Bioinformatics
Volume 21, Number 23, December 1, 2005

LETTER TO THE EDITOR:

DATA AND TEXT MINING:

Christian Blaschke, Alexander Yeh, Evelyn Camon, Marc Colosimo, Rolf Apweiler, Lynette Hirschman, and Alfonso Valencia

Do you do text?

Bioinformatics Advance Access published on September 29, 2005
Bioinformatics 2005 21: 4199-4200; doi:10.1093/bioinformatics/bti695

DISCOVERY NOTE:

STRUCTURAL BIOINFORMATICS:

Ruta Furmonaviciene, Brian J. Sutton, Fabian Glaser, Charlie A. Laughton, Nick Jones, Herb F. Sewell, and Farouk Shakib

An attempt to define allergen-specific molecular surface features: a bioinformatic approach

Bioinformatics Advance Access published on October 4, 2005
Bioinformatics 2005 21: 4201-4204; doi:10.1093/bioinformatics/bti700

SYSTEMS BIOLOGY:

Shinichiro Wachi, Ken Yoneda, and Reen Wu

Interactome-transcriptome analysis reveals the high centrality of genes differentially expressed in lung cancer tissues

Bioinformatics Advance Access published on September 27, 2005
Bioinformatics 2005 21: 4205-4208; doi:10.1093/bioinformatics/bti688

ORIGINAL PAPERS:

GENOME ANALYSIS:

Frédéric Boyer, Anne Morgat, Laurent Labarre, Joël Pothier, and Alain Viari

Syntons, metabolons and interactons: an exact graph-theoretical approach for exploring neighbourhood between genomic and functional data

Bioinformatics Advance Access published on October 10, 2005
Bioinformatics 2005 21: 4209-4215; doi:10.1093/bioinformatics/bti711

SEQUENCE ANALYSIS:

- Eddo Kim and Yossef Kliger
  Discovering hidden viral piracy
  Bioinformatics Advance Access published on October 6, 2005
  Bioinformatics 2005 21: 4216-4222; doi:10.1093/bioinformatics/bti706

- Keun-Joon Park, M. Michael Gromiha, Paul Horton, and Makiko Suwa
  Discrimination of outer membrane proteins using support vector machines
  Bioinformatics Advance Access published on October 4, 2005
  Bioinformatics 2005 21: 4223-4229; doi:10.1093/bioinformatics/bti697

STRUCTURAL BIOINFORMATICS:

- Robert W. Janes
  Bioinformatics analyses of circular dichroism protein reference databases
  Bioinformatics Advance Access published on September 27, 2005
  Bioinformatics 2005 21: 4230-4238; doi:10.1093/bioinformatics/bti690

- Huzefa Rangwala and George Karypis
  Profile-based direct kernels for remote homology detection and fold recognition
  Bioinformatics Advance Access published on September 27, 2005
  Bioinformatics 2005 21: 4239-4247; doi:10.1093/bioinformatics/bti687

- Björn Wallner and Arne Elofsson
  Pcons5: combining consensus, structural evaluation and fold recognition scores
  Bioinformatics Advance Access published on October 4, 2005
  Bioinformatics 2005 21: 4248-4254; doi:10.1093/bioinformatics/bti702

GENE EXPRESSION:

- Hassan M. Fathallah-Shaykh
  Noise and rank-dependent geometrical filter improves sensitivity of highly specific discovery by microarrays
  Bioinformatics Advance Access published on September 22, 2005
  Bioinformatics 2005 21: 4255-4262; doi:10.1093/bioinformatics/bti684

- Stan Pounds and Cheng Cheng
  Sample size determination for the false discovery rate
  Bioinformatics Advance Access published on October 4, 2005
  Bioinformatics 2005 21: 4263-4271; doi:10.1093/bioinformatics/bti699

Ida Scheel, Magne Aldrin, Ingrid K. Glad, Ragnhild Sørum, Heidi Lyng, and Arnoldo Frigessi

The influence of missing value imputation on detection of differentially expressed genes from microarray data

Bioinformatics Advance Access published on October 10, 2005
Bioinformatics 2005 21: 4272-4279; doi:10.1093/bioinformatics/bti708

Yang Xie, Wei Pan, and Arkady B. Khodursky

A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data

Bioinformatics Advance Access published on September 27, 2005
Bioinformatics 2005 21: 4280-4288; doi:10.1093/bioinformatics/bti685

SYSTEMS BIOLOGY:

Maris Lapinsh, Peteris Prusis, Staffan Uhlén, and Jarl E. S. Wikberg

Improved approach for proteochemometrics modeling: application to organic compound—amine G protein-coupled receptor interactions

Bioinformatics Advance Access published on October 4, 2005
Bioinformatics 2005 21: 4289-4296; doi:10.1093/bioinformatics/bti703

DATABASES AND ONTOLOGIES:

Andrew C. R. Martin

Mapping PDB chains to UniProtKB entries

Bioinformatics Advance Access published on September 27, 2005
Bioinformatics 2005 21: 4297-4301; doi:10.1093/bioinformatics/bti694

APPLICATIONS NOTE:

GENOME ANALYSIS:

Alberto M. R. Dávila, Daniel M. Lorenzini, Pablo N. Mendes, Thiago S. Satake, Gabriel R. Sousa, Linair M. Campos, Camila J. Mazzoni, Glauber Wagner, Paulo F. Pires, Edmundo C. Grisard, Maria C. R. Cavalcanti, and Maria Luiza M. Campos

GARSA: genomic analysis resources for sequence annotation

Bioinformatics Advance Access published on October 6, 2005
Bioinformatics 2005 21: 4302-4303; doi:10.1093/bioinformatics/bti705

SEQUENCE ANALYSIS:

Tobias Hindemitt and Klaus F. X. Mayer

CREDO: a web-based tool for computational detection of conserved sequence motifs in noncoding sequences

Bioinformatics Advance Access published on October 4, 2005
Bioinformatics 2005 21: 4304-4306; doi:10.1093/bioinformatics/bti691

GENETICS AND POPULATION ANALYSIS:

Hongyu Yang and Alan R. Gingle
OxfordGrid: a web interface for pairwise comparative map views
Bioinformatics Advance Access published on October 4, 2005
Bioinformatics 2005 21: 4307-4308; doi:10.1093/bioinformatics/bti698

Giovanni Montana
HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients
Bioinformatics Advance Access published on September 27, 2005
Bioinformatics 2005 21: 4309-4311; doi:10.1093/bioinformatics/bti689

DATABASES AND ONTOLOGIES:

Jun Cai, Jing Zhang, Ying Huang, and Yanda Li
ATID: a web-oriented database for collection of publicly available alternative translational initiation events
Bioinformatics Advance Access published on October 10, 2005
Bioinformatics 2005 21: 4312-4314; doi:10.1093/bioinformatics/bti704

Gemma L. Holliday, Gail J. Bartlett, Daniel E. Almonacid, Noel M. O'Boyle, Peter Murray-Rust, Janet M. Thornton, and John B. O. Mitchell
MACiE: a database of enzyme reaction mechanisms
Bioinformatics Advance Access published on September 27, 2005
Bioinformatics 2005 21: 4315-4316; doi:10.1093/bioinformatics/bti693

Erratum
Bioinformatics 2005 21: 4317; doi:10.1093/bioinformatics/bti734 **Erratum**
Bioinformatics 2005 21: 4318; doi:10.1093/bioinformatics/bti735

BIOINFORMATICS LETTER TO THE EDITOR Vol. 21 no. 23 2005, pages 4199–4200
doi:10.1093/bioinformatics/bti695

Data and text mining

Do you do text?

Christian Blaschke[1], Alexander Yeh[2], Evelyn Camon[3], Marc Colosimo[2], Rolf Apweiler[3], Lynette Hirschman[2] and Alfonso Valencia[1,*]

[1]Centro Nacional de Biotecnología, CNB-CSIC, Cantoblanco, Madrid, Spain, [2]The MITRE Corporation, Bedford, MA, USA and [3]EBI-EMBL, Hinxton Campus, UK

Received on May 4, 2005; revised on September 22, 2005; accepted on September 24, 2005
Advance Access publication September 29, 2005

Retrieving information from text has become an important area in bioinformatics, and not too surprisingly, this journal has published more than 30 papers on this topic since the first article published by the journal in 1998. In addition, ISCB (International Society for Computational Biology) (www.iscb.org) has organized special sessions in the ISMB conferences (Intelligent Systems for Molecular Biology, see www.iscb.org/ismb2005) and a specialized interest group (www.pdg.cnb.uam.es/BioLink/) for the last five years. In parallel, major computer science conferences in related areas have begun to include sessions on biology, e.g. the TREC (text retrieval conference) Genomics track (medir.ohsu.edu/~genomics), ICML (International Conference on Machine Learning) and a series of workshops organized in association with the ACL (Association for Computation Linguistics) and Human Language Technology meetings.

The requirements of the text mining community are similar to those in other areas of bioinformatics:

- Availability of high quality input information
- A set of objective metrics for the comparison of different methods
- The need to involve the biologist to keep the focus on developing applications that are suitable for end users and biological databases.

Exactly the same issues have been discussed, and partially solved, in the field of protein structure prediction, in part thanks to the organization of CASP (Critical Assessment of techniques for protein Structure Prediction) (predictioncenter.org) during the last 10 years.

With similar evaluation goals in mind, we organized BioCreAtIvE (critical assessment of information extraction in biology), focusing on two tasks. The first dealt with extraction and normalization of gene or protein names from text for three model organism databases (fly, mouse, yeast). The second task addressed issues of extracting functional annotations from text. Overall, 27 groups participated in the assessment (see www.pdg.cnb.uam.es/BioLINK/BioCreative.eval.html and www.mitre.org/public/biocreative/).

The results and the assessment were discussed in a meeting sponsored by EMBO (European Molecular Biology Organization), in Granada, Spain (www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/). The results for gene/protein name extraction showed that at least four groups provided systems that were able to extract gene names from sentences of MEDLINE abstracts at over 80% balanced precision and recall. For the subtask of recognizing the relation between names and normalized database identifiers, the results ranged from a maximum of 92% balanced precision and recall for yeast to 79% for mouse. These results indicate that this technology may now be mature enough to be used in production environments (e.g. document retrieval). However, the results for gene/protein names lag behind those obtained for identifying persons and locations for online news (90–95%). The identification of the many other entities of interest in biology (chemical compounds, tissues, diseases, species and others) will involve additional challenges.

For the functional annotation task, systems were asked to identify a segment of text as evidence for a GO (Gene Ontology) annotation for a given protein in full text articles. In this case, participants were not given training examples of identified text segments. Annotations and text evidence were reviewed by expert annotators from the GO annotation team (www.ebi.ac.uk/GOA/) for validity. When both the protein name and the GO annotation were given, several systems provided correct evidence for the GO predictions 25–30% of the time. The average performances were lower in a subtask in which the GO codes were not given. Interestingly, two systems provided a higher rate of correct predictions by focusing on high confidence cases. These results indicate that the retrieval of functional information is a challenging problem. We believe, however, that this first BioCreAtIvE assessment has laid the foundation for rapid progress in this area by providing an infrastructure, particularly training and test datasets, which will encourage researchers to test their systems against these datasets. A technical description of the results can be found at www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/handout/, and the full collection of methods and evaluation papers has been recently published (*BMC Bioinformatics,* 2005; **6** Suppl. 1). Indeed, BioCreAtIvE also delivered the associated collection of annotated data provided by the organizers and the corresponding evaluations of results from the participating groups (www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/results/). This collection will complement other datasets such as the GENIA corpus (http://www-tsujii.is.s.u-tokyo.ac.jp) as a valuable resource for training and testing methods.

Our main conclusions from the meeting are that:

A number of groups achieved similar levels of performance, using a variety of technologies. These ranged from those based on

*To whom correspondence should be addressed.

natural language processing and linguistic analysis to machine learning and computational biology approaches used in protein structure prediction, gene finding and the like.

An unbiased assessment, based on clear standards and objective evaluation, has provided a more realistic view of the state of the art than has been available to date from more limited evaluations reported in the literature. Setting up the assessment required a considerable effort in the preparation of the datasets, and the evaluation of the results required a considerable effort by human experts. In spite of this, the size and the quality of the available datasets are the main limitation of BioCreAtIvE and other assessment initiatives.

Indeed, a number of groups representing what could be called traditional bioinformatics are experiencing considerable success in the field and we encourage you to consider exploring this new field: have you tried your best in text mining?

Data and additional information are available at www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/results

ACKNOWLEDGEMENTS

The contribution of Alex Morgan, John Wilbur, Lorrie Tanabe and Vivian Lee was essential for the organization and evaluation of BioCreAtIvE. Essential, as well, was the participation of the 27 groups, and their input to the organization, evaluation and discussion of the results. The datasets for the first task were provided by NCBI (National Center for Biotechnology Information) and MITRE and the datasets for the second by EBI–EMBL (European Bioinformatics Institute–European Molecular Biology Laboratory). The MITRE contributions to BioCreAtIvE were supported in part by NSF (grant EIA-0326404), and those to CNB-CSIC and EBI were supported by the European Commission (grants TEMBLOR QLRT-2001-00015 and ORIEL IST-2001-32688).

BIOINFORMATICS DISCOVERY NOTE Vol. 21 no. 23 2005, pages 4201–4204
doi:10.1093/bioinformatics/bti700

Structural bioinformatics

An attempt to define allergen-specific molecular surface features: a bioinformatic approach

Ruta Furmonaviciene[1], Brian J. Sutton[4], Fabian Glaser[5], Charlie A. Laughton[2], Nick Jones[3], Herb F. Sewell[1] and Farouk Shakib[1,*]

[1]Allergy Research Group, Institute of Infection, Immunity and Inflammation, [2]Molecular Recognition Group, School of Pharmacy and [3]Division of Otorhinolaryngology, School of Medical and Surgical Science, The University of Nottingham, Nottingham, UK, [4]The Randall Division of Cell and Molecular Biophysics, King's College London, London, UK and [5]European Bioinformatics Institute, Wellcome Trust Genome Campus, Cambridge, UK

Received on July 11, 2005; revised and accepted September 28, 2005
Advance Access publication October 4, 2005

ABSTRACT

Allergens are proteins that elicit T helper lymphocyte type 2 (Th2) responses culminating in IgE antibody production and allergic disease. However, we have no answer to the fundamental question of why certain proteins are allergens, while others are not. We hypothesized that analysis of the surface of diverse allergens may reveal common structural features which might enable them to be recognized as Th2-inducing antigens by cells of the innate immune system. We have therefore used the ConSurf server to search for allergen-specific motifs. This has enabled us to identify residue conservation patterns in the homologues of Ara t 8 (plant profilin), Act c 1 (actinidin), Bet v 1 (plant pathogenesis-related protein) and Ves v 5 (venom allergen). The results demonstrate the presence of allergen-specific patches consisting of an unusually high proportion of surface-exposed hydrophobic residues. The patches that have been identified may represent molecular patterns recognizable by cells of the innate immune system.

Contact: farouk.shakib@nottingham.ac.uk

Supplementary Information: http://www.nottingham.ac.uk/immunology/research/BI

INTRODUCTION

Allergens are proteins that elicit powerful T helper lymphocyte type 2 (Th2) responses, culminating in IgE antibody production and the development of allergic conditions such as asthma. However, we have no answer to the fundamental question of how allergens make the immune system respond in this way. Work done in the past few years has clearly established the central role of dendritic cells in the induction of Th2-mediated allergic diseases (Eisenbarth *et al*., 2003), but given the innocuous nature of most allergens a major question remains: why do allergen-activated dendritic cells induce Th2, rather than Th1, responses? Thus, there is considerable interest in defining the nature of the molecular surface of allergenic proteins and the mechanisms involved in their initial recognition, and subsequent Th2 cell polarization, by cells of the innate immune system.

Innate immune responses against pathogens are thought to be triggered when a pathogen-associated molecular pattern (PAMP) is recognized by a pattern recognition receptor (PRR) (Janeway, 1989), such as a toll-like receptor (TLR) (Janeway and Medzhitov, 1999; Reis e Sousa, 2001; Takeda *et al*., 2003), leading to the development of protective Th1 responses. PAMPs represent conserved molecular patterns, also called 'molecular signatures', that are essential for the survival of microbes and are often shared by large groups of microorganisms. It is therefore conceivable that allergens too are endowed with a molecular pattern which, when recognized by a PRR, triggers a deleterious Th2 response, culminating in IgE antibody production and allergy.

A number of previous studies used different algorithms to try to predict allergenicity (WHO, 2001; Zorzet *et al*., 2002; Stadler and Stadler, 2003), but these have only searched for sequence motifs, rather than common surface features. An earlier study (Aalberse, 2000) suggested that most allergens could be classified into four structural families, but again it fell short of demonstrating any common surface motifs.

In an attempt to define the molecular surface features of allergens that might enable them to be recognized as Th2-inducing antigens by cells of the innate immune system, we used the ConSurf server (Glaser *et al*., 2003) to identify the conservation patterns and to search for allergen-specific motifs in diverse allergens. The ConSurf server enables the identification of functionally important regions on the surface of a protein or domain, based on the phylogenetic relations between its close sequence homologues, and projects the data onto a representative crystal structure. Here we report our results.

METHODS

The ConSurf web server (Glaser *et al*., 2003) facilitates the identification of patterns of conserved and variable residues in a protein by estimating the degree of conservation of amino acids within its close sequence homologues, and subsequently mapping the conservation scores onto a representative crystal structure. ConSurf requires a minimum number (five) of close homologous sequences of similar length and a representative crystal structure. Using the three-dimensional (3D) structure of a protein as an input, the ConSurf server obtains the sequence from the PDB (Berman *et al*., 2000) file and carries out a search for close homologous sequences of the protein using PSI-BLAST (Altschul *et al*., 1997). This search is based on using the Swiss-Prot database (Bairoch and Apweiler, 1999) and a default single

*To whom correspondence should be addressed at Division of Immunology, Queen's Medical Centre, Nottingham NG7 2UH, UK

iteration of PSI-BLAST with an E-value cutoff of 0.001. The sequences obtained are then aligned using CLUSTALW (Thompson *et al*., 1994) (with default parameters) and a phylogenetic tree is built using the neighbour joining algorithm (Saitou and Nei, 1987), as implemented in the Rate4Site program (Pupko *et al*., 2002). Conservation scores corresponding to the site's evolutionary rate are calculated using the Bayesian method (Mayrose *et al*., 2004) (ConSurf Version 3.0), which is a significant improvement over previous methods, particularly when only a small number of sequences are available. The Bayesian method also assigns a confidence interval to each of the inferred evolutionary conservation scores (Susko *et al*., 2002). The proteins, with their conservation scores colour-coded on their surface, can finally be visualized online using the Protein Explorer engine (Martz, 2002).

The conservation scale used for visualization is obtained through ConSurf by processing the original continuous conservation scores obtained from Rate4Site. The colour grades (from 1 to 9) are assigned as follows: conservation scores below average (i.e. negative values, which are indicative of slowly evolving, conserved sites) are divided into 4.5 equal intervals. The same 4.5 intervals are used for scores above average (i.e. positive values, which are indicative of rapidly evolving, variable sites). Thus, nine equally sized categories of conservation or grades are obtained. Using this procedure, the 'width' (i.e. the maximum and minimum scores) of each colour grade would vary for different polypeptide chains. Thus, the colouring results of a ConSurf calculation do not indicate the absolute magnitudes of evolutionary distances, but rather the relative degrees of conservation for each residue.

RESULTS

Out of the SDAP (Ivanciuc *et al*., 2003) list of allergens (http://fermi.utmb.edu/SDAP), only four allergen groups (95 allergens in total) met the ConSurf utility criteria of having a minimum number of five homologous allergen sequences (*E*-value cutoff of 0.001) of similar length and a representative structure.

All known homologous sequences for each representative crystal structure [Ara t 8/3NUL (1.60 Å) (Thorn *et al*., 1997), Act c 1/2ACT (1.70 Å) (Baker, 1980), Bet v 1/1BV1 (2.00 Å) (Gajhede *et al*., 1996) and Ves v 5/1QNX (1.90 Å) (Henriksen *et al*., 2001)] were searched using the ConSurf server (Glaser *et al*., 2003). The sequences were then divided into allergens and non-allergens. The protein was considered allergenic if referred to as such in the Swiss-Prot/TrEMBL database (ExPaSy site http://ca.expasy.org/sprot/) or in SDAP. Admittedly, however, some of the proteins that were considered non-allergens might be allergenic in some predisposed individuals and this may introduce some noise into the comparisons, but the fact that we are sure about the allergenic group means that we can be certain when looking for conservation patterns in allergens.

The list of allergens and non-allergens used as input for the ConSurf search is shown in Supplementary Table I. There were 37 allergens and 46 non-allergens in the 3NUL group, 9 allergens and 162 non-allergens in the 2ACT group, 27 allergens and 19 non-allergens in the 1BV1 group, and 22 allergens and 51 non-allergens in the 1QNX group. Although, the number of homologues in the 2ACT allergen group is rather small, however it is still acceptable by the ConSurf server and was included in our analysis since a smaller number of non-allergen homologues did not alter the conservation pattern.

The selected protein sequences were examined for common residue patterns using the ConSurf server, searching for common allergen-specific patterns not present in the 'control', non-allergen group. All allergen-specific and accessible residues that are either highly conserved (red colour, levels 8–9) or highly variable (blue colour, levels 1–2), but which have only average scores (white) in the non-allergen sequence, are underlined in Figure 1. Also identified (underlined) are residues that are highly conserved (red) in the allergen sequence but highly variable (blue) in the non-allergen sequence, or vice versa, provided that they are accessible.

Figure 2 shows that allergen-specific surface residues were frequently hydrophobic: 55% in 3NUL, 48% in 2ACT, 39% in 1BV1 and 44% in 1QNX (chain A).

DISCUSSION

Allergic reactions consist of a series of events that start with recognition of the native allergen structure by antigen presenting cells, such as dendritic cells, and culminates in IgE antibody production and mast cell sensitization and triggering. Our hypothesis is that allergens display molecular patterns that are recognized by PRRs on antigen presenting cells. This encounter between the native allergen and the PRRs provides instructive signals for the immune system to mount an IgE antibody response leading to allergy.

The ConSurf server enables the identification of functionally important regions of the surface of a protein of known 3D structure based on the phylogenetic relations between its close sequence homologues. Four groups of allergens (95 allergens in total) were found with sufficient homologous sequences and at least one crystal structure to which this method could be applied. In each group, the sequence of the allergen of known structure was used in the PSI-BLAST search to identify all known allergenic and non-allergenic homologues. The allergen-specific motifs (i.e. clusters of residues in the allergen but not the non-allergen group) that were identified consisted of both highly conserved and highly variable residues. Interestingly, a previous study which identified functionally important surface patches in the MHC class I peptide-binding groove and in the antigenic surfaces of the influenza haemagglutinin has shown that these patches also consisted of both highly conserved and highly variable residues (ConSurf Gallery: http://consurf.tau.ac.il/gallery.html).

The hypothesis behind the present approach is that if there is a set of receptors that have a rather limited repertoire that recognize allergen-associated molecular patterns (by analogy with recognition molecules of innate immune cells such as the PRR) then we might expect to find common features among allergens that consist of highly conserved and also highly variable residues within defined patches. Our search revealed that there were indeed allergen-specific motifs formed by adjacent conserved and variable residues, which, unusually for surface residues, were mostly hydrophobic. Since the four groups studied here constitute only a small sample of allergens, it may be premature to generalize this result. However, this is the first demonstration of an allergen-specific surface feature displayed by structurally and functionally diverse groups of allergens.

The above findings are in line with the recent notion that the innate immune system might have evolved to detect hydrophobic portions of immunogenic proteins (Seong and Matzinger, 2004) consisting of a string of hydrophobic amino acids, rather than being dependent on the exact amino acid sequence (Berezovskyi and Trifono, 2000). Those authors have therefore argued that what

Fig. 1. Residue conservation amongst allergen (upper line) and non-allergen (lower line) sequences belonging to the four allergen groups that meet the ConSurf utility criteria, namely the homologues of Ara t 8, Act c 1, Bet v 1 and Ves v 5. All allergen-specific residues that are either highly conserved (red) or highly variable (blue), but which have only average scores (white) in the non-allergen sequence, are underlined. Also underlined are residues that are highly conserved (red) in the allergen sequence but highly variable (blue) in the non-allergen sequence, or vice-versa, provided that they are accessible. Shown in green are residues (in the 1QNX chain A sequences) that could not be assessed for conservation because of lack of homologous motifs (i.e. present in <10% of the sequences). Residues that are assigned confidence intervals too large to be trustworthy are coloured in yellow.

makes a protein recognizable as an allergen (i.e. Th2-inducing protein) by antigen presenting cells is its hydrophobic nature, as illustrated by allergens such as lipocalins, lipid transfer proteins and seed storage proteins (Seong and Matzinger, 2004). The hydrophobic residues of such proteins are normally buried, but they may become exposed upon unfolding, such as following the loss of a specific ligand that is integrated into their 3D structure. Examples of such ligands include metal ions (e.g. Ca in parvalbumins) and lipids (e.g. MD-2 proteins, lipocalins and lipid transfer proteins) (Breiteneder and Mills, 2005).

The relevance to allergen recognition by dendritic cells of the residues identified here will clearly need to be validated. This could be done by mutagenesis experiments and the mutant proteins tested for allergenicity, or in the longer term by virtual screening for ligands (PRRs) that bind to the putative molecular patterns identified.

3NUL allergens (profilins)
(Hydrophobic 55%, uncharged hydrophilic 18%, acidic 11% and basic 16%)

.SWQSYVDDH L.CDVEGNHL TAAAILGQDG SVWAQSAKFP

QLKPQEIDGI KKDFEEPGFL APTGLFLGGE KY.VIQGEQG

AVIRGKKGPG GVTIKKTNQA LVFGFYDEP. TGGQCNLVVE

RLGDYLIESE L

2ACT allergens (cysteine proteases)
(Hydrophobic 48%, uncharged hydrophilic 33%, acidic 19% and basic 0%)

LPSYVDWRSA GAVVDIKSQG ECGGCWAFSA IATVEGINKI

TSGSLISLSE QELIDCGRTQ NTRGCDGGYI TDGFQFIIND

GGINTEENYP YTAQDGDCDV ALQDQKYVTI DTYENVPYNN

EWALQTAVTY QPVSVALDAA GDAFKQYASG IFTGPCGTAV

DHAIVIVGYG TEGGVDYWIV KNSWDTTWGE EGYMRILRNV

GGAGTCGIAT MPSYPVKY

1BV1 allergens (plant pathogenesis-related proteins)
(Hydrophobic 39%, uncharged hydrophilic 8%, acidic 23% and basic 31%)

GVFNYETETT SVIPAARLFK AFILDGDNLF PKVAPQAISS

VENIEGNGGP GTIKKISFPE GLPFKYVKDR VDEVDHTNFK

YNYSVIEGGP IGDTLEKISN EIKIVATPDG GSILKISNKY

HTKGDHEVKA EQVKASKEMG ETLLRAVESY LLAHSDAYN

1QNX(A) allergens (venom allergens)
(Hydrophobic 44%, uncharged hydrophilic 23%, acidic 9% and basic 24%)

.....AEAEF NNYCKIKCLK GGVHTACKYG SLKPNCGNKV

VVSYGLTKQE KQDILKEHND FRQKIARGLE TRGNPGPQPP

AKNMKNLVWN DELAYVAQVW ANQCQYGHDT CRDVAKYQVG

QNVALTGSTA AKYDDPVKLV KMWEDEVKDY NPKKKFSGND

FLKTGHYTQM VWANTKEVGC GSIKYIQEKW HKHYLVCNYG

PSGNFKNEEL YQTK

Colour-coding:

VALMGIPFC	RED	Hydrophobic
DE	BLUE	Acidic
RHK	MAGENTA	Basic
STYNQ	GREEN	Uncharged hydrophilic

Fig. 2. The nature of residues constituting allergen-specific surface patches (underlined) found on the crystal structures representing the four allergen groups that met the ConSurf utility criteria. Allergen-specific surface residues are mostly hydrophobic (shown in red).

ACKNOWLEDGEMENTS

This work was primarily funded by Asthma UK (London; Grant ID 02/005) and partially supported by the Nasal Research Fund (Nottingham; Grant ID 7365).

Conflict of Interest: none declared.

REFERENCES

Aalberse,R.C. (2000) Structural biology of allergens. *J. Allergy Clin. Immunol.*, **106**, 228–238.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Bairoch,A. and Apweiler,R. (1999) The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Res.*, **27**, 49–54.

Baker,E.N. Structure of actinidin, after refinement at 1.7 A resolution. *J. Mol. Biol.*, **141**, 441–484.

Berezovskyi,I.N. and Trifono,E. (2000) Protein structure and folding: a new start. *J. Biomol. Struct. Dyn.*, **19**, 397–403.

Berman,H.M. *et al.* (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.

Breiteneder,H. and Mills,E.N. (2005) Molecular properties of food allergens. *J. Allergy Clin. Immunol.*, **115**, 14–23.

Eisenbarth,S.C. *et al.* (2003) The master regulators of allergic inflammation: dendritic cells in Th2 sensitization. *Curr. Opin. Immunol.*, **15**, 620–626.

Gajhede,M. *et al.* (1996) X-ray and NMR structure of Bet v 1, the origin of birch pollen allergy. *Nat. Struct. Biol.*, **3**, 1040–1045.

Glaser,F. *et al.* (2003) ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics*, **19**, 163–164.

Henriksen,A. *et al.* (2001) Major venom allergen of yellow jackets, Ves v 5: structural characterization of a pathogenesis-related protein superfamily. *Proteins*, **45**, 438–448.

Ivanciuc,O. *et al.* (2003) SDAP: database and computational tools for allergenic proteins. *Nucleic Acids Res.*, **31**, 359–362.

Janeway,C.A.,Jr (1989) Approaching the asymptote? Evolution and revolution in immunology *Cold Spring Harb. Symp. Quant. Biol.*, **54**, 1–13.

