laboratório de Biodados, Dept Bioquímica e Imunologia, ICB-UFMG, Brazil
{franc, miguel}@icb.ufmg.br

Abstract. The application of stepwise nucleotide substitutions in $poli_{100}$-codon sequences disposed in-tandem allowed us to verify which codons are more prone to be replaced by stop codons. Observing a poli-codon sequence *in silico* evolved, we defined a 1% Death Lethal dose for each codon, meaning how many substitutions it can resist before mutating to stop codon. Lower $DL1_{stop}$, higher the chance of a codon being replaced by a stop. Due to genetic code degeneracy, different codons codify same amino acids and, therefore, there will be synonymous codons presenting different $DL1_{stop}$ indices. We evaluated all KEGG protein-coding genes of 9 eukaryotic and also synthetic random genes in order to find whether natural selection have selected genes to prefer the utilization of stop-resistant synonymous codons or not. Although this selection has not proven to be a general pattern, we observed the presence of genes using mainly stop-resistant codons, such as many *S. pombe* ribosomal proteins.

Keywords: genetic code, natural selection, codon usage, evolutionary stability, stop codons

1 Introduction

The evolution of protein-coding genes is clearly related to the functionality of the protein, since natural selection must keep all the organisms' metabolism working properly. Thus, individuals presenting a number of non-functional proteins probably will not be able to produce viable offspring. However, proteins evolve from random DNA mutations and therefore, mutations will only be maintained if they do not produce too much modification in the protein structure and/or function. Many amino acid substitution matrices have been described to suggest different effects of amino acid residue replacement on actual proteins [1]. Although these matrices were built considering many factors that might alter protein function, such like residue conservation [2, 3], amino acid chemical properties [4, 5], frequency of amino acid contacts in protein structures [6], residue volume [4], hydropathy [7], frequency of dipeptides [8] and many other characteristics; none of them have tried to weight the value of the worse kind of amino acid substitution possible: the mutation into a stop codon.

Although protein evolution depends on natural selection, mutations in protein-coding genes will only be maintained if they produce functional proteins. Moreover, the evolutionary kinetics of amino acid replacement in protein sequences is clearly

linked to the codon-amino acid attributions given by the genetic code [9-11]. It is well known that genetic code minimizes the effect of mutations by using similar codons to codify for the same amino acid, avoiding non-synonymous mutations [9, 12, 13]. Although somewhat allowing DNA mutation resistance, synonymous codons clearly vary in their mutational distance to stop codons. For example, both UUA and CUU codons codify Leucine residues; however one single mutation in UUA Leucine codon may produce the UGA or UAA stop codons, while CUU would need more than one mutation to be replaced by a stop codon. This way, CUU is more stop-resistant than UUA.

Here, we have evaluated the codon usage of eukaryotic protein-coding genes to check their preference for the usage of stop-resistant codons. Analyzing each group of synonymous codons, we have developed a metric ($DL1_{stop}$) to calculate the distance of each codon to a stop signal. All protein-coding sequences from 9 complete sequenced eukaryotes (174,530 sequences) were downloaded from KEGG database [14, 15] and analyzed. A set of random DNA protein-coding sequences was also produced and compared to the overall data for actual organisms. Moreover, considering a given protein, we were able to build a synthetic putative protein presenting the closer and farther synonymous codon configuration in regard of stop codon distance. Actual and random protein-coding sequences were compared to these best and worst evolutionary stable proteins possible and we have confirmed the well-known observation that genetic code minimizes the effect of DNA mutation, among other interesting observations.

2 Methods

2.1. Production of poli-codon ancestral FASTA sequences

We have produced 64 multi FASTA files (one for each codon) containing 10,000 identical poli-codon sequences of 100 codons each. For example, we have produced an “ATG.fasta” file containing 10,000 sequences, each of them composed of 100 ATG codons disposed in tandem. Similar files were produced for each other of the 63 codons. These files were submitted to an *in silico* molecular evolutionary process as described below.

2.2. In silico evolution of poli-codon sequences and the DL1 index of stop codon attachment

Each one of 10,000 poli100-codon sequences was subject to a series of stepwise substitution mutations on its DNA sequence. We did not produce insertions and deletions and we have evolved the sequences based on Jukes and Cantor-like molecular evolutionary model [16]. This model was based on a series of stepwise mutation rounds and each round was characterized for the mutation of one single base. The base mutated was chosen randomly and, consequently, it presented 25%

chances to be mutated to any other base, including itself. Each poli100-codon sequence was evolved in *n* nucleotide substitution rounds until it gets a single in-frame stop codon on its sequence and the number *n* of mutational steps necessary to produce this stop codon was stored in a MySQL database, counted and averaged for all sequences. The evolving simulated sequences present 100 codons and we have stopped the simulation when we found an in-frame codon mutated to any stop codon. Therefore, this averaged number of mutational steps leading to the appearance of a single stop codon in protein was considered the mutational lethal dose capable to transform 1% of codons in a given protein to a stop codon, and it was called $DL1_{stop}$. Codons presenting higher $DL1_{stop}$ were considered evolutionary stable, since they require more time (measured in number of mutational events) to be replaced by a stop codon. Empirical values of $DL1_{stop}$ were manually smoothed to avoid statistical artifacts produced by random mutations.

2.3. Protein Average Stop Attachment value

In order to calculate the average evolutionary stability of a single protein, we have applied $DL1_{stop}$ index to each codon codifying it. These codon indexes were then summed and averaged, considering the protein size. So, each protein will present an Average Stop Attachment index (1) as follows:

$$ASA = \frac{\sum DL1_{stop}\, codon}{N} \qquad (1)$$

where N is the total number of amino acids of a given protein.

2.4. Relative Stop attachment of proteins

Another index named Relative Stop attachment (*RSa*) was calculated for each protein. This index was produced based in the fact that we know which codons in a group of synonymous codons are the most and less stable ones, considering their $DL1_{stop}$ value. Therefore, we are able to predict a putative synonymous protein (containing the same amino acids of some given sequence) presenting the best and the worst configuration of codons in regard of stop attachment. Here, we have considered the best protein as the one presenting the same amino acids, but codified by the synonymous codons with highest $DL1_{stop}$. On the other way, the worst protein was considered the one containing the synonymous codons with lowest $DL1_{stop}$. Based on this index, we have calculated for actual proteins how close they are from the worst (or the best) configuration stop codon distance. Therefore, protein Relative Stop attachment (2) was defined as:

$$RSa = \frac{(ASA_{protein} - ASA_{worst})}{ASA_{best} - ASA_{worst}} * 100 \qquad (2)$$

the best and worst values have been calculated considering an entirely identical protein, on which the codons are, respectively, the farthest and the closest regarding stop codon distance.

Whether a protein presents the best possible configuration of stable amino acids, it will receive an RSa index of 100, while the worst configuration will be scored as 0. Intermediate values will represent where the protein is in the way from the worst codon configuration possible to the best one.

2.5. Evolutionary stability tests on actual protein-coding genes

We have downloaded 174,530 protein-coding genes of 9 eukaryotic organisms from KEGG Genes database [14]. We have decided to use KEGG since this database allows the retrieving of the coding part of genes (CDS) and permits an easy parsing of codon sequences. Moreover, using KEGG Orthology we were able to compare orthologous groups of proteins regarding their stop codon attachment. Other ortholog databases, such as COG [17] and MultiParanoid [18], does not present the nucleotide sequence of proteins clustered and their usage would need further steps of protein to gene mapping, unnecessary when using KEGG.

Table 1 shows the organisms analyzed in this work and the number of protein-coding genes downloaded from the databases.

Table 1. Organisms and number of protein-coding genes used in this work

Organism	Mnemonic	Number of KEGG protein-coding genes	Number of KO protein-coding genes
Arabidopsis thaliana	ath	26,803	2,880
Caenorhabditis elegans	cel	20,080	3,079
Canis familiaris	cfa	19,809	2,533
Drosophila melanogaster	dme	14,508	1,997
Homo sapiens	hsa	25,719	7,198
Mus musculus	mmu	30,057	8,046
Rattus norvegicus	rno	26,259	5,953
Saccharomyces cerevisiae	sce	6,224	1,788
Schizosaccharomyces pombe	spo	5,071	1,389

All proteins from a given genome were classified and their ASA and RSa indexes were calculated. We have also produced a set of 10,000 randomly generated DNA sequences to compare results of actual proteins to random configuration of nucleotides. The proportion of codons in these random synthetic protein-coding genes respects the genetic code proportion (e.g. six Rs to each W).

3 Results

3.1. $DL1_{stop}$ attachment by codon

We have applied stepwise DNA mutation rounds in a set of 640,000 nucleotide sequences, being each sequence composed of 100 identical codons; 10,000 of such 100 base sequences were assayed for each of the 64 codons. For each mutation round, one single nucleotide had the chance to mutate into each other nucleotide, including itself, at 25% chance. So, these poli100-codons sequences were evolved until they get a single stop codon. As soon as, e. g., one of the 100 codons of a poli-TTT gets mutated to a stop codon, the evolution of this gene was stopped and the number of mutational steps necessary to achieve this stop codon mutation was stored in a database, counted and averaged for each 10,000 sequences produced for this codon. The same procedure was performed for each other codon and $poli_{100}$-codon sequence. Table 2 shows, for each codon, its amino acid codified, smoothed values of stop codon attachment and the class of stop distance it belongs.

As we see in Table 2, there are 9 amino acids (F, Q, Y, C, H, N, K, D and E) on which their synonymous codons present the same smoothed value of $DL1_{stop}$. The presence of these amino acids in a given protein does not modify its closeness to stop codons, since one is not able to produce neither a stop codon closer nor a farther version of this specific protein changing only the codon usage of these 9 amino acids. So, these amino acids may be seen as *evolutionarily innocuous*. Methionine and Tryptophan are also evolutionarily innocuous, since they are codified for one single codon. Otherwise, the other 9 evolutionarily non-innocuous amino acids present codons with different DL1stop values and they are the main responsible to make protein-coding genes closer or farther to stop codon mutations. Leucine (L2, L3, L5, L6, L7) present five distinct classes of DL1stop values for their codons; Serine present four codon classes (S2, S3, S4, S6); Arginine (R3, R5, R6), Threonine (T5, T6, T7), Proline (P5, P6, P7), Glycine (S3, S5, S6), Alanine (A5, A6, A7) and Valine (V5, V6, V7) have three classes and, at least, Isoleucine is codified by codons on class 5 or 7 of stop codon distance.

3.2. Evolutionary Stability indices measured on KEGG proteins

The average and relative stop codon attachment indices (*ASA* and *RSa*) were calculated for each KEGG protein and their distribution can be seen by organism, respectively, in Figures 1 and 2.

As we see in Figure 1, data produced based on mutations in random nucleotide sequences obeys a normal curve. Data for actual proteins are similar to random analysis and they appear just as slightly shifted curves. *Sce*, *cel*, *ath* and *spo* data seem to have their average in a smaller *ASA* value than random data; *cfa*, *rno*, *mmu* and *hsa* (Chordata taxons) have shown similar *ASA* average value than random data, although they present a more wide bell-shaped curve and a group of genes more resistant to

stop than random data at the right portion of the distribution; *dme* curve seems to fit random data.

Table 2. Average mutational distance and classes of D1stop attachment by codon

Codon	AA[a]	$DL1_{stop}$[b]	$DL1_{stop}$ class[c]	Codon	AA[a]	$DL1_{stop}$[b]	$DL1_{stop}$ class[c]
TTT	F	73	4	ATT	I_7	205	7
TTC	F	73	4	ATC	I_7	205	7
TTA	L_2	6	2	ATA	I_5	90	5
TTG	L_3	12	3	ATG	M	123	6
TCT	S_4	73	4	ACT	T_7	205	7
TCC	S_4	73	4	ACC	T_7	205	7
TCA	S_2	6	2	ACA	T_5	90	5
TCG	S_3	12	3	ACG	T_6	123	6
TAT	Y	6	2	AAT	N	90	5
TAC	Y	6	2	AAC	N	90	5
TAA	*	0	1	AAA	K	12	3
TAG	*	0	1	AAG	K	12	3
TGT	C	12	3	AGT	S_6	123	6
TGC	C	12	3	AGC	S_6	123	6
TGA	*	0	1	AGA	R_3	12	3
TGG	W	6	2	AGG	R_5	90	5
CTT	L_7	205	7	GTT	V_7	205	7
CTC	L_7	205	7	GTC	V_7	205	7
CTA	L_5	90	5	GTA	V_5	90	5
CTG	L_6	123	6	GTG	V_6	123	6
CCT	P_7	205	7	GCT	A_7	205	7
CCC	P_7	205	7	GCC	A_7	205	7
CCA	P_5	90	5	GCA	A_5	90	5
CCG	P_6	123	6	GCG	A_6	123	6
CAT	H	90	5	GAT	D	90	5
CAC	H	90	5	GAC	D	90	5
CAA	Q	12	3	GAA	E	12	3
CAG	Q	12	3	GAG	E	12	3
CGT	R_6	123	6	GGT	G_6	123	6
CGC	R_6	123	6	GGC	G_6	123	6
CGA	R_3	12	3	GGA	G_3	12	3
CGG	R_5	90	5	GGG	G_5	90	5

a. Amino acid codified by codon; the number associated represent the different $DL1_{stop}$ classes a single amino acid can belong; b. Smoothed average number of mutations necessary to produce 1% of stop codons in a given $poli_{100}$-codon sequence ($DL1_{stop}$); c. Codon classes by $DL1_{stop}$ (we have considered stop codons themselves as class 1).

Figure 2 also shows random data fitting a normal curve. Data for actual proteins once again have shown to be similar when compared to random analysis and they appear as slightly shifted curves. This time, data for *cfa*, *hsa*, *rno*, *mmu* and *dme* are

shifted from random to the right side, representing more evolutionary stable proteins. On the other hand, *sce* and *cel* seems to be producing more evolutionary instable than random data.

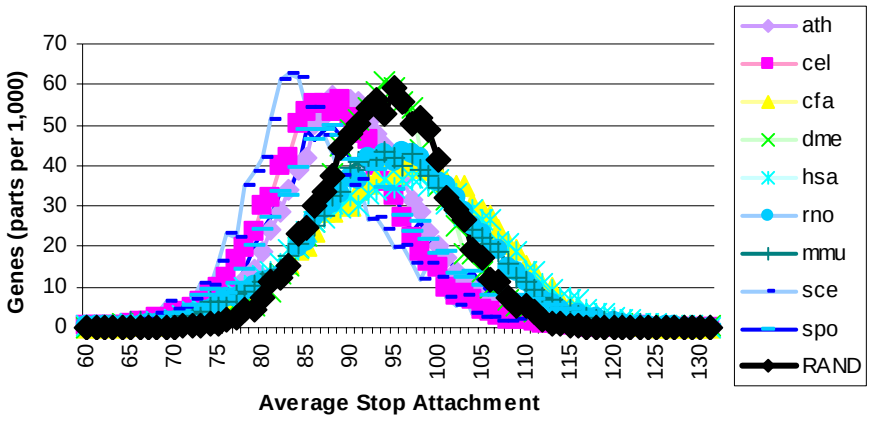


Fig. 1. Distribution of *ASA* values by organism per thousand genes. The *ASA* value represents the average number of mutations a protein can resist per 100 codons without bearing a single stop codon.

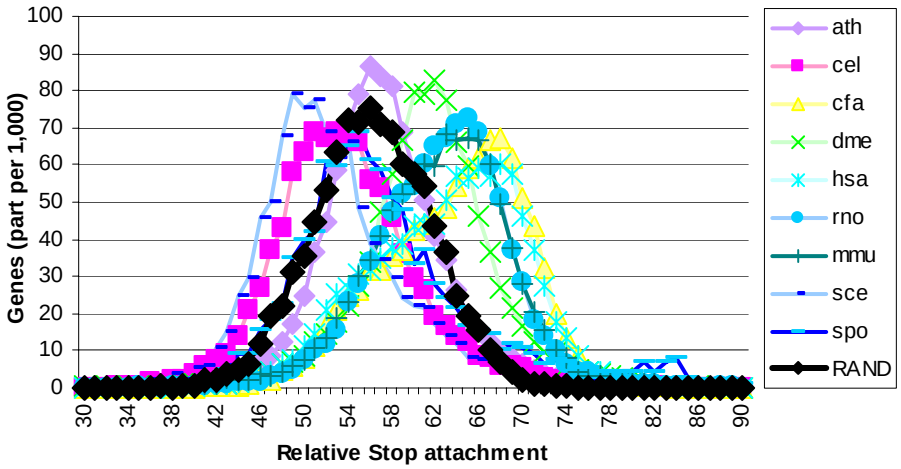


Fig. 2. Distribution of *RSa* values by organism per thousand genes. The *RSa* value represents how the proteins behave tending to be closer (left side, worse stability) or farther (right side, better stability) to stop codon occurrence, considering their putative codon configuration

Table 3 shows average and standard deviation values of *RSa* and *ASA*, including calculated values for best and worst evolutionary stable version of organisms' proteins.

Table 3. Average and Standard Deviation of main stop attachment indices used in this work

Org	RSa Avg	RSa Std	ASA Avg	ASA Std	ASA_{best} Avg*	ASA_{best} Std*	ASA_{worst} Avg*	ASA_{worst} Std*
cfa	64.10	6.54	96.63	9.77	124.26	9.83	46.71	3.75
hsa	63.38	6.98	96.11	11.13	124.47	11.66	46.30	4.35
rno	62.95	6.43	94.78	11.69	122.89	13.24	46.48	5.30
mmu	62.77	6.44	94.96	10.94	123.31	12.23	46.67	4.98
dme	61.91	6.57	94.36	8.61	122.81	9.39	48.09	4.28
spo	58.38	*9.02*	91.21	10.01	121.92	8.16	47.92	3.60
ath	57.70	5.59	90.85	7.96	122.45	8.68	47.65	3.72
RAND	56.76	5.39	94.59	6.96	*131.34*	7.24	46.38	4.09
cel	54.47	6.58	88.65	8.10	121.41	8.95	49.40	3.73
sce	52.83	6.34	86.55	8.53	120.77	9.20	48.09	3.87

* Calculated, not-real data

Data shown in Table 3 confirms Figures 1 and 2 analyses. *ASA* and *RSa* values have shown to be higher in *hsa* and *cfa*. As observed in figures, *RSa* and *ASA* RAND data are not as different from actual data as one might suppose. It is also interesting to verify that best *ASA* theoretical values (ASA_{best}) observed for random data is clearly higher (131.34) than empirical observations of KEGG protein-coding genes. Considering that theoretical data for worst RAND protein (46.38) is similar to empirical ones, this explains why having a high average *ASA* value, random data present a small *RSa*.

3.3. Outlier RSa proteins

Protein-coding genes presenting outlier values of *RSa* and a KO annotation were analyzed in this section. Considering the *RSa* index, random generated data have shown to present values between 37.19 and 78.52, averaged 56.76. We have used this last average random *RSa* score plus (or minus) a number of random standard deviations (5.39, see Table 3) to verify the number of actual KO annotated proteins by organism present in each class (Table 4).

Data in Table 4 shows a clear tendency to be more present in the right side high *RSa* columns, evidencing KO proteins as preferentially selected for higher distance from stop codons. *Spo* has proven to be the organism with highest indices of *RSa* in their KO classified proteins, a curious observation since sce was verified to present a small number of proteins with high *RSa* scores. Amongst these 16 highly scored *spo* proteins, 11 were identified as ribosomal proteins, 3 as subunits of the translation elongation factor 1-alpha, 1 as an alcohol dehydrogenase and a last one as the glycolysis' pathway enzyme pyruvate kinase. These data may be taken with caution since KO classification is under working progress and the proportion of proteome in database may vary.

Table 4. Number of KO annotated genes presenting outlier *RSa* values, considering average and standard deviation of random data

Org	Avg- 6*Std *<24.42*	Avg- 5*Std *<29.81*	Avg- 4*Std *<35.20*	Avg- 3*Std *<40.59*	Avg+ 3*Std *>72.93*	Avg+ 4*Std *>78.32*	Avg+ 5*Std *>83.71*	Avg+ 6*Std *>89.10*
cfa	0	0	0	0	169	4	0	0
hsa	1	1	1	3	451	26	2	0
rno	0	0	0	0	165	9	0	0
mmu	0	0	0	2	149	11	2	0
dme	0	0	5	12	195	34	8	0
spo	0	1	1	2	284	192	85	16
ath	0	0	1	3	62	7	0	0
RAND[a]	0	0	0	15	11	1	0	0
cel	0	0	1	14	48	7	1	0
sce	1	1	3	15	4	1	0	0

a. Number of RAND genes in each class, although RAND genes were not classified into KOs.

4 Discussion

In this paper, we have analyzed the codon usage composition of eukaryotic protein-coding genes considering the closeness between codons and stop codons. It might be expected that natural selection would modify the codon composition of proteins to make them farther from stop codons; maintaining even the highly mutated individuals with a working metabolism. KEGG has shown to be a good data source to use, since it makes available the complete nucleotide coding sequence of protein-coding genes, many of them classified into orthologous groups. $DL1_{stop}$ index has allowed the identification of closer and farther synonymous codons to stop codons. The *ASA* value, defined for each protein, has permitted the evaluation of the average mutation number that a given protein is able to resist per 100 codons without bearing a single stop codon. It is clear for us that *ASA* value might be influenced by different sized proteins and also for different mutation rates on which genes have been submitted. However, an overview of stop codon proximity influence on protein-coding genes has been achieved and further studies on this subject may elucidate better particular aspects of this interesting hypothesis. Whist *ASA* value scores a mutation to stop probability per 100 codons in a given protein, *RSa* interestingly presents the position where a protein is in the way from the worst to best evolutionary stability, considering its amino acid sequence. Using the theory presented here, it is possible to predict a future world where organisms' proteins will be manipulated in embryos in order to turn them as the most evolutionary stable they can be (as shown using ASA_{best} parameter), probably helping to avoid cancers and other diseases caused by DNA somatic mutations. The evolutionary stability of proteins, however, it is clearly not only a function of stop codon proximity and another evolutionary factors, such like the capacity of codons to keep coding the same amino acid after DNA mutations [19], must be considered when thinking about protein-coding genes resistance to mutations.

Moreover, in order to study the evolutionary stability inside single proteins, we suggest the usage of a translation vocabulary on which new proposed symbols will define each class of stop codon closeness (Table 2). Using these data, we are currently producing an extension of the present study trying to correlate stop codon closeness with codon position inside proteins. One might expect that codons used in the C-terminal might present less stop-resistant codons than N-terminal ones (Prosdocimi and Ortega, unpublished).

The similarity observed between random produced nucleotide sequences and actual data evidence that selection for stop-resistant codons have happened generally in a soft and non-directed way (Figures 1 and 2, Table 3), mainly in Chordata phyla. Besides, some proteins seemed to have been directly selected for using codons with a high $DL1_{stop}$ index (Table 4), such like many *spo* ribosomal ones. Some other interesting observations can be made looking into data presented: (1) this kind of stop-resistant codon selection have seemed to be absent or very slight in *ath*, *cel* and *sce*, since they do not present many outlier genes and they have shown the smallest values of *ASA* and *RSa* indices; (2) although presenting the highest average *RSa* value in their proteins (Table 3), *cfa* has not shown many outlier *RSa* genes (Table 4), making us to believe that selection for stop-resistant codons in this organism is made in a broader genomic way; (3) the fact that *spo* present proteins with higher outlier *RSa* scores (Table 4) having a regular *RSa* average can be explained if we observe it presents the higher *RSa* standard deviation (Table 3); (4) the suggestion made when analyzing *spo* outlier *RSa* genes that ribosomal proteins would present higher stop codon distance was not confirmed when analyzing most evolutionary stable KOs (data not shown) and a new hypothesis can be made about this to be true just in unicellular organisms, what may be tested in future, analyzing prokaryotic genomes; (5) the well-known fact that genetic code minimizes the effects of DNA mutations was once more confirmed here, since random produced data has shown an average *RSa* value larger than 50, that might be obtained if genetic code was indifferent to the distance of codons to stop codons.