Janeway,C.A.,Jr and Medzhitov,R. (1999) Lipoproteins take their toll on the host. *Curr. Biol.*, **9**, 879–882.

Martz,E. (2002) Protein explorer: easy yet powerful macromolecular visualization. *Trends Biochem. Sci.*, **27**, 107–109.

Mayrose,I. *et al.* (2004) Comparison of site-specific rate-inference methods for protein sequences: empirical Bayesian methods are superior. *Mol. Biol. Evol.*, **21**, 1781–1791.

Pupko,T. *et al.* Rate4Site: an algorithmic tool for the identification of functional regions on proteins by surface mapping of evolutionary determinants within their homologues. *Bioinformatics*, **18** (Suppl.), S71–77.

Reis e Sousa,C. (2001) Dendritic cells as sensors of infection. *Immunity*, **14**, 495–498.

Saitou,N. and Nei,M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–425.

Seong,S.Y. and Matzinger,P. (2004) Hydrophobicity: an ancient damage-associated molecular pattern that initiates innate immune responses. *Nat. Rev. Immunol.*, **4**, 469–478.

Stadler,M.B. and Stadler,B.M. (2003) Allergenicity prediction by protein sequence. *FASEB J.*, **17**, 1141–1143.

Susko,E. *et al.* (2002) Testing for differences in rates-across-sites distributions in phylogenetic subtrees. *Mol. Biol. Evol.*, **19**, 1514–1523.

Takeda,K. *et al.* (2003) Toll-like receptors. *Annu. Rev. Immunol.*, **21**, 335–376.

Thompson,J.D. *et al.* (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Thorn,K.S. *et al.* (1997) The crystal structure of a major allergen from plants. *Structure*, **15**, 19–32.

WHO (2001), Evaluation of allergenicity of genetically modified foods. *Report of a Joint FAO/WHO Expert Consultation. World Health Organization. Geneva.*

Zorzet,A. *et al.* (2002) Prediction of food protein allergenicity: a bioinformatic learning systems approach. *In Silico Biol.*, **2**, 1–10.

BIOINFORMATICS DISCOVERY NOTE Vol. 21 no. 23 2005, pages 4205–4208
doi:10.1093/bioinformatics/bti688

Systems biology

Interactome-transcriptome analysis reveals the high centrality of genes differentially expressed in lung cancer tissues

Shinichiro Wachi*, Ken Yoneda and Reen Wu

Center for Comparative Respiratory Biology and Medicine and Division of Pulmonary/Critical Care Medicine, University of California, Davis, CA 95616, USA

Received on June 9, 2005; revised on September 7, 2005; accepted on September 21, 2005
Advance Access publication September 27, 2005

ABSTRACT

Motivation: Global protein interaction network (interactome) analysis provides an effective way to understand the relationships between genes. Through this approach, it was demonstrated that the essential genes in yeast tend to be highly connected as well as connected to other highly connected genes. This is in contrast to the genes that are not essential, which share neither of these properties. Using a similar interactome-transcriptome approach, the topological features in the interactome of differentially expressed genes in lung squamous cancer tissues are assessed.

Results: This analysis reveals that the genes that are differentially elevated, as obtained from the microarray gene profiling data, in cancer are well connected, whereas the suppressed genes and randomly selected ones are less so. These results support the notion that a topological analysis of cancer genes using protein interaction data will allow the placement of the list of genes, often of the disparate nature, into the global, systematic context of the cell. The result of this type of analysis may provide the rationale for therapeutic targets in cancer treatment.

Contact: swachi@ucdavis.edu

Supplementary information: Supplementary data for this paper are available on *Bioinformatics* online.

1 INTRODUCTION

To adapt a famous passage of John Donne, no gene is an island. Systems biology, or more specifically network biology, is driven by the gradual realization that a single gene is seldom accountable for a discrete biological function (Barabasi and Oltvai, 2004). In response, contemporary biology has amassed a battery of methods to survey the global features of the cells, from DNA, RNA and proteins to small molecules (Ideker *et al*., 2001; Kitano, 2002).

Transcriptome analysis, enabled by technology such as oligonucleotide microarray, is a simultaneous interrogation of gene expression by measuring the transcriptional activity on a global scale (Lockhart *et al*., 1996). Microarray, as well as the genome project, were the first forays of humanity into the realm of systems biology. Other than the fact that microarray analysis suffers from inherently noisy information (therefore requiring many replicates and often limited by cost and availability of materials), the other problem is the sheer volume of information obtained from this type of experiment (Claverie, 1999). Furthermore, the expression level change of a gene may be corollary to change in another gene and may not be the direct cause of the cellular phenotype. Additional information is required to place these genes in context.

Interactome analysis is a study of interactions between the biological molecules on a global scale (Ito *et al*., 2001). High-throughput mapping of protein interaction allows the global survey of protein interaction of organisms. The resulting maps of proteome-wide protein interactions are called protein networks.

Topological features of the protein networks have been demonstrated to reflect the functionality of the interacting genes. For example, essential genes in yeast tend to be well connected and globally centered in the protein network (Jeong *et al*., 2001; Wuchty and Almaas, 2005). Furthermore, globally centered interactions are more likely to be well conserved and serve as an evolutionary backbone for the network (Wuchty and Almaas, 2005).

Protein network analysis will place the genes identified in microarray experiments in a broader biological context. Since protein networks reflect the functional grouping of these interacting or coordinately induced/suppressed genes, the roles of the subsets of co-expressed genes may be resolved using the combined data. This may be done by evaluating the topological features of the sets of genes identified by microarray experiments.

Studies in cancer have validated the effectiveness of the microarray technique, allowing identification of tumor subclasses and marker genes for diagnosis and treatment of the disease (DeRisi *et al*., 1996; Sorlie *et al*., 2001). However, the genes identified have no further association with other genes that are co-regulated. Integration of protein network data may extend the reach of the established method of analysis by considering the genes in a broader context. Based on this notion, we seek to reveal the biological significance of differentially expressed genes in squamous cell lung cancer that is identified through our recent microarray gene expression profiling study by using interactome-transcriptome analysis. We find high centrality in these differentially induced genes, but not for the genes that are suppressed in cancer.

*To whom correspondence should be addressed.

2 MATERIALS AND METHODS

2.1 Sample preparation

Tissue biopsy samples were collected from five squamous cell lung cancer patients undergoing surgical removal of tumor. Total RNA was doubly extracted from these samples using TRIzol™ reagent (Invitrogen, Carlsbad, CA), following the manufacturer's instructions (Chomczynski and Sacchi, 1987).

2.2 Microarray analysis

The double-extracted total RNA was submitted to our Institute's core microarray facility. cRNA samples were prepared and hybridized to the array (Affymetrix® Hg-U133A™); its signals were then scanned using the protocols suggested by the manufacturer.

Bioconductor (Dudoit *et al*., 2003), a biological data analysis package based on R statistical programming language (Ihaka and Gentleman, 1996), was used for array data analysis and integration with other gene annotations. Robust microarray analysis (RMA) was used for normalization (Irizarry *et al*., 2003). RMA-derived expression values were used for the rest of the analysis (Galfalvy *et al*., 2003). Paired *t*-tests were used to distinguish the genes in which expression levels in the cancer cells differed from the paired normal lung tissue. Paired *t*-tests were possible because the control samples were taken from the normal lung tissues of the same individuals from whom the lung cancer samples were obtained.

2.3 Integration of array data to protein network

Online predicted human interaction database (OPHID, obtained on April 26, 2005) was used for the analysis of human protein interaction (Brown and Jurisica, 2005). Briefly, OPHID contains 16 034 known human protein interactions obtained from various public protein interaction databases, as well as 23 889 additional protein interactions that are predicted.

Genes in the array were matched to those in OPHID using gene symbols and protein sequences. In order to identify which genes in OPHID corresponded to which genes listed in the array data, we have used the following methods: (1) using the gene symbol that is directly indicated in the OPHID protein database and (2) using FASTA program to compare the peptide sequence of the OPHID protein to the peptide sequence of the array probe targets (Pearson and Lipman, 1988). As a result, 2137 genes on the microarray were matched to the protein network from OPHID.

2.4 Connectivity analysis of protein interaction map

Analysis was conducted on the connectivity of genes in the protein interaction map. For each connectivity l, genes in the protein network with exactly l links were selected (n_l). From these genes, differentially expressed genes (DEGs) were counted (n_l'). The fraction n_l' / n_l was calculated (fraction genes, or FG) for each connectivity. Pearson's r was used to measure the correlation between the FG and the connectivity. In order to determine the significance of the correlation, the same number of genes that were differentially expressed was randomly chosen from the protein interaction map. Pearson's r of each set of randomly sampled genes was compared against the r obtained from DEGs to determine the likelihood that the correlation can occur by chance.

2.5 *k*-core analysis of protein interaction map

k-core analysis is an iterative process in which the nodes are removed from the graphs in order of least connected (Wuchty and Almaas, 2005). More specifically, for each iteration of k, given the network from the previous iteration, genes with less than k connections are removed from the graph. This will result in a series of subgraphs that gradually reveal the globally central region of the original network.

In order to measure the centrality of the selected set of genes, the excess retention (ER) of the differentially expressed genes was calculated for each *k*-core. ER is a measure of the degree to which proteins from a particular group are represented relative to the entire protein network. The detailed explanation of ER has been described elsewhere (Wuchty and Almaas, 2005).

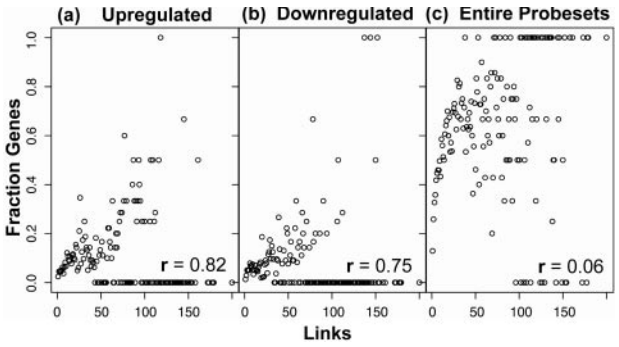

Fig. 1. Correlation of connectivity (links) versus the fractions of select genes with exactly l links. The genes with exactly l links were chosen from the protein network, and then the fraction of selected genes from this subset was calculated. (**a**) Upregulated genes ($n = 360$) in SCC of lung have Pearson's r, which demonstrates high positive correlation ($r = 0.82$). (**b**) Downregulated genes ($n = 270$) in SCC have slightly less correlation ($r = 0.75$). (**c**) Microarray probesets that match the genes in the protein network ($n = 2137$) show no correlation to link number ($r = 0.06$). Thus, using the genes on the microarray does not contribute to bias in the number of links for genes differentially expressed in SCC. (FG values of 1 and 0 are not excluded for r values.)

3 RESULTS

3.1 Expression profiles of lung cancer

To obtain a list of genes that are differentially expressed in cancer, tissue samples from five patients with squamous cell carcinoma (SCC) of the lung were collected. Additionally, an equal number of normal tissues surrounding the tumor were collected from the same patients for comparison. This control minimizes the effect of variation in gene expression between the individuals, yielding a more accurate characterization of the genes differentially expressed in the disease.

Following the normalization of the array data from these samples, genes that were consistently different from normal tissue (paired *t*-test, $P < 0.05$) were selected as DEG. Furthermore, only the genes that can be mapped to the existing protein network were chosen. As a result, 360 DEGs were chosen as genes that were upregulated in squamous carcinoma, whereas 270 DEGs were chosen as genes that were downregulated in squamous carcinoma (see online supplement for the list of genes). DEGs were subsequently mapped to protein interaction maps for further analysis.

3.2 Topological analysis of lung cancer genes

In order to determine the topological features of the DEGs, the edge distribution for DEGs was compared to the rest of the graph (Section 2.4). We find that the genes that are upregulated in lung cancer have a positive correlation with the number of edges associated with them (Fig. 1a). This positive correlation indicates that lung cancer DEGs that are upregulated are highly connected. Downregulated genes have a slightly lower, but positive correlation to connectivity (Fig. 1b).

Because only a subset of genes in the graph is present on the microarray that was used, bias of the selected DEGs is of concern.

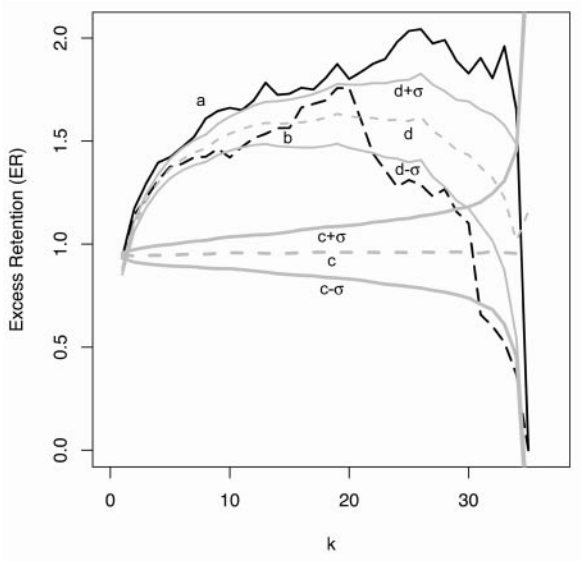

Fig. 2. *k*-core analysis of differentially expressed genes in SCC of lung. *k* is the number of iterative decomposition of the graph (*k*-core subgraph), starting from the outermost edge. Higher *k* represents the central placement of the subset of genes in the original graph ($k = 1$). ER is the degree to which genes from a *k*-core is under- (ER < 1) or over-represented (ER > 1) relative to the original graph. a: ER of the upregulated genes ($n = 360$) in SCC; b: ER of the downregulated genes ($n = 270$) in SCC; c, c + σ, c − σ: mean ER of 1000 randomly selected genes from the entire network ($n = 360$), plus or minus the standard deviation ($\pm\sigma$); d, d + σ, d − σ: mean ER and $\pm\sigma$ for the 1000 randomly selected genes ($n = 360$) present in the microarray used in the experiment that matches the genes in the graph ($n = 2137$).

However, there is no correlation between the genes in the array to the connectivity in the protein network (Fig. 1c). Thus, the gene representation of the microarray itself poses no bias to the connectivity of the DEGs.

We then randomly selected the genes from the graph to see if we can get a high positive correlation. The same number of genes as DEGs was randomly sampled from genes that match the array. Then the correlation was calculated and compared the correlation to that of the DEGs. For the upregulated genes, average of 7.7% equaled or exceeded *r* of DEGs ($\sigma = 1.8$, $n = 11$). For the downregulated genes, average of 86.2% equaled or exceeded *r* of DEGs ($\sigma = 0.8$, $n = 11$). Since the only difference between the two analyses is the number of genes sampled (370 for upregulated genes and 270 for downregulated genes), the significance value is highly sensitive to the number of genes selected. Thus, one can conclude that, at a minimum, for the upregulated genes the positive correlation between the connectivity and the DEGs is not likely due to chance.

3.3 *k*-core analysis of protein network

Using the *k*-core analysis method, we have measured how DEGs were close to the topological center of the protein network (Fig. 2). In this analysis, an increase in excess retention as value of *k* is increased indicates that the selected group of genes is located near the topological center of the protein network. Higher the *k* the ER is maintained, the closer the genes are to the center of the network. For the upregulated genes, the excess retention (ER) increases as *k* approaches 30, then drops precipitously around 35 (Fig. 2a). For the downregulated genes, ER increases as *k* approaches 20 and then descends beyond this point (Fig. 2b). The early peak of the downregulated genes relative to that of the upregulated genes indicates that the upregulated genes are more centrally located in the protein network than the downregulated genes.

In order to test the significance of this finding, ER has been calculated for the same number of nodes that has been randomly selected from the graph. When genes were chosen from the entire graph, ER remained near unity, on average (Fig. 2c). However, when genes that correspond to the microarray probesets were chosen, ER did increase on average (Fig. 2d) up to $k = 20$. A distinct difference in the ER for each *k*-core can be seen between cancer gene and random control (average of 7 out of 1000 randomly chosen genes out of array probes are above the ER for upregulated genes where $1 < k < 35$). Moreover, it is apparent that downregulated genes do not show a significant difference in ER against the genes that were randomly chosen from the arrays. From these results, we can conclude that the upregulated genes are centrally located in the protein network, but the same conclusion cannot be made for the downregulated genes.

4 DISCUSSION

Transcriptional profiling of cancer using microarray has revolutionized the field by allowing researchers to discover tumor subclasses and target genes for diagnosis and therapy. Protein interaction mapping using high-throughput yeast-two-hybrid (HT-Y2H) methods has been hailed as the harbinger for the systematic approach to functional genomics, allowing individual genes to be placed in a global context of cellular functions (Mendelsohn and Brent, 1999). This leads to the question how are genes that are differentially regulated in cancer placed in the global context of the protein interaction map? In this study, we exploit the most recent advances in interactome analysis to answer this question.

Much of the biological diversity of tumors is the result of variation in the transcriptional programs (Perou *et al*., 2000). Microarray analysis of cancer cells has allowed the identification of specific genes or proteins that could serve as molecular targets for improved diagnosis and therapy. The subsequent interpretation of the genes identified by the microarray analysis was analyzed in a context where a single gene was accounted for the individual phenotype. However, this ignores the fact that proteins rarely act in isolation from one another, and their specific functions are determined by association with other proteins.

A rational starting point for taking the proteins into proper context is to use the protein network or interactome. Interactome analysis has made a remarkable progress in the past decade, mainly due to the development of high-throughput screening methods such as HT-Y2H. To date, interactomes of three eukaryotes have been mapped by HT-Y2H (Giot *et al*., 2003; Li *et al*., 2004; Uetz *et al*., 2000). Due to high cost and labor, human HT-Y2H maps may not be available for some time. Predicted mapping of human protein interaction can serve as a surrogate map, years before the complete human protein network based on direct experimental evidence becomes available.

Hypothetical human protein interaction maps are relatively new (Brown and Jurisica, 2005; Lehner and Fraser, 2004). The prediction of protein interaction is based on sequence-based searches for conserved protein interactions, also known as interologs (Matthews

et al., 2001). Because these human interaction maps are hypothetical, it is likely to have many false positives as well as missing protein interactions. There is a claim that approximately half of the predicted interactions using interologs between microorganisms can be experimentally validated (Sharan *et al*., 2005). While the lack of accuracy in the interologs approach may instill little confidence in the predicted map, it is easy to envision its role in the validation of the HT-Y2H global protein network mapping.

The topological analyses using a hypothetical human protein interaction map suggest that the genes upregulated in the squamous lung carcinomas are rich in global hubs. Global hubs are well-connected nodes that are also located near the center of the protein network. Evidence is lacking to make the same claim to the downregulated genes. Centrality of the genes is associated with the essential functions of the genes in the yeast (Jeong *et al*., 2001). It has been shown that the essential genes, those that are lethal when mutated, tend to be well connected. Another study shows that the yeast genes that are not essential but provide a vital function in toxin metabolism also have high number of edges associated with the nodes, albeit less well-connected than that of the essential genes (Said *et al*., 2004). *k*-core analysis has been performed on the yeast essential genes and were shown to be global hubs, whereas non-essential genes were not (Wuchty and Almaas, 2005). The study also indicates that these global hubs are conserved throughout different species.

One can surmise, from the centrality of the cancer-associated genes, the evolutionary constraints to the genes expressed in cancer cells. Whatever the mechanism, there must be a core set of genes that needs to be maintained throughout the course of somatic evolution in the tumor microenvironment. These genes are the 'essential genes' for the cancer and share the topological characteristics that the essential genes do in yeast. This may be manifested by the upregulated genes in our cancer samples that tend to be, but not restricted to, the global hubs. Downregulated genes tend to be less constrained in this way, either because the genes are suppressed by the upregulated genes (secondary effect of the core cancerous gene expression) or because the suppressed genes are overwhelmed by circumvention from the alternate or compensatory pathway that induces the upregulation of the cancerous genes.

Our analysis concludes that squamous cell lung cancer genes share similar topological features for essential proteins. This finding fulfills the somatic evolution model of cancer. Cancerous cells undergo frequent mutation, division and selection, ultimately leading to the fittest 'disregulated' phenotype. Given this, it is reasonable to assume that the genes, which are differentially expressed in contrast to the surrounding normal tissue, are essential for survival and proliferation. Thus, given that the centrality of the genes in a protein network is characteristic of the essential genes in yeast, it follows that the genes essential for cancer cells would also share the topological characteristics of these essential genes.

This is the first time the predicted human protein interaction map has been used for the analysis of cancer. While microarray analysis has been used extensively for the identification of marker genes for various types of cancer, our approach may facilitate the development of drugs that target the genes that may be directly responsible for the disease.

ACKNOWLEDGEMENTS

The authors thank Kenneth Chmiel for technical support and Kento Oki for critical reading of the manuscript. This work was supported in part by the National Institutes of Health (HL35635, HL077315 and HL077902).

Conflict of Interest: none declared.

REFERENCES

Barabasi,A.L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet*., **5**, 101–113.

Brown,K.R. and Jurisica,I. (2005) Online predicted human interaction database. *Bioinformatics*, **21**, 2076–2082.

Chomczynski,P. and Sacchi,N. (1987) Single-step method of RNA isolation by acid guanidinium thiocyanate-phenol-chloroform extraction. *Anal. Biochem*., **162**, 156–159.

Claverie,J.M. (1999) Computational methods for the identification of differential and coordinated gene expression. *Hum. Mol. Genet*., **8**, 1821–1832.

DeRisi,J. *et al*. (1996) Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nat. Genet*., **14**, 457–460.

Dudoit,S. *et al*. (2003) Open source software for the analysis of microarray data. *Biotechniques* (Suppl), 45–51.

Galfalvy,H.C. *et al*. (2003) Sex genes for genomic analysis in human brain: internal controls for comparison of probe level data extraction. *BMC Bioinformatics*, **4**, 37.

Giot,L. *et al*. (2003) A protein interaction map of *Drosophila melanogaster*. *Science*, **302**, 1727–1736.

Ideker,T. *et al*. (2001) A new approach to decoding life: systems biology. *Annu. Rev. Genomics Hum. Genet*., **2**, 343–372.

Ihaka,R. and Gentleman,R. (1996) R: a language for data analysis and graphics. *J. Comput. Graph. Stat*., **5**, 299–314.

Irizarry,R.A. *et al*. (2003) Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res*., **31**, e15.

Ito,T. *et al*. (2001) Exploring the protein interactome using comprehensive two-hybrid projects. *Trends Biotechnol*., **19**, S23–S27.

Jeong,H. *et al*. (2001) Lethality and centrality in protein networks. *Nature*, **411**, 41–42.

Kitano,H. (2002) Systems biology: a brief overview. *Science*, **295**, 1662–1664.

Lehner,B. and Fraser,A.G. (2004) A first-draft human protein-interaction map. *Genome Biol*., **5**, R63.

Li,S. *et al*. (2004) A map of the interactome network of the metazoan *C. elegans*. *Science*, **303**, 540–543.

Lockhart,D.J. *et al*. (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat. Biotechnol*., **14**, 1675–1680.

Matthews,L.R. *et al*. (2001) Identification of potential interaction networks using sequence-based searches for conserved protein–protein interactions or 'interologs'. *Genome Res*., **11**, 2120–2126.

Mendelsohn,A.R. and Brent,R. (1999) Protein interaction methods—toward an endgame. *Science*, **284**, 1948–1950.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Perou,C.M. *et al*. (2000) Molecular portraits of human breast tumours. *Nature*, **406**, 747–752.

Said,M.R. *et al*. (2004) Global network analysis of phenotypic effects: protein networks and toxicity modulation in *Saccharomyces cerevisiae*. *Proc. Natl Acad. Sci. USA*, **101**, 18006–18011.

Sharan,R. *et al*. (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA*, **102**, 1974–1979.

Sorlie,T. *et al*. (2001) Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications,. *Proc. Natl Acad. Sci. USA*, **98**, 10869–10874.

Uetz,P. *et al*. (2000) A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**, 623–627.

Wuchty,S. and Almaas,E. (2005) Peeling the yeast protein network. *Proteomics*, **5**, 444–449.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4209–4215
doi:10.1093/bioinformatics/bti711

Genome analysis

Syntons, metabolons and interactons: an exact graph-theoretical approach for exploring neighbourhood between genomic and functional data

Frédéric Boyer[1], Anne Morgat[2], Laurent Labarre[3], Joël Pothier[4] and Alain Viari[1,*]

[1]INRIA Rhône-Alpes, HELIX Group, 655 avenue de l'Europe, 38334 Montbonnot Cedex, France, [2]Swiss Institute of Bioinformatics, Swiss-Prot Group, 1 rue Michel Servet, CH-1211 Geneva, Switzerland, [3]UMR 8030, Genoscope, Centre National de Séquençage, 2 rue Gaston Crémieux, CP5706–91057 Evry cedex, France and [4]Atelier de BioInformatique, Université Paris VI, 12 rue Cuvier, 75005 Paris, France

Received on June 1, 2005; revised on July 22, 2005; accepted on October 6, 2005
Advance Access publication October 10, 2005

ABSTRACT

Motivation: Modern comparative genomics does not restrict to sequence but involves the comparison of metabolic pathways or protein–protein interactions as well. Central in this approach is the concept of neighbourhood between entities (genes, proteins, chemical compounds). Therefore there is a growing need for new methods aiming at merging the connectivity information from different biological sources in order to infer functional coupling.

Results: We present a generic approach to merge the information from two or more graphs representing biological data. The method is based on two concepts. The first one, the correspondence multigraph, precisely defines how correspondence is performed between the primary data-graphs. The second one, the common connected components, defines which property of the multigraph is searched for. Although this problem has already been informally stated in the past few years, we give here a formal and general statement together with an exact algorithm to solve it.

Availability: The algorithm presented in this paper has been implemented in C. Source code is freely available for download at: http://www.inrialpes.fr/helix/people/viari/cccpart

Contact: Alain.Viari@inrialpes.fr

INTRODUCTION

A large variety of biological data obtained from genome-wide experiments can be represented by graphs (Alm and Arkin, 2003). Best-known examples are protein–protein interaction graphs where vertices are proteins and edges represent physical interactions between them. Metabolic networks are another example of more sophisticated graphs, called bipartite graphs, with two types of vertices representing chemical compounds and reactions and the edges indicate which compound is the substrate or product of a particular reaction. Finally, even the layout of the genes on a chromosome can be described as a particular graph, called an interval graph, where the vertices, representing the genes, are connected if the genes are adjacent (or, more generally, if the linear distance between two genes is less than a given threshold). Central to this graph representation is the notion of adjacency or connectivity (Galperin and Koonin, 2000). For instance a group of connected vertices in a protein–protein interaction graph may be interpreted as a molecular complex, a group of connected reactions may be interpreted as a biochemical pathway and a group of co-oriented adjacent genes may represent an operon. The following step is then to integrate these different biological graphs in order to answer more complex biological questions, for instance to identify which contiguous genes do encode for enzymes catalysing contiguous reactions in the metabolic network or do encode for interacting polypeptides. Although the importance of graph representation to perform comparative analysis has been recognized for long (Ogata *et al*., 2000; Snel *et al*., 2002; Yanai and DeLisi, 2002; Overbeek *et al*., 1999), we still lack a unified framework for this kind of comparisons. The purpose of this paper is to provide such a unified approach and to illustrate it on several instances of biological graphs.

*To whom correspondence should be addressed.

METHODS

For the sake of simplicity, all the definitions will be given hereafter for two graphs but the generalization to the case of $n > 2$ graphs is immediate.

The correspondence multigraph

Let us consider two graphs G_1 and G_2, hereafter referred to as the primary graphs, representing some biological data. Each graph G_i is described by a set of vertices V_i and edges E_i.

$$G_1 = (V_1, E_1); \quad G_2 = (V_2, E_2).$$

One should note that the vertices do not necessarily represent the same kind of data in the two graphs. For instance, V_1 may represent a set of genes whereas V_2 may represent a set of reactions.

Let us note $P = V_1 \times V_2$, the Cartesian product, of V_1 and V_2. That is, the set of all couples of the form (v_1, v_2) where $v_1 \in V_1$ and $v_2 \in V_2$.

Let us consider a particular relation R stating a correspondence (not necessarily one-to-one) between the elements of V_1 and V_2. For instance a gene is R-related to a reaction if the EC number of the gene's product is the same as the EC number of the reaction.

Finally let us note V_{R} the restriction of P to the couples where the two elements are related by R.

$$(v_1, v_2) \in V_{\mathrm{R}} \Leftrightarrow \mathrm{R}(v_1, v_2).$$

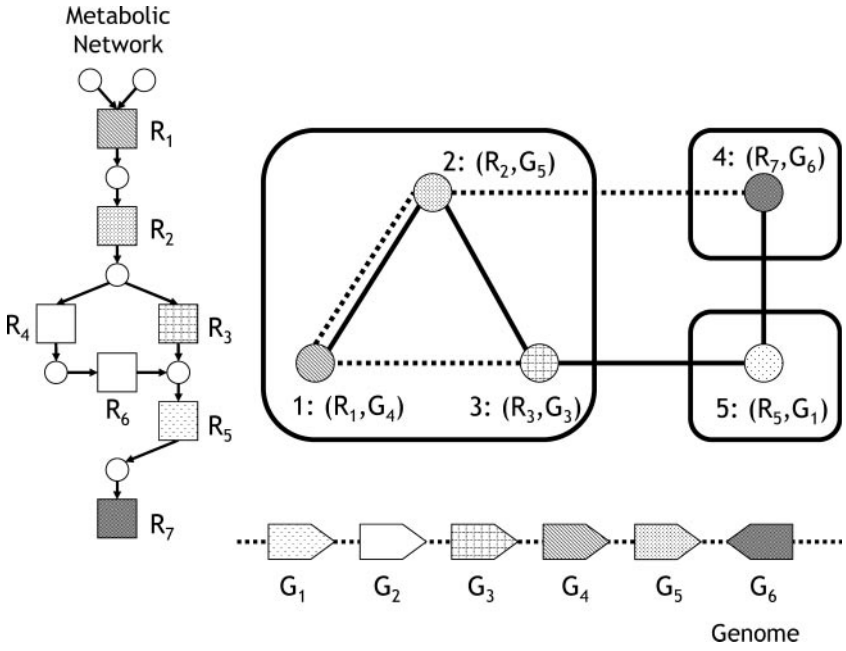

Fig. 1. Illustration of the correspondence multigraph and its CCCs. The vertices of the correspondence multigraph are defined by (gene, reaction) couples and we restrict the set of all possible couples to those where the product of the gene encodes for a polypeptide that catalyses the corresponding reaction. Plain edges denote the reaction connectivity and dashed edges denote the gene connectivity. In this example, there are three CCCs ({1,2,3}, {4} and {5}). They correspond to sets of adjacent genes encoding for polypeptides catalysing adjacent metabolic reactions.

For instance, we can restrict all the possible couples of genes to the couples of orthologous genes or all the possible couples of genes and reactions to those where the gene product catalyses the reaction.

The correspondence multigraph G is defined by the set of vertices V_{R} and two sets of edges E'_1 and E'_2:

$$G = (V_{\mathrm{R}}, E'_1, E'_2)$$

with

$$((v_1, v_2), (w_1, w_2)) \in E'_1 \Leftrightarrow (v_1, w_1) \in E_1$$

$$((v_1, v_2), (w_1, w_2)) \in E'_2 \Leftrightarrow (v_2, w_2) \in E_2.$$

The two previous definitions simply state that two vertices (i.e. couples) of the correspondence multigraph have an edge in E'_1 (respectively E'_2) if and only if their first (respectively second) elements are connected in G_1 (respectively G_2).

An example of correspondence multigraph, involving the comparison of the gene organization with metabolic pathways, is given in Figure 1.

The common connected components

A connected component (CC) of a graph G = (V, E) is a maximal subset of vertices such that every vertex is reachable from each other vertex in the component. In a similar way, a common connected component (CCC) (Gai *et al*., 2003) of a multigraph G = (V_{R}, E'_1, E'_2) is a maximal subset of vertices that are both connected in E'_1 and E'_2.

This definition is illustrated in Figure 1 where the multigraph has three CCCs ({1,2,3}, {4}, {5}). For the biological interpretation, it is important to remember that a CCC is a set of vertices such that every vertex is reachable from each other vertex through every type of edges. In other words, a CCC is composed of components that are connected in every primary graph—possibly by a different network of edges.

ALGORITHM

Computation of the CCCs

Figure 1 illustrates an important point for the practical computation of the CCCs: a CCC is not simply the intersection of separate CCs for each type of edges. In this example, the CCs of (V_{R}, E'_1) is {1,2,3,4,5} (plain edges) and the connected components of (V_{R}, E'_2) are {1,2,3,4} and {5} (dashed edges), therefore the intersection is {1,2,3,4} and {5} but {1,2,3,4} is not a CCC (since 4 is not reachable through plain edges). Now, going one step further, we can reiterate the intersection process on the set {1,2,3,4}. It then splits into two sets {1,2,3} and {4} that are the CCCs of the multigraph. More generally, the intersection of CC provides a partition of vertices with coarser grain than CCC. The idea of an exact CCC computation algorithm is therefore to iteratively refine this partition. The algorithm is initialized with a partition P_0 composed of a single class with all vertices. Then, at each iteration step k, one computes the intersection of CCs within each class of the partition P_k, possibly splitting this class into subclasses. This eventually gives rise to a new partition P_{k+1}. The algorithm stops when the number of classes does not change (i.e. when $\mathrm{P}_k = \mathrm{P}_{k+1}$). The pseudo-code in Figure 2 gives a sketch of a recursive version of this algorithm. The worst-case time complexity of this algorithm is $O(n\ (e{\cdot}n + m))$ where $n = |V_{\mathrm{R}}|$ total number of nodes in the multigraph, e = number of primary graphs ($e = 2$ in the Figure 1 example) and $m = \sum_{i=1}^{e} |E_i|$ = total number of edges in the multigraph. The worst-case complexity corresponds actually to the case where the final partition is composed of n classes (each class being therefore a singleton) and when each of these classes is extracted at each iteration thus requiring n steps to perform the calculation. In practice, the number of steps that

```
Function CCC            /* compute the common connected components of G */
Input  :
  MultiGraph g
Output :
  Partition             /* the CCC partition of G                        */
Variable :
  Partition p, inter
  array of Partition cc
  Multi-graph g'
Begin
  /* compute connected components foreach type of edges   */
  cc ← ConnectedComponents(g)
  /* compute the intersection of all connected components */
  inter ← Intersection(cc);
  if (|inter|=1) then
    /* single class (stable)                              */
    p ← inter
  else
    /* recursively call CCC on each class of intersection */
    /* and combine results                                */
    p ← ∅
    for i = 1 to |inter| do
      g' ← InducedMultiGraph(g,inter[i])
      p ← p ∪ CCC(g')
    end for
  end if
  return p
End
```

Fig. 2. Pseudo-code of a recursive algorithm for computing the CCCs of a multigraph.

are necessary to get the final partition is much lower than n (for all our experiments with random and real data this number rarely exceeded 10 steps, even for very large multigraphs).

Considering insertions/deletions in primary graphs

An edge between two vertices in the correspondence multigraph implies, by construction, that the corresponding elements are connected in the corresponding primary graphs. For many practical applications, this requirement is too stringent. For instance, with genomic graphs, the requirement is that the genes are strictly adjacent on each of the chromosomes whereas we may want to allow some gene insertions/deletions. A straightforward way to do this is to introduce new edges in the primary graphs. More precisely, if we define the distance between two vertices as the minimum number of edges in a path connecting them, one should add an edge between all pairs of vertices lying at a distance $\leq\delta + 1$. δ is therefore an insertion parameter, the case $\delta = 0$ corresponds to the original primary graph with no additional edge.

Related works

In 2000, Ogata *et al*. presented a graph comparison algorithm to detect functionally related enzyme clusters. Although not stated explicitly in the paper, this approach actually aims at finding CCCs between two graphs representing, respectively, the genomic and metabolic data. The correspondence multigraph is implicitly described as a list of correspondence between the vertices of these two primary graphs. Therefore, this approach is very similar in its spirit to the one presented here. An important difference lies in the fact that the proposed algorithm is a heuristic whereas our algorithm provides exact CCCs. Indeed, it can be shown that the solution of this heuristic approach is actually a subset of the exact solution. In other terms, the heuristic may miss solutions. An example of this will be shown later. In a very similar work, Zheng *et al*. (2002) proposed a graph-based method to infer bacterial operons. Again, the idea is to look for clusters of contiguous genes, coding for catalysers of connected reactions in the metabolic graph. These clusters are found by a breadth-first search on the metabolic graph, pruned by the distance in the genomic graph. As in the previous work, the proposed algorithm is a heuristic. As far as we know, Kelley *et al*. (2003) were the first to propose an explicit definition of a multigraph, in the context of protein interaction networks. Each node of the multigraph consists in a couple of proteins (one for each compared protein interaction network) with sufficient sequence similarity (using BlastP). As in our approach, some additional edges are added to the primary graphs ($\delta = 1$) to account for gaps. At this stage, their approach diverges from ours. Instead of working on the multigraph directly, the authors merge the edges to yield a single graph (called a 'network alignment') and then look for paths and densely connected subgraphs in this graph. This approach has been further extended to the case of three primary graphs by Sharan *et al*. (2005). In 2003, Gai *et al*. addressed the problem of identifying CCCs of two or more graphs from the computational point of view. They proposed a non-trivial algorithm combining maximal clique decomposition together with an Hopcroft-like partitioning approach. This algorithm achieves an $O(n \log n + m \log^2 n)$ time complexity. Furthermore, in the case of interval graphs, the complexity reduces to $O((n + m) \log n)$ (Habib *et al*., 2004). We acknowledge that this theoretical complexity is better than the one achieved by our simpler algorithm. However, as we shall see in the Results section, the computation of the CCCs is not the time-limiting step. It is usually done within few seconds whereas the construction of the multigraph may take several minutes up to one hour (e.g. when sequence similarities need to be established).

RESULTS

We will now illustrate the unified framework with three typical biological applications: (1) the identification of neighbouring genes with similar organization in several genomes (syntons), (2) the identification of neighbouring genes coding for enzymes catalysing connected metabolic reactions (metabolons) and (3) the identification of neighbouring genes coding for interacting proteins (interactons). For each application, we shall give a formal definition of the problem in terms of multigraph definition (by specifying $V_1, \ldots, V_n$; $E_1, \ldots, E_n$ and the restriction V_R) and we shall give examples of CCCs obtained with actual data.

Syntons

This problem can be informally stated as finding sets of contiguous genes (syntons) with conserved local organization across n ($n \geq 2$) bacterial genomes. The input is therefore the gene layout on two or more chromosomes together with a gene orthology relationship. We can reformulate this problem as finding the CCCs of the following correspondence multigraph:

$$V_i = \{g_j^i\}_{j=1,N_i} = \text{set of genes in genome } i \quad (1)$$

$$E_i = \{(g_j^i, g_k^i) \mid |\text{rank}(g_j^i) - \text{rank}(g_k^i)| \leq \delta + 1\}, \quad (2)$$

where rank (g_j^i) is the rank of gene g_j^i on chromosome i (taking into account boundaries for circular chromosomes). One should note that definition (2) does not explicitly require the conservation of the genes orientation although this condition can be easily added if needed.

Finally, the restriction condition writes

$$V_R = \{(g_{i_1}^1, \ldots, g_{i_n}^n) \in V_1 \times \cdots \times V_n \mid \text{similar}(g_{i_1}^1, \ldots, g_{i_n}^n)\}. \quad (3)$$

The similar relation has to be defined more precisely. It specifies which n-tuples of genes (one gene per genome) should be considered as nodes of the multigraph. Several definitions are possible (we omit the subscripts for clarity):

$$(g^1, \ldots, g^n) \in V_R \Leftrightarrow \forall\, g^i, g^j \text{orthologous}(g^i, g^j) \quad (4)$$

or

$$(g^1, \ldots, g^n) \in V_R \Leftrightarrow \exists g^i \mid \forall\, g^j \text{orthologous}(g^i, g^j). \quad (5)$$

Definition (4) requires that all genes in an n-tuple should be orthologous, i.e. $(g^1, \ldots, g^n)$ is a clique of the orthologous relation. Definition (5) is less restrictive and only requires that one gene is orthologous to all the other, i.e. $(g^i, \ldots, g^n)$ is a star of the orthologous relation. Other definitions are possible [e.g. $(g^1, \ldots, g^n)$ is a CC of the orthologous relation] depending upon each particular problem. In the example that will be given later on, we choose definition (4).

Again, the orthologous relation should be made more precise. Some readily available classifications [such as COG (Tatusov *et al*., 1997)] can be used here:

$$\text{orthologous}(g^i, g^j) \Leftrightarrow \text{same COG}(g^i, g^j).$$

When no such classification is available one may resort to sequence similarity:

$$\text{orthologous}(g^i, g^j) \Leftrightarrow \text{similar}(g^i, g^j),$$

where similar stands for any sequence similarity measure (an example will be given hereafter).

We illustrate the approach with $n > 2$ genomes by comparing the chromosomal organization of five enterobacteria (*Escherichia coli*, *Shigella flexeneri*, *Salmonella typhimurium*, *Yersinia pestis* and *Photorhabdus luminescens*). As a sequence similarity measure (similar), we compared each gene product of one genome against the others using BlastP (Altschul *et al*., 1990) and retained couples of genes that can be aligned on at least 80% of the length of the shortest sequence with an identity of at least 40%. It should be pointed out that, since this similar relation is not a one-to-one correspondence, the same gene can be part of several syntons. With this definition, the multigraph has 10 039 vertices and the number of edges ranges from 559 938 ($\delta = 0$) to 1 566 968 ($\delta = 10$). This yields from 516 ($\delta = 0$) to 451 ($\delta = 10$) syntons of size $\geq$2 (the number of syntons decreases but the mean size of the syntons increases with δ in this study). As an example, Figure 3a displays one of the two largest syntons (30 genes) found with $\delta = 5$. The time-limiting step is clearly the computation of sequence similarities (90 min) as the multigraph construction took <2 min and the CCCs computation took 10s (PowerBook G4 1.5 GHz).

Metabolons

This problem can be informally stated as finding sets of contiguous genes (metabolons) encoding for enzymes that catalyse connected metabolic reactions (and, therefore, that may be part of the same pathway). The input is therefore the gene layout on a chromosome, a metabolic graph and a correspondence between genes and chemical reactions. We can reformulate this problem as finding the CCCs of the following correspondence multigraph:

$$V_1 = \{g_i\}_{i=1,N_1} = \text{set of genes in the genome under study} \quad (6)$$

$$V_2 = \{r_i\}_{i=1,N_2} = \text{set of chemical reactions in a database.} \quad (7)$$

We note compounds (r) the set of compounds involved in reaction r (either as substrates or products). For practical applications, it is advisable to ignore ubiquitous compounds (H_2O, ATP, ...) that connect too many reactions.

$$E_1 = \{(g_1, g_2) \mid |\text{rank}(g_1) - \text{rank}(g_2)| \leq \delta_1 + 1\} \quad (8) \equiv (2)$$

$$E_2 = \{(r_1, r_2) \mid \text{compounds}(r_1) \cap \text{compounds}(r_2) \neq \varnothing\} \quad (9)$$

This definition ensures that two reactions are connected whenever they share either a common substrate, product or if a substrate of one reaction is a product of the other one. Moreover, as previously described, a delta parameter (δ_2) can be introduced in order to relax the constraint of strict reaction connectivity. According to this, two reactions will be connected if their distance in the primary metabolic graph is $\leq \delta_2$.

Finally, the restriction condition writes

$$V_R = \{(g, r) \in V_1 \times V_2 \mid \text{ECs}(g) \cap \text{ECs}(r) \neq \varnothing\},$$

where ECs(g) [resp. ECs(r)] is the set of EC numbers associated to the product of gene g (resp. to the reaction r). V_R is therefore a set of couples (gene, reaction) such that the gene and the reaction share at least one EC number in their respective annotations.

To illustrate these definitions, we considered the complete genome of *E.coli* (4289 genes) to build the genomic graph (V_1). The metabolic graph (V_2) was built from the KEGG/LIGAND database

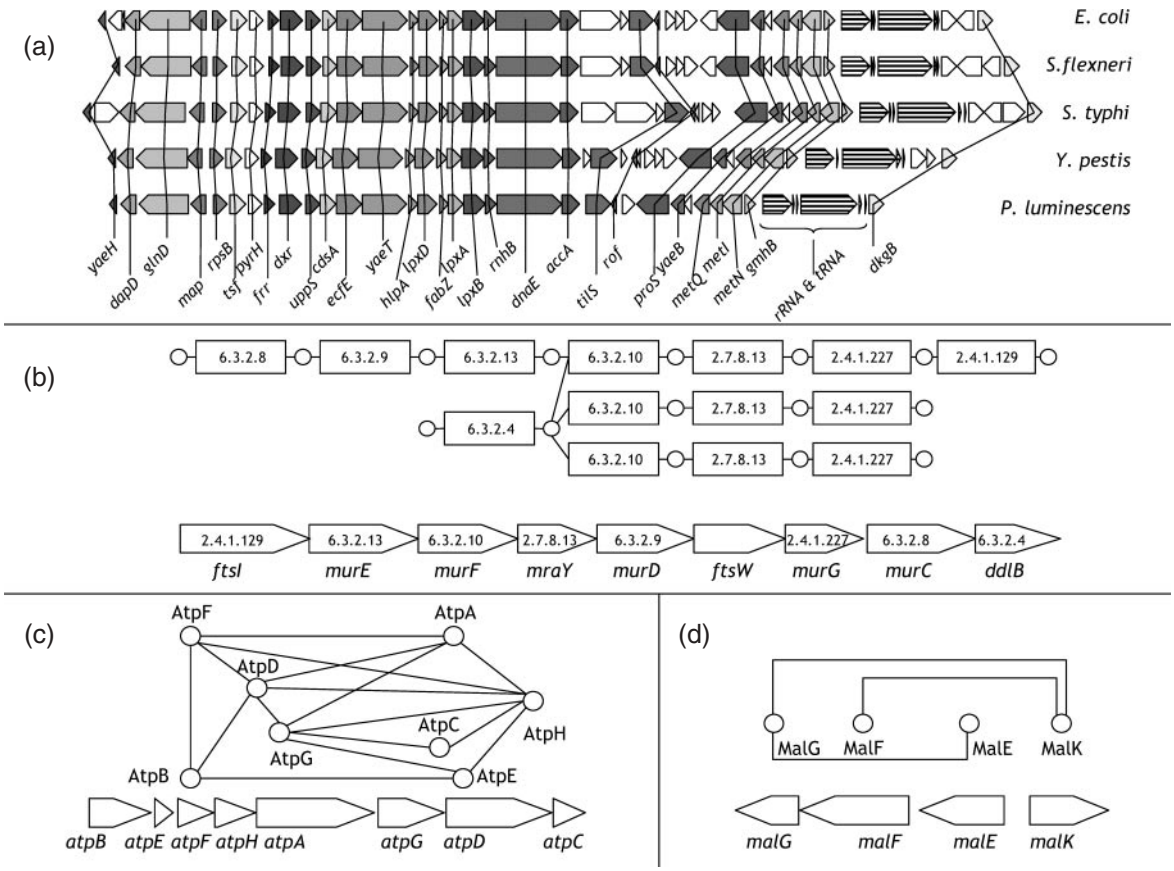

Fig. 3. (**a**) One of the largest synton (30 genes) observed between *E.coli*, *S.flexeneri*, *S.typhimurium*, *Y.pestis* and *P.luminescens* with $\delta = 5$. The vertical lines indicate the correspondence between genes. The striped box indicates rRNA or tRNA genes (that are ignored in this study). (**b**) The second largest metabolon found in *E.coli* with $\delta_1 = 1$ and $\delta_2 = 0$. The metabolon corresponds to a cluster of nine genes involved in cell envelope biosynthesis and cell division (lower part). The corresponding connected reactions are part of the peptidoglycan biosynthesis pathway and are represented on the upper part. The EC numbers and colours indicate the correspondence between reactions and genes. (**c**) and (**d**) Two interactons found in *E.coli* with $\delta_1 = \delta_2 = 0$: ATP synthase complex (c) and maltose ABC transporter (d).

(Kanehisa *et al*., 2004), excluding compounds involved in >20 reactions (ubiquitous compounds). The resulting correspondence multigraph has 2275 vertices and the number of edges ranges from 215 948 ($\delta_1 = 0$ and $\delta_2 = 0$) to 1 645 510 ($\delta_1 = 5$ and $\delta_2 = 5$). This yields from 106 ($\delta_1 = 0$ and $\delta_2 = 0$) to 156 ($\delta_1 = 5$ and $\delta_2 = 5$) metabolons of size ≥ 2 (the size of a metabolon is defined as the number of different genes in the CCC). As an example, Figure 3b displays one of the largest metabolon (8 genes) found with $\delta_1 = 1$ and $\delta_2 = 0$. The total computation time does not exceed 1 min and computation of the CCCs took <5 s.

In order to validate the approach and to examine the impact of the parameters, we performed two different experiments. We first compared the number of metabolons (of size ≥ 2) obtained with $\delta_1 \in [0,5]$ and $\delta_2 \in [0,3]$ for *E.coli* and 100 shuffled *E.coli* genomes (i.e. gene ranks are shuffled). As already suggested by Ogata *et al*. (2000), this empirical approach allows to estimate a *Z*-score and a *P*-value based on the number of observed metabolons. The results are given in Table 1. As expected, the number of metabolons increases with the delta parameters, both on observed and shuffled data. Although, the observed values are highly significant in all cases (*Z*-scores from 69 to 7), one can observe that the significance tends to decrease when increasing the delta parameters. A second, more biologically meaningful, way of evaluating the metabolons is to compare them with already known clusters of genes like operons or regulons. For this purpose, we compared them with the whole set of operons in RegulonDB (Salgado *et al*., 2004). We do not expect a complete match between operons and metabolons because of three main reasons: (1) we do not enforce the gene coorientation whereas in RegulonDB an operon is composed of cooriented genes; (2) we expect a metabolon to match an operon but the converse is false since some operons may not be related to intermediate metabolism (e.g. ABC transporters or ribosomal proteins) and (3) the annotation of enzymes is not comprehensive (in *E.coli* only ~25% of genes are associated to complete EC numbers). For these reasons, we defined two measures of coverage: (1) the fraction of metabolons that are covered by (at least) an operon. We say that a metabolon is covered if it shares at least half of its genes with the operon. (2) the fraction of operons that are covered by (at least) a metabolon. For the reason given above, in this latter case, we restrict to the operons containing at least two enzymes. The results are given in Table 1 (two last columns). The percentage of covered metabolons is rather high (up to 75%) and decreases with the delta parameters. This is due to the fact that the size of the metabolons increases with delta, making them more difficult to be covered by an operon. On the other hand,

Table 1. Number of metabolons found in *E.coli*

δ_1 genomic[a]	δ_2 metabolic[b]	# metabolons[c]	Shuffled genomes[d] # metabolons	SD	*Z*-score	*P*-value[e]	% Covered metabolons[f]	% Covered operons[g]
0	0	105	2.24	1.48	69.56	$\leq 2.0 \times 10^{-4}$	75.00	49.58
	1	106	6.52	2.80	35.51	$\leq 7.9 \times 10^{-4}$	71.58	59.66
	2	124	15.67	4.03	26.85	$\leq 1.4 \times 10^{-3}$	68.75	62.18
	3	136	33.22	5.63	18.26	$\leq 3.0 \times 10^{-3}$	65.32	64.71
1	0	116	4.43	2.25	49.67	$\leq 4.0 \times 10^{-4}$	71.26	54.62
	1	118	12.88	3.61	29.08	$\leq 1.2 \times 10^{-3}$	69.52	63.03
	2	136	30.20	4.88	21.66	$\leq 2.1 \times 10^{-3}$	67.21	65.55
	3	148	59.17	6.08	14.62	$\leq 4.7 \times 10^{-3}$	60.29	70.59
5	0	123	12.36	3.76	29.41	$\leq 1.2 \times 10^{-3}$	68.48	55.46
	1	130	34.49	5.27	18.10	$\leq 3.0 \times 10^{-3}$	66.67	66.39
	2	146	70.47	6.40	11.80	$\leq 7.2 \times 10^{-3}$	59.85	72.27
	3	145	103.49	5.77	7.19	$\leq 1.9 \times 10^{-2}$	52.99	78.15

Comparison with shuffled data and with the RegulonDB database.
[a]Delta parameter on genomic graph.
[b]Delta parameter on metabolic graph.
[c]Observed number of metabolons of size ≥ 2
[d]Observed number of metabolons of size ≥ 2 (#) and SD for 100 shuffled genomes.
[e]According to the Chebyshev's inequality.
[f]Percentage of metabolons that share at least half of their genes with an operon in RegulonDB.
[g]Percentage of operons (containing at least two enzymes) in RegulonDB that share at least half of their genes with a metabolon.

for the same reason, the percentage of covered operon increases. Of course, these results may vary with the organization of the species under study and the quality of available data.

Interactons

This problem can be informally stated as finding sets of contiguous genes (interactons) encoding for proteins that are known to interact with each other (such as components of a molecular complex). The input is therefore the gene layout on a chromosome and the interaction graph between their products. We can reformulate this problem as finding the CCCs of the following correspondence multigraph:

$$V_1 = \{g_i\}_{i=1,N_1} = \text{set of genes in the genome under study} \quad (10)$$

$$V_2 = \{p_i\}_{i=1,N_2} = \text{set of proteins} \quad (11)$$

$$E_1 = \{(g_1, g_2) \mid |\text{rank}(g_1) - \text{rank}(g_2)| \leq \delta_1 + 1\} \quad (12) \equiv (2)$$

$$E_2 = \{(p_1, p_2) \mid \text{interact}(p_1, p_2)\} \quad (13)$$

where interact means that the two proteins are known to interact.

Finally, the restriction condition writes

$$V_R = \{(g,p) \in V_1 \times V_2 \mid \text{product}(g) = p\}$$

where product(g) denotes the product of gene g. V_R is therefore a set of couples (gene, protein) such that the gene encodes for the protein. Edges in the correspondence multigraph indicate that the genes are contiguous and that the proteins physically interact.

To illustrate these definitions, we considered the complete genome of *E.coli* (4289 genes) to build the genomic graph (V_1). The protein–protein interaction (PPI) graph (V_2) was build from the DIP database (Salwinski *et al*., 2004) and was restricted to proteins from *E.coli*. In this case, the multigraph is very small (4289 nodes and 4551 edges for $\delta_1 = 0$ and $\delta_2 = 0$), the total computation time for the multigraph construction and computation of the CCCs is <2 s. Figure 3c and d show two examples of interactons computed with $\delta_1 = 0$ and $\delta_2 = 0$. The first one corresponds to the largest interacton that has been detected (eight genes). It consists of the gene cluster coding for the eight components of the ATP synthase complex. In this case, the protein nodes are strongly linked to each other. For instance AtpH interacts with six other proteins of the interacton. The second example corresponds to an interacton of four genes, associated to the maltose ABC transporter. We have selected this example to emphasize a case of loosely linked nodes. In this case there is no two adjacent genes encode for two interacting proteins (in other terms, no nodes in the subgraph of the multigraph are connected both with genomic and PPI vertices). This is a typical example of a solution missed by Ogata *et al*. heuristic. Indeed this greedy heuristic requires at least two nodes being connected in all primary graphs in order to begin the clustering process. So, in this case, the CCC will be completely missed. More generally Ogata *et al*. heuristic may lead to smaller solutions than the exact algorithm.

CONCLUSION

We have introduced a general framework to represent correspondences between genomic or functional data represented by graphs together with an exact procedure to extract clusters of neighbouring entities (such as genes, proteins, reactions) from this representation. The advantage of this approach is that it allows a fairly general formulation of the problem that can be further adapted to different actual cases. This approach was illustrated in three particular cases. The first example mixes genomic (G) graphs only and could therefore be denoted as the G_n (syntons between $n \geq 2$ genomes) problem. The second example mixes genomic and metabolic

information and could be denoted as a G_1M problem. The extension to the G_nM problem is straightforward and could be used to identify clusters of neighbouring genes in several organisms and associated with connected reactions. Finally, the third example mixes genomic and protein–protein interaction data and could therefore be described as the G_1I_1 problem (we used a subscript to the I graph to indicate that the interactions are species specific). Again, the extension to the G_nI_1 and G_nI_n problems could be envisaged as well. Another potentially interesting case is the I_1M type of problem that related metabolic and interaction data in order to grab, for instance, the channelling (tunnelling) of substrates in enzymatic complexes. Finally, the GIM type of problem allows mixing the three kinds of data together. Other extensions may involve new kinds of primary graphs. This means either new types of primary nodes (e.g. using protein domains instead of complete proteins) or new kind of primary edges [e.g. shared expression patterns or gene fusions (Marcotte *et al*., 1999)]. Finally, we would like to point out another, more algorithmic extension of the current approach that may become important when dealing with more than two graphs. Considering for instance the $G_{n>2}$ problem, the idea is to introduce the notion of quorum that specifies a minimum number of primary graphs (i.e. species) for which the connectivity property holds. This will allow to look for syntons not occurring in all the species but, at least, in a minimum number of them. The precise definition of this extension will be our next task in the future.

ACKNOWLEDGEMENTS

The authors would like to thank Marie-France Sagot and Eric Coissac for helpful discussions during the course of this work.

Conflict of Interest: none declared.

REFERENCES

Alm,E. and Arkin,A.P. (2003) Biological networks. *Curr. Opin. Struct. Biol*., **13**, 193–202.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Gai,A.-T., Habib,M., Paul,C. and Raffinot,M. (2003) Identifying common connected components of graphs. *Report RR-LIRMM-03016*. LIRMM, Université de Montpellier 2, France.

Galperin,M.Y. and Koonin,E.V. (2000) Who's your neighbor? New computational approaches for functional genomics *Nat. Biotechnol*., **18**, 609–613.

Habib,M., Paul,C. and Raffinot,M. (2004) Maximal common connected sets of interval graphs. *Proceedings of Combinatorial Pattern Matching: 15th Annual Symposium, CPM 2004, Istanbul, Turkey, July 5–7 2004, LNCS*, **3109,** pp. 359–372.

Kanehisa,M. *et al*. (2004) The KEGG resource for deciphering the genome. *Nucleic Acids Res*., **32**, D277–D280.

Kelley,B.P. *et al*. (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl Acad. Sci. USA*, **100**, 11394–11399.

Marcotte,E.M. *et al*. (1999) Detecting protein function and protein–protein interactions from genome sequences. *Science*, **285**, 751–753.

Ogata,H. *et al*. (2000) A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Res*., **28**, 4021–4028.

Overbeek,R. *et al*. (1999) The use of gene clusters to infer functional coupling. *Proc. Natl Acad. Sci. USA*, **96**, 2896–2901.

Salgado,H. *et al*. (2004) RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res*., **32**, D303–D306.

Salwinski,L. *et al*. (2004) The database of interacting proteins: 2004 update. *Nucleic Acids Res*., **32**, D449–D451.

Sharan,R. *et al*. (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA*, **102**, 1974–1979.

Snel,B. *et al*. (2002) The identification of functional modules from the genomic association of genes. *Proc. Natl Acad. Sci. USA*, **99**, 5890–5895.

Tatusov,R.L. *et al*. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.

Yanai,I. and DeLisi,C. (2002) The society of genes: networks of functional links between genes from comparative genomics. *Genome Biol*., **3**, research0064.