Here, based on a standard scientific hypothesis about the usage of stop-resistant codons by eukaryotic protein-coding genes, we have developed a completely bioinformatics experiment to test this hypothesis. Therefore, more than being used in data mining and in the choosing of most likely experimental data to be preferentially tested, as it has already proven to be highly useful, bioinformatics shows also its relevance in the production of interesting and new knowledge about biology and evolution in the molecular level.

Acknowledgments. We thank Adriano Barbosa for critical reviewing the manuscript.

References

[1] Bulka B, desJardins M, Freeland SJ. An interactive visualization tool to explore the biophysical properties of amino acids and their contribution to substitution matrices. BMC Bioinformatics. 2006 Jul 3;7:329.

[2] Henikoff S, Henikoff JG. Amino acid substitution matrices. Adv Protein Chem. 2000;54:73-97.

[3] Tyagi M, Gowri VS, Srinivasan N, de Brevern AG, Offmann B. A substitution matrix for structural alphabet based on structural alignment of homologous proteins and its applications. Proteins. 2006 Oct 1;65(1):32-9.
[4] Goodarzi H, Katanforoush A, Torabi N, Najafabadi HS. Solvent accessibility, residue charge and residue volume, the three ingredients of a robust amino acid substitution matrix. J Theor Biol. 2006 Dec 19; [Epub ahead of print]
[5] Grantham R. Amino acid difference formula to help explain protein evolution. Science. 1974 Sep 6;185(4154):862-4.
[6] Miyazawa S, Jernigan RL. A new substitution matrix for protein sequence searches based on contact frequencies in protein structures. Protein Eng. 1993 Apr;6(3):267-78.
[7] Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol. 1982 May 5;157(1):105-32.
[8] Gonnet GH, Cohen MA, Benner SA. Analysis of amino acid substitution during divergent evolution: the 400 by 400 dipeptide substitution matrix. Biochem Biophys Res Commun. 1994 Mar 15;199(2):489-96.
[9] Archetti M. Selection on codon usage for error minimization at the protein level. J Mol Evol. 2004 Sep;59(3):400-15.
[10] Archetti M. Codon usage bias and mutation constraints reduce the level of error minimization of the genetic code. J Mol Evol. 2004 Aug;59(2):258-66.
[11] Hershberg U, Shlomchik MJ. Differences in potential for amino acid change after mutation reveals distinct strategies for kappa and lambda light-chain variation. Proc Natl Acad Sci U S A. 2006 Oct 24;103(43):15963-8.
[12] Haig D, Hurst LD. A quantitative measure of error minimization in the genetic code. J Mol Evol. 1991 Nov;33(5):412-7. Erratum in: J Mol Evol 1999 Nov;49(5):708.
[13] Zhu CT, Zeng XB, Huang WD. Codon usage decreases the error minimization within the genetic code. J Mol Evol. 2003 Nov;57(5):533-7.
[14] Kanehisa M, Goto S, Kawashima S, Nakaya A. The KEGG databases at GenomeNet. Nucleic Acids Res. 2002 Jan 1;30(1):42-6.
[15] Kanehisa M, Goto S, Kawashima S, Okuno Y, Hattori M. The KEGG resource for deciphering the genome. Nucleic Acids Res. 2004 Jan 1;32(Database issue):D277-80.
[16] Cantor CR, Jukes TH. The repetition of homologous sequences in the polypeptide chains of certain cytochromes and globins. Proc Natl Acad Sci U S A. 1966 Jul;56(1):177-84.
[17] Tatusov RL, Koonin EV, Lipman DJ. A genomic perspective on protein families. Science. 1997 Oct 24;278(5338):631-7.
[18] Alexeyenko A, Tamas I, Liu G, Sonnhammer EL. Automatic clustering of orthologs and inparalogs shared by multiple proteomes. Bioinformatics. 2006 Jul 15;22(14):e9-15.
[19] Prosdocimi F, Ortega JM. The codon usage of Leucine, Serine and Arginine reveals evolutionary stability of protein-coding genes. (Submitted to BSB2007)

The codon usage of Leucine, Serine and Arginine reveals evolutionary stability of proteomes and protein-coding genes

Francisco Prosdocimi[1], J. Miguel Ortega[1,*]
[1] Laboratório de Biodados, Dept Bioquímica e Imunologia, ICB-UFMG, Brazil
{franc, miguel}@icb.ufmg.br

Running-head: Evolutionary stability of eukaryotic proteomes

Abstract. DNA substitution mutation rounds applied in poli$_{100}$-codon sequences allowed us to verify which codons are more prone to be replaced by synonymous than non-synonymous ones. Evolving DNA sequences *in silico*, we have defined a DL50 index meaning how many substitution mutations a particular poli-codon sequence can resist before 50% of the translated sequence being mutated to another amino acid. Lower this index, higher the chance of a codon to be replaced by a non-synonymous one. Therefore, in order to keep evolutionarily stable one might expect that proteins will be selected to use synonymous codons with higher amino acid persistence values. Evolutionary stability of KEGG protein-coding genes from 9 eukaryotes were analyzed in order to find whether natural selection have acted on genes to prefer the utilization of more evolutionarily stable synonymous codons or not. A bias to the use of persistent codons was more conspicuous in complex metazoa.

Keywords: genetic code, natural selection, codon usage, evolutionary stability, synonymous codons

1 Introduction

In the evolution of protein-coding genes, their sequences have always being subjected to random DNA mutations. Although DNA consists in the informational cell repository, proteins are the main cell activity effectors and, therefore, natural selection acts on proteins as an effect of DNA mutations. If we consider the Darwinism happening at the molecular level, a DNA mutation producing a non-functional protein will probably be negative selected and the individual bearing this mutation will die or produce less viable offspring.

The way on which DNA mutations is related to the sequence and function of proteins is expressed by the genetic code. Two main theories are discussed about origin and evolution of genetic code and they try to explain why some codons codify for specific amino acids. The first theory suggests that genetic code has been built to

reflect the biosynthetic pathway on which amino acids are produced in biological organisms [1-3]. According to this theory, amino acids presenting similar biochemical synthetic pathways would be codified by similar codons [4]. The alternative theory states that the main force to shape genetic code is the selection for error minimization and similar codons would codify chemical similar amino acids, avoiding proteins to be damaged by DNA mutations [5-7]. The discussion about the main forces to build the genetic code is still in process [8, 9] and new evidences keep being discovered about this topic [10-12]. The present study also tries to bring light for this discussion.

Codons codifying the same amino acid are known as synonymous codons. It has already been proved that synonymous codons differ on their capacity to reduce the effects of mutation errors [12, 13]. So, in the course of evolutionary process, a DNA sequence of a protein-coding gene accumulate mutations and some codons are easily replaced by non-synonymous ones, putatively changing the protein structure, while some others are easily replaced to codons encoding exactly the same amino acid. So, the impact of mutation errors in a protein-coding DNA sequence depends on the codons used to codify amino acids in proteins.

Here, we have evaluated the codon usage of eukaryotic protein-coding genes to check their preference for the utilization of synonymous persistent codons, the ones on which the mutation to a synonymous codon is higher than the others. Analyzing each group of synonymous codons, we have developed a metric (DL50) to calculate the distance of each codon to their synonymous codons. Higher the DL50, higher the chance of this codon to be replaced by a synonymous one and higher the evolutionary stability of a given protein. Protein sequences less susceptible to DNA mutations are said to be more stable along the evolutionary process.

We have analyzed the evolutionary stability of 9 eukaryotic proteomes downloaded from KEGG database [14, 15]. A set of random DNA protein-coding sequences was also produced and compared to the overall data for actual organisms. Moreover, considering a given protein, we were able to build a synonymous protein presenting the worst and the best codon configurations in order to, respectively, allow or avoid non-synonymous mutations. Actual and random protein-coding sequences were compared to these best and worst evolutionary stable proteins and we have observed that amino acid usage of actual protein-coding genes are slightly shifted to the production of instable proteins, although the presence of highly stable sequences was also verified.

2 Methods

2.1. Evolution of poli-codonic sequences and the DL50 index

We have produced 64 multi FASTA files (one for each codon) containing 10,000 identical poli-codon sequences of 100 codons each. For example, we have produced an “CAG.fasta” file containing 10,000 sequences, each of them composed of 100 CAG codons disposed in tandem. Similar files were produced for each other of the 63 codons (see Prosdocimi and Ortega, 2007b [16] for a better description).

Each one of 10,000 poli$_{100}$-codon sequences was subject to a series of substitution mutations rounds on its DNA sequence. We did not produce insertions and deletions and we have evolved the sequences based on a Jukes and Cantor-like molecular evolutionary model [17]. This model was based on a series of stepwise mutation rounds and each round was characterized for the mutation of one single base. The base mutated was chosen randomly and, consequently, it has presented 25% chances to be mutated to any other base, including itself. Each poli$_{100}$-codon sequence was evolved until its translated protein has presented 50% identity with the original poli-amino acid translated sequence (non-mutated).

The evolving simulated sequences present 100 codons and the number of mutational steps leading to this 50% modification of original protein was considered the mutational Death Lethal dose capable to transform 50% of codons in a given protein into non-synonymous codons. This index was called DL50 amino acid. Codons presenting higher DL50 were considered evolutionary stable, since they require more time (measured in number of mutational events) to be replaced by a non-synonymous codon. Empirical values of DL50 were manually smoothed to avoid statistical artifacts produced by random mutations.

2.2. Synonymous Codon Usage analysis

After discovering which synonymous codons were more stable to keep a protein protected from non-synonymous substitutions, we have calculated the codon usage percentage for each amino acid relevant for evolutionary stability. This calculation was done considering the entire proteome of analyzed organisms (see below). The codon usage percentage was calculated inside a group of synonymous codons, e. g., we have calculated the percentage of TTA codons used among all Leucines of a given proteome. Therefore, Synonymous Codon Usage (SCU) was calculated as:

$$SCU_{codon} = \frac{N_{codon}}{N_{AA}} * 100 \qquad (1)$$

where N_{codon} is the number of times a given codon (or couple of codons) is used in the proteins of an organism and N_{AA} is the number of times the amino acid codified by this codon is used.

2.3. Amino acid Codon Usage Efficiency

The Codon Usage Efficiency (*CUE*) considering evolutionary stability of synonymous codons was measured as a point in the way from the worst codon usage configuration possible to the best one. The worst codon usage configuration was considered the one on which the codons used for a given amino acid were the ones more prone to be changed by non-synonymous codons (the ones with lower DL50). Contrarily, the best codon usage configuration was considered when an organism has shown to use more codons that have shown to be closer to synonymous ones, avoiding amino acid mutation. So, in order to calculate the *CUE* of an amino acid, we

have first multiplied the codon usage of all their synonymous codons by their DL50 observed value (SCU_{codon} * $DL50_{codon}$). Moreover, we have also calculated the worst and the best codon usage configuration of each amino acid. The worst codon usage configuration of an amino acid would happen if all the codons codifying it in a given proteome were the synonymous ones with lower DL50. On the other hand, the best evolutionary stable codon usage configuration of an amino acid will be the one where all codons codifying it were the ones presenting the higher DL50 value.

$$CUE_{AA} = \frac{\sum(SCU_{codon} * DL50_{codon}) - (100 * DL50Worst_{AA})}{(100 * DL50Best_{AA}) - (100 * DL50Worst_{AA})} * 100 \qquad (2)$$

where CUE_{AA} represents the efficiency of an amino acid in the way from the worst codon configuration to the best one, considering preferential mutations to synonymous codons

Whether an amino acid presents the best possible configuration of stable codons, it will receive a CUE_{AA} index of 100, while the worst configuration will be scored as 0. Intermediate values will represent where the usage of a given amino acid is in the way from the worst codon configuration possible to the best one.

2.4. Protein average DL50 attachment value

In order to calculate the average evolutionary stability of a single protein, we have applied the DL50 index to each codon codifying it. The codon indexes were then summed and averaged considering the protein size. So, each protein will present an Average Protein Persistence index (1) as follows:

$$APP = \frac{\sum DL50_{codon}}{N} \qquad (3)$$

where N is the total number of amino acids of a given protein

2.5. Evolutionary stability tests on actual protein-coding genes

Data for nine complete genome eukaryotic organisms were downloaded from KEGG Genes database [14]. KEGG was chosen since it is a curated database and it presents only the coding part of genes (CDS), such as we want to study. Table 1 shows the organisms analyzed in this work and the number of protein-coding genes downloaded from the databases.

The SCU_{codon} and CUE_{AA} index of non innocuous evolutionary amino acids regarding synonymous substitutions were calculated for proteomes. A set of 10,000 random generated DNA sequences were produced to function as control and to be compared with actual protein data. Proportion of codons in these synthetic random set has respected the proportion of the genetic code (e.g. six Ls to each M).

Table 1. Organisms and number of protein-coding genes used in this work

Organism	Mnemonic	Number of KEGG protein-coding genes
Arabidopsis thaliana	ath	26,735
Caenorhabditis elegans	cel	20,056
Canis familiaris	cfa	19,779
Drosophila melanogaster	dme	14,064
Homo sapiens	hsa	25,694
Mus musculus	mmu	30,032
Rattus norvegicus	rno	26,227
Saccharomyces cerevisiae	sce	5,863
Schizosaccharomyces pombe	spo	5,043

3 Results

3.1. DL50 persistence value by codon

We have applied stepwise DNA mutation rounds in a set of 640,000 nucleotide sequences, being each sequence composed of 100 identical codons; 10,000 of such 100 base sequences were assayed for each of the 64 codons. For each mutation round, one single nucleotide had the chance to mutate for each other nucleotide, including itself, at 25% chance. So, these $poli_{100}$-codons sequences were evolved until they were 50% different in protein similarity to their original sequence. As soon as, e. g., 50 of the 100 codons of a poli-TTT gets mutated to codify an amino acid different of Phenylalanine, this evolution was stopped and the number of mutational steps necessary to achieve this 50% sequence mutation was counted and averaged for each 10,000 sequences produced for each codon. This average smoothed value of mutational steps was called DL50. Table 2 shows, for each codon, its amino acid codified, smoothed values of DL50 and the class of persistence it belongs.

Table 2 shows that codons containing C as first base have proven to be, in average, the ones presenting the higher persistence value. When G is the first base of the codon, all DL50 are equal 149, except when A is the second base. The values present a higher standard deviation when T is the first base. All codons with A as second base but the stop codon TAA presents a DL50 of 111. When a poli-codon sequence present C as the second base, it needs 149 mutations in order to chance 50% of their codifying amino acids, except if the codon is either TCT_{Ser} or TCC_{Ser}, which DL50 is equal 153. Codons presenting G or T as second letter contain more diverse persistence values than A or C ones. Moreover, excluding the codons codifying for one single amino acid (ATG and TGG), in all other boxes when we look the genetic code, the persistence values are the same when the last base of codon is a pair A-G or T-C.

Table 2. Average mutational distance to synonymous codons and classes of DL50 by codon

Codon	AA[a]	DL50[b]	DL50 class[e]	Codon	AA[a]	DL50[b]	DL50 class[e]
TTT	F	111	2	ATT	I	128	3
TTC	F	111	2	ATC	I	128	3
TTA	L_3	136	3	ATA	I	128	3
TTG	L_3	136	3	ATG	M	97	1
TCT	S_5	153	5	ACT	T	149	4
TCC	S_5	153	5	ACC	T	149	4
TCA	S_4	149	4	ACA	T	149	4
TCG	S_4	149	4	ACG	T	149	4
TAT	Y	111	2	AAT	N	111	2
TAC	Y	111	2	AAC	N	111	2
TAA	*	128	3	AAA	K	111	2
TAG	*	111	2	AAG	K	111	2
TGT	C	111	2	AGT	S_2	111	2
TGC	C	111	2	AGC	S_2	111	2
TGA	*	111	2	AGA	R_3	136	3
TGG	W	97	1	AGG	R_3	136	3
CTT	L_6	156	6	GTT	V	149	4
CTC	L_6	156	6	GTC	V	149	4
CTA	L_7	178	7	GTA	V	149	4
CTG	L_7	178	7	GTG	V	149	4
CCT	P	149	4	GCT	A	149	4
CCC	P	149	4	GCC	A	149	4
CCA	P	149	4	GCA	A	149	4
CCG	P	149	4	GCG	A	149	4
CAT	H	111	2	GAT	D	111	2
CAC	H	111	2	GAC	D	111	2
CAA	Q	111	2	GAA	E	111	2
CAG	Q	111	2	GAG	E	111	2
CGT	R_6	156	6	GGT	G	149	4
CGC	R_6	156	6	GGC	G	149	4
CGA	R_7	178	7	GGA	G	149	4
CGG	R_7	178	7	GGG	G	149	4

a. Amino acid codified by codon; the number associated represent the different DL50 classes a single amino acid can belong; b. Smoothed average number of mutations necessary to produce 50% of amino acid mutations in a given poli100-codon sequence (DL50); c. Codon classes by DL50.

As we see in Table 2, almost all amino acids (except R, L and S) present synonymous codons with the same smoothed value of DL50. The presence of these amino acids in a given protein does not modify its evolutionary stability, since one is not able to produce neither a closer nor a farther version of this protein if one modifies its codon usage. So, these amino acids may be seen as *evolutionarily innocuous*. Otherwise, Arginine, Leucine and Serine present codons with different

DL50 values and they are the main responsible to make proteins more or less stable against DNA mutations. These three hexacodonic amino acids present three distinct two-codon classes of DL50 values for their synonymous codons, where Leucine and Arginine codons present persistence classes of 3, 6 and 7, while Serine codons are less stable and they were classified in the stability classes 2, 4 and 5.

3.2. Synonymous Codon Usage of hexacodonic amino acids

Since the unique amino acids relevant for protein stability along evolution has proven to be the hexacodonic ones (the ones codified by six codons) and since we are considering just synonymous proteins instead of conservative amino acid changes between proteins, we have now analyzed the *SCU* of these amino acids in the organisms studied. Moreover, the present *SCU* analysis was made based on the pairs of codons presenting the same DL50 (Table 2). Codon usage in randomly produced DNA sequences was also taken on account as a matter of comparison. Best and worst possible *SCU* values are also presented (Table 3).

Table 3. Synonymous Codon Usage (*SCU*) of hexacodonic amino acids coupled DL50 codons

Codon	AA	sce	spo	ath	cel	dme	cfa	mmu	rno	hsa	RAND	best	worst
AGT/AGC	S	27,4	26,0	28,7	25,4	38,7	39,2	39,0	39,2	38,9	33,3	0	100
TCA/TCG	S	31,0	28,5	31,0	40,5	29,3	20,4	19,9	19,9	20,9	33,4	0	0
TCT/TCC	S	41,7	45,4	40,3	34,0	32,0	40,4	41,1	40,8	40,2	33,3	100	0
TTA/TTG	L	55,9	51,4	36,6	35,1	23,6	20,2	20,6	20,4	20,5	33,3	0	100
CTT/CTC	L	18,7	32,8	42,1	41,1	25,5	32,9	33,4	33,3	33,2	33,2	0	0
CTA/CTG	L	25,4	15,8	21,3	23,7	50,9	46,9	45,9	46,3	46,3	33,4	100	0
AGA/AGG	R	68,9	34,3	55,9	37,4	22,5	40,4	45,7	45,2	41,5	33,3	0	100
CGT/CGC	R	20,0	42,9	23,1	30,6	46,3	27,2	24,8	24,9	27,1	33,4	0	0
CGA/CGG	R	11,1	22,9	21,0	32,0	31,2	32,5	29,5	29,9	31,4	33,3	100	0

To complement Table 3 data, Table 4 shows the codon usage efficiency by amino acid (CUE_{AA}) regarding mutational resistance on each analyzed organism.

Table 4. CUE_{AA} indices for hexacodonic amino acids and proteomes show the efficiency on codon usage regarding the evolutionary stability of synonymous codons.

AA	sce	spo	ath	cel	dme	cfa	mmu	rno	hsa	RAND	best	worst
S	69,7	71,2	68,4	70,7	58,5	58,9	59,2	58,9	59,1	63,5	100	0
L	34,3	31,4	41,4	43,3	63,0	62,6	61,8	62,1	62,1	49,2	100	0
R	20,6	43,3	32,0	46,6	53,2	45,4	41,3	41,8	44,3	49,2	100	0
Proteome	41,5	48,7	47,2	53,5	58,2	55,6	54,1	54,3	55,2	54,0	100	0

Table 3 and 4 show clearly the situation of codon usage evolutionary stability for each organism and amino acid analyzed. It is interesting to realize different usage pattern considering each amino acid from simple to complex organisms. Serine codon usage presents a higher evolutionary stability in non-Metazoa clades (*sce*, *spo* and *ath*) and Pseudocoelomata (*cel*). Leucine has presented higher *CUE* in Coelomata

clade and Arginine presents atypical stability efficiency between taxa, showing to be higher in *dme*.

Moreover, Serine *CUE* when a random nucleotide pattern is used has shown to be higher than Leucine and Arginine indices in Table 4 and it has happened due to the DL50 evolutionary stability scores of their codons. While Serine present codons with DL50 of 111, 149 and 153; Leucine and Arginine codons present DL50 values ranging 136, 156 and 178 mutation rounds. Since 149 is closer to 153 than 156 to 178, the average random evolutionary stability has shown to be higher.

3.3. Evolutionary Stability indices measured on KEGG proteins

The Average Protein Persistence index (*APP*) was calculated for each protein and their distribution by organism can be seen in Figure 1.

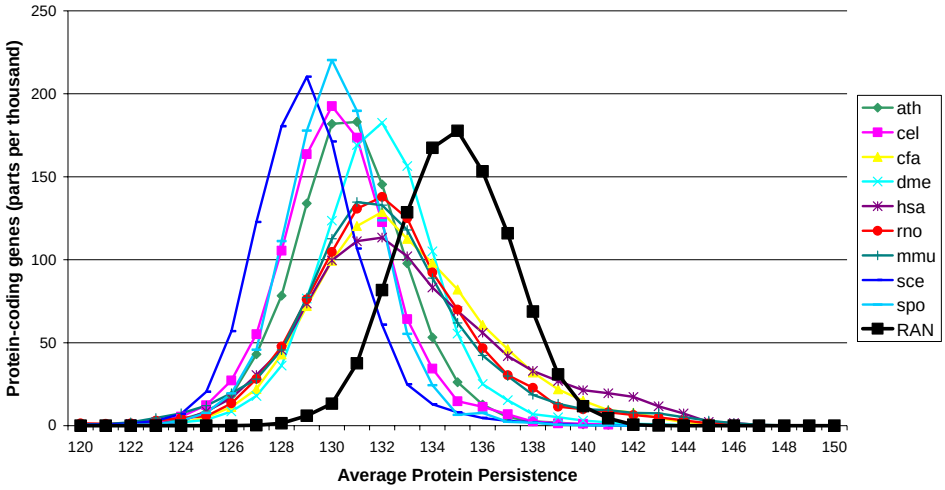


Fig. 1. Number of genes (per thousand) presenting specific *APP* values. The *APP* value represents the average number of mutations a protein can resist per 100 codons without being 50% mutated at protein level.

As we see in Figure 1, the data produced based on mutations in random nucleotide sequences obeys a normal curve. Data for actual proteins are also similar among them. When compared to random analysis, curves for actual proteins have shown to be slightly shifted in direction to a lower *APP* and present a pronounced tail into the right side direction. Moreover, it is also possible to realize a pattern concerning organisms' complexity: as the organism turns more complex, it presents higher diversity of *APP* values (a more wide-shaped curve) and bigger tails into the direction of higher protein codon usage persistence.

4 Discussion

Here we have analyzed the differential persistence of synonymous codons submitted to random DNA mutations along evolution. An initial simulation allowed us to define DL50 persistence values for each codon and those values were then used to calculate the evolutionary stability of eukaryotic proteomes and protein-coding genes.