Zheng,Y. *et al*. (2002) Computational identification of operons in microbial genomes. *Genome Res*., **12**, 1221–1230.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4216–4222
doi:10.1093/bioinformatics/bti706

Sequence analysis

Discovering hidden viral piracy

Eddo Kim[1,2] and Yossef Kliger[1,*]

[1]Compugen Ltd, Tel Aviv 69512, Israel and [2]Faculty of Life Sciences, Bar-Ilan University, Ramat Gan 52900, Israel

Received on July 28, 2005; revised on September 15, 2005; accepted on October 5, 2005
Advance Access publication October 6, 2005

ABSTRACT

Motivation: Viruses and developers of anti-inflammatory therapies share a common interest in proteins that manipulate the immune response. Large double-stranded DNA viruses acquire host proteins to evade host defense mechanisms. Hence, viral pirated proteins may have a therapeutic potential. Although dozens of viral piracy events have already been identified, we hypothesized that sequence divergence impedes the discovery of many others.

Results: We developed a method to assess the number of viral/human homologs and discovered that at least 917 highly diverged homologs are hidden in low-similarity alignment hits that are usually ignored. However, these low-similarity homologs are masked by many false alignment hits. We therefore applied a filtering method to increase the proportion of viral/human homologous proteins. The homologous proteins we found may facilitate functional annotation of viral and human proteins. Furthermore, some of these proteins play a key role in immune modulation and are therefore therapeutic protein candidates.

Contact: kliger@compugen.co.il

1 INTRODUCTION

Viruses must develop means to evade the host immune system. One evasion strategy is the modulation of the host immune response by encoding different immunomodulatory proteins. Often, one can find sequence similarity between these viral immunomodulatory proteins and host proteins, suggesting that these viral proteins were 'pirated' from the host. This phenomenon, named viral piracy, is a strategy predominantly taken by large dsDNA viruses (Seet *et al*., 2003). Members of the Herpesviridae and the Poxviridae families are known to encode proteins that modulate the host immune response, including homologs of cytokines, interferon regulatory factors, chemokines and their receptors.

Epstein–Barr virus (EBV) (Moore *et al*., 1990) and Human Cytomegalovirus (HCMV) (Kotenko *et al*., 2000) harbor their own homolog of IL-10—a powerful immunosuppressor that downregulates the synthesis of pro-inflammatory cytokines and prevents macrophages from acting as antigen presenting cells (Bogdan *et al*., 1991; D'Andrea *et al*., 1993; de Waal Malefyt *et al*., 1991). EBV IL-10 has apparently retained the immune-inhibitory properties associated with the cellular ligand, but not the immune-stimulatory features associated with Human IL-10 (reviewed in Lucas and McFadden, 2004). Kaposi's sarcoma-associated herpesvirus (KSHV) encodes a homolog of IL-6, a multi-functional cytokine with a wide range of activities. Moreover, while normally cells must express both the IL-6 receptor and the gp130 co-receptor to respond to IL-6, the viral IL-6 can induce signaling through gp130 alone and thus stimulate peripheral blood B cells that are unresponsive to the human IL-6 (Breen *et al*., 2001; Mullberg *et al*., 2000). Hence, not only do viruses encode homologs of immune-related proteins, often these homologs have evolved and gained new capabilities not present in the original host protein.

Myxoma virus, a member of the Poxviridae family, takes a different path to suppress the immune response. It encodes a protein homologous to the human TNF receptor. This viral protein lacks the transmembrane domain, and is therefore a soluble receptor, which acts as an inhibitor of the human TNF signaling pathway (Upton *et al*., 1991). In addition, Myxoma virus encodes several serine protease inhibitors, named Serp-1, Serp-2 and Serp-3, that exert anti-inflammatory as well as anti-apoptotic activities (Guerin *et al*., 2001; Macen *et al*., 1993; Petit *et al*., 1996). The Serp-1 protein is currently in clinical trials for the treatment of various inflammatory-related diseases.

Host IL-1 receptor binds IL-1α, IL-1β and the natural competitor IL-1 receptor antagonist. Vaccinia virus, another member of the Poxviridae family, encodes an IL-1 receptor homolog that can only bind IL-1β. Deletion experiments reveal that the Vaccinia virus-encoded IL-1 receptor homolog diminishes the systemic response to infection. This illustrates how the study of the mechanisms, devised by viruses to modulate the host defenses, promotes the understanding of the different roles of IL-1α and IL-1β (Alcami and Smith, 1992).

In addition, Vaccinia virus encodes the Vaccinia virus complement-control protein (VCP), a soluble protein that shares homology with C4-binding protein. VCP inhibits both the classical and the alternative complement pathways and contributes to virulence (Isaacs *et al*., 1992). Variola virus, the most virulent member of the Orthopoxvirus genus, encodes a complement-control protein termed SPICE, which is significantly more potent than VCP at inhibiting the formation of C3/C5 convertases necessary for complement-mediated viral clearance (Rosengard *et al*., 2002).

These examples illustrate that viruses pirate host proteins that modulate the effect of both the adaptive and innate immune systems. The importance of viral pirated proteins as immune modulators is emphasized by the finding that a TNF receptor homolog, and homologs of chemokines were found in primary virus isolates, but not in laboratory-adopted strains (Benedict *et al*., 1999; Cha *et al*., 1996). This finding suggests that these viral proteins are not essential for viral replication, but are important for the survival of the virus within the host.

Computational efforts have been successful in revealing previously unknown viral/human homologs (Holzerlandt *et al*., 2002).

*To whom correspondence should be addressed.

Yet, we hypothesized that many viral/human homologs have not yet been identified, due to low sequence similarity. For an alignment hit, the expectation value (*E*-value) reflects the expected number of random alignments having equal or better scores. Hence, the *E*-value is a conventional way for estimating the authenticity of an alignment hit. In general, homologous proteins gain significant *E*-values, while false hits gain low-significance *E*-values. Our primary objective is to test whether diverged viral/human homologous proteins exist in very-weak alignment hits that are usually ignored. However, weak alignment hits often have low-significance *E*-values, making it difficult to distinguish homologous proteins from false hits. Hence, we developed a method that estimates the number of homologous proteins in a group of alignment hits. This analysis confirmed that homologous proteins exist even in very-low-similarity alignment hits. Obviously, these homologs are masked by many false hits. Thus, we developed a scoring function that can be used to filter out many of the false hits. The detection of novel viral/human homologs facilitates functional annotation of viral and human proteins and may provide therapeutic protein candidates.

2 METHODS

2.1 Data preparation

Human RefSeq NPs were retrieved from the RefSeq ftp site on May 5, 2004 (21 196 sequences) (Pruitt and Maglott, 2001). Sequences of viral proteins belonging to the Herpesviridae and the Poxviridae families were downloaded from the NCBI server on June 27, 2004 (35 893 sequences) and were subject to a non-redundant algorithm (Holm and Sander, 1998), with a cutoff of 90% (resulting in 8185 sequences). Mammalian protein sequences were retrieved from SwissProt (Boeckmann *et al*., 2003) on June 27, 2004 (30 399 sequences).

2.2 Signal peptide prediction

Predicting whether a protein possesses a signal peptide was performed using the neural-network method of the SignalP 3.0 prediction tool with default parameters (Bendtsen *et al*., 2004).

2.3 Identifying sequence similarity between viral and human proteins

Viral proteins were queried against the human RefSeq NPs database with Blastp using default parameters (Altschul *et al*., 1990), except for the substitution matrix, where we used the BLOSUM45 matrix in order to enhance detection of distant homologs. All signal sequences of the human proteins were removed prior to the Blastp run, so the alignments are not dependent on the signal peptide sequences. Alignment hits with length of less than 30 amino acid residues were discarded.

2.4 Calculating evolutionary rate per residue of human proteins

Human RefSeq proteins were searched for mammalian homologs with Blastp using default parameters. For each human protein, alignment hits having Blastp *E*-value < 0.001 were selected (at least 5 sequences, maximum 50). Each group, consisting of a human protein and its mammalian homologs, was subject to multiple alignment using ClustalW (Thompson *et al*., 1994). The multiple alignment was used as input to the Rate4Site empirical Bayesian method (Mayrose *et al*., 2004), using default parameters, but with no branch length optimization to reduce computing time. This method calculates the evolutionary rate for each of the residues of a protein, based on the multiple alignment. The use of multiple alignments may introduce artifacts related to the non-equal representation of proteins in databases (del Sol Mesa *et al*., 2003). The Rate4site method addresses this difficulty by the construction of an evolutionary tree. The use of an evolutionary tree diminishes the effect of redundant sequences by weighting clusters of closely related sequences differently (Pupko *et al*., 2002).

3 RESULTS

We hypothesized that high sequence divergence impedes the discovery of many viral pirated proteins, because these are hidden in low sequence similarity alignments, where they are masked by many false hits. In order to examine this hypothesis, we developed a method to estimate the amount of viral/human homologs in a set of alignment hits.

3.1 A method to assess the amount of homologous proteins in a set of alignment hits

Our primary goal was to estimate the number of homologous proteins in a set of alignment hits. To this end, a protein feature, which is expected to be conserved among homologous proteins, was needed. Ideally such a protein feature should be easily detected and encompass 50% of the proteins to minimize the fraction of random alignment hits in which both proteins either have or lack the feature. The signal sequence is an abundant protein feature, encompassing ∼20% of proteins, which can be reliably detected using the SignalP tool (Bendtsen *et al*., 2004; Zhang and Henzel, 2004). Thus, we tested whether the presence of a signal sequence is a feature conserved among homologous proteins.

Viral proteins were queried against human proteins with Blastp. For each viral protein, only the best alignment hit was analyzed. We divided the alignment hits into five partitions according to their *E*-values. The fraction of the alignment hits in which both the human and the viral proteins either have or lack a signal sequence was calculated (Fig. 1). For comparison, the expected fraction of random alignment hits, in which both the human and viral proteins either have or lack a signal sequence, is plotted as well (Fig. 1, 'Random' column). As expected, the stronger the sequence similarity (low *E*-values), the higher the chance that both proteins will either have or lack a signal sequence. The results are statistically significant for all partitions where *E*-value < 5 ($P < 1e{-}4$, χ^2 test). Thus, the signal sequence is, in general, a conserved feature of homologous proteins. Therefore, we used this finding to estimate the amount of homologous proteins.

Alignment hits could be classified into one of four subgroups according to the presence (or absence) of a signal sequence in the human and viral proteins. Alignment hits, where both human and viral proteins have (or lack) a signal sequence, correspond to the YY (or NN) subgroup. Alignment hits, where the human protein has (or lacks) a signal sequence but the viral protein lacks (or has) one, correspond to the YN (or NY) subgroup.

A set of alignment hits is comprised of homologous proteins and random hits. It is possible to calculate the theoretical maximum number of random hits in a given alignment hits set from the observed distribution of the alignment hits in the four subgroups (YY, YN, NY and NN) and the expected distribution of random alignment hits. In order to calculate the distribution of random alignment hits in the four subgroups, we generated a random alignment hits set comprising viral/human insignificant alignment hits (*E*-value > 5). As expected, the null hypothesis, that the observed distribution of the insignificant alignment hits in the four subgroups

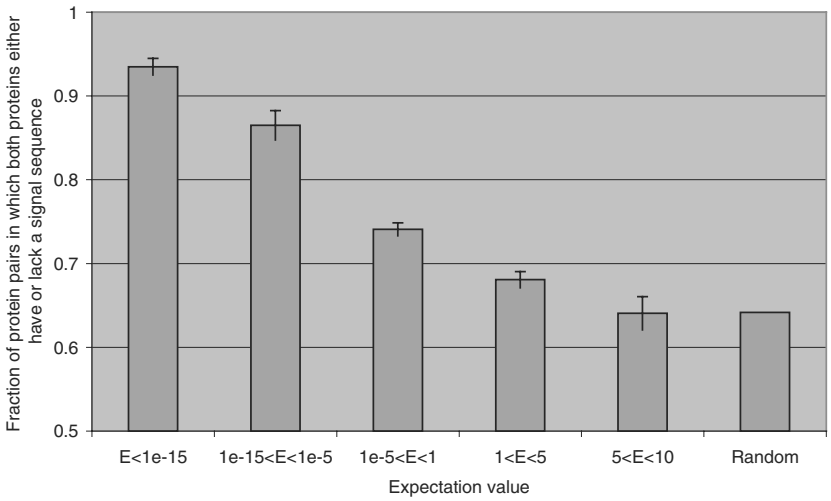

Fig. 1. Human/Viral homologous proteins tend to conserve their cellular topology. Viral/Human Blastp alignment hits were divided into five partitions according to their *E*-value. For each *E*-value partition the fraction of alignment hits, in which both the human and viral proteins either have or lack a signal sequence, is plotted. The expected fraction for random pairing of proteins is also plotted. The null hypothesis that the fraction of protein pairs in which both proteins either have or lack a signal sequence can be expected by chance is rejected for all partitions where *E*-value < 5 (*P*-value < 1e–4, χ^2 test). This null hypothesis could not be rejected for the partition where *E*-value is between 5 and 10 (*P*-value = 0.96).

Table 1. Estimated number of homologous human/viral proteins

% Identity	Sub-partition	YY	YN	NY	NN	*P*-value	Fraction of homologs
35–100	Enriched half	67	29	30	235	<1e−20	0.590
35–100	Depleted half	25	60	52	224	0.115	NS
35–100	Entire group	92	89	82	459	<1e−15	0.372
10–35	Enriched half	194	277	181	1183	<1e−20	0.438
10–35	Depleted half	102	352	232	1149	<1e−7	0.280
10–35	Entire group	296	629	413	2332	<1e−20	0.359

NS, non-significant.
The distribution of the alignment hits in the four subgroups (YY, YN, NY and NN) is indicated for the Identity > 35% and the 10% < Identity < 35% alignment sets, as well as for the enriched/depleted sub-partitions. The *P*-value reflects the chance that this distribution is the one expected if all alignment hits were random. The estimated fraction of homologous proteins is also indicated. Note that this fraction is an underestimation, as explained in Section 3.1.

is similar to the one that would be expected if all hits were random ones, could not be rejected (*P*-value = 0.99, χ^2 test). Since this group of random alignment hits is large (590 alignment hits), the distribution of any set of random alignment hits in the four subgroups should be similar to the observed one.

Hence, in a given set of alignment hits, we defined the subset of random alignment hits as the largest that could still maintain the distribution we calculated for random data. This way, we can calculate the number of homologous proteins (Table 1, 'entire groups'). The described method will always result in an underestimation of the number of homologous proteins, because we consider the theoretical maximum number of random hits.

3.2 Filtering method

We examined alignment hits within and below the twilight zone (10% < Identity < 35%) (Blake and Cohen, 2001; Rost, 1999), in order to discover unknown distant homologs. Of course, in this range, many false alignment hits mask the homologous proteins. In order to enrich the proportion of viral/human homologs, we looked for a feature that differentiates between homologs and false alignment hits. Next, the quality of this feature was determined according to its ability to enrich the data with homologous proteins, estimated as described in Section 3.1.

3.3 Relating evolutionary rate per residue with viral piracy

The feature examined was the tendency of viruses to conserve the same amino acid residues that are conserved among mammalian orthologs and paralogs. Viral proteins are subject to major sequence changes, due to their constant need to evade the host immune system. Nevertheless, in order for a viral pirated protein to maintain its function, amino acid residues that are crucial for the structure or function of the protein are expected to be conserved. We assigned a mammalian evolutionary rate for all residues of the human proteins, for all viral/human alignment hits sharing significant sequence similarity (*E*-value < 1e–5). For each of the 20 natural amino acid residues, separately, we gathered the evolutionary rates at all positions, where the amino acid residue is identical in the viral and human proteins. Similarly, we gathered the evolutionary rates at all positions, where this amino acid residue is not identical in the viral and human proteins. The two lists of evolutionary rates—for the identical and not identical positions—were compared to determine whether viral pirated proteins tend to conserve the same residues that are conserved among mammals. The same was performed for each of the 20 amino acid residues.

We expect that residues that are not important will often gain high evolutionary rates, while important residues will often gain low evolutionary rates. Indeed, we found that residues that are identical in the viral/human homologs have lower mammalian evolutionary rates than residues that are not identical. This is true for each of the

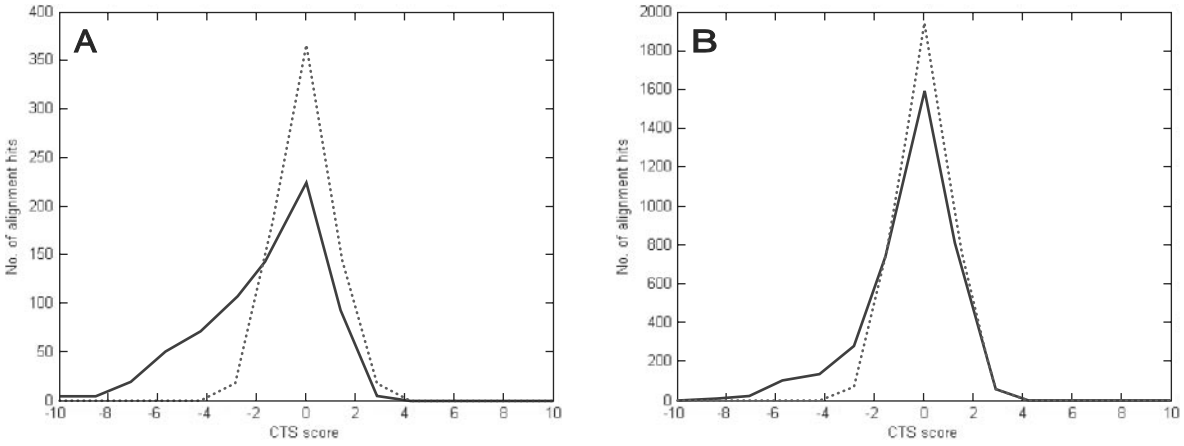

Fig. 2. The CTS scoring function distinguishes between homologous proteins and false hits. The distribution of the CTS scores of the alignment hits of the different partitions is plotted in solid line. For comparison, the distribution of CTS based on random alignments is plotted in dotted line. (**A**) Alignment hits of 35–100% identitiy. (**B**) Alignment hits of 10–35% identity.

20 natural amino acids. In order to estimate the statistical significance of the results, the null hypothesis that the two lists of mammalian evolutionary rates—for the identical and non-identical positions—are of the same distribution was rejected (*P*-value < 1e–7 for each of the 20 natural amino acids, Mann–Whitney test).

This procedure was performed on alignment hits with *E*-value < 1e–5, where most hits are of viral/human homologous proteins. For comparison, the same procedure was performed on insignificant alignment hits. The results revealed that the difference, between the mammalian evolutionary rates of residues that are identical in the viral and human aligned proteins and the residues that are not, was insignificant.

The identification of the important residues in proteins depends on the quality of the pairwise and multiple alignments (Casari *et al*., 1995). However, there is no reason to suspect that alignment errors will bias the results in a specific manner. Indeed, our results indicate that, in general, the accuracy of the alignments is sufficient for this procedure.

In conclusion, the tendency of viral proteins to maintain the residues that are conserved among mammals is a feature that distinguishes between homologous proteins and false alignment hits. Thus, we used this feature to enrich the proportion of homologous proteins.

3.4 Calculating a 'conservation tendency score' for each viral/human alignment hit

For each viral/human alignment hit the mammalian evolutionary rate scores, of all the residues of the human proteins, were computed by the Rate4Site program. We gathered all the evolutionary rate scores of residues that were identical between the human and the viral proteins. Similarly, we gathered all the evolutionary rate scores of residues that were not identical in the alignment. In order to test the hypothesis that these two score groups are similar, a Mann–Whitney test was applied. Next, we assigned the score of the corresponding Mann–Whitney test [conservation tendency score (CTS)] to each of the alignment hits.

In order to confirm that this method enriches the proportion of alignment hits comprising viral/human homologous proteins, we divided the list of alignment hits into two equally sized sub-partitions according to their CTSs. The enriched sub-partition comprised the alignment hits with better CTSs, and the depleted sub-partition comprised the alignment hits with worse CTSs. We then divided each of the sub-partitions to the four subgroups (YY, YN, NY and NN) and calculated the fraction of homologs, as well as the chance that the distribution is the same as would be expected for random alignment hits. The results revealed that for significant alignment hits (>35% sequence identity) the enriched sub-partition has a higher fraction of homologs than the depleted sub-partition (Table 1, Fig. 2A).

Even though most of these alignment hits are easy to detect by sequence alignment tools, some alignment hits are not trivial to explain. One such alignment hit is a VEGF homolog encoded by Bovine popular stomatitis virus (GI:41057442, *E*-value = 2e–25, Identity = 54%, CTS = –3.86), a member of the Poxviridae family. Human VEGF is a well-studied protein that participates in angiogenesis, induces endothelial cell proliferation, promotes cell migration and is a vascular permeability factor (Ferrara, 2001). However, it was not obvious what advantage the virus achieves by encoding a VEGF homolog. A recent publication suggests that human VEGF has also a regulatory effect on T cells and that human VEGF could promote the Th1-Th2 shift (Lee *et al*., 2004). This finding suggests that by encoding the VEGF homolog, Bovine popular stomatitis virus promotes the Th1-Th2 shift, thereby inhibiting the anti-viral cellular immune response.

Next, we examined weak alignment hits—with Identity < 35%—and found that a significant amount of homologs exists in this group as well. We then confirmed that the CTS-based method is able to decrease the fraction of false alignment hits in this set of alignments, thereby enriching the fraction of homologous proteins (Table 1, Fig. 2B).

The aim of this study is to find novel viral piracy events, therefore we focused on weak alignment hits having sequence identity <35% and *E*-value > 1e–5. Remarkably, the results revealed that ~29% of these alignment hits, which are usually ignored, are of homologous proteins (Table 2, 'entire group'). Furthermore, Figure 3 reveals that the CTS-based scoring function could also differentiate between weak hits of homologous proteins and false ones. Therefore, we

Table 2. Estimated number of homologous human/viral proteins with Identity = 10–35% and *E*-value > 1e−5

Partition	YY	YN	NY	NN	*P*-value	Fraction of homologs
Entire group	247	594	398	1945	<1e−15	0.288
Top 100 proteins (best CTS)	18	7	9	66	<1e−7	0.643

The distribution of the alignment hits in the four subgroups (YY, YN, NY and NN) is indicated for the entire group, as well as for the 100 proteins with the best CTS. The *P*-value reflects the chance that this distribution is the one expected if all alignments were random hits. The estimated fraction of homologous proteins is also indicated. Note that this fraction is an underestimation, as explained in Section 3.1.

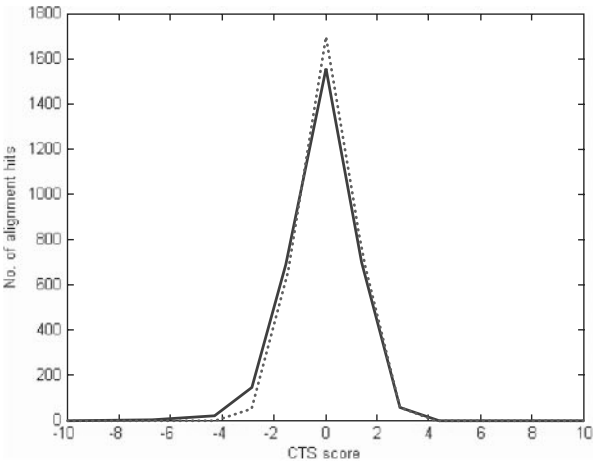

Fig. 3. The CTS scoring function distinguishes between homologous proteins and false hits, even in very weak alignment hits (Identity = 10–35% and *E*-value > 1e–5). The distribution of the CTS scores of the alignment hits is plotted in solid line. For comparison, the distribution of CTS based on random alignments is plotted in dotted line.

applied the CTS-based scoring function and focused on the 100 alignment hits having the best CTS, where we estimated that ∼64% are of homologous proteins (Table 2).

4 DISCUSSION

Few viral proteins, which were pirated from their mammalian host, are currently in clinical and pre-clinical studies for the treatment of various diseases associated with excessive inflammatory or immune responses (reviewed in Lucas and McFadden, 2004). Most annotated viral/human homologs, such as homologs of soluble TNF receptor, IL-6 and others share high sequence similarity. In order to find novel viral/human homologs, we examined low-to-moderate sequence similarity alignment hits having 10–35% identity and *E*-value > 1e–5. Our results indicated that ∼29% of the alignment hits are of homologous proteins (Table 2). We assigned each of the alignment hits a CTS, and examined the 100 proteins having the best CTS among this group. The results revealed that ∼64% of these alignment hits are of homologous proteins.

Of special interest are the secreted viral proteins, as they may be used as therapeutics. Thus, we focused on the YY subgroup of the top 100 proteins of the 10–35% identity and *E*-value > 1e–5 alignment hits set. The tendency of homologous proteins to maintain a functional signal sequence is the feature that was used to assess the amount of homologous proteins in a list of alignment hits (Fig. 1). As this feature distinguishes between homologs and random hits, focusing on the YY subgroup results in a higher fraction of homologous proteins than the one calculated. The following are several examples of secreted and membranal viral/human homologs that have been detected.

Human IL-6 (NP_000591) shares very low sequence similarity (*E*-value = 1.4; Identity = 20%) with Cercopithecine herpesvirus 17 R2 protein (GI:18653819). This alignment hit has a good CTS score of –3.36, supporting the reports, which were based on genomic location and followed by functional analysis, that this viral protein is an IL-6 homolog (Desrosiers *et al*., 1997; Kaleeba *et al*., 1999).

Human IL-10 (NP_000563) shares very high sequence similarity (*E*-value = 3e–53; Identity = 92%) with Human EBV IL-10 homolog (GI:114886). In addition, human IL-10 shares moderate sequence similarity (*E*-value = 7e–9; Identity = 27%) with Human herpesvirus 5 UL111A (GI:39841977). Support for the authenticity of the latter alignment hit comes from the CTS scoring method, which gives a score of –2.54. Indeed, this moderate sequence similarity alignment, which was not detected using a computational analysis that used a PSSM representing herpesvirus protein motifs (Holzerlandt *et al*., 2002), was shown to bind to the human IL-10 receptor and compete with human IL-10 for the receptor binding sites (Kotenko *et al*., 2000).

Human interferon gamma receptor 1 (NP_000407) shares moderate sequence similarity (*E*-value = 1e−5; Identity = 20%) with lumpy skin disease virus LSDV008 putative soluble interferon gamma receptor (GI:22595541). This alignment hit has a good CTS score of –2.51. Our finding is consistent with the prediction by Kara *et al*. (2003) that LSDV008 is an interferon gamma receptor. Several members of the Poxviridae family encode a soluble IFN-gamma receptor that efficiently blocks the binding of IFN-gamma to cellular receptors, thus inhibiting the functions of IFN-gamma (reviewed in Alcami and Smith, 1996).

Human endothelial lipase precursor (NP_006024) shares low sequence similarity (*E*-value = 0.3; Identity = 24%) with Gallid herpesvirus 2 lipase (GI:10180702). This alignment hit has a good CTS score of –3.82. Despite the low sequence similarity, this viral protein retained several conserved lipase features (Tulman *et al*., 2000).

Human OX-2 membrane glycoprotein (CD200) (NP_005935) shares low sequence similarity with Yaba-like disease virus 141R protein (GI:12085124; *E*-value = 0.0002; Identity = 30%), Yaba-monkey tumor virus 141R protein (GI:38229299; *E*-value = 0.012; Identity = 28%) and Myxoma m141R protein (GI:4097179; *E*-value = 0.025; Identity = 33%), all are members of the Poxviridae. Despite the low sequence similarity, these alignment hits have good CTS scores of –3.39, –3.06 and –2.53, respectively. OX-2 homolog of Myxoma virus has been shown to have an immunomodulatory effect and to diminish the activation level of circulating T lymphocytes. Moreover, while this OX-2 homolog is necessary for the full development of a lethal infection in rabbits, it is not required for efficient virus replication in susceptible cell lines, *in vitro* (Cameron *et al*., 2005). Herpesviruses have also adopted OX-2 homologs. KSHV, a gamma-herpesvirus, encodes the K14 protein, an OX-2 homolog that shares 40% sequence

similarity with the human protein. This OX-2 homolog binds to CD200 receptor with identical affinity and kinetics and inhibits TNF-alpha production in a similar manner to the human protein (Foster-Cuevas *et al*., 2004). Viruses of distinct families adopted OX-2, probably by independent events, indicating that it is a good point for the viruses to intervene in the regulation of the immune system. Thus, we suggest OX-2 as a target for drug intervention in inflammatory disorders. Callitrichine herpesvirus 3 C2 protein (GI:24943099) also shares low sequence similarity (E-value = 0.038; Identity = 22%) with Human OX-2 membrane glycoprotein. Even though the sequence similarity is weak, this alignment hit has a good CTS score of –1.89. Callitrichine herpesvirus 3 is a gammaherpesvirus, a subfamily whose members were shown to encode OX-2 homologs, consistent with this finding.

Homologous proteins are of the same evolutionary origin. Various applications motivate the search for homologous proteins. An ideal homology detection method would identify all homologs for every target sequence in a test set, while also having no false predictions. However, some homologous proteins share only low sequence similarity and therefore are difficult to distinguish from false alignment hits. Herein, we presented a method to assess the amount of homologous proteins in a set of alignment hits. In addition, we presented a scoring function that can be used to enrich the proportion of homologous proteins in that set of alignment hits. This technique is applicable for the identification of homologous proteins sharing low-to-moderate sequence similarity. We implemented this technique to detect diverged homologs between human and large dsDNA viruses. Our results are enriched with viral/human homologous proteins, enabling manual reviewing aimed at identifying the alignment hits that are a result of events of viral piracy.

ACKNOWLEDGEMENTS

We are thankful to T. Pupko and I. Mayrose for the Unix version of Rate4Site. We are also grateful to P. Akiva, I. Borukhov, E. Eisenberg, E. Gofer, M. Havilio, I. Hecht, D. Gerber, Y. Kinar, G. Kojekaro, G. Kol, T. Lapidot, E. Levanon, A. Novik, R. Sarid, R. Sorek, A. Toporik, N. Tsabar, L. Tsirulnikov and A. Wool for helpful suggestions and fruitful discussions.

Conflict of Interest: none declared.

REFERENCES

Alcami,A. and Smith,G.L. (1992) A soluble receptor for interleukin-1 beta encoded by vaccinia virus: a novel mechanism of virus modulation of the host response to infection. *Cell*, **71**, 153–167.