Although in the course of evolution proteins frequently modify their amino acidic composition, here we prefer to consider proteins as entities evolving without amino acid replacements. This decision was done based in the fact that many matrices exists considering many different aspects of amino acid substitution, such like residue conservation [18, 19], amino acid chemical properties [20-22], frequency of amino acid contacts in protein structures [23], residue volume [22], hydropathy [24], frequency of dipeptides [25] and many other characteristics [26]. Therefore, in order to avoid the choice of a specific matrix, we have preferred to characterize the evolution of proteins analyzing first just the completely conservative substitutions between synonymous codons.

We have proven that the codon usage of hexacodonic amino acids hides a clue about how natural selection operates to keep proteins more evolutionary stable and resistant to mutations. So, using the *APP* index (Figure 1), we were able to evaluated the strength of natural selection operating in a given protein, while analyzing CUE_{AA} (Table 4) this index for all proteome, we are able to have a glimpse about the overall proteins from the eukaryotic organisms studied.

Cel data (Table 3, 4 and Figure 1) have shown to be more similar to non-Metazoa organisms' than to Metazoa ones. Other evidences have already been shown confirming this data [27] and it may be purposed that the great modifications occurred during Metazoa evolution have happened after the acquisition of coeloma by the Coelomate organisms.

Although presenting lower evolutionary stability than random data, actual organisms' data have shown to present a tail reaching the right side of *APP* distribution. The proteins presenting these high *APP* values were probably directed selected to resist in conditions of high DNA mutation rates. We are currently producing a database to be released presenting highly stable proteins of eukaryotic organisms.

The observation that *APP* is bigger when considering random data shows (1) that the amino acid usage in actual proteins is not random and (2) it brings to a lower evolutionary stability than the one suggested by randomly picking codons in the genetic code. So, it is possible to suppose that the codon to amino acid attribution in the genetic might be very efficient when the code was built, but when organisms have evolved and proteins became more complex, maybe it has not shown to be that efficient any more. This last observation may be taken with caution, but there is evidence confirming it [10] and further studies must be made to elucidate better this subject.

Another interesting hypothesis is the supposition that natural selection have selected proteins in the other way from evolutionary stability. So, it is possible to suppose that instable evolutionary proteins would be preferentially selected than stable ones, since they show a more plastic and adaptive DNA sequence. These

evolving instable proteins, even if they cause deaths in many individuals due to malfunction, they would be able to accumulate amino acid modifications that might bring an improvement in the protein function along time. Therefore, evolutionary instable proteins may produce better organism's or species' adaptation to a continuous modifying environment. And, so, considering the environment seems more variable for an unicellular and simple organisms than to complex ones, this would explain why non-Metazoa clades present more instable proteins and also why Metazoa codons have being replaced from instable to stable ones.

The evolutionary stability and mutational resistance study on protein-coding genes has proven to be a very interesting discipline and it presents a new paradigm on bioinformatics, where genome analysis change its way from a descriptive initiate into a truly scientific effort. More than studying just the evolutionary stability of synonymous codons, we are presently developing strategies to bring together this data with the similarity of codons to stop codons [16] and also trying to consider some aspects of amino acid replacement. Moreover, we are currently obtaining data for a higher number of organisms, trying to derive clade specific patterns of protein stability along evolution. At least, new models of molecular evolution (such like Kimura two parameters and gamma distribution) are currently being simulated in order to verify better the adequacy of them into actual organisms' proteins.

References

[1] Wong JT. A co-evolution theory of the genetic code. Proc Natl Acad Sci U S A. 1975 May;72(5):1909-12.
[2] Di Giulio M. The extension reached by the minimization of the polarity distances during the evolution of the genetic code. J Mol Evol. 1989 Oct;29(4):288-93.
[3] Di Giulio M. The coevolution theory of the origin of the genetic code. J Mol Evol. 1999 Mar;48(3):253-5.
[4] Di Giulio M. Genetic code origin and the strength of natural selection. J Theor Biol. 2000 Aug 21;205(4):659-61.
[5] Woese CR. On the evolution of the genetic code. Proc Natl Acad Sci U S A. 1965 Dec;54(6):1546-52.
[6] Epstein CJ. Role of the amino-acid "code" and of selection for conformation in the evolution of proteins. Nature. 1966 Apr 2;210(5031):25-8.
[7] Haig D, Hurst LD. A quantitative measure of error minimization in the genetic code. J Mol Evol. 1991 Nov;33(5):412-7. Erratum in: J Mol Evol 1999 Nov;49(5):708.
[8] Freeland SJ, Knight RD, Landweber LF, Hurst LD. Early fixation of an optimal genetic code. Mol Biol Evol. 2000 Apr;17(4):511-8.
[9] Di Giulio M. The origin of the genetic code cannot be studied using measurements based on the PAM matrix because this matrix reflects the code itself, making any such analyses tautologous. J Theor Biol. 2001 Jan 21;208(2):141-4.
[10] Archetti M. Codon usage bias and mutation constraints reduce the level of error minimization of the genetic code. J Mol Evol. 2004 Aug;59(2):258-66.
[11] Archetti M. Selection on codon usage for error minimization at the protein level. J Mol Evol. 2004 Sep;59(3):400-15.
[12] Archetti M. Genetic robustness and selection at the protein level for synonymous codons. J Evol Biol. 2006 Mar;19(2):353-65.

[13] Hershberg U, Shlomchik MJ. Differences in potential for amino acid change after mutation reveals distinct strategies for kappa and lambda light-chain variation. Proc Natl Acad Sci U S A. 2006 Oct 24;103(43):15963-8.
[14] Kanehisa M, Goto S, Kawashima S, Nakaya A. The KEGG databases at GenomeNet. Nucleic Acids Res. 2002 Jan 1;30(1):42-6.
[15] Kanehisa M, Goto S, Kawashima S, Okuno Y, Hattori M. The KEGG resource for deciphering the genome. Nucleic Acids Res. 2004 Jan 1;32(Database issue):D277-80.
[16] Prosdocimi F, Ortega JM. About a preference for stop-resistant codons in eukaryotic protein-coding genes. (Submitted to BSB2007)
[17] Cantor CR, Jukes TH. The repetition of homologous sequences in the polypetide chains of certain cytochromes and globins. Proc Natl Acad Sci U S A. 1966 Jul;56(1):177-84.
[18] Henikoff S, Henikoff JG. Amino acid substitution matrices. Adv Protein Chem. 2000;54:73-97.
[19] Tyagi M, Gowri VS, Srinivasan N, de Brevern AG, Offmann B. A substitution matrix for structural alphabet based on structural alignment of homologous proteins and its applications. Proteins. 2006 Oct 1;65(1):32-9.
[20] McLachlan AD. Tests for comparing related amino-acid sequences. Cytochrome c and cytochrome c 551 . J Mol Biol. 1971 Oct 28;61(2):409-24.
[21] Grantham R. Amino acid difference formula to help explain protein evolution. Science. 1974 Sep 6;185(4154):862-4.
[22] Goodarzi H, Katanforoush A, Torabi N, Najafabadi HS. Solvent accessibility, residue charge and residue volume, the three ingredients of a robust amino acid substitution matrix. J Theor Biol. 2006 Dec 19; [Epub ahead of print]
[23] Miyazawa S, Jernigan RL. A new substitution matrix for protein sequence searches based on contact frequencies in protein structures. Protein Eng. 1993 Apr;6(3):267-78.
[24] Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol. 1982 May 5;157(1):105-32.
[25] Gonnet GH, Cohen MA, Benner SA. Analysis of amino acid substitution during divergent evolution: the 400 by 400 dipeptide substitution matrix. Biochem Biophys Res Commun. 1994 Mar 15;199(2):489-96.
[26] Rice DW, Eisenberg D. A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. J Mol Biol. 1997 Apr 11;267(4):1026-38.
[27] Prosdocimi F, Mudado MA, Ortega JM. A set of amino acids found to occur more frequently in human and fly than in plant and yeast proteomes consists of non-essential amino acids. Comput Biol Med. 2007 Feb;37(2):159-65.

Finding Normalizers Genes by Means of Homology Searches on Expressed Sequence Tags and Oligonucleotide Array Data

{Track: computational biology / bioinformatics. Category: full paper}

Saulo Pinto[1, 2] and J. Miguel Ortega[1*]

[1] Laboratório de Biodados, Departamento de Bioquímica e Imunologia, Instituto de Ciências Biológicas, Universidade Federal de Minas Gerais, Brazil.
[2] Instituto de Informática, Pontifícia Universidade Católica de Minas Gerais, Brasil.
{saulo@pucminas.br, miguel@icb.ufmg.br}

Abstract. The decision about which gene to use as a reference in gene expression analysis is a difficult one. A reference gene is used mostly as a normalizer of gene expression levels and hence it has to have stable expression levels through a wide range of tissues, conditions and, to be universal, organisms. Here, we present and test a methodology suited to find such normalizers genes. Our results confirm that some well known housekeeping genes are not good normalizers in many tissues whereas other ones are, and point to the direction of showing that there is no such universal normalizer gene.

Keywords: endogenous reference genes, normalizer gene, EST, KOG, oligonucleotide array.

1 Introduction

Normalization of gene expression data by the expression of a control or reference gene (a *normalizer*) is a widely used technique in Molecular Biology. For example, in RT-PCR assays normalization is fundamental to correct sample-to-sample variation and to provide a reliable way to compare gene expression levels [6]. If the normalizer is present in the organism's genome it is called endogenous, otherwise it is exogenous. In some cases is not viable to use an exogenous normalizer, for instance, in the construction of cDNA libraries or in PCR-based analysis. In those cases an endogenous normalizer should be used. It's important to observe that an endogenous control gene is preferred since its expression is influenced by the same manipulations of the other genes being studied [4]. However, beyond to be endogenous, a normalizer gene must present some features in order to be a reliable one.

Traditionally, *housekeeping* (or maintenance) genes are used to normalize expression data. Housekeeping genes are those constitutively expressed in order to keep cellular function [12]. Such genes must be expressed in a large number of tissues [11, 12], experimental conditions [6], developmental stages [7, 12], and to be universal,

* Corresponding author.

different organisms. However, a couple of studies have pointed that some of those commonly used normalizer genes underwent very significant changes in their expression in many experimental conditions [4, 6, 7] which can lead to incorrect data interpretation. So, in order to be a well suited universal normalizer, a gene must present two main attributes: (1) keep its expression constant or at least in low, predictable or known variation (2) through the vast majority of tissues, conditions, and organisms. However, previous works point that there is no such universal normalizer.

Nowadays, the detection of the two main characteristics of normalizers is being performed through standalone RT-PCR [3, 4, 7] and in conjunction with high-density oligonucleotide expression arrays [6, 12]. Here, we present a methodology that utilize ESTs data from the NCBI's KOG database [10], NCBI's GEO oligonucleotide arrays data series [1], and homology searches in a statistical analysis to detect normalizers genes, considering an extra point that is almost always not considered in studies of differential gene expression: the number of differently expressed genes detected when normalization by a certain gene is applied.

2 Materials and Methods

2.1 Data

Expressed Sequence Tags. Data was obtained from an additional work from our group: K-EST[1] (KOG Expression/Sampling Tool [5] and data to be published elsewhere). Briefly EST expression data for four model organisms [*A. thaliana* (ath), *C. elegans* (cel), *D. melanogaster* (dme) and *H. sapiens* (hsa)] were obtained through BLAST searches, being analyzed 360833 ESTs from *A. thaliana*, 293530 from *C. elegans*, 370672 from *D. melanogaster* and 360398 from *H. sapiens* which were grouped into 4852 KOGs, 1021 TWOGs and 4181 LSEs and used in the *R* statistics calculation [8] — look at the Appendix for *R* calculation specifics. In order to apply the *R* statistics to these data each organism's dataset was considered to be a library and the KOG's expression behavior was investigated over organisms.

Oligonucleotide Array. Data from the GSE2361 GEO series were downloaded and processed. The series comes from an experiment built on the Affymetrix HG-U133A oligonucleotide array platform and comprises 36 samples representing the expression of 36 normal human tissues: heart, thymus, spleen, ovary, kidney, skeletal muscle, pancreas, prostate, small intestine, colon, placenta, bladder, breast, uterus, thyroid, skin, salivary gland, testis, trachea, adrenal gland, bone marrow, cerebellum, amygdala, caudate nucleus, corpus callosus, hippocampus, thalamus, pituitary gland, spinal cord, brain, fetal brain, liver, fetal liver, stomach, lung, fetal lung as described by [2]. The Affymetrix MAS 5.0 software calculates a *signal* value to characterize the expression level of each sequence represented in the array. The signal value is calcu-

[1] http://www.biotec.icb.ufmg.br/K-EST_novo

lated and so a *call* to indicate whether the corresponding sequence is confidently present (P), absent or not reliable (A) or marginally detected (M).

2.2 Finding Candidates

The finding of KOG candidates to be normalizers was started by applying the *R* statistics [8] to the EST data in the following way:

```
for each KOG, k, in the dataset do
  normalize the dataset using k as the normalizer;
  sort the data set according to R values in
  decreasing order to build a ranking;
  for N = 1, 10, 50, 100 do
    Calculate the arithmetic mean of the N greatest
    R values in the ranking;
```

It's important to remark that the premise explored here is: candidates that "produce" the smaller maximum *R* values when used as normalizers are better in differential expression analysis. Since genes with greatest *R* are the most differentially expressed over a set of libraries [8], a normalizer that produces a large number of genes with big *R* values is worse than one that produces a small number of such genes. This feature is important in projects where differentially expressed genes must be further investigated by costly techniques — look at the appendix for *R* calculation specifics.

After that, a *variation rate* (*vr*) was calculated for each successive pair of KOGs (k_j and k_{j+1}, where k_j has greater *R* (mean *R* if $N > 1$) value than k_{j+1}) in the rankings using the Formula 1. The topmost KOGs were selected scanning down the ranking until *vr* values got stable (with no significant variation (< 0.01) from two successive KOGs).

$$vr_j = \frac{\left(R_{k_{j+1}} - R_{k_j}\right)}{R_{k_j}} \,. \qquad (1)$$

After choosing candidates, all FASTA sequences that compose each KOG were retrieved from a lab's local database used by K-EST. BLAST searches were performed against the FASTA of every full sequence (not only the probe sets sequences) represented in the oligonucleotide array. These sequences were downloaded from the Affymetrix website[2] in order to form new groups of sequences representing, each one, a KOG in the GeneChip data in the subsequent steps. Only alignments that reached at least 80% in similarity and score equals or greater than 100 (as reported by the blastall program) were considered for the inclusion of a sequence in the group representing a KOG in the GeneChip data analysis.

[2] http://www.affymetrix.com/products/arrays/specific/hgu133.affx

2.2 Oligonucleotide Array Data Analysis

As detailed above, in order to study the behavior of each candidate KOG and confirm its election as a normalizer, GeneChip sequences that aligned to the KOG's original sequences were grouped together to represent KOGs in the GeneChip and the data for the GSE2361 series was analyzed. For each KOG the signal of all its group sequences in each array sample that was flagged as present (P) by the Affymetrix MAS 5.0 software was summated and the arithmetic mean calculated to be the representative value of the KOG in each of the 36 tissues composing the series. For each KOG the coefficient of variation and the fold (maximum divided by the minimum value) were separately calculated for the whole set of 36 tissues and for seven subsets: the CNS (Central Nervous System) tissues, non-CNS tissues, fetal tissues (lung, liver, brain), fetal and adult counterpart tissues, non-fetal non-CNS tissues, male/female-specific tissues (prostate, testis and uterus, breast, colon, placenta, ovary, respectively).

3 Results

3.1 Candidates KOGs

Table 1 shows the candidate KOGs chosen by the application of methodology presented here. KOG's identifiers, descriptions and their expressions in the four model organisms according to K-EST and the maximum *R* values are listed. Table 2 presents a portion of the rankings generated for the values of *N* used to find the candidates. The four rankings are very alike for the topmost nine KOGs. Since in the $N = 1$ ranking the *vr* dropped below the fifth KOG, it was decided to include KOGs in the other rankings too (following the same criteria exposed in the Methods section). Apart from that, the examination of results in Table 2 suggests that using only the sorting of the greatest *R* value ($N = 1$) is well suited to select candidates and the use of others *N* greatest *R* values should not introduce improvement nor distortion in the results, but this decision could exclude important candidates as the other following results show.

Table 1. The nine KOGs that are candidates to be normalizers and their (EST) expression in the four model organism.

KOG	Description	Occurrences per 100K ESTs				
		Max R	ath	cel	dme	hsa
KOG0052	Translation elongation factor EF-1 alpha/Tu	1175.66	1545	3104	1881	3382
KOG1376	Alpha tubulin	3111.17	579	608	1478	1278
KOG0676	Actin and related proteins	3549.65	802	1226	337	2492
KOG1375	Beta tubulin	4303.11	742	554	698	924
KOG0657	Glyceraldehyde 3-phosphate dehydrogenase	4542.37	1901	353	527	1125
KOG0019	Molecular chaperone (HSP90 fa-	5387.64	317	800	500	738

	mily)					
KOG0857	60s ribosomal protein L10	5439.23	376	322	352	731
KOG0001	Ubiquitin and ubiquitin-like proteins	5690.34	1224	503	202	765
KOG0815	60s acidic ribosomal protein P0	6076.47	472	270	603	712

The results using the methodology agree to the lists of genes commonly used as references in expression analyses as reported in the literature [3, 5, 11, 12]. Excepting to translation elongation factor (KOG0052), the *R* value didn't vary more than two fold for the eight other candidates in the Table 1. This is important since the genes represented by the KOG0052 are usually reported as highly expressed [4] and in the dataset utilized here it is too (it's mean is 68% greater than the second largest) as presented in the Table 1. So, the *R* statistics shows to be not much dependent of the rate of expression of the genes used as normalizers, but sensitive to the expression variation over libraries.

Table 2. The top of rankings generated in order to find the candidates presented in Table 1.

N = 1	*vr*	**N = 10**	*vr*	**N = 50**	*vr*	**N = 100**	*vr*
KOG0052	2.25	KOG0052	13.29	KOG0052	1.76	KOG0052	2.22
KOG1376	0.14	KOG1376	0.19	KOG1376	0.20	KOG1376	0.23
KOG0676	0.21	KOG1375	0.02	KOG0676	0.02	KOG1375	0.03
KOG1375	0.06	KOG0676	0.08	KOG1375	0.05	KOG0676	0.02
KOG0657	0.19	KOG0019	0.14	KOG0019	0.05	KOG0019	0.06
KOG0019	0.01	KOG0657	0.01	KOG0001	0.20	KOG0001	0.19
KOG0857	0.05	KOG0001	0.23	KOG0657	0.14	KOG0657	0.16
KOG0001	0.07	KOG0857	0.09	KOG0857	0.07	KOG0857	0.06
KOG0815	0.01	KOG0815	0.02	KOG0815	0.01	KOG0815	0.02

3.2 Oligonucleotide Array Confirmation

KOGs in the GeneChip. Table 3 presents the groups of sequences found in the array for each candidate KOG through the homology searches. A "pipe" character (vertical bar) in the genes' title represented by the sequences in the chip means that the titles were put together in the same row for space reasons. The same is true for gene symbols containing parentheses. For instance, "*ubiquitin B | C | D*" and "*UB(B, C, D)*" are shorts for the common names "Ubiquitin B", "Ubiquitin C", "Ubiquitin D" and symbols UBB, UBC, UBD, respectively.

Overall performance. In order to analyze the stability of the candidate KOG's groups, the tissues were separated in subsets as presented in Table 4: all 36 tissues in the GeneChip series (*All*), CNS-specific (*CNS*), not belonging to CNS (*non-CNS*), fetal (*Fetal*), non-fetal not belonging to CNS (*Non-Fetal non-CNS*), fetal and non-fetal

counterparts (*Fetal and non-Fetal*), male-specific (*Male*), and female-specific (*Female*). Values in italics highlight the smallest fold in a subset (in a table column) and values bold-faced highlight the smallest fold for a KOG (in a table row) through tissues subsets. If there's a column or a row with more than one "smallest" value every one is highlighted.

Table 3. The groups of gene sequences in the GeneChip that were found for each KOG.

KOG Id	Sequence Gene Title(s)	Gene Symbol(s)
kog0001	ubiquitin-like 4	UBL4
	ubiquitin B | C | D	UB(B,C, D)
	2'-5'-oligoadenylate synthetase-like	OASL
	interferon, alpha-inducible protein (clone IFI-15K)	G1P2
	ubiquitin A-52 residue ribosomal protein fusion product 1	UBA52
	ribosomal protein S27a	RPS27A
	Finkel-Biskis-Reilly murine sarcoma virus (FBR-MuSV) ubiquitously expressed (fox derived); ribosomal protein S30	FAU
kog0052	eukaryotic translation elongation factor 1 alpha 1 | alpha 2	EEF1(A1, A2)
kog0676	ARP1 actin-related protein 1 homolog A, centractin α (yeast)	ACTR1A
	actin like protein	LOC81569
	actin-related protein 10 homolog (S. cerevisiae)	ACTR10
	actin, gamma 1	ACTG1
	actin, beta pseudogene 9	ACTBP9
	calcitonin gene-related peptide-receptor component protein	RCP9
	actin-like 7A | 7B	ACTL(7A, 7B)
	actin, alpha 1, skeletal muscle	ACTA1
	actin, alpha 2, smooth muscle, aorta	ACTA2
	actin, alpha, cardiac muscle	ACTC
	actin, gamma 2, smooth muscle, enteric	ACTG2
	actin-like 7B	ACTL7B
	actin, beta	ACTB
Kog0019	heat shock 90kDa protein 1, alpha | beta	HSPC(A, B)
	heat shock protein 75	TRAP1
Kog0657	glyceraldehyde-3-phosphate dehydrogenase	GAPD
	glyceraldehyde-3-phosphate dehydrogenase, spermatogenic	GAPDS
Kog0815	ribosomal protein, large, P0	RPLP0
	chromosome 20 open reading frame 41	C20orf41
Kog1376	tubulin, alpha, ubiquitous	K-ALPHA-1
	tubulin, alpha 1 (testis specific)	TUBA1
	NDRG family member 3	NDRG3
	NDRG family member 3	NDRG3
	tubulin, alpha 2 | alpha 3 | alpha 8	TUBA(2, 3, 8)
Kog0857	ribosomal protein L10 | L10-like	RPL10(L)
Kog1375	tubulin, beta polypeptide 4, member Q	TUBB4Q
	tubulin, beta 1 | 2 | 4	TUB(B1, B2, B4)

	tubulin beta MGC4083	MGC4083
	tubulin, beta polypeptide	TUBB
	beta 5-tubulin	OK/SW-cl.56

It's remarkable that the Ubiquitin KOG (*KOG0001*) is the most stable over all tissues and over five of eight sets. In fact it is among the three most stable in every subset, except to the fetal one, indicating that the genes composing this KOG's group should be considered the best suited as references in the broader sense of spanning the maximum number of tissues. It has the best (smallest) fold value over all sets analyzed (Male, 1.08), but this value must be carefully considered because only two tissues compose that subset and almost all KOGs have low values in it. On the other side, the β-tubulin KOG (*KOG1375*) demonstrated to be the worst (or the second worst) over seven subsets getting not too bad only in the male-specific tissues. Beyond general remarks about the behavior of genes or groups of genes in a broad range of different specialized tissues is very important to researchers working with specifics kinds of tissues to know about specific normalizers.

Table 4. The folds of the KOGs selected by the methodology over tissues subsets.