Alcami,A. and Smith,G.L. (1996) Soluble interferon-gamma receptors encoded by poxviruses. *Comp. Immunol. Microbiol. Infect. Dis.*, **19**, 305–317.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Bendtsen,J.D. *et al*. (2004) Improved prediction of signal peptides: SignalP 3.0. *J. Mol. Biol.*, **340**, 783–795.

Benedict,C.A. *et al*. (1999) Cutting edge: a novel viral TNF receptor superfamily member in virulent strains of human cytomegalovirus. *J. Immunol.*, **162**, 6967–6970.

Blake,J.D. and Cohen,F.E. (2001) Pairwise sequence alignment below the twilight zone. *J. Mol. Biol.*, **307**, 721–735.

Boeckmann,B. *et al*. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, **31**, 365–370.

Bogdan,C. *et al*. (1991) Macrophage deactivation by interleukin 10. *J. Exp. Med.*, **174**, 1549–1555.

Breen,E.C. *et al*. (2001) Viral interleukin 6 stimulates human peripheral blood B cells that are unresponsive to human interleukin 6. *Cell Immunol.*, **212**, 118–125.

Cameron,C.M. *et al*. (2005) Myxoma virus M141R expresses a viral CD200 (vOX-2) that is responsible for down-regulation of macrophage and T-cell activation *in vivo*. *J. Virol.*, **79**, 6052–6067.

Casari,G. *et al*. (1995) A method to predict functional residues in proteins. *Nat. Struct. Biol.*, **2**, 171–178.

Cha,T.A. *et al*. (1996) Human cytomegalovirus clinical isolates carry at least 19 genes not found in laboratory strains. *J. Virol.*, **70**, 78–83.

D'Andrea,A. *et al*. (1993) Interleukin 10 (IL-10) inhibits human lymphocyte interferon gamma-production by suppressing natural killer cell stimulatory factor/IL-12 synthesis in accessory cells. *J. Exp. Med.*, **178**, 1041–1048.

de Waal Malefyt,R. *et al*. (1991) Interleukin 10 (IL-10) and viral IL-10 strongly reduce antigen-specific human T cell proliferation by diminishing the antigen-presenting capacity of monocytes via downregulation of class II major histocompatibility complex expression. *J. Exp. Med.*, **174**, 915–924.

del Sol Mesa,A. *et al*. (2003) Automatic methods for predicting functionally important residues. *J. Mol. Biol.*, **326**, 1289–1302.

Desrosiers,R.C. *et al*. (1997) A herpesvirus of rhesus monkeys related to the human Kaposi's sarcoma-associated herpesvirus. *J. Virol.*, **71**, 9764–9769.

Ferrara,N. (2001) Role of vascular endothelial growth factor in regulation of physiological angiogenesis. *Am. J. Physiol. Cell Physiol.*, **280**, C1358–1366.

Foster-Cuevas,M. *et al*. (2004) Human herpesvirus 8 K14 protein mimics CD200 in down-regulating macrophage activation through CD200 receptor. *J. Virol.*, **78**, 7667–7676.

Guerin,J.L. *et al*. (2001) Characterization and functional analysis of Serp3: a novel myxoma virus-encoded serpin involved in virulence. *J. Gen. Virol.*, **82**, 1407–1417.

Holm,L. and Sander,C. (1998) Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, **14**, 423–429.

Holzerlandt,R. *et al*. (2002) Identification of new herpesvirus gene homologs in the human genome. *Genome Res.*, **12**, 1739–1748.

Isaacs,S.N. *et al*. (1992) Vaccinia virus complement-control protein prevents antibody-dependent complement-enhanced neutralization of infectivity and contributes to virulence. *Proc. Natl Acad. Sci. USA*, **89**, 628–632.

Kaleeba,J.A. *et al*. (1999) A rhesus macaque rhadinovirus related to Kaposi's sarcoma-associated herpesvirus/human herpesvirus 8 encodes a functional homologue of interleukin-6. *J. Virol.*, **73**, 6177–6181.

Kara,P.D. *et al*. (2003) Comparative sequence analysis of the South African vaccine strain and two virulent field isolates of Lumpy skin disease virus. *Arch. Virol.*, **148**, 1335–1356.

Kotenko,S.V. *et al*. (2000) Human cytomegalovirus harbors its own unique IL-10 homolog (cmvIL-10). *Proc. Natl Acad. Sci. USA*, **97**, 1695–1700.

Lee,C.G. *et al*. (2004) Vascular endothelial growth factor (VEGF) induces remodeling and enhances TH2-mediated sensitization and inflammation in the lung. *Nat. Med.*, **10**, 1095–1103.

Lucas,A. and McFadden,G. (2004) Secreted immunomodulatory viral proteins as novel biotherapeutics. *J. Immunol.*, **173**, 4765–4774.

Macen,J.L. *et al*. (1993) SERP1, a serine proteinase inhibitor encoded by myxoma virus, is a secreted glycoprotein that interferes with inflammation. *Virology*, **195**, 348–363.

Mayrose,I. *et al*. (2004) Comparison of site-specific rate-inference methods for protein sequences: empirical Bayesian methods are superior. *Mol. Biol. Evol.*, **21**, 1781–1791.

Moore,K.W. *et al*. (1990) Homology of cytokine synthesis inhibitory factor (IL-10) to the Epstein–Barr virus gene BCRFI. *Science*, **248**, 1230–1234.

Mullberg,J. *et al*. (2000) IL-6 receptor independent stimulation of human gp130 by viral IL-6. *J. Immunol.*, **164**, 4672–4677.

Petit,F. *et al*. (1996) Characterization of a myxoma virus-encoded serpin-like protein with activity against interleukin-1 beta-converting enzyme. *J. Virol.*, **70**, 5860–5866.

Pruitt,K.D. and Maglott,D.R. (2001) RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Res.*, **29**, 137–140.

Pupko,T. *et al*. (2002) Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. *Bioinformatics*, **18**(Suppl. 1), S71–S77.

Rosengard,A.M. *et al*. (2002) Variola virus immune evasion design: expression of a highly efficient inhibitor of human complement. *Proc. Natl Acad. Sci. USA*, **99**, 8808–8813.

Rost,B. (1999) Twilight zone of protein sequence alignments. *Protein Eng.*, **12**, 85–94.

Seet,B.T. *et al*. (2003) Poxviruses and immune evasion. *Annu. Rev. Immunol.*, **21**, 377–423.

Thompson,J.D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.

Tulman,E.R. *et al*. (2000) The genome of a very virulent Marek's disease virus. *J. Virol*., **74**, 7980–7988.

Upton,C. *et al*. (1991) Myxoma virus expresses a secreted protein with homology to the tumor necrosis factor receptor gene family that contributes to viral virulence. *Virology*, **184**, 370–382.

Zhang,Z. and Henzel,W.J. (2004) Signal peptide prediction based on analysis of experimentally verified cleavage sites. *Protein Sci*., **13**, 2819–2824.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4223–4229
doi:10.1093/bioinformatics/bti697

Sequence analysis

Discrimination of outer membrane proteins using support vector machines

Keun-Joon Park[1,2], M. Michael Gromiha[1,*], Paul Horton[1] and Makiko Suwa[1]

[1]Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST), AIST Tokyo Waterfront Bio-IT Research Building, 2–42 Aomi, Koto-ku, Tokyo 135–0064, Japan and [2]Laboratory of Functional Analysis *in silico*, Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokane-dai Minato-ku, Tokyo 108–8639, Japan

Received on June 27, 2005; revised and accepted September 27, 2005
Advance Access publication October 4, 2005

ABSTRACT

Motivation: Discriminating outer membrane proteins from other folding types of globular and membrane proteins is an important task both for dissecting outer membrane proteins (OMPs) from genomic sequences and for the successful prediction of their secondary and tertiary structures.

Results: We have developed a method based on support vector machines using amino acid composition and residue pair information. Our approach with amino acid composition has correctly predicted the OMPs with a cross-validated accuracy of 94% in a set of 208 proteins. Further, this method has successfully excluded 633 of 673 globular proteins and 191 of 206 α-helical membrane proteins. We obtained an overall accuracy of 92% for correctly picking up the OMPs from a dataset of 1087 proteins belonging to all different types of globular and membrane proteins. Furthermore, residue pair information improved the accuracy from 92 to 94%. This accuracy of discriminating OMPs is higher than that of other methods in the literature, which could be used for dissecting OMPs from genomic sequences.

Availability: Discrimination results are available at http://tmbeta-svm.cbrc.jp

Contact: michael-gromiha@aist.go.jp

INTRODUCTION

Outer membrane proteins (OMPs) perform a variety of functions, such as mediating non-specific, passive transport of ions and small molecules, selectively allowing the passage of molecules such as maltose and sucrose (Schirmer *et al*., 1995; Forst *et al*., 1998; Schulz, 2000; Wimley, 2003) and are involved in voltage-dependent anion channels (Mannella, 1998). These proteins contain β-strands as their membrane spanning segments and are found in the outer membranes of bacteria, mitochondria and chloroplasts (Schulz, 2002). A comparative analysis on the distribution of amino acid residues in α-helical and β-barrel membrane proteins shows that the membrane part of OMPs is more complex than that of transmembrane helical proteins due to the intervention of many charged and polar residues in the membrane (Gromiha *et al*., 1997; Gromiha, 1999). Consequently, the success rate of discriminating β-barrel membrane proteins from other proteins is significantly lower than that of α-helical membrane proteins (Hirokawa *et al*., 1998; Chen and Rost, 2002).

Recently, several methods have been proposed for discriminating OMPs from amino acid sequences (Gnanasekaran *et al*., 2000; Wimley, 2002; Martelli *et al*., 2002; Liu *et al*., 2003; Bagos *et al*., 2004; Natt *et al*., 2004; Garrow *et al*., 2005). Gnanasekaran *et al*. (2000) developed a structure-based sequence alignment method for discriminating β-stranded membrane proteins and reported an accuracy of 80%. Wimley (2002) analyzed the architecture of 15 OMPs and proposed a method based on hydrophobicity for identifying β-barrel membrane proteins in genomic sequences. It has been reported that this method correctly identified 75% of the OMPs (Liu *et al*., 2003). Martelli *et al*. (2002) used 12 OMPs and developed a Hidden Markov Model (HMM) for picking up the β-barrel membrane proteins and reported an accuracy of 84% in a set of 145 OMPs. Liu *et al*. (2003) analyzed the amino acid composition in the membrane spanning regions of 12 β-barrel membrane proteins and applied the information for discrimination, which showed 85% accuracy when tested with 241 OMPs. Bagos *et al*. (2004) developed an algorithm based on HMM for discriminating OMPs and reported an accuracy of 89% in a set of 133 OMPs. Natt *et al*. (2004) used a set of 16 OMPs and proposed a machine learning technique for discrimination, which showed an average accuracy of 90% in a set of randomly selected 100 globular and 16 OMPs. Garrow *et al*. (2005) proposed a modified *k*-nearest neighbor algorithm and reported an accuracy of 92.5% using weighted amino acids and evolutionary information. Martelli *et al*. (2003) reviewed the performance of a few methods for the discrimination and prediction of membrane protein structures. All these methods used minimal information for the analysis and the prediction accuracy is rather modest.

Further, few methods have been suggested to screen OMPs from genomic sequences (Zhai and Saier, 2002; Berven *et al*., 2004; Bigelow *et al*., 2004). Zhai and Saier (2002) developed a β-barrel finder program based on secondary structure, hydropathy and amphipathicity parameters and used it for identifying OMPs in *Escherichia coli* genome. This algorithm has recognized 10 families correctly and missed the proteins from 4 OMP families. Berven *et al*. (2004) proposed a program for identifying OMPs using two factors: (1) C-terminal pattern typical of many integral β-barrel proteins and (2) integral β-barrel score based on the extent to which the sequence contains stretches of amino acids typical of

*To whom correspondence should be addressed.

transmembrane β-strands. Bigelow *et al*. (2004) introduced a profile-based HMM for discriminating OMPs and suggested the probable OMPs in genomic sequences of 72 Gram-negative bacteria.

Classification based on support vector machines (SVMs) has several applications in bioinformatics and computational biology. It has been widely used to predict protein secondary structures (Nguyen and Rajapakse, 2005b), solvent accessibility (Yuan *et al*., 2002; Kim and Park, 2004; Nguyen and Rajapakse, 2005a), protein–protein binding sites (Bradford and Westhead, 2004; Res *et al*., 2005), remote protein homology detection (Busuttil *et al*., 2004), protein domains (Vlahovicek *et al*., 2005) protein subcellular localization (Hua and Sun, 2001; Park and Kanehisa, 2003; Nair and Rost, 2005) and gene and tissue classification from microarray expression data (Brown *et al*., 2000). The biological and bioinformatics applications of SVMs have been reviewed in Byvatov and Schneider (2003) and Yang (2004).

In our earlier work, we have developed a statistical method based on amino acid composition and residue pairs for discriminating OMPs (Gromiha and Suwa, 2005; Gromiha *et al*., 2005). It showed an accuracy of 89% in a dataset of 377 OMPs (Gromiha and Suwa, 2005). As SVMs have a wide range of applications and perform well in prediction algorithms, we have developed a method using SVMs for discriminating OMPs. We have examined the performance of SVMs using different kernel functions and parameters, and various sequence features represented by the composition of amino acid residues and residue pairs. We observed that SVMs could discriminate the OMPs at an accuracy of 92% with amino acid composition and the accuracy is improved to 94% using residue pair information. This method has the ability to correctly pick up the OMPs and exclude other folding types of globular proteins at high accuracy levels.

MATERIALS AND METHODS

Datasets

All protein sequences were collected from the dataset of Gromiha and Suwa (2005), extracted from the PSORT-B database (Gardy *et al*., 2003) for membrane proteins and the PDB40D_1.37 database of SCOP (Murzin *et al*., 1995; Berman *et al*., 2000) for globular proteins. The dataset included a total of 377 OMP sequences, 674 globular protein sequences and 268 α-helical membrane proteins. In these datasets, OMPs and α-helical membrane proteins have many redundant sequences and the globular proteins have been filtered with <40% sequence similarity.

Removal of highly similar sequences

Sequences with a high degree of similarity to other sequences were removed by all-to-all sequence similarity check using the program CD-HIT (Li *et al*., 2001), which produces a non-redundant protein database, using a greedy incremental algorithm as implemented by Holm and Sander (1998). We produced non-redundant protein datasets for all types of proteins by CD-HIT with full-length matches of <40% sequence identity. We did not consider the proteins containing B, X or Z in the amino acid sequence.

The total number of proteins in the final dataset are 208 OMPs, 673 globular proteins and 206 α-helical membrane proteins and the sequences are available at http: www.cbrc.jp/~gromiha/omp/dataset2.html.

Support vector machines

SVM is a learning algorithm (Cristianini and Shawe-Taylor, 2000), which from a set of positively and negatively labeled training vectors learns a classifier that can be used to classify new unlabeled test samples. SVM learns the classifier by mapping the input training samples $\{x_1, \ldots, x_n\}$ into a possibly high-dimensional feature space and seeking a hyperplane in this space which separates the two types of examples with the largest possible margin, i.e. distance to the nearest points. If the training set is not linearly separable, SVM finds a hyperplane, which optimizes a trade-off between good classification and large margin.

For actual implementation we used the freely downloadable SVM-light package by Joachims (1999). We tested linear, polynomial and RBF (radial basis function) kernels with various parameters.

Compositions of amino acids and amino acid pairs

Each protein in the training dataset of N proteins is characterized by a vector $\vec{x}_i$ ($i = 1, \ldots, N$) representing certain sequence features, together with the positive label or the negative labels for discriminating two different groups (e.g. globular and OMPs). In addition to the amino acid composition, we considered the amino acid pair composition. The vector $\vec{x}_i$ has 20 elements for the amino acid composition and 400 elements for the amino acid pair composition. Amino acid composition is defined as the ratio between the number of occurrences of a specific amino acid residue and the total number of residues in a protein.

For each ordered pair of amino acids, the amino acid pair composition C_{ij} is defined as the number of occurrences of amino acid i followed by j divided by the total number of adjacent pairs (i.e. the length of the sequence minus one).

Feature selection

At first, we considered selection with the Fisher discriminant ratio (FDR). It is defined as

$$\mathrm{FDR}_i = \frac{(\mu^i_{\mathrm{OMP}} - \mu^i_{\mathrm{m}})^2}{(\sigma^i_{\mathrm{OMP}})^2 + (\sigma^i_{\mathrm{m}})^2}, \quad (1)$$

where μ^i_{OMP} and σ^i_{OMP} are the mean and variance of the i-th amino acid (20 residues) or amino acid pair (400 residue pairs) composition in OMPs, respectively. μ^i_{m} and σ^i_{m} denote the mean and variance of amino acid i or amino acid pair i composition in globular proteins, α-helical membrane proteins or non-OMPs. In this study, the group of non-OMPs means the sum of globular proteins and α-helical membrane proteins. Selecting features with the highest FDR values is often employed as a simple technique for feature selection (Liu *et al*., 2003).

Another selection procedure is done according to backward and forward selection for handling datasets with amino acid composition and amino acid pair composition. In this work, backward selection (elimination) started with the complete set of 20 amino acid composition features. It evaluates all the subsets by eliminating the composition of one amino acid from the complete set and selects the one with the best performance measure of Matthews correlation coefficient (MCC) [Equation (5)]. It then evaluates all the subsets with one feature less than the best subset from the previous step and selects the second best. The process stops when decreasing the size of current best subset leads to a lower prediction rate. Forward selection for the 400 amino acid pair composition features was started from the result of backward selection subset of amino acid composition features. It evaluates all the one-additional feature subsets and selects the one with the best prediction rate. It then builds all the two-additional feature subsets that include the features already selected from the first step and finds the best one. This process continues until increasing the size of the current subset leads to a lower performance measure. Although FDR has been used for feature selection in some studies, we adopted cross-validated classification accuracy for the selection criteria in this work.

Since we have small number of proteins in our dataset, it is expected that training with the complete set of pair composition features (400D) may cause overfitting. Hence, we performed forward selection with amino acid pair composition features to find a good small feature set.

5-Fold cross-validation test

The prediction performance was examined by the 5-fold cross-validation test, in which the three types of proteins were randomly divided into five subsets of approximately equal size. This means that the data were partitioned into training and test data in five different ways. After training the SVMs with a collection of four subsets, the performance of the SVMs was tested against the fifth subset. This process was repeated five times so that every subset was once used as the test data.

In order to assess the accuracy of prediction methods we use four measures. The sensitivity, specificity and overall accuracy are defined by

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3}$$

$$\text{overall accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{4}$$

where TP, FP, TN and FN refer to the number of true positives, false positives, true negatives and false negatives proteins, respectively.

The MCC (Matthews, 1975) is defined as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}. \tag{5}$$

The value of MCC is one for a perfect prediction and zero for a completely random assignment. Sensitivity measures our ability to correctly predict OMPs, while specificity measures our ability for correctly reject non-OMPs.

RESULTS AND DISCUSSION

Amino acid composition in OMPs, globular and α-helical membrane proteins

Table 1 shows the result of amino acid composition for 20 amino acid residues in OMPs, globular and α-helical membrane proteins. The FDR values [Equation (1)] also have been computed and the results are presented in Table 2. The residues Ser, Glu, Cys and His show high FDR values (FDR > 0.2) between OMPs and globular proteins. These residues also show significant differences (>1) in percent composition between globular proteins and OMPs (Table 1). The formation of disulfide bonds between Cys residues requires an oxidative environment and such disulfide bridges are not usually found in intracellular proteins (Branden and Tooze, 1999). The analysis of the three-dimensional (3D) structures of 15 β-barrel OMPs shows the presence of just 8 (0.1%) Cys residues and none of them is in the membrane part (Gromiha and Suwa, 2003). Hence, the occurrence of Cys is significantly higher in globular proteins than in OMPs. Glu is a strong helix former (Chou and Fasman, 1978) and this tendency influences its higher occurrence in globular proteins than OMPs.

The composition of Ser shows the opposite tendency that the composition is higher in OMPs than globular proteins. The structural analysis of several OMPs shows that Ser plays an important role in the stability and function of OMPs (Gromiha and Suwa, 2005). In outer membrane protein A (OmpA, PDB code no. 1QJP), the interior of β-strands contain an extended hydrogen bonding network of charged and polar residues (Pautsch and Schulz, 2000). In outer membrane protease OmpT (PDB code no. 1I78), the side chains of the residues Ser22, Gln228 and Asn258, located above the membrane, form hydrogen bonds to main chain atoms in the β-barrel (Vandeputte-Rutten *et al*., 2001). Interestingly, none of the residues, which have high composition in globular proteins (Glu, His and Cys), is involved in such patterns (Pautsch and Schulz, 2000; Vandeputte-Rutten *et al*., 2001). Further, the importance of Ser to the stability and function of OMPs has been reported for outer

Table 1. Amino acid composition for the 20 amino acid residues in outer membrane, globular and α-helical membrane proteins

Category	Residue	Composition (%) Outer membrane (208)	Globular (673)	α-Helical membrane (206)
Aliphatic	Ala	9.4	8.4	10.3
	Gly	8.7	7.7	8.3
	Ile	4.7	5.8	7.5
	Leu	8.9	8.5	12.7
	Pro	3.7	4.5	4.3
	Val	6.7	7.2	8.2
Aromatic	Phe	3.7	3.8	5.5
	Tyr	4.1	3.4	2.8
	Trp	1.2	1.3	2.0
Negative charged	Asp	5.9	5.8	3.3
	Glu	4.9	6.7	3.7
Positive charged	Arg	5.2	5.1	4.4
	His	1.2	2.2	1.7
	Lys	4.9	6.2	3.4
Polar	Asn	5.4	4.4	3.1
	Gln	4.7	3.9	3.2
	Ser	8.0	5.8	5.9
	Thr	6.3	5.7	5.2
Sulfur containing	Cys	0.4	1.5	0.8
	Met	1.7	2.2	3.6

Table 2. FDR of each amino acid between OMP and globular or α-helical membrane proteins [see Equation (1)]. High FDR values in globular proteins (>0.2) are shown in bold.

Residue	FDR (Fisher discriminant ratio) Globular	α-Helical membrane
Ala	0.033	0.039
Arg	0.003	0.070
Asn	0.142	1.281
Asp	0.000	1.487
Cys	**0.267**	0.198
Gln	0.078	0.341
Glu	**0.312**	0.231
Gly	0.074	0.012
His	**0.247**	0.109
Ile	0.114	0.900
Leu	0.014	1.083
Lys	0.109	0.302
Met	0.096	1.528
Phe	0.000	0.496
Pro	0.082	0.066
Ser	**0.495**	0.745
Thr	0.039	0.254
Trp	0.004	0.295
Tyr	0.070	0.343
Val	0.018	0.293

membrane cobalamin transporter (BtuB, PDB code no. 1NQE), anion-selective porin (Omp32, PDB code no. 1E54), etc. (Zeth *et al*., 2000; Chimento *et al*., 2003a,b).

The amino acid composition of 20 residues in OMPs and α-helical membrane proteins shows that Met, Asp, Asn and Leu have a significant difference (FDR > 1.0) between them (Table 2). The group of OMPs showed higher composition of Asp and Asn than that in α-helical membrane proteins. The occurrence of residues Leu and Met is higher in α-helical membrane proteins than that in OMPs. Further, from Table 1, the residues Ala, Ile, Leu, Val, Phe, Trp and Met have higher composition in α-helical membrane proteins than OMPs. This result reflects the presence of hydrophobic stretches of amino acid residues in the membrane part of α-helical membrane proteins.

Kernel selection

We begin with the selection of a kernel from three possibilities: the simple linear kernel, the polynomial kernel and the RBF kernel. The performance of each classifier was measured by examining how well the classifier identified positive and negative examples in the test sets, according to the 5-fold cross-validation test. In this analysis, the RBF kernel showed the best performance with overall prediction rate or MCC values. Various values of the parameter γ were also tested for the RBF kernel, and our choice was $\gamma = 0.02$ or 0.03 for feature selection. The parameter C, which controls the trade-off between training error and margin, was set to 0.5 or 0.6 in this work. The RBF kernel is defined by

$$K(\vec{x}, \vec{y}) = \exp(-\gamma \|\vec{x} - \vec{y}\|^2), \quad (7)$$

where $\gamma = 1/\sigma^2$ and σ is called the width of the (Gaussian) kernel. Instead of explicitly mapping the objects to the possibly high-dimensional feature space, SVM usually works implicitly in the feature space by only computing the corresponding kernel $K(\vec{x}, \vec{y})$ between any two objects x and y.

Discrimination of OMPs and globular proteins

Table 3 shows the results of the 5-fold cross-validation tests for the RBF kernel SVM classifiers with the parameters $\gamma = 0.03$ and $C = 0.5$ using amino acid composition. The initial overall prediction rate with 20 amino acid composition was 91.1% and the value of MCC was 0.757. Starting from this 20 amino acid composition feature vector, we executed backward feature selection, and the overall prediction rate increased to 92.5%. This reduced set has a 17D feature vector with the exclusion of residues Ala, Glu and Lys. This result reveals that the feature selection with FDR values moderately improved the performance.

We performed a second feature selection (forward) for 400 amino acid pair composition features starting from the 17D feature subset. The second feature selection chose 8 pair compositions, and the final version yields a 25D feature vector. The information of the additional eight pair compositions improved the value of MCC from 0.798 to 0.846. The selected pair compositions were EL, AA, AT, SS, AG, AI, ID and YE.

Discrimination of OMPs and α-helical membrane proteins

Table 4 shows the results of the 5-fold cross-validation tests for the RBF kernel SVM classifiers with the parameters $\gamma = 0.03$ and $C = 0.6$ using amino acid composition. We observed that SVMs could discriminate the OMPs at an overall accuracy of 94.9% with 20 amino acid compositions and the prediction rate improved to 95.9% after backward selection. After backward elimination, the composition information of residues Arg, Asp, Tyr, Cys and His were removed from the training and test datasets (15D feature vector). The performance of this discrimination including overall accuracy and MCC was better than the discrimination of OMPs and globular proteins. In our interpretation, the results shown in Tables 3 and 4 indicate that discrimination of OMPs and α-helical membrane proteins seems to be easier than that of OMPs and globular proteins. It is reasonable that the discrimination of OMPs and α-helical membrane proteins at high accuracy is consistent with higher FDR values for α-helical membrane proteins compared with globular proteins (Table 2). Further, the information about the occurrence of hydrophobic residues in the membrane part of α-helical membrane proteins can discriminate this class of proteins at high accuracy as reported in Mitaku *et al*. (2002).

We have included the information about amino acid pair composition with the result of first feature selection (composition of 15 amino acids) and we observed that there is no improvement in the prediction rate (Table 4). This result revealed that the composition of reduced amino acids could discriminate the groups of OMPs and α-helical membrane proteins successfully.

Table 3. Discrimination of OMPs and globular proteins

Composition	Prediction rate (%)			
	Sensitivity	Specificity	Overall	MCC
Amino acid residues (20D)	82.7	93.8	91.1	0.757
Residue pairs (400D)	83.2	97.3	94.0	0.830
Residues and residue pairs (420D)	63.5	100.0	91.4	0.755
Reduced amino acids[a] (17D)	87.5	94.1	92.5	0.798
Reduced amino acids and residue pairs[b] (25D)	88.0	96.4	**94.4**	**0.846**

The number given in parentheses indicates the number of variables.
[a]After backward feature selection, the composition features of Ala, Glu and Lys were removed from the training and test sets (17D feature vectors).
[b]The second (forward) feature selection included eight kinds of amino acid pair compositions (EL, AA, AT, SS, AG, AI, ID and YE).
Highest prediction rate and MCC are shown in bold.

Table 4. Discrimination of OMPs and α-helical membrane proteins

Composition	Prediction rate (%)			
	Sensitivity	Specificity	Overall	MCC
Amino acid residues (20D)	98.6	91.3	94.9	0.901
Residue pairs (400D)	99.0	90.3	94.7	0.897
Residues and left residue pairs (420D)	95.7	89.8	92.8	0.856
Reduced amino acids[a] (15D)	99.0	92.7	**95.9**	**0.920**
Reduced amino acids left and residue pairs[b] (15D)	99.0	92.7	**95.9**	**0.920**

[a]After backward feature selection, the composition features of Arg, Asp, Tyr, Cys and His were removed from the training and test sets (15D feature vector).
[b]None of the residue pairs was selected by the second feature selection.
Highest prediction rate and MCC are shown in bold.

Table 5. Discrimination of OMPs and non-OMPs

Composition	Prediction rate (%)			
	Sensitivity	Specificity	Overall	MCC
Amino acid residues (20D)	87.5	92.6	91.6	0.752
Residue pairs (400D)	86.5	96.4	94.5	0.823
Residues and residue pairs (420D)	79.3	99.0	95.2	0.840
Reduced amino acids[a] (18D)	89.9	92.5	92.0	0.767
Reduced amino acids and residue pairs[b] (28D)	90.9	94.7	**93.9**	**0.816**

Non-OMPs (879) consist of globular proteins (673) and α-helical membrane proteins (206).

[a]The first feature selection was backward selection for 20 kinds of amino acid and the composition features of Ala and Glu were removed.

[b]In the second feature selection, the composition features QA, DF, DA, KK, EF, NK, DR, YN, FF and LI were added to training and test sets ($20 - 2 + 10 = 28$D feature vector) by forward selection. The overall prediction rate and MCC are shown in bold.