KOG Id	**All**	**CNS**	**non-CNS**	**Fetal**	**Non-Fetal non-CNS**	**Fetal and non-Fetal**	**Male**	**Female**
Kog0001	*2.12*	*1.74*	2.12	1.79	*1.74*	*1.79*	***1.08***	1.32
Kog0052	2.44	2.13	*2.03*	1.35	2.03	1.93	1.28	***1.18***
Kog0676	2.67	2.31	2.67	**1.49**	2.55	1.93	1.79	1.56
Kog0019	4.34	1.75	4.34	2.04	4.15	2.30	**1.11**	2.62
Kog0657	4.56	2.02	4.56	***1.15***	4.56	2.94	1.61	2.06
Kog0815	7.33	5.43	3.35	**1.23**	3.35	3.84	2.01	**1.23**
Kog1376	6.02	3.71	3.81	1.97	3.58	6.02	**1.18**	1.78
Kog0857	9.44	6.00	6.54	**1.40**	6.54	3.50	3.19	2.02
Kog1375	17.33	14.46	6.07	4.07	3.69	10.17	**1.26**	2.46

Specific tissues subsets performance. Considering the ten CNS-specific tissues only (Figure 1 and Table 4), Ubiquitin (KOG0001) (the best in agreement to [12]) and HSP90 chaperons (KOG0019) show up in stability and GAPDH (KOG0657) as being highly expressed and very stable too (except in the cerebellum and whole and fetal brain where its high expression degraded its stability shown in the other more specific CNS tissues ranging from amygdala to spinal cord). In fact, the stability and high expression pattern of GAPDH KOG is present in fetal tissues, which include fetal brain, as shown in Figure 2.

Three fetal tissues were analyzed (Figure 2 and Table 4): brain, liver, and lung. The GAPDH KOG was the most stable together with two KOGs whose components are involved in the cell's translational machinery: the eukaryote elongation factors (EEF, KOG0052) and the ribosomal protein P0 (KOG0815). These three are by far the most expressed groups in the fetal tissues. Since its well known role in energy production pathways and its now accepted roles in DNA replication, RNA exportation

from cell nucleus and cytoskeletal organization [9] it is expected for GAPDH to be highly expressed in fetal tissues where the metabolism is accelerated compared to adulthood one. It's very interesting that only KOG0052 stay between the three most stable when the non-fetal tissues are analyzed together with fetal ones (Figure 3).

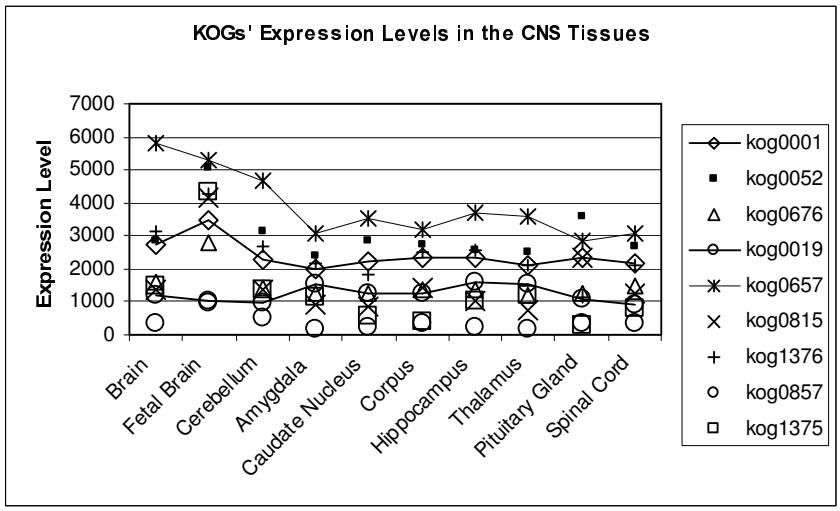


Fig. 1. KOGs expression levels in the GeneChip for the CNS tissues only. The three best KOGs [*KOG001 (Ubiquiin family), KOG0019 (HSP90 family), and KOG0657 (GAPDH)*] are shown as continuous lines. Except for the expression level magnitude all eight KOGs have roughly similar patterns of expression overall.

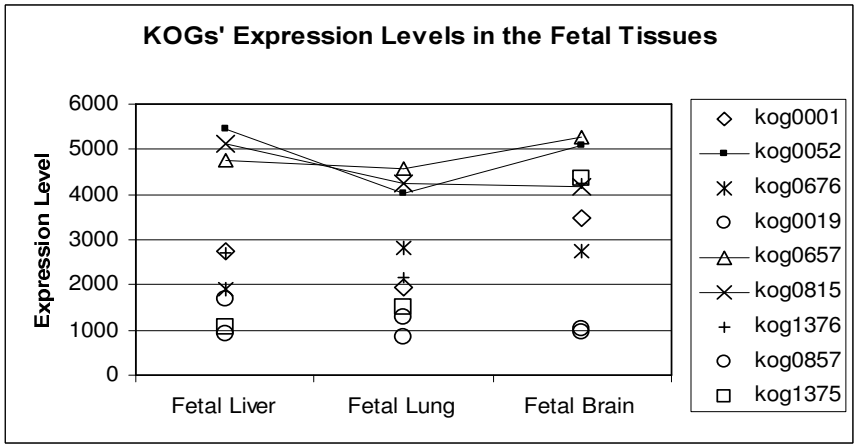


Fig. 2. Fetal tissues expression considering the eight KOGs. *Ubiquiin family* (*KOG0001*), eukaryotic *translation elongation factor* (*KOG0052*), and *the 60s acidic ribosomal protein P0 (KOG0815), and KOG0657 (GAPDH)*] are shown using continuous lines since they are the three best stable.

Looking not only to fetal tissues, but examining their adulthood counterparts (Figure 3 and Table 4), another view can be taken. It's remarkable that almost every KOG expression level shrink from fetal to adulthood tissues with some exceptions. The Ubiquitin KOG gets back as the least varying followed by the EEF and the Actin KOGs. However, those stabilities are a kind of fake. For example, the Ubiquitin has its expression diminished from lung to fetal lung and increased from brain to fetal brain. This balance did the fold and the coefficient of variation (not shown) to be the smallest overall but do not mean stability or an expected behavior at all.

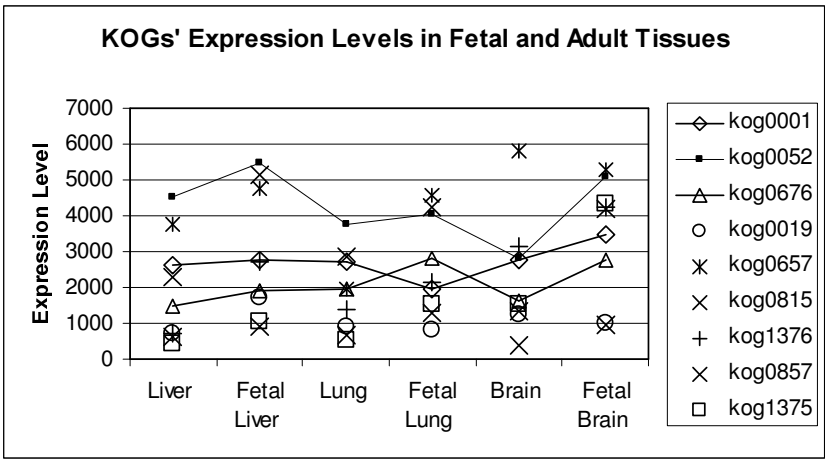


Fig. 3. Comparison between fetal tissues and their adulthood counterparts. It's notable the general behavior: expression levels are higher in fetal than in non-fetal tissues pairs, exceptions due to *GAPDH* (*KOG0657*) and RPLP0 (*KOG0815*) whose expression level is higher in the brain than in fetal brain and *Ubiquitin* (KOG0001) that drops down from lung to fetal lung.

Considering only female-specific tissues (Figure 4 and Table 4) the three most stable are the three most expressed (together with the highly expressed GAPDH KOG in all tissues but breast): Ubiquitin (KOG0001), EEF (KOG0052), and RPLP0 (KOG0815). Its impressive the stability shown by EEF. Its coefficient of variation is only 6.8% that is the smallest in all analyzed sets (discarding the small Male subset), suggesting its components as the normalizers when the expression of this kind of tissues have to be studied.

Considering the two male-specific tissues (prostate and testis) the striking difference to the other subsets is the stability of KOGs for α-tubulin (KOG1376) and Heat Shock protein (KOG0019) — Figure 5 and Table 4. Another notable feature is the higher expression level of almost every KOG in the prostate. Only Ubiquitin and β-tubulin did not present such behavior.

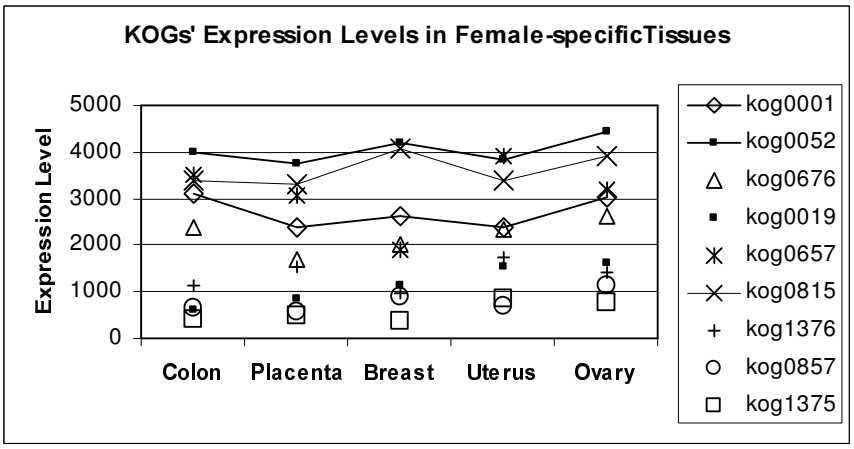


Fig. 4. Female-specific expression levels of the nine KOGs' groups. The KOG for Eukaryote Elongation Factor (*KOG0052*) is the most stable.

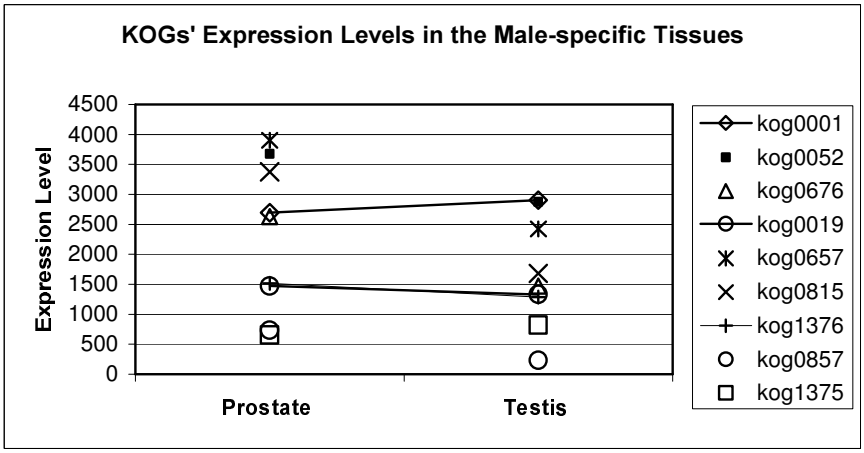


Fig. 5. The behavior of the KOGs in the two male-specific tissues analyzed (*KOG0019* and *KOG1376* lines are almost overlapped). Remarkable is the general trend of the expression levels being smaller in the *Testis* than in the *Prostate*.

3 Conclusions

This work presented results of an ongoing effort to implement a straightforward method to find the most stable genes through the utilization of different kinds of data

produced by different technologies and stored as SAGE and cDNA libraries and Genechips series. The results of the tissues subsets' analysis suggest that there is no "universal" normalizer gene in the sense of one that is stable through a large number of different kinds of tissues, in agreement with past results [11, 12]. For instance, the EEF KOG was indicated as the most stable through the four model organisms, but in the human tissues it was not that good. It is, in a general look, worse than Ubiquitin. But the results show that some genes are very stable in some sets of tissues. For example, the EEF and GAPDH KOGs components are very stable in the human female and fetal tissues, respectively; whereas the Ubiquitin KOG in the human CNS and male tissues.

The work presented here is essentially a bioinformatician one. Although it might occur that our conclusion is apparent as long as EST and GeneChip data described here are used, the list of genes found by our methodology seems to agree with the empirically chosen and, most important, between both sources of data investigated here. Other comparisons with other data from different technologies such as SAGE and cDNA libraries and more accurate statistical validation is on the way.

4 Direction and Future Works

A simple methodology to study and find expression level-stable genes was presented here. The results are intermediate in the sense that other microarray series for the four model organisms are being analyzed at the moment. Besides, SAGE (Serial Analysis of Gene Expression) data are been fed in a database and will be included in the subsequent analysis. Another analysis to be included is to analyze the KOGs data from K-EST considering the subsets of tissues as have been done to the GeneChip data in order to get another validation of the election of normalizers genes.

References

1. (Barret *et al.* 2006) Barrett, T., Troup, D.B., Wilhite, S.E. , Ledoux, P. , Dmitry Rudnev , Carlos Evangelista , Irene F. Kim , Alexandra Soboleva , Maxim Tomashevsky , and Ron Edgar : NCBI GEO: mining tens of millions of expression profiles—database and tools update. Nucl. Acids Res. (2006) 35: D760-D765.
2. Ge, X., Yamamoto, S., Tsutsumi, S., Midorikawa, Y., Ihara, S., Wang, S.M., Aburatani, H.: Interpreting expression profiles of cancers by genome-wide survey of breadth of expression in normal tissues. Genomics 86 (2005) 127-141.
3. Goossens K., van Poucke, M., van Soom, A., Vandesompele, J., van Zeveren, A., Peelman, L.J.: Selection of reference genes for quantitative real-time PCR in bovine preimplantation embryos. BMC Developmental Biology (2005), 5:27.
4. Hamalainen, H.K., Tubman, J.C., Vikman, S., Kyrola, T., Ylikoski, E., Warrington, J.A., Lahesmaa, R.: "Identification and validation of endogenous reference genes for expression profiling of T helper cell differentiation by quantitative real-time RT-PCR.", Anal Biochem. Dec 1;299(1):63-70, 2001.
5. Mudado, M.A. & Ortega, J.M.: "A picture of gene sampling/expression in model organisms using ESTs and KOG proteins", *Genet. Mol. Res.* 5 (1): 242-253, 2006.

6. Pohjanvirta, R., Niittynen, M., Linden, J., Boutros, P.C., Moffat, I.D., Okey, A.B., "Evaluation of various housekeeping genes for their applicability for normalization of mRNA expression in dioxin-treated rats", Chemico-Biological Interactions Volume 160, Issue 2 , 134-149, 2006.
7. Robert, C., McGraw, S., Massicote, L, Pravetoni, M., Gandolfi, F., Sirard, M.: Quantification of Housekeeping Transcript Levels During the Development of Bovine Preimplantation Embryos. Biology of Reproduction 67, 1465-1472 (2002).
8. Stekel, D.J., Git, Y., Falciani, F.: "The Comparison of Gene Expression from Multiple cDNA Libraries", *Gen. Res.* 10:2055-2061, 2000.
9. Tatton et al. 2000 Tatton, W.G., Chalmers-Redman, R.M., Elstner, M., Leesch, W., Jagodzinski, F.B., Stupak, D.P., Sugrue, M.M., Tatton, N.A.: Glyceraldehyde-3-phosphate dehydrogenase in neurodegeneration and apoptosis signaling. J. Neural Transm. Suppl. (2000) (60):77-100.
10. 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, 41, 2003.
11. Vandesompele, J., De Preter, K., Pattyn, F., Poppe, B., Van Roy, N., De Paepe, A., Speleman, F.: "Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes.", Genome Biol. Jun 18;3(7), 2002.
12. Warrington, J. A., Nair, A., Mahadevappa, M., Tsyganskaya, M.: "Comparison of human adult and fetal expression and identification of 535 housekeeping/maintenance genes", Physiol Genomics, 2:143-147, 2000.

Appendix: The *R* Statistics

In differential gene expression analysis, a set of genes whose expression shows large variation over a set of libraries is searched for. In such kind of analysis a measure must be used to compare the expression of a gene in libraries being studied. Stekel *et al.* 2000 [8] describe the derivation of the *R* statistics (Equation 1) that is suited to evaluate differential expression over any number of libraries at once.

$$R_j = \sum_{j=1}^{m} x_{i,j} \log\left(\frac{x_{i,j}}{N_i f_i}\right). \qquad (2)$$

In the above formula, f_i is the global frequency of the gene j over all libraries i. N_i is total abundance for library I, and $x_{i,j}$ is the abundance of gene j in the library i.
Here we considered the set of all clusters of orthologous groups for eukaryotic genomes (KOGs) in four model organisms: *A. thaliana*, *C. elegans*, *D. melanogaster*, and *H. sapiens*, that can be searched using K-EST, to apply the *R* statistics. In order to do so, we treated each organism's dataset as a different library and each KOG as being a gene. Hence, we calculated the *R* values for each KOG over the four libraries (organisms).

On The Improvement of Transcriptome Annotation After Clustering and Assemblage of Incremental Number of ESTs

Maurício A. Mudado and J. Miguel Ortega

Laboratório de Biodados, Departamento de Bioquímica e Imunologia, Instituto de Ciências Biológicas, UFMG, Av. Antônio Carlos 6627, Belo Horizonte - MG, Brasil
{mudado,miguel}@icb.ufmg.br

Abstract. EST clustering is a widely used procedure in transcriptome projects and is a common sense to improve annotation. In this work we demonstrate a method to test the BLAST annotation of four Model Organism's ESTs and its uniques assembled with the TGICL assemblage software, with the KOG database. Increased numbers of ESTs were used and results show that clustering is reduced by using 5K ESTs but approaches saturation around 80%, by using over 50K ESTs. Compared to non-clustered ESTs, annotation of assembled ESTs shows better results and improves as increased number of ESTs is used. Compared to non-clustered ESTs, results for *C. elegans* and *D. melanogaster* show an increment in the annotation, by diminishing no hit annotations (around 0.8 fold) and raising correct annotation by around (1.2 fold) in both organisms. Thus, a 20% improvement of correct annotation is attained by EST clustering and assemblage

Keywords: Transcriptome, EST, Clustering, TGICL, BLAST, annotation, KOG

1 Introduction

The clustering of a transcriptome is a widely used procedure, initiated by the construction of the Human Unigene [1]. In Unigene, single-pass partial cDNA sequences also known as Expressed Sequence Tag or EST [2] are compared to each other and to available cDNA sequences with the program MegaBLAST, a greedy version [3] of BLAST software [4] developed to increase the speed up the clustering procedure. Other initiatives relied on the use of the assemblage software Cap3 [5] to simultaneously cluster and assemble clustered sequences into consensus sequences also known as contigs. Besides not being designed for clustering, the main difficulty inherent from the use of Cap3 for processing large EST collections is the intense use of memory. Some researchers used to break transcriptomes in reasonable samples to generate contigs and later submit their contigs to subsequent rounds of assemblage with Cap3 [6]. TIGR has produced Transcript Index (TI) for several organisms using such approach, but its bioinformatics team has later developed a software package known as TGICL Tool, which contains a initial step that allows for cluster generation

in a similar manner to the Unigene procedure, based on MegaBLAST comparisons of the EST sequences, and in a second stage by running Cap3 on each cluster, producing contigs and singlets (non clustered ESTs), which constitutes the uniques or transcript index (TI) sequences [7].

A common sense in the literature is that the assemblage of individual EST sequences into contigs shall improve annotation, due to the fact that the larger the sequence, higher the score in BLAST comparisons to public available databases. However, no exhaustive experimental investigation has been conducted on this issue. Our group has developed an approach that suits to this demand. The procedure makes use of KOG database, in which proteins from diverse Model Organisms are clustered into groups of orthologs and paralogs [8], In a first round, ESTs are assigned to the cognate organism proteins, providing with a positive control for the annotation and, in a second round, already assigned ESTs are annotated by KOG entries from other organisms, and the annotation is compared to the initial assignment step. Thus, resultant annotation can lay in three categories: correct, changed and speculated. The last occurs when a EST is not assigned to the cognate organism KOG entry, but the database speculate an annotation for it, by aligning it to a KOG entry from other organism. Together with these three categories for annotated ESTs there are also ESTs in the "no hit" category, which can be either too short to provide a hit of alternatively representing genes not present in KOG dataset, and "assigned but no hit" during annotation procedure, which might concentrate genes that are specific to the analyzed organism (e.g. *A. thaliana*, the only plant in KOG database).

Here we present a test of TIGCL clustering and assemblage of incremental number of ESTs from *C. elegans* and *D. melanogaster* and tests of annotation with KOG database. Assemblage was sensible to the input number of ESTs (reduced with 5K but saturating by 150K ESTs). We confirm that the generation of contigs improved annotation without adding potential errors that could result from chimerical assemblage of distinct genes. This effect, as calculated in terms of total ESTs analyzed, led to a very small increase in changed annotation (up to 1.08%) for 150K. Moreover, an important bias on clustering and assemblage of genes that are prompted to correct annotation drives the apparent result that correct annotation is poorer (0.71 fold) if uniques are annotated as opposite to individual ESTs (around 1.2 fold). Furthermore, similar results have been obtained with the *A. thaliana* and *H. sapiens* ESTs.

2 Methods

2.1 Sequences

Large sets of ESTs from four model organisms were downloaded from GenBank at the NCBI web site (http://www.ncbi.nlm.nig.gov): 360,833 for *Arabdopsis thaliana* (Ath); 302,080 for *Caenorhabditis elegans* (Cel); 375,360 for *Drosophila melanogaster* (Dme) and 365,619 for *Homo sapiens* (Hsa). ESTs were filtered for health tissues and organs.

The KOG database was filtered for the 88,613 classified KTL proteins from seven Model Organisms, found in the "kog", "twog" and "lse" files at (ftp://ftp.ncbi.nih.gov/pub/COG/KOG/). A MySQL database was populated with this data and used to select the respective fasta sequences from the "kyva" file ate the same site. The KTL proteins were divided into 60,758 KOG, 4,451 TWOG and 23,404 LSE proteins.

2.2 BLASTs

BLAST software (version 2.2.13) was used in the alignment of uniques against KTL proteins. tBLASTn was used with the following parameters:
-m 8 -b 1000000 -e 1e-10 -F f. These parameters activate the tabular output of BLAST, allowing up to 1 million hits to one protein (default is 250) and deactivates the low-complexity filter, respectively. The low-complexity filter was deactivated in order to allow tBLASTn to achieve 100% identity in the alignments.

2.3 Clustering

The software TGICL (http://compbio.dfci.harvard.edu/tgi) was used to cluster the ESTs and generate uniques. TGICL was run with the following parameters: –p 95 –l 40 –v 30. These parameters put together sequences which overlap with at least 95% similarity, at least 40 bp identical and 30 bp distance from overlap to sequence end. Also, TGICL script was changed to include the following parameters to run the tclust software: SCOV=70 PID=95. These parameters force building of high stringency clusters with at least 95% of identity and 70% coverage of the shorter sequence. The PERL package Math::Random (http://www.cpan.org) was used in order to select random subsets of ESTs for clustering.

3 Results and Discussion

3.1 Clustering randomly selected ESTs

ESTs were randomly selected in incremental sets of 5K, 10K, 50K, 100K and 150K, with the Math::Random PERL package. TGICL was run with these subsets in order to know if assemblage of ESTs saturates and how many ESTs are needed in order to achieve a clustering plateau. As seen in Fig.1 the percentage of ESTs in clusters (non-singlets) raises exponentially from ~40% to ~70% when using 5K to 50K ESTs for Cel, Dme and Ath. Clustering then stalls to 80% when using more than 100K ESTs. This result proves that assemblage is dependent of the number of ESTs used for clustering with TGICL. *H. sapiens* had a much lower clustering percentage compared to the other organisms ESTs, probably because of greater number of 3'-5' ESTs.