Table 6. Top 10 amino acid pair compositions (by FDR) between OMPs and, globular, α-helical membrane proteins or non-OMPs

Globular		α-Helical		Non-OMPs	
Amino acid pair	FDR	Amino acid pair	FDR	Amino acid pair	FDR
LS	0.178	LI	1.000	SS	0.166
LG	0.163	LL	0.894	SY	0.148
SL	0.160	IL	0.694	GY	0.131
SS	0.146	AI	0.660	QS	0.128
SA	0.140	FL	0.625	LS	0.126
EE	0.138	LV	0.604	IL	0.125
SY	0.126	IF	0.603	SN	0.123
AS	0.126	IV	0.572	SA	0.122
GY	0.111	IA	0.526	NS	0.115
QS	0.109	LM	0.490	AQ	0.113

Discrimination of OMPs and non-OMPs

We have examined the discrimination ability of OMPs and non-OMPs by the present method. For this purpose, we defined a non-OMPs dataset by combining the globular and α-helical membrane proteins. Table 5 shows the result of 5-fold cross-validation tests for the RBF kernel SVM classifiers with the parameters $\gamma = 0.02$ and $C = 0.5$ using amino acid and amino acid pair composition. Again, we executed a two-step feature selection, consisting of first the backward selection and then the forward selection. Backward selection reduced the amino acid composition features to 18 by excluding the residues Ala and Glu. This backward selection improved the sensitivity of OMPs from 87.5 to 89.9% and the overall accuracy from 91.6 to 92%.

After backward elimination, we did forward selection for 400 amino acid pair composition to the 18 amino acid composition feature subset. As a result of forward selection, the final feature vector of our method contained 10 additional amino acid pair compositions of QA, DF, DA, KK, EF, NK, DR, YN, FF and LI. Interestingly, most of these amino acid pairs contain either a charged or an aromatic residue. The inclusion of representative amino acid pairs, such as two like/oppositely charged (KK, DR), hydrophobic (LI), aromatic (FF) and the combination of charged and hydrophobic (DA), polar and charged (NK), charged and aromatic (EF), aromatic and polar (YN), polar and hydrophobic (QA) and charged and hydrophobic (DA) improved the prediction accuracy. However, the inclusion of amino acid pairs that show significant difference between OMPs and non-OMPs (Table 6) did not improve the accuracy. This might be due to the fact that such information is already available in the feature selection of 18 amino acid composition. Another reason might be the fact that the hyperplane of an SVM is constructed by combination of several features, whereas the FDR reflects the difference of only one feature. The additional information about the 10 amino acid pairs increased the MCC from 0.767 to 0.816 and the accuracy from 92 to 94%.

The combination of all the amino acid and residue pair compositions raised the correlation up to 0.84 and the overall accuracy is 95.2% (Table 5). Although the accuracy and correlation are high it has 420 variables (20 amino acids and 400 residue pairs). Generally, fitting the data with a minimum number of variables increases the robustness of the results. In the present work we selected 28 feature variables (18 amino acids and 10 residue pairs), giving an accuracy of 93.9%. This is almost as high as the accuracy (95.2%) obtained when using all 420 features. Thus we recommend using the selected features for discrimination.

Implementation

The prediction method presented in this paper is implemented as a computer program named TMBETA-SVM and the web service is made available at http://tmbeta-svm.cbrc.jp. The program predicts OMPs based on the compositions of amino acids and amino acid pairs, using the SVM classifiers with the RBF kernel and the parameters $C = 0.5$ and $\gamma = 0.02$. The datasets used in this work are also available at http://www.cbrc.jp/~gromiha/omp/dataset2.html.

Comparison with other methods

Liu *et al*. (2003) proposed a method based on the amino acid composition of residues in transmembrane β-strand segments of 12 proteins in PDB to discriminate β-barrel membrane proteins and claimed an accuracy of 85.4% on a set of 241 OMPs. As the membrane spanning segments are used to compute the amino acid composition, this method could identify the OMPs, which have a high content of amino acid residues in the membrane but it missed the proteins with fewer membrane spanning β-strand segments. Martelli *et al*. (2002) devised an HMM method using 12 OMPs in PDB and tested the method on 145 OMPs, which yielded an accuracy of 84%. Bagos *et al*. (2004) used an HMM for discriminating β-barrel OMPs and reported an accuracy of 88.8% for a set of 133 OMPs. The method based on amino acid composition showed an accuracy of 89% on a dataset of 377 OMPs (Gromiha and Suwa, 2005). In this work, we have used a set of 208 OMPs, 673 globular proteins, and 206 α-helical membrane proteins. The OMPs were discriminated with an accuracy of 94% from the pool of 1087 sequences, while correctly excluding 96% of the transmembrane α-helical proteins. This compares favorably to an accuracy of 90% reported by an HMM (Martelli *et al*., 2002). Although the direct comparison of accuracies reported by different methods is not appropriate (due to differences in datasets and validation procedures) it may give some information about the performance of different methods. We have examined the discriminative power of the program TMB-Hunt (Garrow *et al*., 2005), which claimed

the highest accuracy of 92.5%, using the publicly available web server and the same dataset of 1087 proteins used in the present work. We observed that this program discriminated the OMPs with an accuracy of 89.2% whereas the cross-validation accuracy obtained by the present work is 93.9%. The MCC obtained with TMB-Hunt is 0.729 and that obtained with our method is 0.816. The high accuracy achieved by the present method is due to the effectiveness of the method as well as the information gained from the large dataset of globular proteins and OMPs.

CONCLUSIONS

We have developed an SVM-based method for discriminating OMPs using amino acid composition and residue pairs. The influence of amino acids and residue pairs for improving the accuracy has been analyzed. We found that the selection of 18 amino acid residues could discriminate the OMPs at an accuracy of 92%. Further, the inclusion of 10 residue pairs raised the accuracy to 94% and the correlation from 0.77 to 0.82. We have developed a web server for discriminating OMPs, which takes the amino acid sequence as input, and the predicted type of the protein is displayed as output. The program is available online at http://tmbeta-svm.cbrc.jp/

Conflict of Interest: none declared.

REFERENCES

Bagos,P.G. *et al.* (2004) A hidden Markov model method, capable of predicting and discriminating β-barrel outer membrane proteins. *BMC Bioinformatics*, **5**, 29.

Berman,H.M. *et al.* (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Berven,F.S. *et al.* (2004) BOMP: a program to predict integral β-barrel outer membrane proteins encoded within genomes of Gram-negative bacteria. *Nucleic Acids Res.*, **32**, W394–W399.

Bigelow,H.R. *et al.* (2004) Predicting transmembrane beta-barrels in proteomes. *Nucleic Acids Res.*, **32**, 2566–2577.

Bradford,J.R. and Westhead,D.R. (2005) Improved prediction of protein–protein binding sites using a support vector machines approach. *Bioinformatics*, **21**, 1487–1494.

Branden,C. and Tooze,C. (1999) *Introduction to Protein Structure*. Garland Publishing Inc., New York.

Brown,M.P.S. *et al.* (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl Acad. Sci. USA*, **97**, 262–267.

Busuttil,S. *et al.* (2004) Support vector machines with profile-based kernels for remote protein homology detection. *Genome Inform. Ser. Workshop Genome Inform.*, **15**, 191–200.

Byvatov,E. and Schneider,G. (2003) Support vector machine applications in bioinformatics. *Appl. Bioinformatics*, **2**, 67–77.

Chen,C.P. and Rost,B. (2002) State-of-the-art in membrane protein prediction. *Appl. Bioinformatics*, **1**, 21–35.

Chimento,D.P. *et al.* (2003a) Substrate-induced transmembrane signaling in the cobalamin transporter BtuB. *Nat. Struct. Biol.*, **10**, 394–401.

Chimento,D.P. *et al.* (2003b) The *Escherichia coli* outer membrane cobalamin transporter BtuB: structural analysis of calcium and substrate binding, and identification of orthologous transporters by sequence/structure conservation. *J. Mol. Biol.*, **332**, 999–1014.

Chou,P.Y. and Fasman,G.D. (1978) Prediction of the secondary structure of proteins from their amino acid sequence. *Adv. Enzymol.*, **47**, 45–148.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, Cambridge, MA.

Forst,D. *et al.* (1998) Structure of the sucrose-specific porin ScrY from *Salmonella typhimurium* and its complex with sucrose. *Nat. Struct. Biol.*, **5**, 37–46.

Gardy,J.L. *et al.* (2003) PSORT-B: improving protein subcellular localization prediction for Gram-negative bacteria. *Nucleic Acids Res.*, **31**, 3613–3617.

Garrow,A.G. *et al.* (2005) TMB-Hunt: a web server to screen sequence sets for transmembrane beta-barrel proteins. *Nucleic Acids Res.*, **33**, W188–W192.

Gnanasekaran,T.V. *et al.* (2000) Profiles from structure based sequence alignment of porins can identify beta stranded integral membrane proteins. *Bioinformatics*, **16**, 839–842.

Gromiha,M.M. (1999) A simple method for predicting transmembrane alpha helices with better accuracy. *Protein Eng.*, **12**, 557–561.

Gromiha,M.M. and Suwa,M. (2003) Variation of amino acid properties in all-beta globular and outer membrane protein structures. *Int. J. Biol. Macromol.*, **32**, *93–98.*

Gromiha,M.M. and Suwa,M. (2005) A simple statistical method for discriminating outer membrane proteins with better accuracy. *Bioinformatics*, **21**, 961–968.

Gromiha,M.M. *et al.* (1997) Identification of membrane spanning beta strands in bacterial porins. *Protein Eng.*, **10**, 497–500.

Gromiha,M.M. *et al.* (2005) Application of residue distribution along the sequence for discriminating outer membrane proteins. *Comput. Biol. Chem.*, **29**, 135–142.

Hirokawa,T. *et al.* (1998) SOSUI: classification and secondary structure prediction system for membrane proteins. *Bioinformatics*, **14**, 378–379.

Holm,L. and Sander,C. (1998) Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, **14**, 423–429.

Hua,S. and Sun,Z. (2001) Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, **17**, 721–728.

Joachims,T. (1999) Making large-scale SVM learning practical. In Schölkopf,B., Burges,C. and Smola,A. (eds) *Advances in Kernel Methods—Support Vector Learning*. MIT Press, Cambridge, MA.

Kim,H. and Park,H. (2004) Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor. *Proteins*, **54**, 557–562.

Li,W. *et al.* (2001) Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, **17**, 282–283.

Liu,Q. *et al.* (2003) Identification of β-barrel membrane proteins based on amino acid composition properties and predicted secondary structure. *Comput. Biol. Chem.*, **27**, 355–361.

Mannella,C.A. (1998) Conformational changes in the mitochondrial channel protein, VDAC and their functional implications. *J. Struct. Biol.*, **121**, 207–218.

Martelli,P.L. *et al.* (2002) A sequence-profile-based HMM for predicting and discriminating β-barrel membrane proteins. *Bioinformatics*, **18**, S46–S53.

Martelli,P.L. *et al.* (2003) The prediction of membrane protein structure and genome structural annotation. *Comp. Funct. Genomics*, **4**, 406–409.

Matthews,B.W. (1975) Comparison of predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Mitaku,S. *et al.* (2002) Amphiphilicity index of polar amino acids as an aid in the characterization of amino acid preference at membrane-water interfaces. *Bioinformatics*, **18**, 608–616.

Murzin,A.G. *et al.* (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Nair,R. and Rost,B. (2005) Mimicking cellular sorting improves prediction of subcellular localization. *J. Mol. Biol.*, **348**, 85–100.

Natt,N.K. *et al.* (2004) Prediction of transmembrane regions of beta-barrel proteins using ANN- and SVM-based methods. *Proteins*, **56**, 11–18.

Nguyen,M.N. and Rajapakse,J.C. (2005a) Prediction of protein relative solvent accessibility with a two-stage SVM approach. *Proteins*, **59**, 30–37.

Nguyen,M.N. and Rajapakse,J.C. (2005b) Two-stage multi-class support vector machines to protein secondary structure prediction. *Pac. Symp. Biocomput.*, 346–357.

Park,K.-J. and Kanehisa,M. (2003) Prediction of protein subcellular locations by support vector machines using compositions of amino acid pairs. *Bioinformatics*, **19**, 1656–1663.

Pautsch,A. and Schulz,G.E. (2000) High-resolution structure of the OmpA membrane domain. *J. Mol. Biol.*, **298**, 273–282.

Res,I. *et al.* (2005) An evolution based classifier for prediction of protein interfaces without using protein structures. *Bioinformatics*, **21**, 2496–2501.

Schirmer,T. *et al.* (1995) Structural basis for sugar translocation through maltoporin channels at 3.1 A resolution. *Science*, **267**, 512–514.

Schulz,G.E. (2000) $\beta-$Barrel membrane proteins. *Curr. Opin. Struct. Biol.*, **10**, 443–447.

Schulz,G.E. (2002) The structure of bacterial outer membrane proteins. *Biochim. Biophys. Acta*, **1565**, 308–317.

Vandeputte-Rutten,L. *et al.* (2001) Crystal structure of the outer membrane protease OmpT from *Escherichia coli* suggests a novel catalytic site. *EMBO J.*, **20**, 5033–5039.

Vlahovicek,K. *et al.* (2005) The SBASE domain sequence resource, release 12: prediction of protein domain-architecture using support vector machines. *Nucleic Acids Res*., **33** (Database Issue), D223–D225.

Wimley,W.C. (2002) Toward genomic identification of beta-barrel membrane proteins: composition and architecture of known structures. *Protein Sci*., **11**, 301–312.

Wimley,W.C. (2003) The versatile β−barrel membrane protein. *Curr. Opin. Struct. Biol*., **13**, 404–411.

Yang,Z.R. (2004) Biological applications of support vector machines. *Brief Bioinformatics*, **5**, 328–338.

Yuan,Z. *et al.* (2002) Prediction of protein solvent accessibility using support vector machines. *Proteins*, **48**, 566–570.

Zeth,K. *et al.* (2000) Crystal structure of Omp32, the anion-selective porin from *Comamonas acidovorans*, in complex with a periplasmic peptide at 2.1 A resolution. *Structure*, **8**, 981–992.

Zhai,Y. and Saier,M.H.,Jr (2002) The β−barrel finder (BBF) program, allowing identification of outer membrane β−barrel proteins encoded within prokaryotic genomes. *Protein Sci*., **11**, 2196–2207.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4230–4238
doi:10.1093/bioinformatics/bti690

Structural bioinformatics

Bioinformatics analyses of circular dichroism protein reference databases

Robert W. Janes

School of Biological and Chemical Sciences, Queen Mary, University of London, Mile End Road, E1 4NS, UK

Received on August 2, 2005; revised on September 21, 2005; accepted on September 23, 2005
Advance Access publication September 27, 2005

ABSTRACT

Motivation: Circular dichroism (CD) spectroscopy has become established as a key method for determining the secondary structure contents of proteins which has had a significant impact on molecular biology. Many excellent mathematical protocols have been developed for this purpose and their quality is above question. However, reference database sets of proteins, with CD spectra matched to secondary structure components derived from X-ray structures, provide the key resource for this task. These databases were created many years ago, before most CD spectrophotometers became standardized and before it was commonplace to validate X-ray structures prior to publication. The analyses presented here were undertaken to investigate the overall quality of these reference databases in light of their extensive usage in determining protein secondary structure content from CD spectra.

Results: The analyses show that there are a number of significant problems associated with the CD reference database sets in current use. There are disparities between CD spectra for the same protein collected by different groups. These include differences in magnitudes, peak positions or both. However, many current reference sets are now amalgamations of spectra from these groups, introducing inconsistencies that can lead to inaccuracies in the determination of secondary structure components from the CD spectra. A number of the X-ray structures used fall short on the validation criteria now employed as standard for structure determination. Many have substantial percentages of residues in the disallowed regions of the Ramachandran plot. Hence their calculated secondary structure components, used as a foundation for the reference databases, are likely to be in error. Additionally, the coverage of secondary structure space in the reference datasets is poorly correlated to the secondary structure components found in the Protein Data Bank. A conclusion is that a new reference CD database with cross-correlated, machine-independent CD spectra and validated X-ray structures that cover more secondary structure components, including diverse protein folds, is now needed. However, that reasonably accurate values for the secondary structure content of proteins can be determined from spectra is a testament to CD spectroscopy being a very powerful technique.

Contact: r.w.janes@qmul.ac.uk

1 INTRODUCTION

Circular dichroism (CD) spectroscopy has become an invaluable research technique, used by many labs worldwide, for gaining information about protein structure, dynamics and interactions both with other proteins and with ligands. This is possible because different types of secondary structures give rise to characteristic CD spectra, which differ in their peak positions and intensities, and to a first approximation a spectrum can be considered to arise from the weighted sum of these components. The information content available from CD is wavelength-range dependent and analyses of spectral data can determine the number of independent eigenvectors needed to reconstruct the original spectrum. For data down to wavelengths of ∼190 nm this number is between three and four (Hennessey and Johnson, 1981). However, because secondary structure components are not independent of each other (Pancoska *et al*., 1992), solutions for a greater number of components than there are independent eigenvectors can be found (Hennessey and Johnson, 1981; Wallace and Janes, 2001).

There are a number of methods that have been developed for deconvoluting CD spectra into the calculated secondary structure components present in the protein. These include as examples, linear least-squares (Chen and Yang, 1971; Brahms and Brahms, 1980), parameterized fit (Provencher and Glöckner, 1981), singular-value decomposition (Hennessey and Johnson, 1981), non-linear least-squares (Wallace and Teeters, 1987) and self-consistent variable selection methods (Sreerama and Woody, 1993; Johnson, 1999; Sreerama and Woody, 2000). These methods are based on very sound mathematical approaches. They have been enhanced and refined over the years, and because of this they can yield reasonable results for the calculation of secondary structure content. Many of these methods are to be found in DICHROWEB (Lobley and Wallace, 2001; Lobley *et al*., 2002; Whitmore and Wallace, 2004), a package designed to aid in the determination of secondary structure content and used world wide. Such is the wide-scale use of CD spectroscopy in research that the creation of the Protein Circular Dichroism Data Bank (PCDDB) has been proposed, which will act as a repository and resource for CD spectra and associated data (Wallace *et al*., 2005).

Synchrotron radiation circular dichroism (SRCD), first developed in 1980 (Sutherland *et al*., 1980), has recently become a potentially

valuable tool for substantially extending the wavelength range of available data due to the increased photon flux of the source over conventional CD (cCD) machines at the lower wavelength limits. For data collected over the full SRCD range, down to ~160 nm, the information content rises to at least seven or eight eigenvectors. These data may be deconvoluted into as many as 12 different secondary, and perhaps supersecondary, structure types thereby enabling a much more detailed resolution of structural features than has been possible from a cCD source (Wallace and Janes, 2001).

Empirical determination of the secondary structure components from CD spectral data employs reference databases. These are either a combination of CD spectra from a set of proteins with known secondary structure content, obtained from their X-ray crystallography structures, or principal component spectra derived from a set of individual spectra. Examples of these databases are Chang *et al*. (1978), Bolotina *et al*. (1980a,b), Brahms and Brahms (1980), Provencher and Glöckner (1981), Compton and Johnson (1986), Pancoska and Keiderling (1991) and Sreerama *et al*. (2000). These reference datasets were created early in the development of CD spectroscopy as a technique for proteins, and significantly more recent databases are for the most part combinations of older ones, and include no or limited new protein secondary structure types and few, if any, new protein constituents. For the major reference databases available, the lowest wavelength data included are to 178 nm. Of note, for SRCD measurements, while a higher number of resolvable secondary structure types should potentially be determinable and with a greater degree of accuracy than is possible from cCD sources, currently no databases are capable of covering down to the full wavelength range available to this technique.

The work presented here analyses the current CD reference databases for the quality of the CD data and X-ray structures used, for their breadth of secondary structure types covered and their effectiveness at covering fold space.

2 METHODS

2.1 CD spectra

CD spectral data were obtained from reference databases at the CDPRO (Sreerama *et al*., 2000) program website (http://lamar.colostate.edu/~sreeram/CDPro/). Additional spectra were obtained from the Brahms and Brahms (1980) reference dataset (provided by Prof. Jon B. Applequist, personal communication) and the Supplementary data (Pancoska and Keiderling, 1995). An SRCD spectrum for γ-crystallin came from Paul Evans and Dr Christine Slingsby. As some original spectra were collected at non-integral wavelengths, an in-house program was used to interpolate these to integral wavelengths for comparison purposes. This did not alter the spectral characteristics in any way however (data not shown). Brahms and Brahms spectra were reported in mean residue ellipticity (θ) values and were therefore scaled to match the delta epsilon ($\Delta\varepsilon$) units used in CDPRO by dividing by 3298.

2.2 X-ray structure data

The X-ray structure data in Tables 1 and 2 are derived from Pancoska and Keiderling (1995), (their original Table 1 set of structures) and *Exp32* reference set from Sreerama *et al*. (2000). Original Protein Data Bank (PDB) (Bernstein *et al*., 1977; Berman *et al*., 2000) files were used for subsequent analyses, even when these had been superseded, as the data within the reference databases are still derived from this original material. Atomic co-ordinates were from PDB files (http://www.pdb.org/) or from the archive site for obsolete structures (http://pdbobs.sdsc.edu/index.cgi).

2.3 X-ray structure analyses

The resolution was obtained from the PDB files. Structural fold information was from the CATH (class, architecture, topology and homologous superfamily) protein topology website (Orengo *et al*., 1997; Pearl *et al*., 2000, 2005). The DSSP (definition of secondary structure of proteins) program (Kabsch and Sander, 1983) was used (http://bioweb.pasteur.fr/seqanal/interfaces/dssp-simple.html) to assign secondary structure content to these proteins. Percentages of secondary structure, derived from the DSSP output, were determined using an in-house program as were the percentages and numbers of 'missing' (undetermined) residues in these structures. These missing residues were checked against the sequence data of the native protein in each case (http://us.expasy.org/srs5/). PROCHECK (Laskowski *et al*., 1993) was used to derive the percentages of residues in fully, additionally and generously allowed and disallowed regions of the Ramachandran plot.

2.4 Evaluating the correlation coefficient of alpha helix and beta sheet content of all PDB proteins against *Exp32*

The values for percentage alpha helical against beta sheet content of all proteins in the PDB were binned into 'ten percentile tranches' (0–9.99, 10–19.99, etc.), not including any nucleic acid material but leaving in all homologue proteins. The reasoning was that any of these proteins could have their CD spectra recorded and they were therefore eligible for inclusion. The *Exp32* set for proteins were binned in a similar way to enable a direct comparison. To quantify the coverage of 'secondary structure space' of the *Exp32* set compared with the whole PDB, the standard Pearson r^2 correlation coefficient was calculated. To create an idealized set of data maximizing the coverage of these secondary structure components for a reference set containing only 32 proteins, each bin of PDB data was reduced by an overall scaling term such that the total of proteins then approximated to 32. Rounding these new values to the nearest integer, and rounding one value of >0.49 manually to one, created the desired idealized 32 protein set.

2.5 Evaluating the correlation coefficient of fold space

In a manner similar to that for the secondary structure components, the fold space of single-domain proteins was obtained from the CATH database for all proteins in the PDB. These data were correlated with those from the *Exp32* set of proteins and a hypothetical set of proteins was also generated to characterize the quality of fold space coverage (in fact comprising 31, as there was one protein unclassified in the *Exp32* set).

3 RESULTS AND DISCUSSION

3.1 Quality of CD spectra in reference databases

The CD spectra used in many of the current reference databases are derived from amalgamations of previously created datasets from different groups, with the aim of broadening the secondary structure types represented by the proteins in these sets. As examples, *Exp32* used as a stand-alone set in SELCON3 (Sreerema and Woody, 1993; Sreerama *et al*., 1999) and as part of many sets in CDPRO (Sreerama *et al*., 2000, http://lamar.colostate.edu/~sreeram/CDPro/), and analysed here, contains 29 CD spectra from Johnson (a gift as stated in Sreerama *et al*., 1999) and 3 from Sreerama *et al*. (1999). The set CDDATA56, currently the largest used in CDPRO containing 56 CD spectra, is a combination from Johnson (Sreerama

Table 1. Set of reference proteins used by Pankoska *et al*., 1995

PDB Code	Res Å	CATH Code	%H	%E	%T	%G	%B	%S	%O	%M	%F	%A	%Ge	%D
4adh[a]	2.40	3.90.180.10 3.40.50.720[b]	21.12	20.58	14.70	3.74	2.13	11.49	26.20	0	78.3**	17.2	1.9	2.5*
1ca2	2.00	3.10.200.10	8.20	28.90	12.50	8.20	1.17	14.06	26.95	1.15	88.4*	11.6	0	0
2cga	1.80	2.40.10.10	7.34	32.04	14.89	6.12	3.06	9.38	27.14	0	85.3*	12.3	0.9	1.4*
5cha	1.67	2.40.10.10	9.32	32.62	12.50	2.54	2.75	11.65	28.60	2.07	84.8*	15	0.2	0
3can	2.40	2.60.120.200	0.00	40.50	9.28	0.00	0.00	19.83	30.37	0	69.2**	24	3.8	2.9*
1cyt[a]	2.00	1.10.760.10	43.20	0.00	13.10	0.00	1.94	9.22	32.52	0	76.2**	19.2	4.1	0.6*
3est	1.65	2.40.10.10	5.41	34.16	17.08	5.41	3.33	7.08	27.50	0	87.9*	12.1	0	0
2grs[a]	2.00	3.50.50.60(2) 3.30.390.30[c]	27.11	18.65	10.41	2.16	1.95	17.35	22.34	3.55	72.9**	20.3	4.3	2.5*
1hco	2.70	1.10.490.10	52.96	0.00	18.81	9.75	0.00	6.62	11.84	0	77.1**	20.9	1.6	0.4*
1rei	2.00	2.60.40.10	0.00	49.06	14.01	2.80	0.46	10.74	22.89	0	87.9*	9.9	0	2.2*
4ldh[a]	2.00	3.40.50.720 3.90.110.10[b]	33.73	11.24	14.28	3.03	2.43	10.63	24.62	0	65.2**	23.5	6.1	4.8*
7lyz	2.50	1.10.530.10	30.23	7.75	20.93	9.30	3.10	12.40	16.27	0	80.5*	17.7	1.8	0
1mbn	2.00	1.10.490.10	77.12	0.00	9.80	0.00	0.00	1.96	11.11	0	83.9*	15.3	0.7	0
8pap[a]	2.80	3.90.70.10	23.11	16.50	8.49	1.41	1.88	18.39	30.18	0	75.6**	23.3	0.6	0.6*
1rhd	2.50	3.40.250.10	27.64	10.92	16.38	2.04	2.38	10.92	29.69	0	77.1**	20.1	2	0.8*
1rn3[a]	1.45	3.10.130.10	17.74	38.70	11.29	3.22	2.41	10.48	16.12	0	81.7*	18.3	0	0
1rns[a]	2.00	3.10.130.10	17.74	39.51	7.25	3.22	2.41	10.48	19.35	13.88	67.3**	23.9	7.1	1.8*
1sbt	2.50	3.40.50.200	30.18	17.81	15.27	0.00	1.81	10.18	24.72	0	74.3**	21.7	3.5	0.4*
2sod	2.00	2.60.40.200	0.66	36.75	12.08	1.15	2.31	20.86	26.15	0	66.5**	24.4	4.4	4.7*
2tln[a]	2.30	3.10.170.10 1.10.390.10[b]	29.74	15.18	14.55	3.79	0.94	15.50	20.25	0	77.0**	18.1	4.4	0.4*
1tim	2.50	3.20.20.90	43.72	16.80	6.07	2.22	0.80	9.31	21.05	0	77.3**	17.8	3.6	1.4*
3pti[a]	1.50	4.10.410.10[d]	13.79	24.13	6.89	6.89	1.72	17.24	29.31	0	87.0*	13	0	0
3ptn	1.70	2.40.10.10	7.17	32.28	14.79	2.69	2.69	15.24	25.11	0	86.2*	13.8	0	0

Some of these are now used for SELCON3 and CDPRO as part of the set containing 56 proteins. The table columns are PDB code, resolution of the structure, CATH code, percentages alpha helix (%H), beta sheet (%E), turn (%T), 3_{10}-helix (%G), bridge (%B), bend (%S) and other (%O) [previously called random coil], as defined by DSSP (Kabsch and Sander, 1983), missing residues (%M), and percentages of residues in the fully (%F), additionally (%A), generously (%Ge) allowed and disallowed (%D) regions of the Ramachandran plot. Note that pi-helix (%I) is not included as no proteins had this secondary structure component. The asterisks are used as flags within PROCHECK to indicate potential problems with the structures. For the fully allowed (%F) region, residue percentages <90% have one asterisk, whilst <80% have two asterisks. In the disallowed (%D) region, residue percentages above zero have one asterisk, and those >5% have two asterisks.

[a]Indicates that these PDB files have now been superseded by superior structure files.

[b]This is a two domain protein.

[c]This is a three domain protein.

[d]Refers to 4pti which supersedes 3pti.

et al., 1999) (29), Sreerama *et al*. (1999) (3), Yang *et al*. (1978) (a gift as stated in Provencher and Glöckner, 1981) (6), Pancoska and Keiderling (1991) (5) and membrane proteins from Park *et al*. (1992) (13). An inherent problem is that the spectra used were often obtained from individual non-commercial CD machines, or machines modified to collect CD data, with, at the time, limited cross-reference calibration. CD spectra for the same protein from the different sets used in these amalgamated databases have different spectral features in a number of cases, as illustrated in Figures 1 and 2. Figure 1 shows the CD spectra of superoxide dismutase from three original databases (Johnson, 1999; Pancoska and Keiderling, 1995 and Brahms and Brahms, 1980). The spectra are considerably different from each other, and yet are from the same source material. In addition, each of these spectra is equated to secondary structure components from the same PDB file (2sod). Only one of these spectra is now used in the current datasets, but it is unclear why one was chosen over the rest and which one is the 'actual' spectrum of the protein? Figure 2a–c show the spectra from Brahms and Brahms (1980) compared with that from Johnson (1999). Here the differences are not so pronounced as in Figure 1, being either wavelength shifts in the spectra, magnitude shifts in the peaks, ratio differences between peaks or a combination of these, but they are nevertheless serious when being used as the basis for empirical calculations of secondary structure content for novel proteins. Figure 2d compares a spectrum of γ-crystallin from a database set with an SRCD spectrum of the same protein (Evans *et al*., 2004). Although the spectra have comparable characteristics there is a 10 nm shift between them, the SRCD spectrum being down-wavelength from that in the database. Whilst CD spectra from single-source reference databases were possibly 'internally consistent' when used as an isolated set, when they became components within an amalgamated set, their differences, as exemplified here, created problems with consistency within these combined sets. To ensure consistency, cross-calibration and cross-checking on a diverse range of machines are of vital importance to remove possible machine bias within the data (Miles *et al*., 2003, 2005).