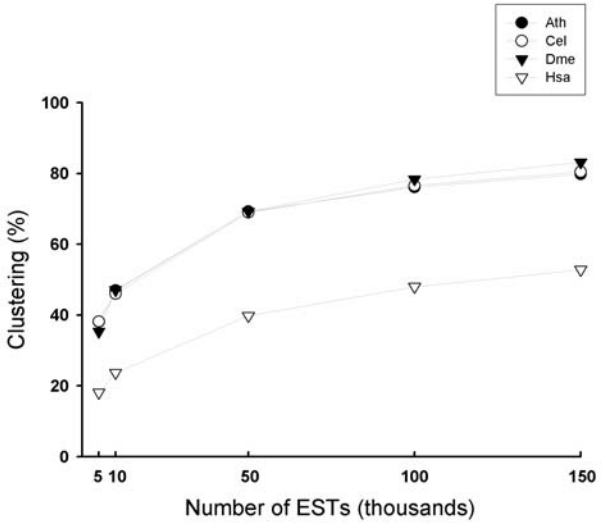


Fig. 1. Percentage of ESTs in clusters (*clustering*) of incrementing number of ESTs selected at random. *A. thaliana* (*Ath*), *C. elegans* (*Cel*), *D. melanogaster* (*Dme*) and *H. sapiens* (|*Hsa*) are shown (*full circles, open circles, full inverted triangles, open inverted triangles*).

3.2 Measuring the annotation quality of uniques

Uniques were then aligned with tBLASTn with the KTL proteins from KOG database (see methods). The annotation experiment for the uniques with the KOG database has two steps. First, uniques are assigned to its proper organism's KTL proteins by selecting the best hits from tBLASTn alignments (Fig.2, right side). Second, uniques are annotated by removing the proper organism's proteins from the database, aligning the uniques with the remaining six organism's proteins with tBLASTn (Fig.2 left side) and always selecting the best hits.

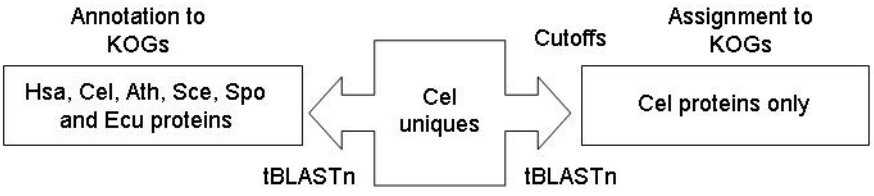


Fig. 2. Schema of the assignment and annotation of *C. elegans* uniques with KOG proteins. The uniques are assigned to Cel's own KOG proteins with the use of similarity cutoffs (*right side)* and annotated to all KOG proteins but Cel's KOG proteins (*left side)*. All organisms' uniques passed by the same pipeline.

Five types of annotation are allowed: correct, changed, speculated, 'assigned but no hit' and 'no hit'. When the assignment and the annotation of a unique are both to the same KOG ID, the annotation is correct and when they are not, the annotation is changed. When a unique annotates to a KOG ID but did not assign we say that the database is speculating an annotation. When a unique is assigned to a KOG ID but didn't annotate to any KOG ID, we say that it is 'assigned but no hit'. Finally, when there is nor assignment neither annotation, it is defined as 'no hit' (see Table 2).

Table 1. Types of annotation.

Type of Annotation	Assignment	Annotation	KOG ID
Correct	+	+	Same
Changed	+	+	Different
Speculated	-	+	Any
Assign. But no Hit	+	-	Any
No Hit	-	-	-

Fig. 3A and C shows a comparison of the quality of annotation of the uniques (white symbols), formed from subsets of 5K, 50K, 100K and 150K ESTs from Cel and Dme, to the direct annotation of the same set of non-clustered ESTs (no TGICL used) (black symbols). Fig. 3B and D shows results from a similar experiment, where the quality of annotation is shown by directly computation of the number of ESTs that are comprised by the uniques formed previously (white symbols). The same comparison against non-clustered ESTs is shown (black symbols).

As seen in Fig. 3 A through D, non-clustered ESTs (black symbols) tend to have the same pattern of annotation (almost linear) in all sets of ESTs compared to the annotation of uniques (white symbols). Non-clustered ESTs from Cel and Dme show ~44% and 41% of correct annotation and around 32% of no hit annotation and almost 1.5 % of changed annotation for both organisms in all sets of ESTs annotated. On the other hand, the result for uniques and the clustered ESTs comprised by these uniques, shows that incrementing the number of clustered ESTs leads to an input of novel information to the annotation process.

The annotation of uniques shows that no hit annotation raises 6% and 6.8% (up to 1.24 and 1.34 fold compared to non-clustered ESTs) and correct annotation diminishes 7% and 8% (up to 0.7 fold for both organisms, compared to non-clustered ESTs), for Cel and Dme respectively, by using up to 150 K ESTs (Fig. 3 A and C). The assigned but no hit annotation is also higher in uniques by 6.4% in Cel and 0.1% in Dme (1.2 and 1.0 fold respectively), compared to non-clustered ESTs in the same range of ESTs. However, the annotation of the ESTs comprised by these uniques (Fig. 3 B and D) depicts a different picture. Clustering is already effective by using 5K ESTs in both Cel and Dme (see small increment in correct annotation and diminishing of assigned but no hit and no hit annotations, compared to non-clustered ESTs). By using up to 150K ESTs there is an augment in correct annotation of 5.4% and 4.8% (up to 1.21 and 1.20 fold compared to non-clustered ESTs) and a diminishing of no hit annotation of 3.1% and 5.9% (up to 0.67 and 0.77 fold compared to non-clustered ESTs) for Cel and Dme respectively. The assigned but no hit annotation is also

diminished by 6.9% and 3% (0.78 and 0.87 fold) for Cel and Dme respectively, in the same range (150K ESTs). Changed annotation is only augmented by 1.08% and 0.89% for Cel and Dme respectively. Speculative annotation was unchanged, with values around 2% and was suppressed from the graphic. The difference in annotation between uniques and its ESTs (Fig. 3 A and C versus Fig. 3 B and D) is due to a numerical artifact. Although the number of correct annotated uniques is diminishing and no hit uniques are rising, the number of correct annotated ESTs comprised by these uniques is also rising at the same time. There is a lower number of contigs represented by a great number of ESTs annotated correctly, against a higher number of contigs represented by a fewer number of ESTs with no hit annotations. Results were similar to Ath and Hsa (data not shown, see supplementary in www.biodados.icb.ufmg.br).

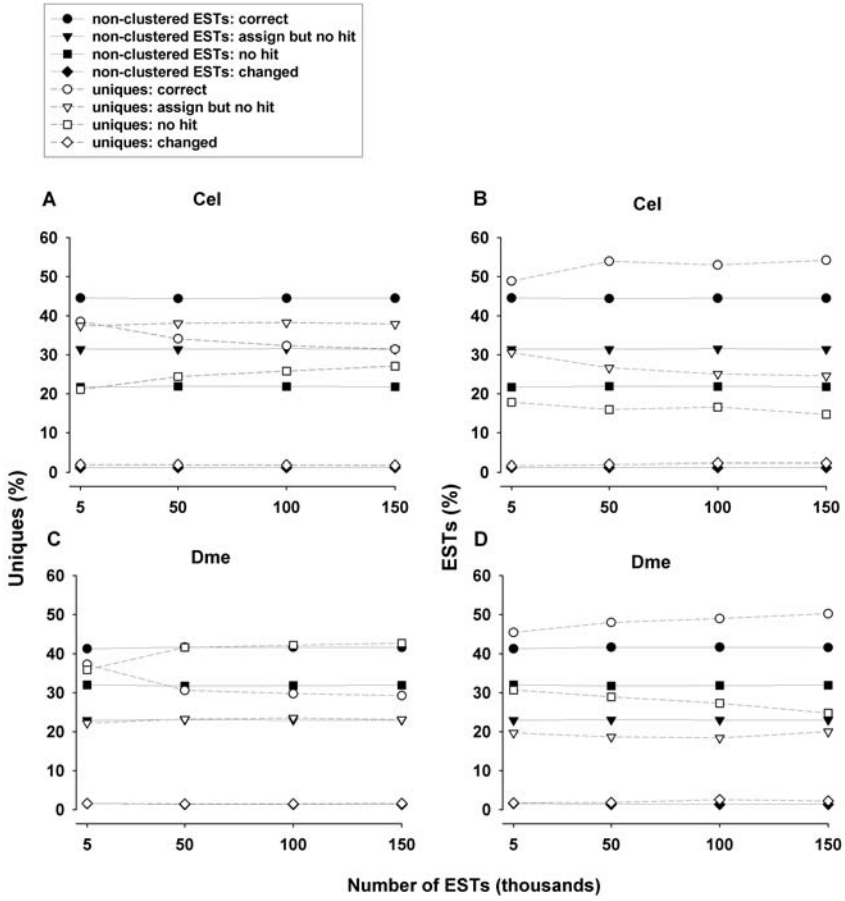


Fig. 3. Comparison of the annotation of uniques (*A and C*) and the ESTs (*B and D*) comprised by these uniques (*white symbols)* with non-clustered ESTs (*black symbols*). Increasing numbers of ESTs were used (*5K, 50K, 100K and 150K*). Annotation results are shown in percentage by symbols (*circle:correct; inverted triangle:assigned but no hit; square:no hit and lozenge:changed annotations*).

In conclusion, this work showed that EST clustering with the software TGICL approaches saturation at around 80% by using over 50K ESTs. Furthermore, we demonstrated a method to evaluate the annotation of uniques generated by TGICL and its ESTs with the KOG database. Results showed, by comparison with non-clustered ESTs, that clustering ESTs with TGICL improved annotation, by diminishing assigned but no hit and no hit annotations (from 0.67 to 0.87 fold) and rising correct annotated ESTs - around 1.2 fold - for both organisms. Thus, a 20% improvement of correct annotation is attained by EST clustering and assemblage. Changed annotation is only slightly augmented (up to 1.08% of all ESTs). Annotation had better improvement as the number of input ESTs to TGICL was increased up to 150K.

Although the clustering of ESTs into contigs is expected to yield a gain in accuracy, this issue has not yet been investigated since the proper positive control was not available. That was possible using KOG clusters. The evaluations presented here indicate that the clustering of ESTs improve the accuracy of annotation for the user as the project generates large amounts of ESTs, thus justifying the analysis of individual ESTs as they are generated if the goal is to produce short amount of data (e.g. under 10K ESTs). TGICL as well as Unigene approach, by clustering ESTs with a BLAST search prior to Cap3 assemblage, improves the scalability of the process, since only increased clusters are subject to novel rounds of Cap3 assemblage.

Acknowledgements
Research supported by CAPES, FAPEMIG and CNPq/MCT.

References

1. Wheeler, D. L., et al.: Database Resources of the National Center for Biotechnology. Nucl Acids Res 31 (2003) 28-33
2. Adams, M.D. *et al*. Complementary DNA sequencing: Expressed sequence tags and human genome project. Science. 252 (1991) 1651−1656
3. Zhang, Z., Schwartz, S., Wagner, L., Miller, W. J.: A greedy algorithm for aligning DNA sequences. Comput Biol. 7(2000) 203-14
4. Altschul, S. F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. 215 (1990) 403-410
5. Huang, X., Madan, A.: CAP3: A DNA sequence assembly program. Genome Res. 9 (1999) 868-77.
6. Liang, F., Holt, I., Pertea, G., Karamycheva, S., Salzberg, S. L., Quackenbush, J.: An optimized protocol for analysis of EST sequences. Nucleic Acids Res. 28 (2000) 3657–3665
7. Lee, Y., Tsai, J., Sunkara, S., Karamycheva, S., Pertea, G., Sultana, R., Antonescu, V., Chan, A., Cheung, F., Quackenbush, J.: The TIGR Gene Indices: clustering and assembling EST and known genes and integration with eukaryotic genomes. Nucleic Acids Res. 33 (2005) D71-4
8. Tatusov, R. L., Fedorova, N. D., Jackson, J. D., et al.: The COG database: an updated version includes eukaryotes. BMC Bioinformatics. 4 (2003) 41.

PHEIO, a Java/MySQL based phylogenetic editor for NCBI Taxonomy tree

Velloso H, Pena IA and Ortega JM

Laboratório de Biodados, Departamento de Bioquímica e Imunologia, Instituto de Ciências Biológicas, UFMG, Av. Antônio Carlos 6627, Belo Horizonte - MG, Brasil
hvmelo@gmail.com, bellinhaap@yahoo.com.br, miguel@icb.ufmg.br

Abstract. We have developed PHEIO (PHylogenetic EdItOr), a visual phylogenetic tree editor. PHEIO was developed using Java and uses MySQL as database management system. PHEIO loads NCBI taxonomy database into a local server and allows the user to create and edit taxonomic groups, furthermore, it turns taxonomy relationships into phylogenetic ones. PHEIO receives as input, files containing a list of taxon names and traces the path back to the root for each taxon, furthermore it creates and display, in a user-friendly interface, one graphical phylogenenic tree. When a taxon is selected in the active tree, related information is displayed, such as: scientific name, phylogenetic rank, taxonomy id, and the distance of a node to the tree root. PHEIO also provides a taxon search system. Bacterial evolutionary trees were used as a model. In conclusion, the user friendly interface combined with a powerful database allows one to apply phylogenetic relationships to automated evolutionary analysis

Keywords: *Phylogenetic editor*, phylogeny analisys, homology, taxonomy.

Introduction

Bioinformatics tools are ideally supposed to handle large analysis providing either qualified information at the end of the processing of information, but also, and maybe more important, a source of information designed to automate other tasks. In this sense, many databases are known as secondary ones, such as COG [1], Kegg orthology [2], OrthoMCL-DB [3], in which groups of homologous proteins are clustered. In the Taxonomy area, a remarkable effort has been done at NCBI to provide information for all taxa that have any sequence information in GenBank [4]. A database for Phylogenetic information, Treebase [5], is of great notoriety and represents a repository for phylogenetic studies, accessed under a web environment.

However, the implementation of a myriad of bioinformatics tools might relay on a database with taxonomy and phylogenetic information. For example, evolutionary analysis of genes currently depends on the selection of a root organism, closer to the nearest common ancestral. Taxonomy trees such as NCBI's may be able to provide information of other genus for the species that is under investigation, but the actual need for this task is a phylogenetic tree, thus, increasing the requirement for a phylogenetic editor.

There are some good phylogeny editor programs such as PhyloDraw which is a drawing tool for creating phylogenetic trees (http://pearl.cs.pusan.ac.kr/phylodraw/); Phyfi, who draws phylogenies using the Newick format representing trees in computer-readable and form makes use of the correspondence between trees and nested parentheses (http://cgi-www.daimi.au.dk/cgi-chili/phyfi/go).

However, a complete environment for the development of phylogenetic trees is still Kegg orthologylacking. Here we present PHEIO, a PHylogenetic EdItOr. PHEIO uses as input a MySQL local database imported from NCBI Taxonomy. A Java interface allows multiple curators to edit the Taxonomy tree, adding evolutionary information, turning into a Phylogenetic tree as output, which is stored in the database and can be queried for several complementary purposes.

Methods

PHEIO was developed entirely in Java 1.4.2 using as IDE Eclipse-SDK 3.2.1. PHEIO connects to a either local or remote MySQL 5.0.2 database using mysql-connector-java 5.0.4. The database was populated with information on taxonomic names and relationships. This information was entirely extracted from NCBI dump files (names.dmp and nodes.dmp) downloaded by FTP (ftp://ftp.ncbi.nih.gov/pub/taxonomy/). Based on these files, parent-child relationships were reconstructed into two tables: *name* and *node*. A third table called *user_group* was created to manage phylogenetic groups entered by the user. Using this information PHEIO can build a complete taxonomic tree using Java Swing's JTree, as below.

A graphical interface was built using Java's Swing API. Swing is a library of GUI (graphical user interface) controls that includes widgets such text boxes, buttons, split panes and trees. Swing supports pluggable look and feel – not by using the native platform's facilities, but by roughly emulating them. Thus, Swing grants uniform behavior on all platforms. For PHEIO we used mainly the following Swing components: JFrames, JButtons, JSplitPanes and JTrees.

PHEIO's graphical interface was built on a JFrame containing a JSplitPane. The upper section of the pane was reserved for a customized JTree to show the taxonomic and philogenetic relationships. The inferior section of the pane shows information about a taxon/node selected by the user.

Results

PHEIO starts showing a initial tree containing only the root node "all species". Afterwards, the user can insert nodes into the tree in two ways: (i) loading a file containing a list of taxon names or (ii) searching for specific taxa in the database. PHEIO then builds the customized JTree. The JTree will show only the nodes the user desires, so we refer to it as the "working tree". When the user loads a list of taxon names (figures 1 and 2), PHEIO searchs for these names in the *name* table and for each taxon found, PHEIO traces the path back until it reachs the working tree.

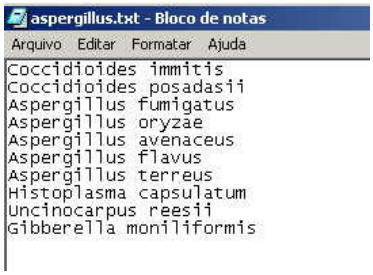


Fig. 1: list of a taxon names

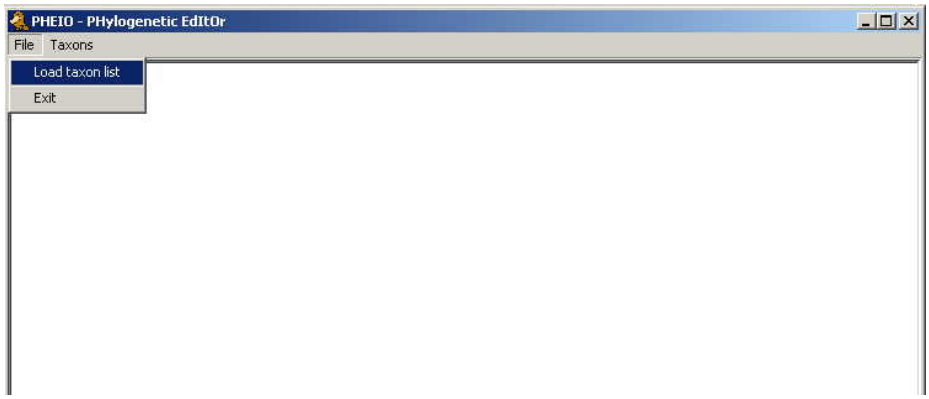


Fig. 2: loading taxon list on PHEIO

As a result, we have a tree containing all the taxa listed in the file, their common ancestors and their traces back to the root (figure 3).If a taxon listed in the file is not found in the database, PHEIO's log shows a warning message and this specific taxon is discarded. PHEIO also provides a taxon search dialog (figure 4) where the user can insert the name or part of the name of the taxon and the system will show all the taxons in the database which contain that text. Afterwards, the user choose a taxon from the resulting list to be inserted in the working tree.

When the user selects a node, PHEIO shows information about that node, such as scientific name, rank, taxId and the distance of that node to the root (degree of primitivity) (figure 3, bottom panel).

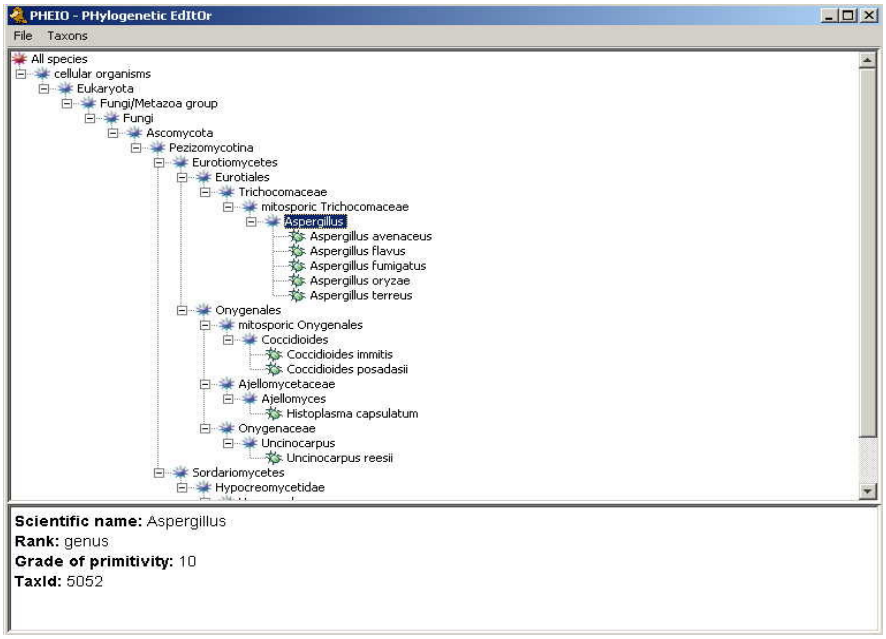


Fig. 3: PHEIO Taxonomy tree with listed species, their common ancestors information and about that node, such as scientific name, rank, taxId and the distance of that node to the root (Grade of primitivity).

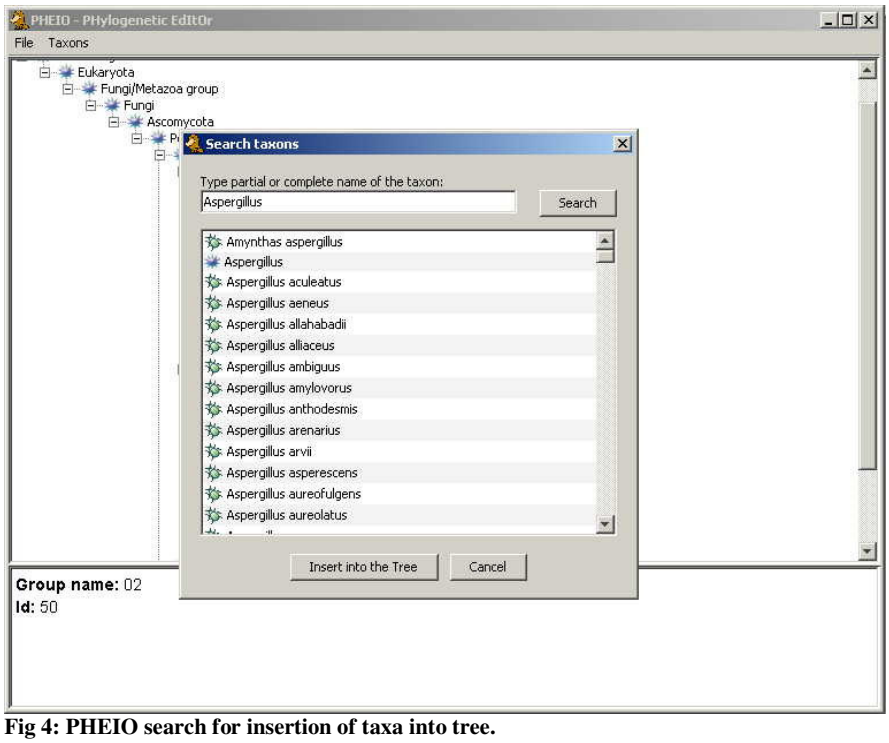


Fig 4: PHEIO search for insertion of taxa into tree.

The main purpose of PHEIO is to create phylogenetic relationships in NCBI taxonomic trees. This is possible through the possibility of creating and managing taxa groups. Thus, the user can create phylogenies in PHEIO based on Treebase and other phylogeny databases, as in the example below.

Figure 5 shows a taxonomic tree generated from a list of Pezizomycotina sub-phylum species.

Based on phylogenetic analysis from Genome, rRNA and other sources, the user creates a new group called "01" including *A. oryzae, A. terreus, A. flavus and A. avenaceus* (figures 5, 6 and 7).