Table 2. *Exp32*—a set of reference proteins used in SELCON3 and many other amalgamation sets[a]

PDB Code	Res Å	CATH code	%H	%E	%T	%G	%B	%S	%O	%M	%F	%A	%Ge	%D
4mbn	2.00	1.10.490.10	74.50	0.00	5.88	5.88	0.00	1.96	11.76	0	92.0	7.3	0.7	0.0
2mhb	2.00	1.10.490.10	67.24	0.00	9.05	8.71	0.00	3.83	11.14	0	89.9*	8.1	2.0	0.0
2hmz	1.66	1.20.120.50	64.60	0.00	8.62	5.53	0.00	4.20	17.03	0	88.0*	11.1	1.0	0.0
2lzm	1.70	1.10.530.40	66.46	8.53	7.31	0.00	0.60	7.31	9.75	0	95.3	4.7	0.0	0.0
3tim	2.80	3.20.20.90	39.35	15.46	10.84	5.42	1.60	5.42	21.88	0.40	89.1*	10.4	0.5	0.0
6ldh	2.00	3.40.50.720	39.81	16.10	8.51	3.95	0.91	9.72	20.97	0	85.3*	10.9	2.7	1.7*
1lys	1.72	1.10.530.10	31.00	6.20	24.03	10.85	4.65	7.75	15.50	0	90.3	8.8	0.4	0.4*
8tln	1.60	3.10.170.10	37.10	16.35	10.37	4.08	0.62	13.52	17.92	0	87.8*	11.9	0.0	0.4*
5cyt	1.50	1.10.760.10	40.77	0.00	16.50	0.00	1.94	8.73	32.03	0	91.8	8.1	0.0	0.0
3pgk	2.50	3.40.50.1260	34.45	11.08	4.09	0.00	0.48	22.89	26.98	0	59.9**	20.3	10.6	9.2**
1eri	2.70	3.40.580.10	28.35	18.77	15.32	5.36	1.53	9.57	21.07	5.43	86.2*	12.5	0.9	0.4*
1fx1	2.00	3.40.50.360	29.25	21.76	17.00	2.72	1.36	11.56	16.32	0.67	84.8*	12.0	0.8	2.4*
1sbt	2.50	3.40.50.200	30.18	17.81	15.27	0.00	1.81	10.18	24.72	0	74.3**	21.7	3.5	0.4*
3gpd	3.50	—	24.85	20.80	10.62	2.54	1.04	14.97	25.14	0	62.8**	21.5	7.5	8.2**
9pap	1.65	3.90.70.10	23.11	16.98	10.84	2.83	4.24	12.26	29.71	0	88.4*	11.6	0.0	0.0
2sbt	2.80	3.40.50.200	21.45	13.81	17.45	0.00	0.00	15.63	31.63	0	59.6**	30.2	7.6	2.7*
3rn3	1.45	3.10.130.10	17.74	33.06	14.51	3.22	2.41	10.48	18.54	0	87.0*	13.0	0.0	0.0
2psg	1.80	2.40.70.10	10.81	38.64	11.35	9.72	1.08	8.64	19.72	0	90.8	8.6	0.6	0.0
1beb	1.80	2.40.128.20	9.93	42.62	11.21	7.37	0.00	12.17	16.66	3.70	87.0*	10.9	0.4	1.8*
5cha	1.67	2.40.10.10	9.32	32.62	12.50	2.54	2.75	11.65	28.60	1.69	84.8*	15.0	0.2	0.0
1azu	2.70	2.60.40.420	11.29	25.80	13.70	0.00	2.41	20.16	26.61	1.56	60.6**	31.2	5.5	2.8*
3est	1.65	2.40.10.10	5.41	34.16	17.08	5.41	3.33	7.08	27.50	0	87.9*	12.1	0.0	0.0
4gcr	1.47	2.60.20.10	2.87	45.97	8.04	6.32	2.29	12.06	22.41	0	91.4	8.6	0.0	0.0
2pab	1.80	2.60.40.180	7.01	50.00	12.28	0.00	0.43	10.08	20.17	10.24	81.8*	15.7	1.5	1.0*
2ctv	1.95	2.60.120.200	0.00	46.41	11.81	3.79	0.84	14.34	22.78	0	87.5*	12.5	0.0	0.0
1rei	2.00	2.60.40.10	0.00	49.06	14.01	2.80	0.46	10.74	22.89	0.93	87.9*	9.9	0.0	2.2*
1tnf	2.60	2.60.120.40	0.00	44.73	8.11	1.97	1.09	16.88	27.19	3.18	67.2**	25.6	4.7	2.6*
2sod	2.00	2.60.40.200	0.66	36.75	12.08	1.15	2.31	20.86	26.15	0	66.5**	24.4	4.4	4.7*
2abx	2.50	2.10.60.10	0.00	10.81	1.35	0.00	0.00	30.40	57.43	0	14.8**	36.9	23.8	24.6**
1col	2.40	1.10.490.30	75.12	0.00	6.59	3.04	0.00	1.26	13.95	3.43	95.2	4.8	0.0	0.0
1ema	1.90	2.40.155.10	3.55	50.66	15.11	6.22	1.33	8.44	14.66	4.20	93.2	6.8	0.0	0.0
1ifc	1.19	2.40.128.20	11.45	58.77	13.74	0.00	0.00	3.05	12.97	0	93.2	6.8	0.0	0.0

[a]The headings to the columns in Table 2 are as described for Table 1.

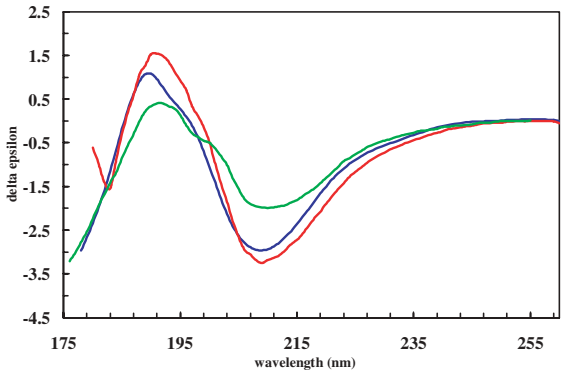

Fig. 1. Superoxide dismutase spectra of the same protein from three CD reference databases Johnson (blue), Pancoska and Keiderling (red) and Brahms and Brahms (green). These spectra are equated to the same secondary structure data derived from PDB file 2sod.

3.2 Wavelength range of CD reference spectra

CD spectra from different reference databases were collected over different wavelength ranges, as illustrated in Figures 1 and 2. The information content is directly proportional to the wavelength range: the shorter the range the less the information available. The data from Johnson (1999) for example have a wavelength range 178–240 nm, while those of the CDDATA56 set, which incorporates the Johnson data, are only over 190–240 nm as other contributing groups collected over a shorter range. This reduction in range represents a significant difference in the available information content, decreasing rather than increasing the number of secondary structure components that can be derived from the data (Wallace and Janes, 2001). In addition, none of the current reference sets covers the range obtainable by SRCD, down to ∼160 nm, and any new reference database would need to address this current shortcoming.

3.3 Quality of X-ray structures in the reference databases

The majority of the X-ray structures used in current reference databases were taken from the limited number available in the PDB (Bernstein *et al*., 1977; Berman *et al*., 2000) in the early 1980s, and some are from earlier. This has an inherent and serious weakness associated with it. Many of these structures were determined long before any systematic refinement protocols, checking and validation programs like PROCHECK (Laskowski *et al*., 1993) were available.

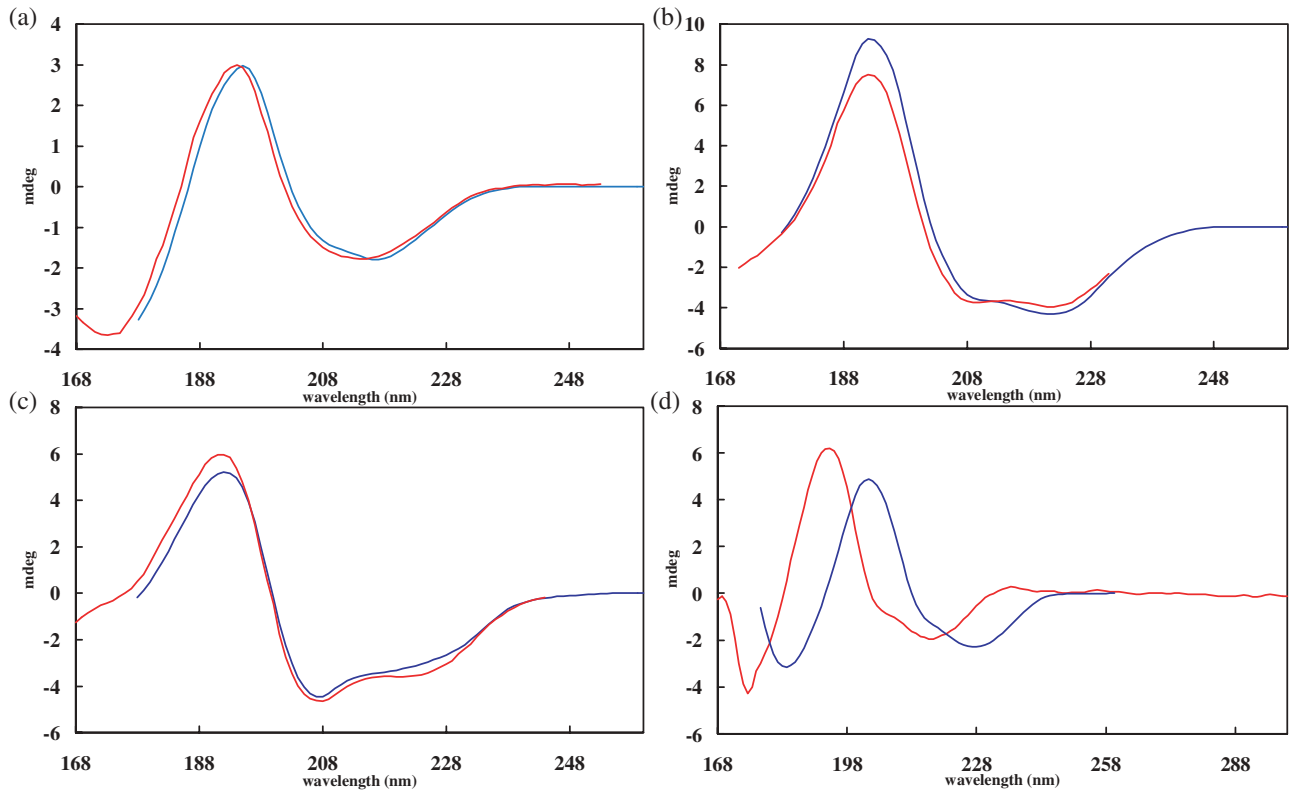

Fig. 2. (**a**)–(**c**) CD spectra of three proteins from current reference database sets in CDPRO (blue) in comparison with the spectra of the same protein from a set now not used (Brahms and Brahms, 1980) (red). The proteins are (a) prealbumin, (b) lactate dehydrogenase and (c) lysozyme. The spectra in (**d**) are of γ-crystallin from the reference database sets used in CDPRO (blue) and SRCD (red) spectral data recorded by Evans *et al*. (2004).

With limited validation, to a lesser or greater extent flaws do exist in a number of these structures which went undetected at that time. However, they are still used either within a stand-alone set or as part of an amalgamation set for determining secondary structure content from a CD spectrum. Data on some of these reference set proteins are presented in Tables 1 and 2 as illustrative examples. The Tables give the PDB code for the given protein at the time of their database inception, CATH database code, resolution of the structure, percentage secondary structure content, percentage residues missing (undetermined) in the structure and percentage residues in the fully, additionally and generously allowed and disallowed regions of the Ramachandran plot, as defined by PROCHECK. Another issue is that it is assumed that protein crystal and solution structures are the same despite the environments being markedly different. Any structural differences that might result from different conditions, e.g. concentrations, salts, pH, etc., could also compound the database inaccuracies in secondary structure determination from CD data.

3.4 Ramachandran plot quality of structures

For the Ramachandran plot, PROCHECK defines a threshold for well-resolved, accurate structures as >90% of their residues being located in the most favoured region, and this is flagged should the value fall below this level. If the value <80% then a double flag is issued to draw attention to potentially more serious problems within the structure. Table 1 is a reference dataset from Pancoska *et al*. (1995), and some members of this set are now used in amalgamation sets in CDPRO. All 23 (100%) proteins of this set are under the 90% threshold for the most favoured region. Additionally, 13 of 23 structures (57%) of these proteins are under the 80% threshold. Of these 5 proteins are now used in amalgamated datasets (2cga, 4adh, 1ca2, 2grs and 1rhd) of which 3 (60%) are below the 80% threshold. In Table 2, the *Exp32* set, 22 of 32 structures (69%) have less than the 90% threshold for residues in the most favoured region. Of these, 8 (25%) are doubly flagged, as being <80%, substantially less than optimal. Failing to reach these thresholds indicates there may be some degree of error in their determined conformations, which means that secondary structure contents derived from them must also be in error. Many of these structures would not be publishable today given the validation procedures now employed. Figures 3 and 4 illustrate some of the problems associated with the structures comprising the reference databases. Here, only 14.8 and 59.9% of residues are located in the most favoured region of the Ramachandran plot. Indeed, in the first case, 24.6% of the residues are in the disallowed region of the plot, indicating a larger number of residues wholly incorrect than correct.

X-ray structures solved at low resolution can potentially have regions where errors arise from an inability to follow accurately the electron density. Some structures with less than optimal resolution are in the reference sets. In Table 2 the five structures with the lowest percentage amino acids in the fully allowed region are all solved at a resolution of 2.5 Å or worse. Clearly, the more incorrect the conformation the more incorrect the percentages of secondary structure types derived, and this questions their reliability for use

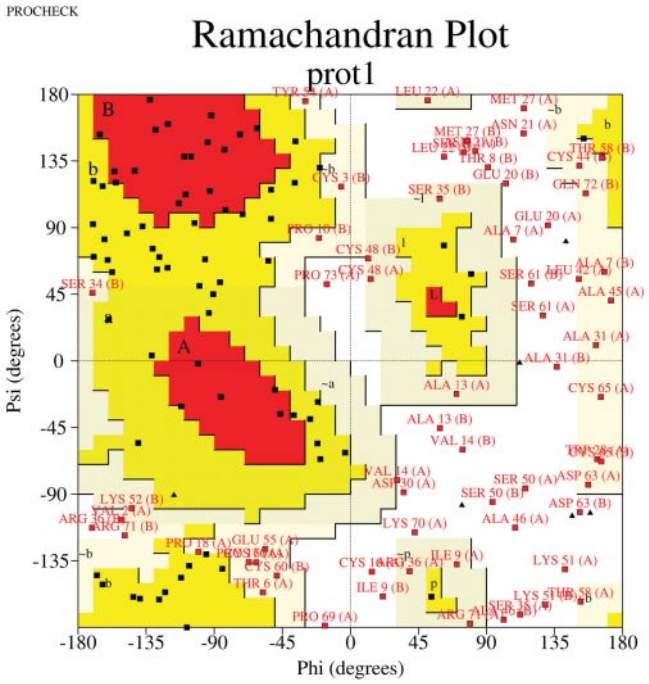

Fig. 3. A Ramachandran plot output, modified from PROCHECK (Laskowski *et al*., 1993) of 'Prot1', a protein used in current reference databases. The areas marked are fully (red), additionally (yellow), generously (fawn) allowed and disallowed (white) regions for amino acid ϕ/φ angles. Shown in red lettering are those residues in the generously allowed and disallowed regions.

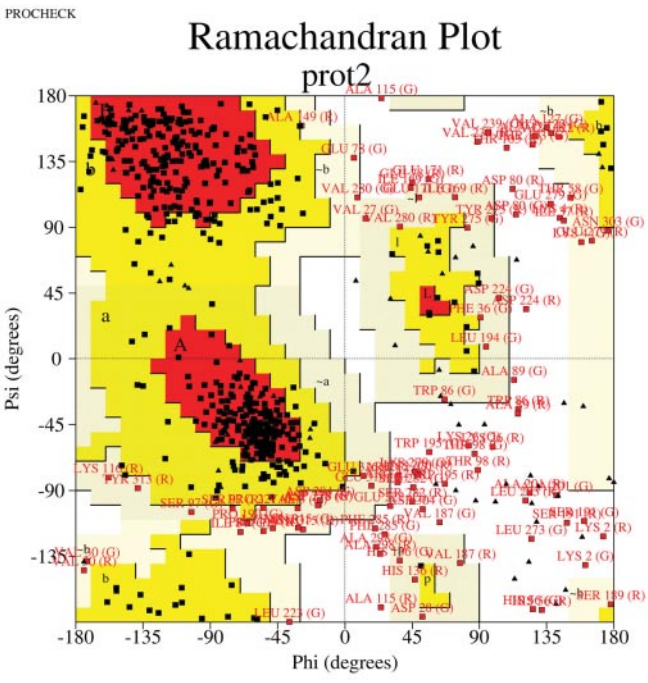

Fig. 4. A Ramachandran plot output (modified from PROCHECK) of 'Prot2', another protein in the current reference databases. Refer Figure 3 for a description of what is depicted.

in the reference datasets. Only a few of the structures have serious mistakes, nevertheless, inclusion of small errors will introduce a degree of inaccuracy, which in turn will lead to erroneous calculation of secondary structure components for empirical determination from a CD spectrum.

3.5 Coverage of secondary structure space

The numbers of X-ray structures used within the reference databases are limited, especially so when compared with those in the PDB. For their optimal utilization it would be important for the sets accurately to reflect the secondary structures present in the PDB. Figure 5 shows as an example, a plot of the coverage of alpha helical against beta sheet content for (a) all protein structures in the PDB, (b) the *Exp32* set and (c) a theoretical idealized set also containing 32 proteins. The *Exp32* reference set does not cover the same secondary structure space as found in the PDB. Correlating the two sets of data, as described in the Methods, gives a value for r^2 of 0.55. This is significantly lower than it could be and is again reflective of the limits imposed in having minimal numbers of protein structures available at the inception of these databases. By comparison, the r^2 term is 0.95 for the idealized set, indicating the coverage of secondary structure space is more extensive, even for such a small dataset.

Increasing the number to that in CDDATA56 (not incorporating the membrane proteins, so this becomes a 43 protein set) gains little in the secondary structure coverage, the r^2 value now becoming 0.62. However, increasing the number of available proteins in the reference set even allows improvement for an idealized set, the r^2 becoming 0.97. Optimizing this correlation with the PDB is one way of ensuring that for a given number of proteins within a reference dataset, they will be as representative as is possible of the whole PDB.

3.6 Coverage of fold space

The maximum number of structures in the current reference databases used is 56, and this includes 13 membrane protein structures together with their related spectra. Whether this is an advisable move is a matter of debate (Wallace *et al*., 2003; Sreerama and Woody, 2004). With SRCD sources the increased information content may enable analysing for the fold of a protein, therefore is it interesting to consider how broad based the coverage of fold space in the current reference databases is? This was not an issue at their inception, where interest lay only in the secondary structure content of the proteins, but it would be an important factor to consider for any new reference databases. Table 2 gives the CATH entry code for the structures in *Exp32*. Figure 6 shows these data in relation to those for all single-domain proteins in the PDB and to a hypothetical set of proteins with the same number as that in *Exp32* (31 here as 1 was unclassified in this set). While some of the more populated CATH classes are well represented in the *Exp32* set, others that should be present to ensure a good coverage of fold classes are clearly lacking. The r^2 correlation coefficient is 0.81 for *Exp32* in relation to the entire PDB. In contrast, the hypothetical set of proteins with the same number of components as in *Exp32* has

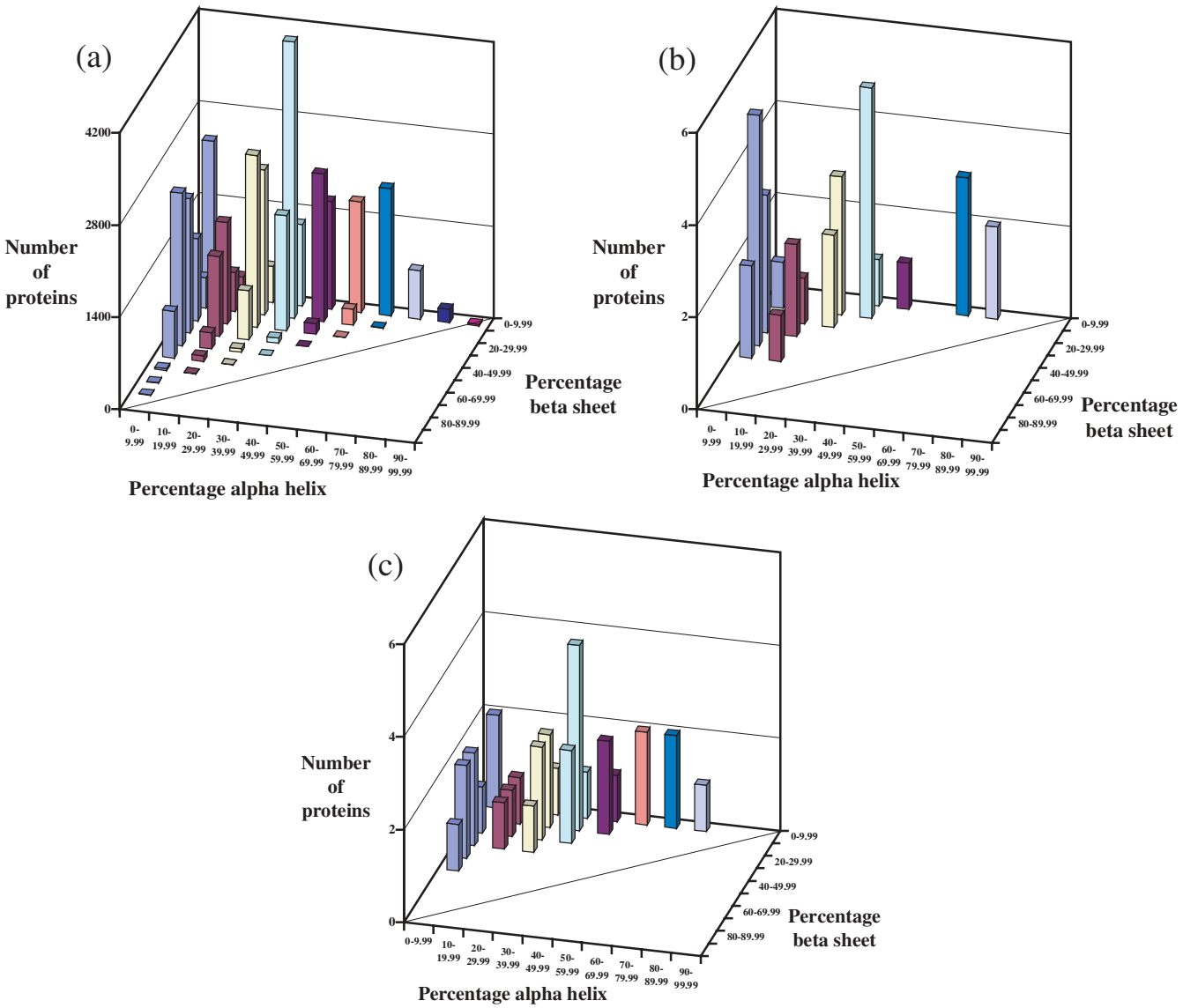

Fig. 5. Plot of alpha helix against beta sheet content for (**a**) all proteins in the PDB compared with (**b**) those proteins found in the CD reference set *Exp32* used in SELCON3, and as part of most other reference databases, and (**c**) for an idealized reference dataset also with 32 proteins, with high correlation to the PDB, as described in the text. Beyond the diagonal base line represents possible regions for alpha/beta content.

an r^2 of 0.97 demonstrating that a better coverage of fold space is possible.

In Table 1, there are four structures present that are multi-domain proteins, containing two or more recognized CATH topologies in their structures, and two of these are used in amalgamated sets in CDPRO. Again, this was not an issue at the time of database inception. However, inclusion of multi-domain proteins into future reference database sets, which might be aimed at analysing for fold recognition (Wallace and Janes, 2001), would lead to difficulties in interpretation of such fold classes and so single-domain proteins would be the optimum to be used in such sets.

3.7 Completeness of structures

The *Exp32* set of structures in Table 2 has 11 (34%) of them that are incomplete, having undetermined regions, maybe from inherent structural flexibility. These missing residues are predominantly lost from the N- and C-termini, and two have >5% of missing structure (5.43% for 1eri and 10.24% for 2pab) representing 15 and 18 residues, respectively. How missing structural content is accounted for in each of the databases is not always clear, especially because definitions pertinent to secondary structure features are also in their infancy and so totals counted for such features are sometimes not the same as those from current calculations. The main method assumes missing residues can be considered as 'other' (previously referred to as 'random coil') and thus adds them to that component's total. This is feasible only if the number of missing residues is insufficient to form any type of secondary structure component. Highly flexible regions with proportionately larger numbers of missing residues might prevent determination of other more structured areas containing secondary structure

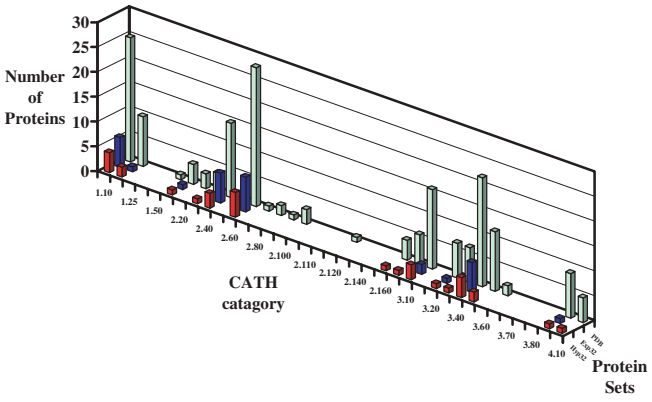

Fig. 6. Population of CATH fold space for single-domain proteins in the PDB (light blue) in comparison with data for the *Exp32* set (blue) and with a hypothetical set of proteins, labelled *Hyp32* (red) with the same number of proteins as that of the *Exp32* set.

components that would be missed as a result. Another method might be to ignore the missing content and to take the known portion of determined structure as being the total content. Whichever of these two methods were to be used, neither is valid as it is making unsupported assumptions in one case and introducing direct errors in the other, and hence both are unsatisfactory. Indeed, there is no satisfactory answer for dealing with missing residues in protein structures being used for a CD reference dataset other than to use only 'complete' proteins, structures whose entire length of chain is resolvable. It should be put into perspective, however, that at the time of inception of the databases many limitations hampered the selection of structures.

4 CONCLUSIONS

Despite the internal errors associated with the CD spectra and X-ray structures found in many reference databases, reasonably accurate values for secondary structure content of proteins can be determined from CD data. This is particularly true for mainly alpha helical proteins, likely due to the lack of variance in the geometry of this secondary structure component in different proteins. Accurate secondary structure determinations for many unknown proteins are also possible because the reference databases contain several of the most popular protein folds. Determination of beta sheet-containing proteins is usually less accurate, due to both the lesser intensity of CD signal from this component relative to that from alpha helices and the greater diversity of topologies of such a component found within proteins. Other less common secondary structural components, such as 3_{10} and PPII helices, also tend to be less accurately determined because of their limited representation within the reference databases.

CD spectroscopy is used to determine the secondary structure content of proteins and many excellent mathematical approaches have been developed for this procedure. All rely on reference databases to obtain accurate values for calculating this content, but the databases themselves must have minimal errors otherwise this accuracy could be compromised. From the analyses presented there are some problems associated with the current CD reference databases, both with the CD spectra and with the X-ray structures used. Some of the CD spectra are potentially erroneous representations of the referenced protein, with restrictions in their wavelength range covered, and many of the X-ray structures are not of the highest quality because of limitations in structure validation procedures. Also the structures are limited in their range of secondary structure types represented and coverage of secondary structure space. These analyses suggest that there is an urgent need to create a new, more comprehensive CD reference database containing cross-validated CD spectra collected and cross-checked on a number of different cCD spectrophotometers and SRCD beamlines to ensure machine independence. These should be for proteins with X-ray structures that have a broad coverage both of different secondary structure types and of secondary structure space, whose quality has been assured by the many available validation programs. Additionally, with the number of SRCD facilities increasing worldwide and improvements in cCD machine optics, any new reference database should extend to lower wavelength limits to enable analyses of these extra data. In summary, the recommendations for the content of a future reference database would be as follows: to contain ~80 proteins; with complete X-ray structures (i.e. those with no missing residues) whose quality has been confirmed by programs such as PROCHECK; a broad range of secondary structure and fold types represented; created with SRCD spectra to achieve low wavelength data (at least 170 nm), matching the CD data to the protein organism/sequence of the X-ray structure; and with full calibration/validation of these CD spectra. Such a database would enhance the quality and accuracy of secondary structure component determination, ensuring that CD spectroscopy remains a very powerful technique.

ACKNOWLEDGEMENTS

I thank Prof. B. A. Wallace for many useful discussions. I thank Prof Jon B. Applequist for the CD data from some of the database sets, and Paul Evans and Dr Christine Slingsby for the SRCD spectral data for γ-crystallin. I also thank Dr Alison Cuff for provision of the single-domain proteins CATH data. This work was supported by a BBSRC grant (B19312).

Conflict of Interest: none declared.

REFERENCES

Berman,H.M. *et al*. (2000) The protein data bank. *Nucleic Acids Res*., **28**, 235–242.

Bernstein,F.C. *et al*. (1977) The protein data bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol*., **112**, 535–542.