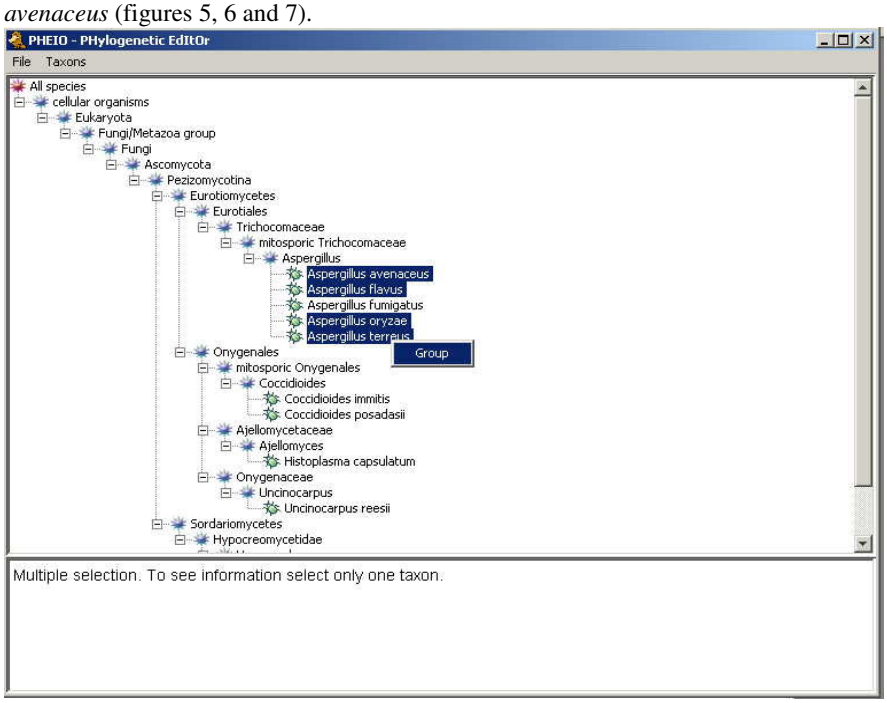


Fig. 5: Grouping *A. oryzae, A. terreus, A. flavus* and *A. avenaceus*.

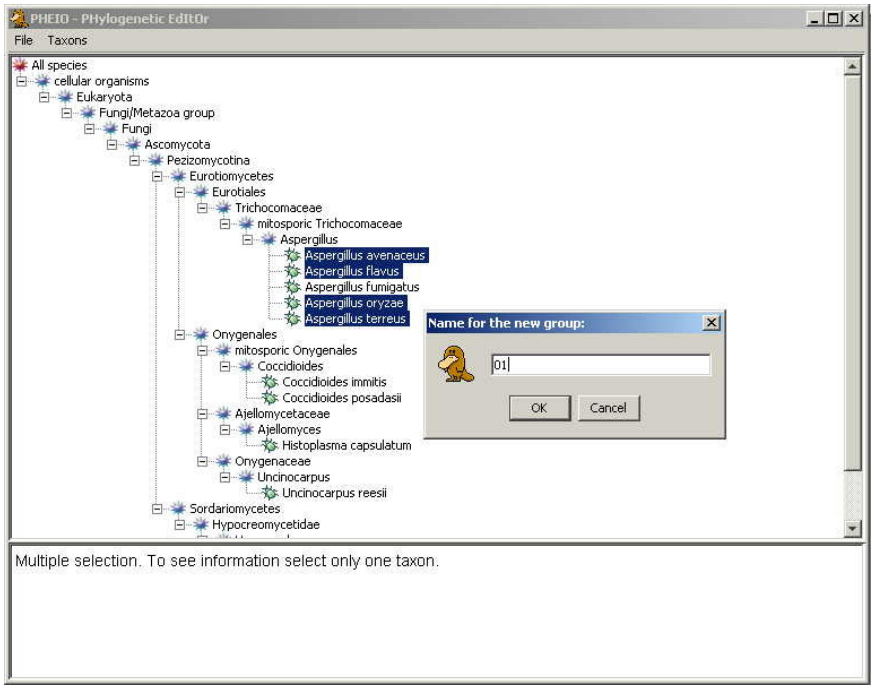


Fig. 6: To nominate the formed group.

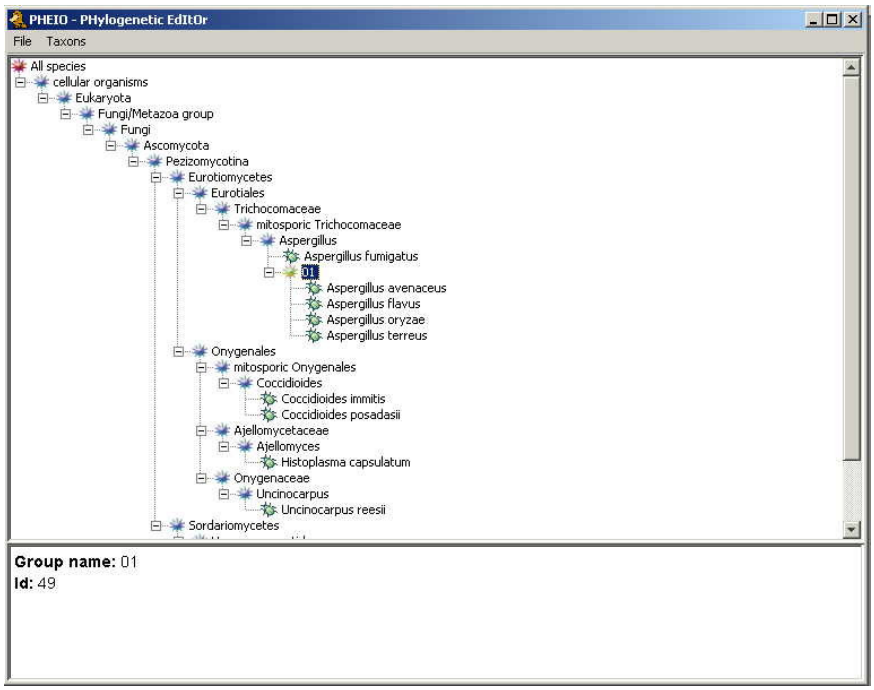


Fig 7: Created the group "01".

Using the same procedure described above, groups 02, 03, and 04 are created (figure 8). Groups can also be renamed and excluded. As the curator edits the NCBI taxonomy tree, the resultant phylogenetic information is automatically saved in PHEIO MySQL database, data being stored in the complementary table "usergroup". A new version under development will store this information in a interchangeable XML document, thus allowing multiple users to merge their editions. Moreover, divergences among entries from NCBI taxonomy and curator's introduced phylogeny will be supported.

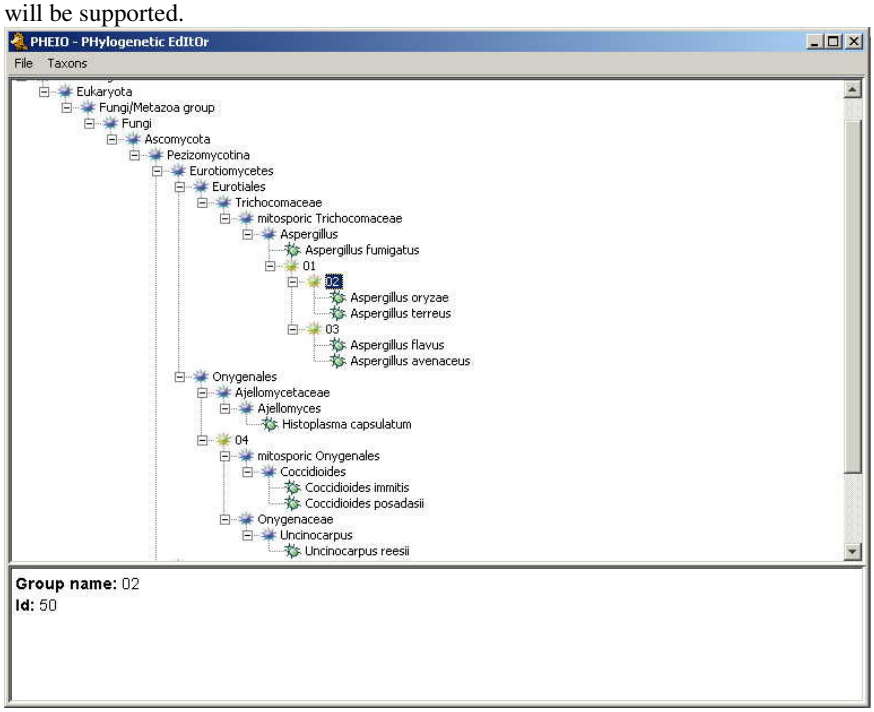


Fig. 8 Phylogeny tree of species from Pezizomycotina sub-phylum.

There is also a log system where all the steps of the building of the tree and insertion of nodes are listed. Possible problems and warnings are also listed (e.g. the input file contains a taxon not present in NCBI Taxonomy tree).

The choice of Java for implementation of PHEIO provided a high-level object-oriented programming language which provides a large and useful API. Java API and object-oriented programming characteristics allows one to build applications quickly and does not require special care about memory allocation, data structures, types and other low-level details. Moreover, Java has the "write once, run anywhere" characteristic.

Thus, a program written in Java runs in any platform that supports a Java Virtual Machine (Windows, Linux, Unix, MacOS and others).

In conclusion, PHEIO represents a useful phylogenetic editor to a standardized taxonomy tree, producing an edited database that can be promptly used by other applications. In our knowledge, PHEIO is the first visual editor to allow such functionalities. PHEIO is being currently used by our group to edit phylogenetic relationships amongst diverse bacterial taxa.

References

[1] Tatusov L R., Koonin E V., Lipman D J.: A Genomic Perspective on Protein Families. Science 278, 631 (1997)
[2] Kanehisa M., Goto S.: KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucl Acids Res, 2000, Vol.28, no.1
[3] Chen F., Mackey AJ., Stoeckert CJ Jr., Roos DS.: OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. Nucleic Acid Res. 2006 Jan 1;34 (Database issue) D363-8
[4] Benson DA., Karsch-Mizrachi I., Lipman DJ., Ostell J., Wheeler DL.: GenBank. Nucleic Acid Res. 2006 Jan 1;34(Database issue):D16-20.
[5] Morell, V. 1996. TreeBASE: the roots of phylogeny. Science 273: 569.

A tool for visualizing and analyzing EST collections

Delane P.O. Dias, Rosane Minghim *, Fernando V. Paulovich, Guilherme P. Telles **

Universidade de São Paulo, ICMC, São Carlos, Brazil
{delane,rminghim,paulovic,gpt}@icmc.usp.br

Abstract. Expressed Sequence Tags (ESTs) are samples of gene sequences, which play the role of templates in synthesis of proteins. Since the amount of collected ESTs in the past few years is enormous, the use of mining and visualization techniques has become very attractive to help study the associations between different data streams. This work proposes a methodology and a tool for visualization of ESTs as a graph for aiding biologists acquire knowledge about these sequences. The methodology includes clustering of ESTs using an assembly program followed by the transformation the groups formed in nodes of a graph. BLAST is used to align among, later adding edges between the most similar sequences. For graph visualization and exploration, we adapted a public software connected to a database. The result is a robust and open source interactive tool for Windows and Linux, that we hope will expand the suit of tools available to genomic exploration.

1 Introduction

In this note we report on a graphical tool to visualize and analyze large collections of ESTs. The tool is still under development and many improvements are yet to be done. The main features are viewing ESTs collections as graphs and the ability to modify clusters of ESTs.

Visualization is the computer science branch that aims at taking advantage of the human visual capability in revealing patters on data. User interaction is capital in visual exploratory endeavor, and may refine automatic mining of patterns and relations in the data set [12].

Many visualization techniques exist. Some are of general application and some are best for particular types of data. Of particular interest is graph visualization. A graph consists of vertices, that usually represent objects, and edges, each connecting two vertices, that usually represent relations between objects. Drawing graphs is a difficult problem by itself. Some strategies exist that make graph visualization and interaction pleasing for the human user [1].

Some software exist for visualization of biological objects of many kinds that relate to our approach. Becker and Rojas [2] employ graph-based techniques in

* Corresponding author. Telephone: +55 16 3373 9730 Fax: +55 16 3373 9751.
** The authors wish to acknowledge the financial support of CNPq and FAPESP.

order to create visual representations of metabolic pathways. Heber et al. [8] describe a method to visualize combinations of EST variants. Other tools adopt the sequence perspective to display biological objects [11]. CLANS [6], Crosslink [3] and Phylographer [10] offer graph-based visualization of ESTs. We intend to go beyond these tools functionality and allow ESTs cluster reorganization and further data exploration.

An EST results from the sequencing of a complementary DNA obtained from an RNA extracted from cells. If we neglect errors introduced by the process itself, each EST represents a portion of an RNA molecule that was transcribed from the cellular DNA and is assumed to be active at the extraction time. EST technology allows assessing the active transcripts in the cell at low cost and became popular in the last years. ESTs collections abound and continue to grow [7]. RNA extraction is a random process. RNA molecules that have many copies in the cell are sampled many times and represented many times as ESTs in a collection. Rare molecules may not be present in the collection at all.

From a computational point of view, an EST is a sequence of letters `A`, `C`, `G` and `T`. Every sequence of letters may also be paired to a sequence of base quality values, one for each letter, that quantifies the level of confidence that the letter represents the correct nucleotide in the molecule.

A common task when dealing with EST collections is clustering, that is, building groups of ESTs representing the same transcript. This is useful to reduce redundancy for data analysis and also to estimate the chance of obtaining novel transcripts from the same culture of cells.

2 Our visualization tool

The problem we addressed with our tool is the following. Given a collection of EST sequences (and their qualities, if available), construct a visual representation of clusters of ESTs at transcript level that allows exploration and modification of the clusters. Such tool is intended to support biological analysis of ESTs.

Our program performs three main tasks: (1) sequence processing and clustering, (2) graph construction and display, and (3) interaction support.

At the very first stage sequences are processed for artifacts removal. ESTs artifacts are sequencing errors introduced by the laboratory processes, the sequencing technology and contamination, that should be removed to avoid introducing relations that have little biological meaning among the sequences [4]. After trimming, ESTs are assembled by CAP3 [9]. Consensus sequences are extracted from CAP3 output and compared to each other using Blast [5]. These tasks are performed by Perl scripts. The output of CAP3 and Blast are processed and a set of input files is built for the visualization tool.

During visualization, contigs built by CAP3 are represented as vertices, and there is an edge between two vertices if and only if their consensus sequences have an alignment with Blast average e-value lower or equal to some threshold e. By average e-value we define the mean e-value of a bidirectional Blast hit.

The graph is displayed and may be interacted with using our visualization tool[1]. Vertices are drawn by a force-based algorithm that tries to planarize the graph. We extended the freely available software TouchGraph [14] to create such graph. Visible edges may be filtered by e-value – the user chooses the maximum e-value and the edges that present values greater than that value are hidden. Vertices may be selected for reassembly. Reassembling a set of vertices involves invoking CAP3 for the sequences represented by the vertices and rebuilding vertices and edges related to such sequences. Sequences or group of sequences can be selected and inspected in a separate window. Searching for sequences is also possible. Figure 1 shows a screenshot of this tool.

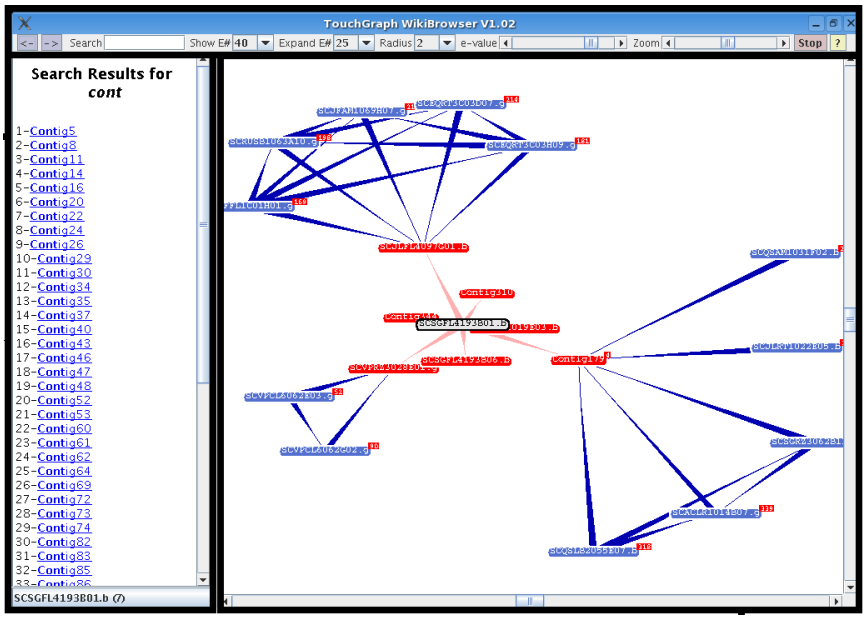


Fig. 1. The tool screenshot, and its main component windows: search window (left), functionalities tool bar (top), and graph presentation window (right).

We believe that our tool is a valuable framework for reaching new insights into ESTs matching and grouping in a flexible and interactive way, allowing to explore ESTs collections in order to reveal relations otherwise hard to find because they belong to different clusters such as alternatively spliced genes and homologous genes. It is also possible to identify spurious relations induced by malformed clusters.

[1] see http://infoserver.lcad.icmc.usp.br/infovis2/GraphVisualization

3 Concluding remarks

The tool presented in this work represents a novel combination of methodologies to allow exploration of ESTs data sets. It provides a methodology to transform such data sets into a graph representation and explore the assembled associations between sequences to find patterns within the data.

Following this methodology, we intend to expand the tool such that new filters for adding and removing graph nodes are provided and new search and comparison techniques are developed. For instance, it is possible to build two different graphs from the same data set (by changing the parameters for the grouping and mounting algorithms) and then coordinate the maps to find relationships between them.

Additionally, we can use multidimensional visualizations such as projections and additional metrics for EST exploration coordinated with this base graph in a similar way as proposed before for visual mining of text collections [13, 15].

References

1. G. D. Battista, P. Eades, R. Tamassia, and I. Tollis. Annotated bibliography on graph drawing algorithms. *Computational Geometry: Theory and Applications*, 4(5):235–282, 1994.
2. M.Y. Becker and I. Rojas. A graph layout algorithm for drawing metabolic pathways. *Bioinformatics*, 17(5):461–467, 2001.
3. T. Dezulian, M. Schaefer, R. Wiese, et al. Crosslink: visualization and exploration of sequence relationships between (micro) RNAs. *Nucleic Acids Research*, 34:W400–W404, 2006.
4. C. Baudet e Z. Dias. New est trimming strategy. In *Proc. of BSB 2005*, volume 3594 of *Lecture Notes in Bioinformatics*, pages 206–209, 2005.
5. S. F. Altschul et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.
6. T. Frickey and A. Lupas. CLANS: a java application for visualizing protein families based on pairwise similarity. *Bioinformatics*, 20(18):3702–3704, 2004.
7. Genbank. www.ncbi.nlm.nih.gov/Genbank/index.html, April 2007.
8. S. Heber, M. Alekseyev, S. Sze, H. Tang, and P.A. Pevzner. Splicing graphs and EST assembly problem. *Bioinformatics*, 18(1):181–188, 2002.
9. X. Huang and A. Madan. CAP3: A DNA sequence assembly program. *Genome Research*, 9:868–877, 1999.
10. A. Kozik. Phylographer. www.atgc.org/PhyloGrapher, April 2007.
11. S. H. Nagaraj, R. B.Gasser, and S. Ranganathan. A hitchhiker's guide to expressed sequence tag (est) analysis. *Briefings in bioinformatics*, 8(1):6–21, 2006.
12. M.C.F. Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *IEEE Trans. on Vis. and Comp. Graphics*, 9(3):378–394, 2003.
13. F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Visual mapping of text collections through a fast high precision projection technique. In *Proc. of IV'06*, pages 282–290, 2006.
14. A. Shapiro. Touchgraph LLC. www.touchgraph.com, April 2007.
15. G. P. Telles, R. Minghim, and F. V. Paulovich. Normalized compression distance for visual analysis of document collections. *Computers & Graphics*, 31(3):327–337, 2007.

A New Approach to the Integration of Proteomics Experimental Data

Alessandra Faria-Campos[1], Herbert Fernandes[2], Rodrigo Gomes[1], Breno Rates[1], Adriano Pimenta[1], Glória Franco[1], Sérgio Campos[2]
alessa@icb.ufmg.br

[1]Departamento de Bioquímica e Imunologia
Instituto de Ciências Biológicas
[2]Departamento de Ciência da Computação
Instituto de Ciências Exatas

Universidade Federal de Minas Gerais
Belo Horizonte - Minas Gerais - Brazil

Abstract. Proteomics databases are currently a very important research topic in bioinformatics. Proteomics is the identification and quantitative analysis of proteins expressed in different conditions or life stages of a cell or organism. Proteomic analysis technologies are mainly chromatography, bi-dimensional electrophoresis and mass spectrometry. To identify a protein it is often necessary to perform a series of experiments, comparing the results of such analysis to those found in proteomics databases. Most existing proteomics databases are usually related to only one type of experiment or represent processed results, instead of raw data. Because of this, researchers frequently have to resort to several data repositories in order to be able to perform the identification of the proteins. In this paper we propose an integrated proteomics database that stores raw and processed data, which are indexed allowing them to be retrieved together or individually. The proposed database, named BNDb for Biomolecules Network Database, is implemented using a MySQL server and is being used to store data from the centipede *Scolopendra viridicornis*, the parasite *Schistosoma mansoni*, the scorpion *Tityus serrulatus* and the spider *Phoneutria nigriventer*. The database construction uses a relational approach and data indexes. The proposed data model uses groups of tables for each data subtype, which store details regarding experimental procedures as well as raw data, analysis results and linked publications. BNDb also stores sequence data publicly available which can be associated to newly identified proteins present in the database. BNDb represents a new contribution to proteomics data management providing a useful service for the scientific community.

Keywords: Proteomics database, data model

1 Introduction

One of the great challenges that face science today, and in particular the biological sciences is the management of increasingly large amounts of data generated by high throughput experiments. A research area that has greatly benefited from the development of new and improved technologies for large scale experiments is proteomics. Proteomics is the identification and quantitative analysis of proteins expressed in different conditions or life stages of a cell or organism (Wilkins et al., 1997). Proteomics data are frequently stored in spreadsheets or published in databases, an approach that presents limited integration regarding raw and processed data. Existing proteomics databases are usually related to only one type of data or represent processed results, instead of raw data. Because of this, proteomics researchers frequently have to resort to several different data repositories in order to be able to extract biological information from the experimental data (Yates, 1998; Gras and Muller, 2001). Thus, several initiatives, including those associated to the Human Proteomics Organization (HUPO), have discussed the importance of unified databases harboring raw and processed data and of a system to improve the management of both types of data (Reif et al., 2004; Rohlff, 2004, Martens et al., 2005).

In this work we propose an integrated proteomics database that stores both raw and processed data, which are indexed allowing them to be retrieved together or individually. The proposed database, named BNDb for Biomolecules Network Database, stores proteomics data produced by members of the Minas Gerais Proteomics Network (Brasil) and sequence data publicly available. The database has been implemented using a MySQL server and uses a relational approach and data indexes which speeds up the process of data retrieval and the correlation with sequence data previously identified.

BNDb addresses the problems outlined above in two different ways. First, it imports experimental data directly from the equipment used to perform the experiment. The output files generated by the equipment are parsed by BNDb and the data is inserted in the database without user assistance, eliminating the bottleneck of inserting data manually into the computer. Moreover, an relational database is used to store this data and retrieve it efficiently. The relational database allows us to use standard database techniques to identify relationships between different sets of data and to retrieve these associated data in a straightforward manner. Currently our database uses SQL queries to retrieve the data, meaning

that we can identify sets of data in different experiments that have common attributes directly. This enables us to relate experiments that have the same data values or that may have been previously unknown to be related. It is important to notice that the focus of BNDb is to store and analyze proteomics experimental data. It is possible to use tools that offer a level of flexibility in the design of the database such as workflow based tools and object oriented databases that is not present in BNDb. Workflow based systems allow users to easily change the protocols and data being stored; object oriented databases make it simpler to change the structure of the database. However, this flexibility is not needed in proteomics, since the types of experiments is fixed for this type of analysis, and the flexibility offered by those tools does not add to the task at hand. Moreover, flexibility often comes at a price, often the ease of modification of tables and experiment steps make the system less efficient, and the fact that these changes are made through a user interface, makes it impossible to implement more complex analyses that cannot be directly modeled through the user interface. The objective of BNDb rather than enabling its final user to make a restricted set of changes to the model, is to make it possible to implement very sophisticated analysis algorithms and tools which can be easily added directly to the database, but cannot be described in terms of an end-user interface.

Some other approaches have been used to store proteomics experimental data. PEDRo is a data schema for how to store and share proteome data (Taylor et al., 2003; Garwood et al., 2004). PRIDE is a database that stores protein and peptide identifications, another attempt that aims primarily to disseminate and make data publicly available (Martens et al., 2005b). Another implementation, 2DDB aims to store and analyze quantitative proteomics data (Malmstrom et al., 2006). None of the mentioned systems however, proposes to integrate project managing, dissemination of raw and analyzed data and cross-referencing of proteomics results with available sequence data as proposed by BNDb.

2 The Data Model

The final objective of the proteomics analysis is to uniquely identify the set of proteins that are present in a given sample. In order to do so, several experiments are typically performed such as bi-dimensional electrophoresis (2D-PAGE), liquid chromatography (LC) and mass spectrometry (MS). It is important to notice that these experiments are usually related, since a sample separation by 2D-PAGE or LC usually precedes

the MS analysis. The data model has to take this into consideration in order to represent the order and relationship of the experiments correctly.

The major components of the data model of the BNDb database can be seen in Figure 1, where the main entity is the *experiment*. Experiments are associated with projects, and those are associated with researchers that belong to them. Known DNA,EST or protein sequences are also associated to experiments. Experiments can be of different types: *Liquid Chromatography*,*1D/2D Gels* and *Mass Spectrometry* and each experiment can have raw data and results stored in the database.

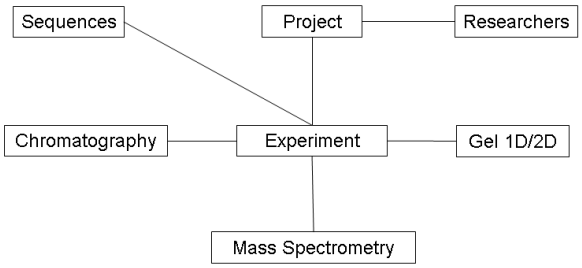


Fig. 1. Main components of the data model.

A simplification of BNDb data model can be seen in Figure 2 (some auxiliary tables are not shown for clarity). In the database, an experiment can be of type *chromatography*, *gel electrophoresis*, or *mass spectrometry*. In each case the corresponding table has an entry associated with the experiment. Each type of experiment has one or more specific results, either chromatography peaks, gel spots or m/z values. These results are also stored in separate tables, and are associated with the experiment. Image and chromatogram files are stored outside of the database, but a link to these files is stored in the database along with numeric results.

The experiment is the main component of the model because BNDb is aimed primarily at assisting researchers in cross-linking their experi-

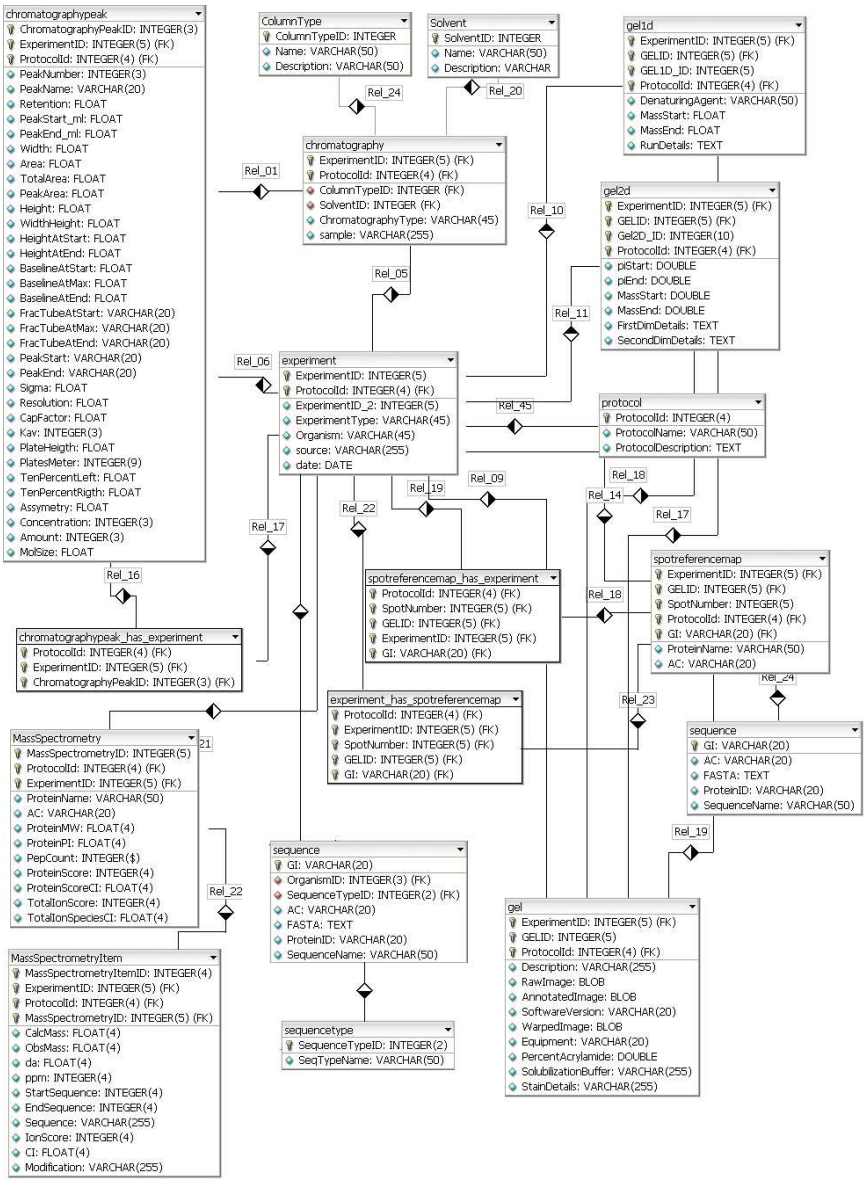


Fig. 2. The data model.

mental data. The common data flow starts when a new experiment is performed and its data is entered in the *experiment* table. Each experiment receives an internal ID that identifies it uniquely. According to the type of experiment the tables related to that type of experiment are also filled

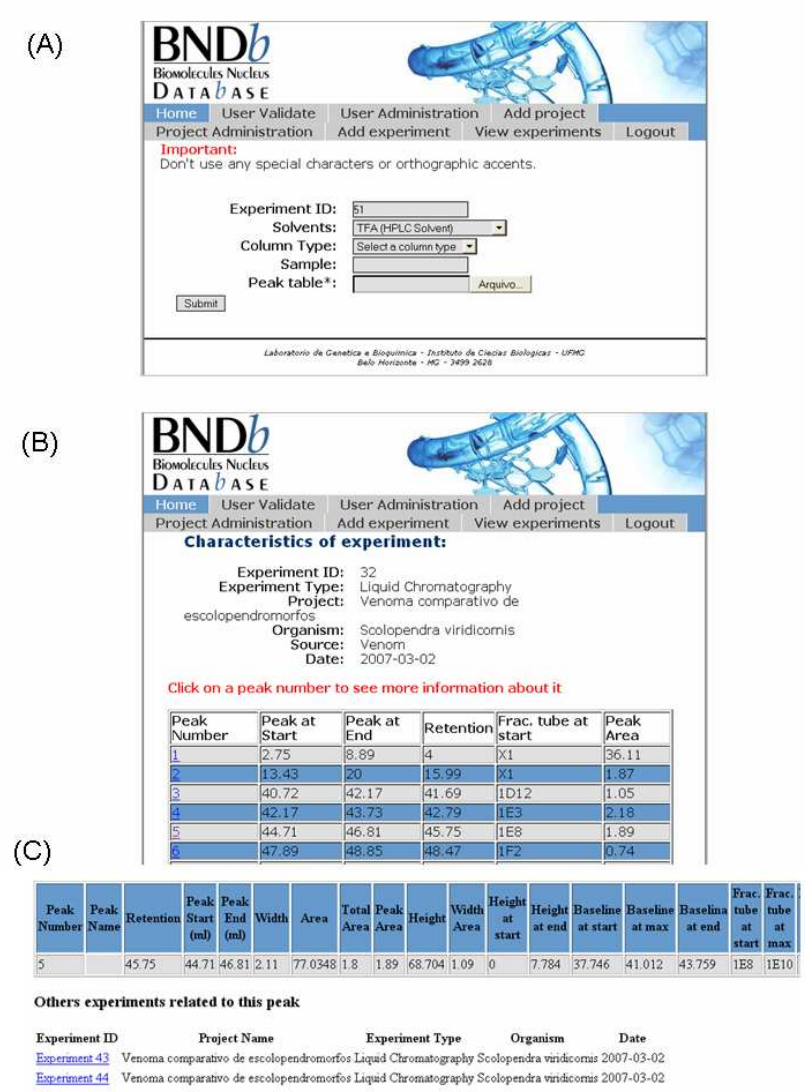


Fig. 3. User interface to upload data and retrieve it from the database. (A) Data upload screen; (B) Experiment analysis screen; (C) Peak analysis screen including cross-links to related experiments.

and associated to the internal ID. Experiments are often related to other experiments, such as when a particular result prompts the researcher to perform a more detailed analysis on a peak or spot. BNDb then stores the associated experiments by cross-linking the internal IDs which allows

it to establish not only a relational, but also a temporal link, so that the sequence of experiments performed can be followed later. Figure 3 shows how this complex set of operations can be visualized by the user. Through the user web interface the researcher can visualize the relationships of his experiment without worrying about the details of database construction. The BNDb database is accessed through a web based interface that uses php scripts to communicate with the database. Experimental data, however, is not directly inserted in the database using the web interface. Instead, the users uploads the experiment results file using the web interface and parsers are used to interpret this data and insert it in the database. This makes importing data faster and less error prone, since the files generated from the equipments that perform the experiments are read automatically by the parsers.

Through the association of the tables in the database it is possible to identify exactly how experiments are related to one another, assisting the researchers in controlling the flow of experiments performed, and also in explaining the results and how they were obtained. It is important to notice that this is possible in BNDb because of the use of a relational database, since it enables the user to search for related items regardless of where they have been stored. Hierarchical storage methods such as PEDRo may have difficulty following some of the relationships due to the nature of their model.

The *project* component allows BNDb to store data of different projects and researchers, enabling each to work independently with exclusive access to their data. BNDb stores not only projects and project members, but also the researcher that has performed each experiment, making possible a fine control of the experiment flow and using security restriction to garantee acess to specific data.

Sequence tables are also included in this model. These tables contain information about nucleotide and protein sequences as well as the sequence in FASTA format. This sequence can be exported from the database to be used in other bioinformatics tools such as similarity searches (using BLAST) or evolutive analysis. The sequence data have been obtained directly from the National Center for Biotechnology Information (NCBI). Sequence data stored in BNDb is an important aspect of this model, because it allows researchers to access all data available on a certain protein from its nucleotide sequence to the proteomic analysis, making it simpler and faster to perform the analysis. To our knowledge this feature is not available in other proteomic databases.

We are currently finishing the implementation of the web interface, as well as the parsers for importing experiment data. The BNDb will then be populated by proteomics data generated by experiments from several laboratories at UFMG and Fiocruz. We are also currently working on importing sequence data from NCBI and to have it inserted in the database.

The database has currently been filled with data from the proteomic analysis of the venom from the arthropod *Scolopendra viridicornis*, a common Brazilian centipede. In order to assess the complexity of the venom of *Scolopendra viridicornis*, a pooled venom sample (1 mg) was subjected to bi-dimensional liquid chromatography. This technique consists of the sequential use of ion-exchange fractionation (first dimension), followed by further purification by reversed phase (RP) chromatography (second dimension) of the fractions obtained in the first step. After the RP step, the fractions were analyzed by electrospray ionization quadrupole/time-of-flight mass spectrometry (ESI-Q-TOF/MS). The fractions which contained proteins and peptides purified to a homogeneous state were subjected to N-terminal sequencing by automated Edman's degradation. Then, similarity searches were performed by the Fasta3 tool against the Uniprot and Swiss-Prot data Bank. Details on the methodology were provided by Rates and co-workers (2007). In Figure 4 we can see the initial chromatogram (4A) and the subsequent ones produced from a second analysis of the individual resulting peaks(4B). The user can then choose one peak from individual experiments and visualize the spectrum corresponding to the MS analysis (4C).

3 Implementation

The BNDb database has been implemented using a MySQL server version 4.0.21 running over a Pentium 2.5 GHZ machine using Linux Suse distribution 9.2. The database construction uses a relational approach and data indexes to associate experiments to each other and to the results and those to projects. The software DBDesigner 4.5.6 has been used for the data model project. The proposed data model uses groups of tables for each data subtype, which store all details regarding the experimental procedure as well as raw data, analysis results and linked publications resulting from an specific experiment. The data model proposed has been designed to store data from proteomic analysis of the centipede *Scolonependra viridicornis* parasite *Schistosoma mansoni*, and

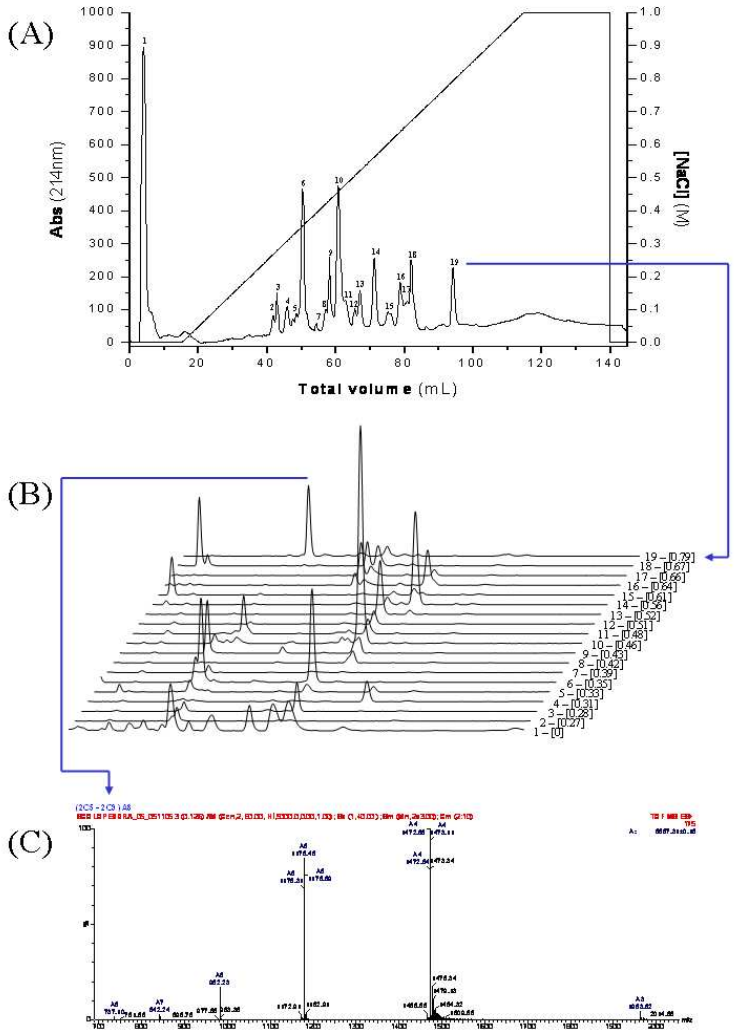


Fig. 4. Example of cross-linked experiments in a typical proteomics analysis. (A) Initial chromatogram from the analysis of *Scolopendra viricornis* venom; (B) Subsequent chromatograms produced from the analysis of the individual peaks in A; (C) Spectrum resulting from analysis of a peak selected in B.

analysis of the venom from the scorpion *Tittyus serrulatus* and the spider *Phoneutria nigriventer*.

The proposed database also stores sequence data from these organisms, so that diverse information regarding an organism can be retrieved

automatically. It contains sequence data publicly available which will be associated to identifications performed in new samples. This data is stored in FASTA format along with identifications as gi or accession numbers.

4 Discussion and Conclusions

In this paper we have presented BNDb, the Biomolecules Network Database, a proteomics integration database. BNDb stores raw and analyzed data along with project management attributes. It also connects the information produced by different types of experiments as well as sequence data in order to be able to present to the researcher a complete picture of the experimental process, making it easier to access all data related to specific proteins. This speeds up the proteomics analysis and increases its reliability.

The number of proteomics databases available nowadays is high and is still increasing. However, most of them have been designed to store processed and curated data or store one type of experiment only. This means that the proteomics data is only stored in the database after the experiments have been completed and their results completely analyzed and curated. These databases assist researchers in comparing their experiments with others that have already been completed. Consequently, these databases perform a different task than BNDb, and in fact are complementary to it. The focus of BNDb is not in storing the final results (even though this is also done), but rather to help researchers in tracking of the data generated by individual experiments and how this data relates to other experimental data even before the experiment results have been processed.

From the other proteomic databases available, two data models relate more closely to the BNDb model, 2DDB and PEDRo. Both store proteomics raw experimental data in a similar way as BNDb. 2DDB, however, focus on the protein identification and how to identify experiments that relate to a given protein. 2DDB is based on a core data model describing fundamentals such as experimental description and identified proteins (Malmstrom et al., 2006). It is efficient in determining the path through which a protein has been identified but raw data are not the focus of the project, and in fact this data is not part of the core data model of 2DDB. As a consequence, 2DDB helps researchers after the protein identification has been performed but is less helpful during the experimental phase, when it is necessary to store raw data and experiments attributes independent of which protein it relates to since this is not known at the

time. 2DDB also does not store sequence data. PEDRo stores proteomics data based on the experiment order in which it was generated (Garwood et al., 2004). It is not, however, a relational database system as are BNDb and 2DDB. Instead, it stores *and processes* data only in a XML format file. As a consequence, it is not so efficient in storing large volumes of data. Besides, the XML storage imposes a natural indexing of the data, since these files are read and stored sequentially. As a consequence, it is very simple to retrieve information in the same order as it was stored, but if one needs to correlate data in a different order using an XML file becomes inefficient, since all information must be reordered in main memory to allow a different indexing. In our case, data is cross-referenced in different ways depending on the analysis being performed, and since the order changes frequently, no predetermined order would be efficient. We use a relational database to store data, because relational databases are designed to allow information retrieval in multiple orders efficiently. So, if a researcher using the PEDRo system wants to access data based on other criteria than the established order, access is not efficient, particularly for large databases, because PEDRo uses the XML format not only for storage, but also for processing the data.

For data capture, PEDRo database makes extensive use of XML for capturing, transmitting, storing and searching proteomics data. The data-capture process uses a software tool which prompts users for values for different fields, and includes facilities for importing substantial data files, such as those representing peak lists. The tool constructs data-entry forms from the XML schema definition of the PEDRo model. The result of the data capture process is thus an XML file that corresponds to the PEDRo schema. BNDb on the other hand, uses a web server as the interface for data capture. Simple forms constructed in php language are made available for data entry. The researcher uploads the result files generated directly by the experiment using this interface. The files are processed through parsers that are used to interpret this data directly out of the experiment results. Using a web based interface gives an increased portability to data capture since users do not need to install any specific software to have access to database for data importing and exporting. Also, this system makes data importing and storing faster and less error prone, since the files are imported automatically, processed by the parsers and inserted directly on the database.

BNDb has been designed to store data from experiments of several different organisms. At the moment data from *S. vidicornis*, *S. mansoni*, *T. serrulatus* and *P. nigriventer* proteomic studies are being collected to

feed the database. These experiments have been performed in different laboratories by different research groups, demonstrating the usefulness of the BNDb, which can provide assistance in the proteomics research for a large scientific community. Moreover, it demonstrates the capability of the database to store data from different formats and research groups, emphasizing also its flexibility.

The construction of a new data model for proteomics data importing and storing represents an important contribution for proteomics and bioinformatics. We have developed a tool that combines a powerful storage engine (the relational database) and a friendly access interface, aiming to assist proteomics researches directly at data handling and storage.

5 References

Garwood K, McLaughlin T, Garwood C, Joens S et al. (2004). PEDRo: A database for storing, searching and disseminating experimental proteomics data. BMC Genomics 5:68.

Gras R and Muller M (2001). Computational aspects of protein identification by mass spectrometry. Curr Opin Mol Ther 3:526-32.

Kalia A and Gupta RP (2005). Proteomics: a paradigm shift. Crit Rev Biotechnol 25:173-198.

Malmstrom L, Marko-Varga G, Westergren-Thorsson G, Laurell T et al. (2006). 2DDB - Abioinformatics solution for analysis of quantitative proteomics data. BMC Bioinformatics 7:158.

Martens L, Nesvizhskii AI, Hermjakob H, Adamski M et al. (2005). Do we want our data raw? Including binary mass spectrometry data in public proteomics data repositories. Proteomics 5:3501-3505.

Martens L, Hermjakob H, Jones P, Adamski M et al. (2005b), PRIDE: the proteomics identifications database. Proteomics 5:3537-45.

Rates, B., M. P. Bemquerer, M. Richardson, M. H. Borges, R. A. Morales, M. E. De Lima e A. M. Pimenta (2007). Venomic analyses of Scolopendra viridicornis nigra and Scolopendra angulata (Centipede, Scolopendromorpha): Shedding light on venoms from a neglected group. Toxicon 49:810-26.

Reif DM, White BC and Moore JH (2004). Integrated analysis of genetic, genomic and proteomic data. Expert Rev Proteomics 1:67-75.

Rohlff C (2004). New approaches towards integrated proteomic databases and depositories. Expert Rev Proteomics 1:267-274.

Taylor CF, Paton NW, Garwood KL, Kirby PD et al (2003). A systematic approach to modeling, capturing, and disseminating proteomics experimental data. Nat Biotechnol 21:247-254.

Yates JR (1998). Database searching using mass spectrometry data. Electrophoresis 19:893-900.

Wilkins MR, Williams KL, Appel, RD and Hochstrasser (1997). Proteome Research: New Frontiers in Functional Genomics. Springer -Verlag, Berlim.

Coffea arabica class 1 and class 2 resistance gene related sequences within the Brazilian Coffee Genome EST database

E.V.S. Albuquerque[1]*, M.S. Silva[2], C.C. Teixeira[1], N.F. Martins[1], M.A. Campos[3] and M.F.Grossi-de-Sá[1]

[1]Embrapa Recursos Genéticos e Biotecnologia - LIMPP, Brasília-DF, Brazil; [2]Embrapa Cerrados, Planaltina-DF, Brazil; [3]Universidade Federal de Lavras, Lavras-MG, Brazil; *Corresponding author, E-mail: erikavsa@cenargen.embrapa.br

ABSTRACT

Coffee is a traditional agricultural product in the context of the Brazilian economy which accounts for 2,4% of the total value of exportations and for the generation of 8 million jobs. Brazil produces 40% of the commercialized coffee in the international market and is the world's second largest coffee consumer, particularly of the *Cofffea arabica* species. Although the *C. arabica* production and consumption correspond to approximately 70% of the coffee market, it is highly susceptible to pests and diseases. Therefore, there is a raising interest of genetic breeding programs in developing *C. arabica* varieties with resistance to pests and diseases. A large number of plant resistance genes (*R* genes) have already been isolated and classified into six categories denoted class 1-class 6. The majority of the *R* genes belongs to the class 2 and encodes proteins which contain the following domains in its sequence: nucleotide binding site (NBS), leucine-rich repeat (LRR) and an N-terminal putative leucine-zipper (LZ) or other coiled-coil (CC) sequence. Class 1 comprises the *R* genes homologous to the type member gene of this class, which is the gene denoted *pto*, that encodes a protein possessing a serine/threonine kinase catalytic region and a myristylation motif at its N-terminus. Well described aminoacid sequences corresponding to class 1 and 2 *R* genes were used to screen the *Coffee Functional Genome* Brazilian Database (CafEST) for class 1 and 2 *R* gene homologous EST sequences from *C. arabica*. The selected ESTs in this search were grouped into clusters (contigs ou singlets), which were subsequently analyzed in terms of homology with *R* genes available within public databases. Multi-alignments among the consensus deduced amino acid sequences of contigs representing probable CafEST *C. arabic*a *R* genes were used to generate phylograms. The results indicate that both putative class 1 and 2 *C. arabic*a *R* genes are considerably represented within the CafEST, since a high number of class 1 and 2 *R* gene related ESTs was retrieved upon screening of this database, this is, 525 ESTs homologous to class 1 *R* genes clustered into 262 clusters (118 contigs and 144 singlets) and 449 ESTs homologous to class 2 *R* genes clustered into 190 clusters (82 contigs and 108 singlets), resulting in a total of 973 ESTs and 452 clusters (200 contigs and 252 singlets). Moreover, the phylograms show that the CafEST *C. arabica* contig sequences may be grouped in four groups of putative class 1 *R* genes and four groups of putative class 2 *R* genes, besides showing these contig sequences are considerably homologous to the sequences used to screen the CafEST. Undergoing analysis of typical R protein domains may add new information to the presented data. This study will help the future development of molecular markers correlated with genetic markers of resistance in coffee and the future isolation of complete *R* gene sequences from *C. arabica*, which may be used for plant genetic transformation.

Key words: *Coffea arabica*, plant resistance, class 1 *R* gene, class 2 *R* gene, NBS, LRR, functional genomics.

INTRODUCTION

Coffee is a woody shrub from the *Rubiaceae* family with a long biological cycle. Among more than 80 species of the subgenus *Coffea* studied so far, *Coffea arabica* is the species mostly commercially cultivated and consumed. Although *C. arabica* production and consumption corresponds to 70 % of the coffee market, it is highly susceptible to pests and diseases. Therefore there is a raising interest of genetic breeding programs in developing *C.arabica* varieties with increased resistance to pests and diseases.

A large number of plant resistance genes (*R* genes) have already been isolated and were classified into six categories denoted class1-class 6. The majority of the *R* genes belongs to the class 2 and contains a nucleotide binding site (NBS), a leucine-rich repeat (LRR) and an N-terminal putative leucine-zipper (LZ) or other coiled-coil (CC) sequence. Class 1 comprises the *R* genes homologous to the type member of this class, the tomato *R* gene denoted *pto*, which presents a serine/threonine kinase catalytic domain and a myristylation motif at its N-terminus (Martin, *et al*., 2003).

In the present study, it is described the identification of *C. arabica* sequences possibilbly coding for class 1 and class 2 *R* genes from the Brazilian Coffee Genome EST database (CafEST). Thorough analysis of these sequences will provide important data about coffee plant resistance mechanisms. The identification and validation of the function of the putative *C. arabica* class 1 and class 2 *R* genes from the CafEST support the future development of molecular markers to assist coffee breeding programs, besides supporting the future isolation of coffee *R* genes, aiming the generation of transgenic plants resistant to pests and diseases.

MATERIAL AND METHODS

All the *C. arabica* sequences used in the present work correspond to sequenced ESTs and clusters (contigs and singlets) obtained from the Brazilian Coffee Genome EST database (CafEST: http://cafe.lge.ibi.unicamp.br/) and are derived from cDNA libraries specific for different genotypes, organs (leaf, stem, fruit, flower and root), growth and stress conditions (Vieira et al, 2006). The CafEST was screened for *R* genes of class 1 and 2, separately, by using known homologous sequences of class 1 and 2 *R* genes through Basic Local Alignment Tool (BLAST) tBLASTn searches (Altschul *et al.*, 1997). The sequence of the *R* gene *pto* (gi|557882) from *Lycopersicon pimpinellifolium* was used as homologous sequence of class 1 *R* gene. The following sequences, which represent subclasses of the class 2 *R* genes, were also used to screen the CafEST database: Bs2 (gi|6456755) from *Capsicum chacoense*; Dm3 (gi|4106975) from *Lactuca sativa*; Gpa2 (gi|6164969) from *Solanum tuberosum*; Hero (gi|26190258) from *Lycopersicon esculentum*; HRT (gi|7110565) from *Arabidopsis thaliana*; I2 (gi|4689223) from *L. esculentum*; Mi1.1 (gi|3449378) from *L. esculentum*; Mi1.2 (gi|3449380) from *L. esculentum*; Pib (gi|37777009) from *Oryza sativa*; Pi-ta (gi|12642090) from *O. sativa*; R1 (gi|17432423) from *Solanum demissum*; RP1 (gi|5702196) from *Zea mays*; RPM1 (gi|963017) from *A. thaliana*; Rpp8 (gi|29839585) from *A. thaliana*; Rpp13 (gi|7229451) from *A. thaliana*; Rps2 (gi|15236112) from *A. thaliana*, Rps5 (gi|3309619) from *A. thaliana*, Rx1 (gi|8515762) and Rx2 (gi|5911745) from *S. tuberosum*, Xa1 (gi|2943742) from *O. sativa* and Sw5 (gi|15418714) from *L. esculentum*. The *C. arabica* ESTs, retrieved from the screening, were clustered into contigs and singlets for each *R* gene class separately by using the Contig Assembly Program (CAP3). The deduced amino acid sequences and the corresponding open reading frames (ORFs) of the *C. arabica* contigs were obtained by using the NCBI ORF Finder (http://www.ncbi.nlm.nih.gov/projects/gorf/). The amino acid sequences of the *C. arabica* contigs and of the homologous sequence mentioned before were analyzed, separately for each *R* gene class, by multi-alignment by using the EMBL-EBI ClustalW program (http://www.ebi.ac.uk/clustalw/). The multi-alignments demonstrating the similarities among the amino acid sequences were used to generate phylogram representantions, separately for each *R* gene class, by using the EMBL-EBI ClustalW program. The resulting phylograms were visualized by using the TreeView program (http://darwin.zoology.gla.ac.uk/~rpage/treeviewx/).

RESULTS AND DISCUSSION

The searches for class 1 and class 2 *R* gene homologues of *C. arabica* within the CafEST, using *e-values* inferior to 1 x e^{-4}, resulted in the identification of 525 ESTs homologous to class 1 *R* genes and 449 ESTs homologous to class 2 *R* genes, retrieving a total of 973 ESTs. Clusterizations of the ESTs, performed for each class separately, resulted in a total of 118 contigs and 144 singlets of putative class 1 *R* gene and 82 contigs and 108 singlets of putative class 2 *R* genes. Therefore, the screening of the CafEST for class 1 *R* genes by using a single homologous sequence (i.e. *pto*) was efficient since it retrieved a considerable number of *C. arabica* class 1 *R* gene related sequences. Several contigs of putative class 1 *R* genes presented long stretches of amino acid sequences with high homology to the Pto protein sequence, as indicates the phylogram on Figure 1. Within the phylograms, the deduced amino acid sequences of the CafEST contigs homologous to class 1 R proteins were grouped into four main groups, the largest one comprising the Pto from tomato. The amino acid sequence similarity among the CafEST contigs of putative class 2 *R* genes, presented in the phylogram in Figure 2, demonstrates that the CafEST comprises ESTs representing all the sub-classes of class 2 *R* genes, i.e., Bs2, Dm3, Gpa2, Hero, HRT, I2, Mi1.1, Mi1.2, Pib, Pi-ta, R1, RPM1, Rpp8, Rpp13, Rps2, Rps5, RX1, RX2, Sw5 and Xa1. Within the phylogram of deduced amino acid sequences of CafEST contigs homologous to class 2 *R* genes, there are four main groups, being the first group homologous to sequences Pi-ta, Rps2, Rps5, Dm3, I2, RP1, HRT, Rpp8 e Rpm1, a second group comprising a single contig with no homology to other R proteins used to generate the phylogram (C36), a third group homologous to the sequences Bs2, R1, Hero, Mi1.1, Mi1.2, Gpa2, RX1, RX2, Sw5, Pib and Rpp13, and a fourth group homologous to the sequence Xa1. Interestingly, it is possible that the contig C36 represents a *C. arabica* subclass of class 2 *R* genes distinct from the known and well-described subclasses. Furthermore, both phylograms indicate that the contigs generated are considerably homologous to the known R protein sequences used to screen the CafEST.

CONCLUDING REMARKS AND PERSPECTIVES

Here we present the screening results of the CafEST for *C. arabica* putative class 1 and class 2 *R* genes and the analysis of their sequence homology with other well studied class 1 and class 2 *R* genes. It was demonstrated that the *C. arabica* putative class 1 and 2 *R* genes are vastly represented within the CafEST and that, in relation to amino acid sequence homology, both the class1 and the class 2 *R* genes grouped into four distinct groups. The present study supports the development of molecular markers for plant resistance to be applied in coffee genetic breeding programs. Moreover, our data is useful for the future isolation of *C. arabica* complete *R* genes to be used in the generation of transgenic plants resistant to pests.

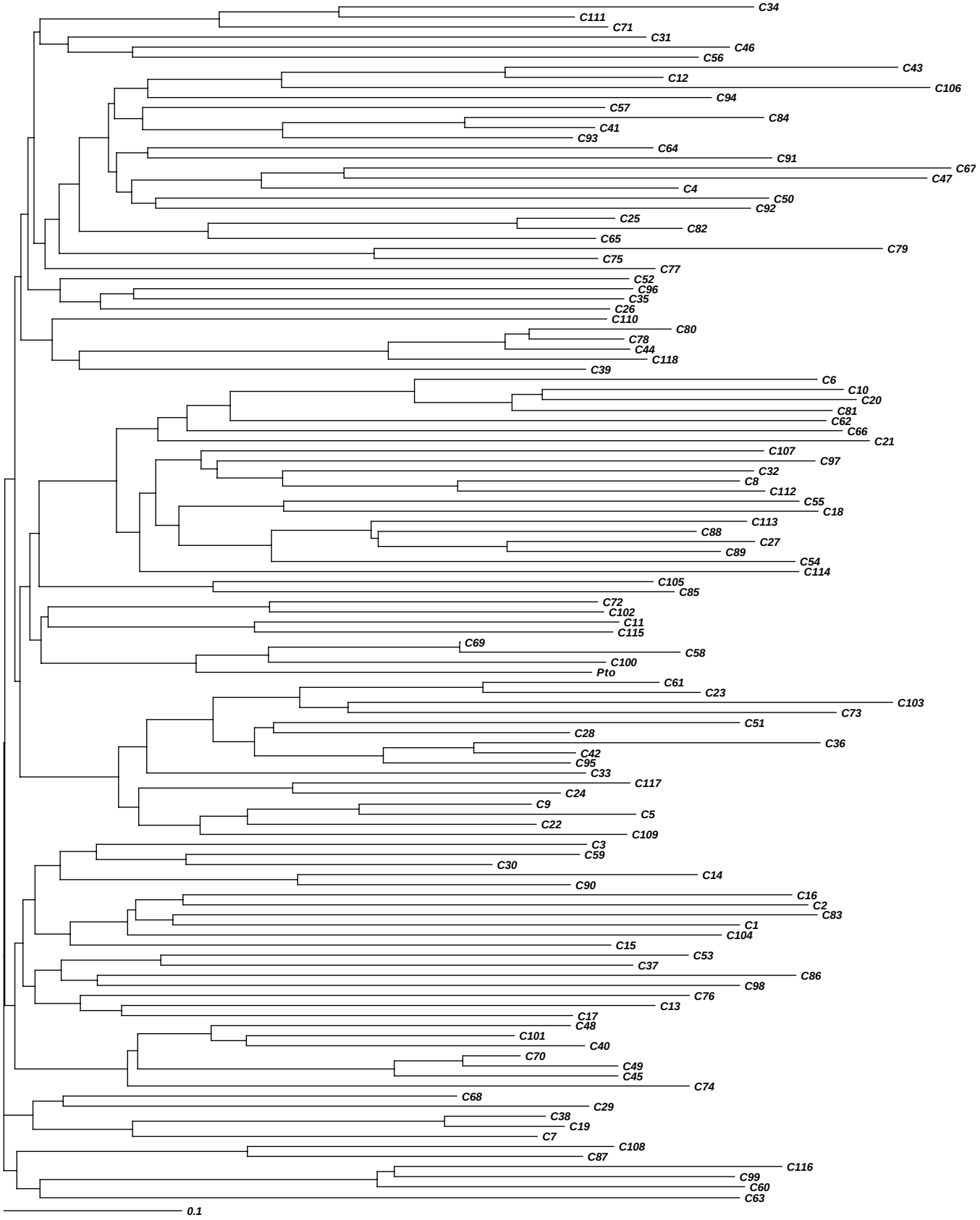


Figure 1. Phylogram representation of similarities among amino acid sequences related to class 1 R proteins. Sequences analyzed include *C. arabica* contigs from the CafESt (C#) and a sequence of a known class 1 R protein used to screen the CafEST (Pto).

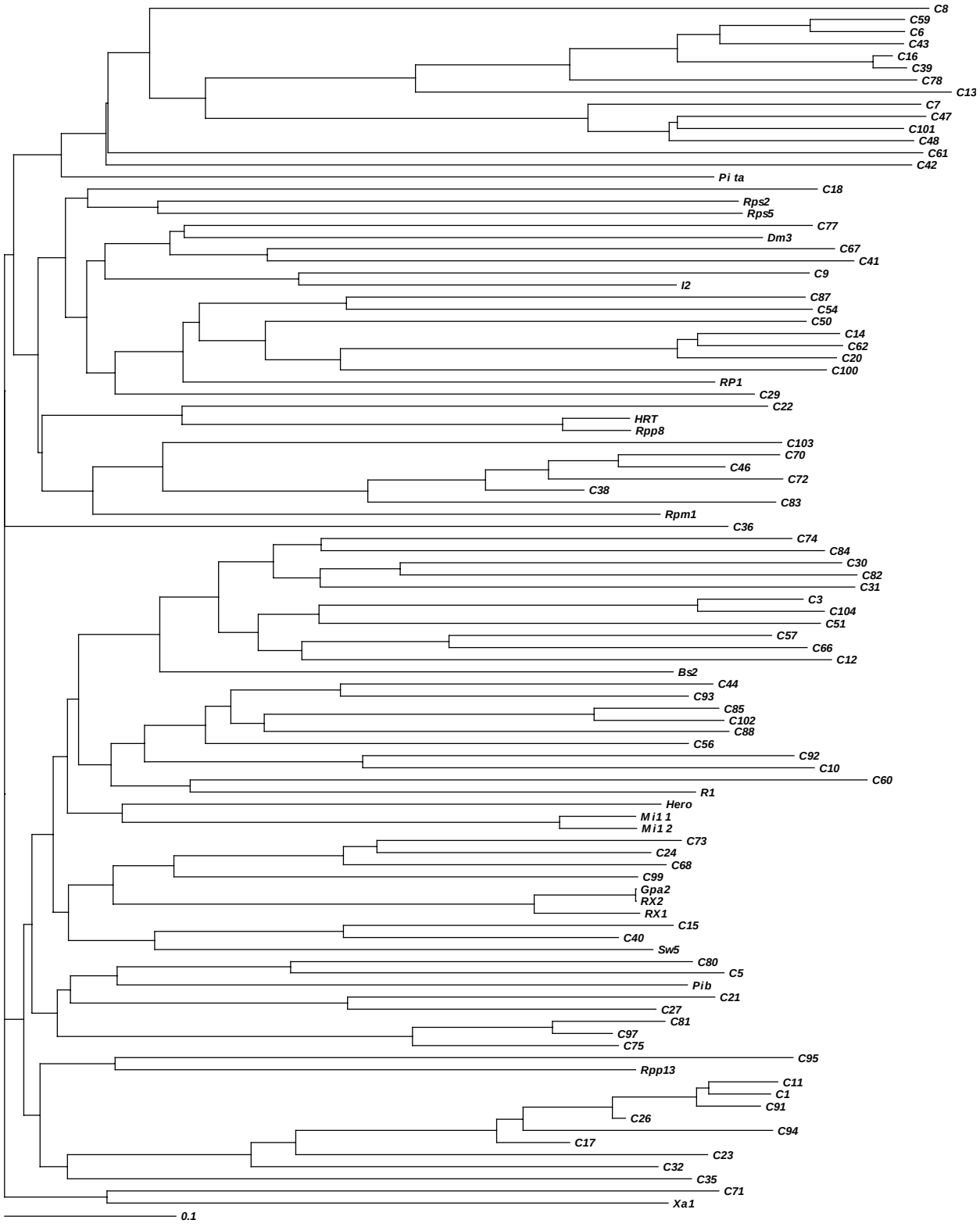


Figure 2. Phylogram representation of similarities among amino acid sequences related to class 2 R proteins. Sequences analyzed include *C. arabica* contigs from the CafESt (C#) and sequences of known class 2 R proteins used to screen the CafEST (Bs2, Dm3, Gpa2, Hero, HRT, I2, Mi1.1, Mi1.2, Pib, Pi-ta, R1, RP1, RPM1, Rpp8, Rpp13, Rps2, Rps5, Rx1, Rx2, Xa1, Sw5).

REFERENCES

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D.J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25:3389-3402.

Martin, G.B., Bogdanove, A.J. & Sessa, G. (2003). Understanding the functions of plant disease resistance proteins. *Annu. Rev. Plant Biol.* 54: 23-61.

Vieira, L.G.E., Andrade, A.C., Colombo, C.A., Araújo, A.H., Mehta, A. *et al.* (2006). Brazilian coffee genome project: an EST-based genomic resource. *Braz. J. Plant Physiol*. 18(1):95-108.

208

References

1. Felipe Albrecht, Jomi Hubner, and Alberto Dávila. A distributed algorithm for phylogenetics inference. In *BSB 2007 Poster Proceedings*, pages 66–69, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
2. Érika Valéria Saliba Albuquerque, Marilia Santos Silva, Cristiane de Camargo Teixeira, Natália Florêncio Martins, Magnólia de Araújo Campos, and Maria Fátima Grossi de Sá. *Coffea arabica* class 1 and class 2 resistance gene related sequences within the brazilian coffee genome est database. In *BSB 2007 Poster Proceedings*, pages 204–207, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
3. Nalvo F. Almeida, Marcel Y. Nakazaki, Andrey A. Tamura, Luciana Y. Hiratsuka, André C. Lima, Said S. Adi, Carlos J.M. Viana, and Leandro P. Brazil. EGGview: Visualization of EGG comparative data using GBrowse. In *BSB 2007 Poster Proceedings*, pages 24–27, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
4. Henrique Velloso, Izabella Pena and J. Miguel Ortega. PHEIO, a java/mySQL based phylogenetic editor for NCBI taxonomy tree. In *BSB 2007 Poster Proceedings*, pages 179–186, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
5. Mauro A. A. Castro, José C. M. Mombach, Rita M. C. de Almeida, and José C. F. Moreira. Impaired expression of NER gene network in sporadic solid tumors. In *BSB 2007 Poster Proceedings*, pages 10–13, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
6. Rocio L. Cecchini, Axel J. Soto, Gustavo E. Vazquez, and Ignacio Ponzoni. A genetic algorithm for detection of relevant descriptors in ADMET prediction. In *BSB 2007 Poster Proceedings*, pages 62–65, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
7. Sergio Manuel Serra da Cruz, Fábio Coutinho, Alberto Dávila, Maria Luiza Machado Campos, and Marta Mattoso. Experiencing GARSA as a scientific workflow on grid environment. In *BSB 2007 Poster Proceedings*, pages 103–114, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
8. José Antônio Fernandes de Macêdo, Sérgio Lifschitz, Fábio Porto, Philippe Picouet, Antonio Basilio de Miranda, and Thomas Dan Otto. Towards a conceptual modeling language for biological domains. In *BSB 2007 Poster Proceedings*, pages 128–137, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
9. Delane P. O. Dias, Rosane Minghim, Fernando V. Paulovich, and Guilherme P. Telles. A tool for visualizing and analyzing EST collections. In *BSB 2007 Poster Proceedings*, pages 187–190, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
10. Zanoni Dias and Cid C. de Souza. Polynomial-sized ILP models for rearrangement distance problems. In *BSB 2007 Poster Proceedings*, pages 74–85, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).

209

11. Alessandra Faria-Campos, Herbert Fernandes, Rodrigo Gomes, Breno Rates, Adriano Pimenta, Glória Franco, and Sérgio Campos. A new approach for the integration of proteomics experimental data. In *BSB 2007 Poster Proceedings*, pages 191–203, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
12. Elmer A. Fernández and Mónica Balzarini. A tool for cluster number estimation in SOM-based gene expression pattern analysis. In *BSB 2007 Poster Proceedings*, pages 14–23, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
13. Mario Guarracino, Davide Feminiano, Francesca Del Vecchio Blanco, and Salvatore Cuciniello. Using gene expression analysis to relate disease, genes, and therapeutics. In *BSB 2007 Poster Proceedings*, pages 86–98, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
14. Tiago J. S. Lopes and Guilherme P. Telles. Finding clusters in tridimensional gene expression datasets. In *BSB 2007 Poster Proceedings*, pages 99–102, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
15. Maurício Mudado and J. Miguel Ortega. On the improvement of transcriptome annotation after clustering and assemblage of incremental number of ESTs. In *BSB 2007 Poster Proceedings*, pages 172–178, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
16. Vinícius Schmitz Pereira Nunes, Alexandre Rossi Paschoal, Clarice Augusta Carvalho Cardoso, Ana Tereza Ribeiro Vasconcelos, and Cláudia Augusta de Moraes Russo. Biota-RIO: a database of animal biodiversity in the state of rio de janeiro. In *BSB 2007 Poster Proceedings*, pages 58–61, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
17. Saulo Pinto and J. Miguel Ortega. Finding normalizers genes by means of homology searches on expressed sequence tags and oligonucleotide array data. In *BSB 2007 Poster Proceedings*, pages 160–171, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
18. Francisco Prosdocimi and J. Miguel Ortega. About a preference for stop-resistant codons in eukaryotic genomes. In *BSB 2007 Poster Proceedings*, pages 138–148, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
19. Francisco Prosdocimi and J. Miguel Ortega. The codon usage of leucine, serine and arginine reveals evolutionary stability of proteomes and protein-coding genes. In *BSB 2007 Poster Proceedings*, pages 149–159, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
20. Daniel Ramiro, Anne Sophie Petitot, Miriam Maluf, and Diana Fernandez. Phylogenetic analysis of the wrky transcription factors gene superfamily in coffee plants. In *BSB 2007 Poster Proceedings*, pages 70–73, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
21. Axel J. Soto, Ignacio Ponzoni, and Gustavo E. Vazquez. Predicting physicochemical properties for drug design using clustering and neural network learning. In *BSB*

210

2007 Poster Proceedings, pages 46–57, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).

22. Marc Sporleder, Octavio Zegarra Aliaga, Vilma Hualla Mamani, Reinhard Simon, and Jürgen Kroschel. *Phthorimaea operculella* granuloviruse: sequence analysis of 5 genes from, 16 geographical isolates. In *BSB 2007 Poster Proceedings*, pages 40–45, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
23. Alan Talevi, Carolina L. Bellera, and Luis E. Bruno-Blanch. Topological indices and graph theory: a useful tool for the characterization of peptides? In *BSB 2007 Poster Proceedings*, pages 28–38, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).
24. Martha Torres, Cristiano Vieira, Glauber Gonalves, and Zilton Junior. IGRAFU, a user-friendly tool based on clusters of pcs for reconstructing phylogenetic trees. In *BSB 2007 Poster Proceedings*, pages 115–127, Angra dos Reis, Brasil, August 2007. Held in conjunction with the International Workshop on Genomic Databases (IWGD 2007).