Bolotina,I.A. *et al*. (1980a) Determination of the secondary structure of proteins from the circular-dichroism spectra.1. Protein reference spectra for alpha structure, beta structure and irregular structure. *Mol. Biol*., **14**, 701–709.

Bolotina,I.A. *et al*. (1980b) Determination of the secondary structure of proteins from the circular-dichroism spectra. 2. Consideration of the contribution of beta-bends. *Mol. Biol*., **14**, 709–715.

Brahms,S. and Brahms,J. (1980) Determination of protein secondary structure in solution by vacuum ultraviolet circular dichroism. *J. Mol. Biol*., **138**, 149–178.

Chang,T.C. *et al*. (1978) Circular dichroic analysis of protein conformation: inclusion of beta-turns. *Anal. Biochem*., **91**, 13–31.

Chen,Y.H. and Yang,J.T. (1971) A new approach to the calculation of secondary structures of globular proteins by optical rotatory dispersion and circular dichroism. *Biochem. Biophys. Res. Commun*., **44**, 1285–1291.

Compton,L.A. and Johnson,W.C.,Jr (1986) Analysis of protein circular dichroism spectra for secondary structure using a simple matrix multiplication. *Anal. Biochem*., **155**, 155–167.

Evans,P. *et al*. (2004) The P23T cataract mutation causes loss of solubility of folded gammaD-crystallin. *J. Mol. Biol*., **343**, 435–444.

Hennessey,J.P.,Jr and Johnson,W.C.,Jr (1981) Information content in the circular dichroism of proteins. *Biochemistry*, **20**, 1085–1094.

Johnson,W.C. (1999) Analyzing protein circular dichroism spectra for accurate secondary structures. *Proteins*, **35**, 307–312.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Laskowski,R.A. *et al*. (1993) PROCHECK—a program to check the stereochemical quality of protein structures. *J. Appl. Cryst*., **26**, 283–291.

Lobley,A. and Wallace,B.A. (2001) DICHROWEB: a website for the analysis of protein secondary structure from circular dichroism spectra. *Biophysical J*., **80**, 373a.

Lobley,A. *et al*. (2002) DICHROWEB: an interactive website for the analysis of protein secondary structure from circular dichroism spectra. *Bioinformatics*, **18**, 211–212.

Miles,A.J. *et al*. (2003) Calibration and standardisation of synchrotron radiation circular dichroism and conventional circular dichroism spectrophotometers. *Spectroscopy*, **17**, 653–661.

Miles,A.J. *et al*. (2005) Calibration and standardisation of synchrotron radiation and conventional circular dichroism spectrometers. Part 2: Factors affecting magnitude and wavelength. *Spectroscopy*, **19**, 43–51.

Orengo,C.A. *et al*. (1997) CATH—a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1108.

Pancoska,P and Keiderling,T.A. (1991) Systematic comparison of statistical-analyses of electronic and vibrational circular dichroism for secondary structure prediction of selected proteins. *Biochemistry*, **30**, 6885–6895.

Pancoska,P. *et al*. (1992) Relationships between secondary structure fractions for globular proteins. Neural network analyses of crystallographic datasets. *Biochemistry*, **31**, 10250–10257.

Pancoska,P. *et al*. (1995) Comparison of and limits of accuracy for statistical analyses of vibrational and electronic circular dichroism spectra in terms of correlations to and predictions of protein secondary structure. *Protein Sci*., **4**, 1384–1401.

Pearl,F.M. *et al*. (2000) Assigning genomic sequences to CATH. *Nucleic Acids Res*., **28**, 277–282.

Pearl,F. *et al*. (2005) The CATH domain structure database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic Acids Res*., **33**, D247–D251.

Provencher,S.W. and Glöckner,J. (1981) Estimation of globular protein secondary structure from circular dichroism. *Biochemistry*, **20**, 33–37.

Sreerema,N. and Woody,R.W. (1993) A self-consistent method for the analysis of protein secondary structure from circular dichroism. *Anal. Biochem*., **209**, 32–44.

Sreerama,N. and Woody,R.W. (2000) Estimation of protein secondary structure from circular dichroism spectra: comparison of CONTIN, SELCON and CDSSTR methods with an expanded reference set. *Anal. Biochem*., **287**, 252–260.

Sreerema,N. *et al*. (1999) Estimation of the number of helical and strand segments in proteins using CD spectroscopy. *Protein Sci*., **8**, 370–380.

Sreerama,N. *et al*. (2000) Estimation of protein secondary structure from circular dichroism spectra: inclusion of denatured proteins with native proteins in the analysis. *Anal. Biochem*., **287**, 243–251.

Sutherland,J.C. *et al*. (1980) Versatile spectrometer for experiments using synchrotron radiation at wavelengths greater than 100 nm. *Nucl. Instrum. Methods*, **172**, 195–199.

Wallace,B.A. and Janes,R.W. (2001) Synchrotron radiation circular dichroism spectroscopy of proteins: secondary structure, fold recognition and structural genomics. *Curr. Opin. Chem. Biol*., **5**, 567–571.

Wallace,B.A. and Teeters,C.L. (1987) Differential absorption flattening optical effects are significant in the circular-dichroism spectra of large membrane-fragments. *Biochemistry*, **26**, 65–70.

Wallace,B.A. *et al*. (2005) *Proteins*, in press.

Whitmore,L. and Wallace,B.A. (2004) DICHROWEB: an online server for protein secondary structure analyses from circular dichroism spectroscopic data. *Nucleic Acids Res*., **32**, W668–W673.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4239–4247
doi:10.1093/bioinformatics/bti687

Structural bioinformatics

Profile-based direct kernels for remote homology detection and fold recognition

Huzefa Rangwala and George Karypis*

Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA

Received on May 12, 2005; revised on June 15, 2005; accepted on September 20, 2005
Advance Access publication September 27, 2005

ABSTRACT

Motivation: Protein remote homology detection is a central problem in computational biology. Supervised learning algorithms based on support vector machines are currently one of the most effective methods for remote homology detection. The performance of these methods depends on how the protein sequences are modeled and on the method used to compute the kernel function between them.

Results: We introduce two classes of kernel functions that are constructed by combining sequence profiles with new and existing approaches for determining the similarity between pairs of protein sequences. These kernels are constructed directly from these explicit protein similarity measures and employ effective profile-to-profile scoring schemes for measuring the similarity between pairs of proteins. Experiments with remote homology detection and fold recognition problems show that these kernels are capable of producing results that are substantially better than those produced by all of the existing state-of-the-art SVM-based methods. In addition, the experiments show that these kernels, even when used in the absence of profiles, produce results that are better than those produced by existing non-profile-based schemes.

Availability: The programs for computing the various kernel functions are available on request from the authors.

Contact: karypis@cs.umn.edu

1 INTRODUCTION

Breakthroughs in large-scale sequencing have led to a surge in the available protein sequence information that has far out-stripped our ability to experimentally characterize their functions. As a result, researchers are increasingly relying on computational techniques to classify these sequences into functional and structural families based on sequence homology.

Although satisfactory methods exist to detect homologs with high levels of similarity, accurately detecting homologs at low levels of sequence similarity (remote homology detection) still remains a challenging problem. Some of the most popular approaches for remote homology prediction compare a protein with a collection of related proteins using methods such as protein family profiles (Gribskov *et al*., 1987), PSI-BLAST (Altschul *et al*., 1997), and hidden Markov models (HMMs) (Krogh *et al*., 1994; Baldi *et al*., 1994; Karplus *et al*., 1998). These schemes produce models that are generative in the sense that they build a model for a set of related proteins and then check to see how well this model explains a candidate protein.

In recent years, the performance of remote homology detection has been further improved through the use of methods that explicitly model the differences between the various protein families (classes) and build discriminative models. In particular, a number of different methods have been developed that build these discriminative models using support vector machines (SVM) (Vapnik, 1998) and have shown, provided there are sufficient data for training, to produce results that are in general superior to those produced by either pairwise sequence comparisons or approaches based on generative models (Jaakkola *et al*., 2000; Liao and Noble, 2002; Leslie *et al*., 2002, 2003; Ben-Hur and Brutlag, 2003; Hou *et al*., 2003, 2004; Saigo *et al*., 2004; Kuang *et al*., 2005).

A core component of an SVM is the kernel function, which measures the similarity between any pair of examples. Different kernels correspond to different notions of similarity and can lead to discriminative functions with different performance. One approach for deriving a kernel function is to first choose an appropriate feature space (potentially derived from the input space directly), represent each sequence as a vector in that space and then take the inner product (or a function derived from them) between these vector-space representations as a kernel for the sequences.

One of the early attempts with such feature-space-based approaches is the SVM-Fisher method (Jaakkola *et al*., 2000), in which a profile HMM model is estimated on a set of proteins belonging to the positive class and used to extract a vector representation for each protein. Another approach is the SVM-pairwise scheme (Liao and Noble, 2002), which represents each sequence as a vector of pairwise similarities to all sequences in the training set. A relatively simpler feature space that contains all possible short subsequences ranging from 3 to 8 amino acids (*k*mers) is explored in a series of papers [Spectrum kernel (Leslie *et al*., 2002), Mismatch kernel (Leslie *et al*., 2003), and profile kernel (Kuang *et al*., 2005)]. All three of these methods represent a sequence X as a vector in this feature space and differ on the scheme they employ to actually determine if a particular dimension u (i.e. *k*mer) is present

*To whom correspondence should be addressed.

(i.e. has a non-zero weight) in X's vector or not. The Spectrum kernel considers u to be present if X contains u as a substring, the Mismatch kernel considers u to be present if X contains a substring that differs with u in at most a predefined number of positions (i.e. mismatches) and the profile kernel considers u to be present if X contains a substring whose PSSM-based ungapped alignment score with u is above a user-supplied threshold. An entirely different feature space is explored by the SVM-Isites (Hou *et al*., 2003) and SVM-HMMSTR (Hou *et al*., 2004) methods that take advantage of a set of local structural motifs (SVM–Isites) and their relationships (SVM-HMMSTR).

An alternative to measuring pairwise similarity through a dot-product of vector representations is to calculate an explicit protein similarity measure. The recently developed LA-Kernel method (Saigo *et al*., 2004) represents one such example of a direct kernel function. This scheme measures the similarity between a pair of protein sequences by taking into account all the optimal local alignment scores with gaps between all of their possible subsequences. The experiments presented in Saigo *et al*. (2004) show that this kernel is superior to previously developed schemes that do not take into account sequence profiles and that the overall classification performance improves by taking into account all local alignments.

In this paper we develop new kernel functions that are derived directly from explicit similarity measures and utilize sequence profiles. We present two classes of such kernel functions. The first class, referred to as window based, determines the similarity between a pair of sequences by using different schemes to combine ungapped alignment scores of certain fixed-length subsequences. The second, referred to as local alignment based, determines the similarity between a pair of sequences using Smith–Waterman alignments and a position independent affine gap model, optimized for the characteristics of the scoring system. Both kernel classes utilize profiles constructed automatically via PSI-BLAST and employ a profile-to-profile scoring scheme we develop by extending a recently introduced profile alignment method (Mittelman *et al*., 2003).

Experiments on two benchmarks derived from SCOP, one designed to detect remote homologs and the other designed to identify folds, show that these new kernels produce results that are substantially better than those produced by all other state-of-the-art SVM-based methods. In addition, the experiments show that these newly proposed kernels, even when used in the absence of profiles, produce results that are better than those produced by existing non-profile-based schemes.

2 METHODS AND ALGORITHMS

2.1 SVM and kernel functions

Key to our algorithm for protein classification is its learning methodology, which is based on support vector machines. Given a set of positive training sequences $\mathcal{S}^+$ and a set of negative training sequences $\mathcal{S}^-$, an SVM learns a classification function $f(X)$ of the form

$$f(X) = \sum_{X_i \in \mathcal{S}^+} \lambda_i^+ \mathcal{K}(X, X_i) - \sum_{X_i \in \mathcal{S}^-} \lambda_i^- (X, X_i), \tag{1}$$

where λ_i^+ and λ_i^- are non-negative weights that are computed during training by maximizing a quadratic objective function, and $\mathcal{K}(.,.)$ is called the kernel function that is computed over the various training set and test set instances. Given this function, a new sequence X is predicted to be positive or negative depending on whether $f(X)$ is positive or negative. In addition, the value of $f(X)$ can be used to obtain a meaningful ranking of a set of instances, as it represents the strength by which they are members of the positive or negative class.

2.2 Sequence profiles

The inputs to our classification algorithm are the various proteins and their profiles. A protein sequence X of length n is represented by a sequence of characters $X = \langle a_1, a_2, \ldots, a_n \rangle$ such that each character corresponds to 1 of the 20 standard amino acids. The profile of a protein X is derived by computing a multiple sequence alignment of X with a set of sequences $\{Y_1, \ldots, Y_m\}$ that have a statistically significant sequence similarity with X (i.e. they are sequence homologs). In this paper we obtain the profiles using PSI-BLAST (Altschul *et al*., 1997) as it combines the steps of finding the homologous sequences and computing their multiple alignment, is very fast, and has been shown to produce reasonably good results. However, the profile-based kernels developed here can be used with other methods of constructing sequence profiles as well.

The profile of a sequence X of length n is represented by two $n \times 20$ matrices. The first is its position-specific scoring matrix PSSM_X that is computed directly by PSI-BLAST using the scheme described in (Altschul *et al*., 1997). The rows of this matrix correspond to the various positions in X and the columns correspond to the 20 distinct amino acids. The second matrix is its position-specific frequency matrix PSFM_X that contains the frequencies used by PSI-BLAST to derive PSSM_X. These frequencies (also referred to as target frequencies (Mittelman *et al*., 2003)) contain both the sequence-weighted observed frequencies [also referred to as effective frequencies (Mittelman *et al*., 2003)] and the BLOSUM62 (Henikoff and Henikoff, 1992) derived-pseudocounts (Altschul *et al*., 1997). For each row, the frequencies were scaled so that they add up to one. In the cases in which PSI-BLAST could not produce meaningful alignments for certain positions of X, the corresponding rows of the two matrices were derived from the scores and frequencies of BLOSUM62.

2.3 Profile-based sequence similarity

Many different schemes have been developed for determining the similarity between profiles that combine information from the original sequence, position-specific scoring matrix, or position-specific target and/or effective frequencies (Mittelman *et al*., 2003; Wang and Dunbrack Jr, 2004; Marti-Renom *et al*., 2004). In our work we use a scheme that is derived from PICASSO (Heger and Holm, 2001; Mittelman *et al*., 2003). Specifically, the similarity score between the i-th position of protein's X profile, and the j-th position of protein's Y profile is given by

$$S_{X,Y}(i, j) = \sum_{k=1}^{20} \text{PSFM}_X(i, k)\, \text{PSSM}_Y(j, k) + \sum_{k=1}^{20} \text{PSFM}_Y(j, k)\, \text{PSSM}_X(i, k), \tag{2}$$

where $\text{PSFM}_X(i,k)$ and $\text{PSSM}_X(i,k)$ are the values corresponding to the k-th amino acid at the i-th position of X's position-specific score and frequency matrices. $\text{PSFM}_Y(j,k)$ and $\text{PSSM}_Y(j,k)$ are defined in a similar fashion.

Equation (2) determines the similarity between two profile positions by weighting the position-specific scores of one sequence according to the frequency at which the corresponding amino acid occurs in the second sequence's profile. Note that by construction, Equation (2) leads to a symmetric similarity score. The key difference between Equation (2) and the corresponding scheme used in Mittelman *et al*. (2003) (referred to as PICASSO3), is that our measure uses the target frequencies, whereas the scheme of (Mittelman *et al*., 2003) was based on effective frequencies. Our experiments (not included here) indicate that target frequencies lead to better results.

2.4 Window-based kernels

The first class of profile-based kernel functions that we developed determines the similarity between a pair of sequences by combining the ungapped alignment scores of certain fixed length subsequences (referred to as $w$mers). Given a sequence X of length n and a user-supplied parameter w, the $w$mer at position i of X ($w < i \leq n - w$) is defined to be the $(2w + 1)$-length subsequence of X centered at position i. That is, the $w$mer contains x_i, the w amino acids before, and the w amino acids after x_i. We will denote this subsequence as $w\text{mer}_X(i)$.

2.4.1 All fixed-width wmers (AF-PSSM) The AF-PSSM kernel computes the similarity between a pair of sequences X and Y by adding up the alignment scores of all possible $w$mers between X and Y that have a positive ungapped alignment score. Specifically, if the ungapped alignment score between two $w$mers at positions i and j of X and Y, respectively is denoted by $w\text{score}_{X,Y}(i,j)$, n and m are the lengths of X and Y, respectively and and $\mathcal{P}_w$ is the set of all possible $w$mer-pairs of X and Y with a positive ungapped alignment score, i.e.

$$\mathcal{P}_w = \{(w\text{mer}_X(i), w\text{mer}_Y(j)) \,|\, w\text{score}_{X,Y}(i,j) > 0\}, \tag{3}$$

for $w + 1 \leq i \leq n - w$ and $w + 1 \leq j \leq m - w$, then the AF-PSSM kernel computes the similarity between X and Y as

$$\text{AF-PSSM}_{X,Y}(w) = \sum_{(w\text{mer}_X(i),\, w\text{mer}_Y(j)) \in \mathcal{P}_w} w\text{score}_{X,Y}(i,j). \tag{4}$$

The ungapped alignment score between two $w$mers is computed using the profile-to-profile scoring method of Equation (2) as follows:

$$w\text{score}_{X,Y}(i,j) = \sum_{k=-w}^{w} S_{X,Y}(i+k, j+k). \tag{5}$$

Note that both the AF-PSSM kernel and the profile kernel (Kuang *et al*., 2005) determine the similarity between a pair of sequences by considering how all of their fixed-length subsequences are related in view of sequence profiles. However, unlike the feature space-based approach employed by Profile, the AF-PSSM kernels determine the $w$mer-based similarity of two sequences by comparing all of their possible $w$mers directly. This allows AF-PSSM to precisely determine whether two $w$mers are similar or not and provide better quantitative estimates of the degree to which two $w$mers are similar.

2.4.2 Best fixed-width wmer (BF-PSSM) In determining the similarity between a pair of sequences X and Y, the AF-PSSM kernel includes information about all possible $w$mer-level local alignments between them. In light of this observation, it can be thought of as a special case of the LA kernels proposed by Saigo *et al*. (2004), which compute the similarity between a pair of sequences as the sum of the optimal local alignment scores with gaps between all possible subsequences of X and Y. The results reported in Saigo *et al*. (2004) show that taking into account all possible alignments leads to better results.

To see whether or not this is true in the context of the profile-derived $w$mer-based kernels, we developed a scheme that attempts to eliminate this multiplicity by computing the similarity between a pair of sequences based on a subset of the $w$mers used in the AF-PSSM kernel. Specifically, the BF-PSSM kernel selects a subset $\mathcal{P}'_w$ of $\mathcal{P}_w$ [as defined in Equation (3)] such that (1) each position of X and each position of Y is present in at most one $w$mer-pair and (2) the sum of the $w$scores of the selected pairs is maximized. Given $\mathcal{P}'_w$, the similarity between the pair of sequences is then computed as follows:

$$\text{BF-PSSM}_{X,Y}(w) = \sum_{(w\text{mer}(X,i),\, w\text{mer}(Y,j)) \in \mathcal{P}'_w} w\text{score}_{X,Y}(i,j). \tag{6}$$

The relation between $\mathcal{P}'_w$ and $\mathcal{P}_w$ can be better understood if the possible $w$mer-pairs in $\mathcal{P}_w$ are viewed as forming an $n \times m$ matrix, whose rows correspond to the positions of X, columns to the positions of Y, and values correspond to their respective $w$scores . Within this context, $\mathcal{P}'_w$ corresponds to a matching of the rows and columns (Papadimitriou and Steiglitz, 1982) whose weight is high (bipartite graph matching problem). Since the selection forms a matching, each position of X (or Y) contributes at most one $w$mer in Equation (6), and as such, eliminates the multiplicity present in the AF-PSSM kernel. At the same time, since we are interested in a highly weighted matching, we try to select the best $w$mers for each position.

In our algorithm, we use a greedy algorithm to incrementally construct $\mathcal{P}'_w$ by including the highest weight $w$mers that is not in conflict with the $w$mers already in $\mathcal{P}'_w$. This algorithm terminates when we cannot include in $\mathcal{P}'_w$ any additional $w$mers.

Note that an alternate way of defining $\mathcal{P}'_w$ is to actually look for the maximum weight matching (i.e. the matching whose weight is the highest among all possible matchings). However, the complexity of the underlying bipartite maximum weight matching problem is relatively high [$O(n^2m + nm^2)$ (Papadimitriou and Steiglitz, 1982)], and for this reason we use the greedy approach.

2.4.3 Best variable-width wmer (BV-PSSM) In fixed-width $w$mer-based kernels the width of the $w$mers is fixed for all pairs of sequences and throughout the entire sequence. As a result, if w is set to a relatively high value, it may fail to identify positive scoring subsequences whose length is smaller than $2w + 1$, whereas if it is set too low, it may fail to reward sequence pairs that have relative long similar subsequences.

To overcome this problem, we developed a kernel, referred to as BV-PSSM, which is derived from the BF-PSSM kernel but operates with variable width $w$mers. In particular, given a user-supplied width w, it considers the set of all possible $w$mer-pairs whose length ranges from one to w, i.e.

$$\mathcal{P}_{1\ldots w} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_w, \tag{7}$$

and among them, it uses the greedy scheme employed by BF-PSSM to select a subset $\mathcal{P}'_{1\ldots w}$ of $w$mer-pairs that form a high weight matching. The similarity between the pair of sequences is then computed as follows:

$$\text{BV-PSSM}_{X,Y}(w) = \sum_{(w\text{mer}(X,i),\, w\text{mer}(Y,j)) \in \mathcal{P}'_{1\ldots w}} w\text{score}_{X,Y}(i,j). \tag{8}$$

Since for each position of X (and Y), $\mathcal{P}'_{1\ldots w}$ is constructed by including the highest scoring $w$mer for i that does not conflict with the previous selections, this scheme can automatically select the highest scoring $w$mer whose length can vary from one up to w; thus, achieving the desired effect.

2.5 Local alignment-based kernels (SW-PSSM)

The second class of profile-based kernels that we examine compute the similarity between a pair of sequences X and Y by finding an optimal alignment between them that optimizes a particular scoring function. There are three general classes of optimal alignment-based schemes that are commonly used to compare protein sequences. These are based on global, local and global–local (also known as end-space free) alignments (Gusfield, 1997). Our experiments with all of these schemes indicate that those based on optimal local alignments [also referred to as Smith–Waterman alignments (Smith and Waterman, 1981)] tend to produce somewhat better results. For this reason we use this method to derive a profile-based alignment kernel, which is referred to as SW-PSSM.

Given two sequences X and Y of lengths n and m, respectively, the SW-PSSM kernel computes their similarity as the score of the optimal local alignment in which the similarity between two sequence positions is determined using the profile-to-profile scoring scheme of Equation (2), and a position-independent affine gap model. The actual alignment is computed using the $O(nm)$ dynamic programming algorithm developed by Gotoh (1982).

Within this local alignment framework, the similarity score between a pair of sequences depends on the particular values of the affine gap model [i.e. gap-opening (go) and gap-extension (ge) costs] and the intrinsic

characteristics of the profile-to-profile scoring scheme. In order to obtain meaningful local alignments, the scoring scheme that is used should produce alignments whose score must on average be negative with the maximum score being positive (Smith and Waterman, 1981). A scoring system whose average score is positive will tend to produce very long alignments, potentially covering segments of low biologically relevant similarity. However, if the scoring system cannot easily produce alignments with positive scores, it may fail to identify any non-empty similar subsequences.

To ensure that the SW-PSSM kernel can correctly account for the characteristics of the scoring system, we modify the profile-to-profile scores calculated from Equation (2) by adding a constant value. This scheme, commonly referred to as zero-shifting (Wang and Dunbrack Jr, 2004), ensures that the resulting alignments have scores that on the average are negative while allowing for positive maximum scores. In our scheme, the amount of zero-shifting, denoted by *zs*, is kept fixed for all pairs of sequences, as a limited number of experiments with sequence pair-specific *zs* values did not produce any better results.

2.6 From similarity measures to Mercer kernels

Any function $\mathcal{K}(\cdot,\cdot)$ can be used as a kernel as long as for any number n and any possible set of distinct sequences $\{X_1,\ldots,X_n\}$, the $n \times n$ Gram matrix defined by $G_{i,j} = \mathcal{K}(X_i, X_j)$ is symmetric positive semidefinite. These functions are said to satisfy Mercer's conditions and are called Mercer kernels, or simply valid kernels.

The similarity based functions described in the previous sections can be used as kernel functions by setting $\mathcal{K}(X_i, X_j)$ to be equal to one of AF-PSSM$_{X_i,X_j}$, BF-PSSM$_{X_i,X_j}$ BV-PSSM$_{X_i,Xj}$ or SW-PSSM$_{X_i,Xj}$. However, the resulting functions will not necessarily lead to valid Mercer kernels, because G may not be positive semidefinite.

To overcome this problem we used the approach described in Saigo *et al*. (2004) to convert a symmetric matrix defined on the training set instances into positive definite by subtracting from the diagonal of the training Gram matrix its smallest negative eigenvalue. The resulting matrix is identical to the similarity based Gram matrix at all positions expect those along the main diagonal. We also experimented with the empirical kernel map approach proposed in Scholkopf and Smola (2002), but we find that the eigenvalue-based scheme produced superior results.

3 EXPERIMENTAL DESIGN

3.1 Dataset description

We evaluated the classification performance of the profile-based kernels on a set of protein sequences obtained from the SCOP database (Murzin *et al*., 1995). We formulated two different classification problems. The first was designed to evaluate the performance of the algorithms for the problem of homology detection when the sequences have low sequence similarities (i.e. the remote homology detection problem), whereas the second was designed to evaluate the extent to which the profile-based kernels can be used to identify the correct fold when there are no apparent sequence similarities (i.e. the fold detection problem).

3.1.1 Remote homology detection (superfamily detection) Within the context of the SCOP database, remote homology detection was simulated by formulating it as a superfamily classification problem. The same dataset and classification problems (The dataset and classification problem definitions are available at http://www.cs.columbia.edu/compbio/svm-pairwise) have been used in a number of earlier studies (Liao and Noble, 2002; Hou *et al*., 2004; Saigo *et al*., 2004) allowing us to perform direct comparisons on the relative performance of the various schemes. The data consisted of 4352 sequences from SCOP version 1.53 extracted from the Astral database, grouped into families and superfamilies. The dataset was processed so that it does not contain any sequence pairs with an *E*-value threshold $<10^{-25}$. For each family, the protein domains within the family were considered positive test examples, and protein domains within the superfamily but outside the family were considered positive training examples. This yielded 54 families with at least 10 positive training examples and 5 positive test examples. Negative examples for the family were chosen from outside of the positive sequences' fold, and were randomly split into training and test sets in the same ratio as the positive examples.

3.1.2 Fold detection Employing the same dataset and overall methodology as in remote homology detection, we simulated fold detection by formulating as a fold classification within the context of SCOP's hierarchical classification scheme. In this setting, protein domains within the same superfamily were considered to be as positive test examples, and protein domains within the same fold but outside the superfamily were considered as positive training examples. This yielded 23 superfamilies with at least 10 positive training and 5 positive test examples. Negative examples for the superfamily were chosen from outside of the positive sequences' fold and split equally into test and training sets (The classification problem definitions are available at http://bioinfo.cs.umn.edu/supplements/remote-homology/). Since the positive test and training instances were members of different superfamilies within the same fold, this new problem is significantly harder than remote homology detection, as the sequences in the different superfamilies did not have any apparent sequence similarity (Murzin *et al*., 1995).

3.2 Profile generation

The position-specific score and frequency matrices used by the profile-based scoring method of Equation (2) were generated using the latest version of the PSI-BLAST algorithm (available in NCBI's blast release 2.2.10), and were derived from the multiple sequence alignment constructed after five iterations using an e value of 10^{-3} (i.e. we used `blastpgp -j 5 -e 0.001`). The PSI-BLAST was performed against NCBI's nr database that was downloaded in November of 2004 and contained 2 171 938 sequences.

3.3 SVM learning

We use the publicly available support vector machine tool SVM^{light} (Joachims, 1999) that implements an efficient soft margin optimization algorithm. Following the approach used by the LA-Kernel (Saigo *et al*., 2004), for any given positive semi-definite kernel Gram matrix $\mathcal{K}(.,.)$ to be tested, we first normalize the points to unit norm in the feature space and separate them from the origin by adding a constant value of one.

3.4 Evaluation methodology

We measured the quality of the methods by using the receiver operating characteristic (ROC) scores, the ROC50 scores, and the median rate of false positives (mRFP). The ROC score is the normalized area under a curve that plots true positives against false positives for different possible thresholds for classification (Gribskov and Robinson, 1996). The ROC50 score is the area under the ROC curve up to the first 50 false positives. Finally,

Table 1. Comparative performance of the window-based kernel functions that rely on sequence profiles

Kernel	Superfamily level			Fold level		
	ROC	ROC50	mRFP	ROC	ROC50	mRFP
AF-PSSM (1)	0.965	0.692	0.022	0.851	0.275	0.143
AF-PSSM (2)	0.978	0.816	0.013	0.909	0.338	0.075
AF-PSSM (3)	0.976	**0.833**	0.014	0.904	0.340	0.080
AF-PSSM (4)	0.956	0.816	0.019	0.911	**0.374**	0.067
BF-PSSM (1)	0.967	0.794	0.025	0.906	0.359	0.082
BF-PSSM (2)	0.980	**0.854**	0.015	0.928	**0.419**	0.059
BF-PSSM (3)	0.977	0.853	0.016	0.918	0.408	0.069
BF-PSSM (4)	0.965	0.830	0.031	0.918	0.414	0.060
BV-PSSM (1)	0.965	0.808	0.027	0.900	0.423	0.088
BV-PSSM (2)	0.973	**0.855**	0.018	0.927	0.475	0.052
BV-PSSM (3)	0.966	0.851	0.022	0.936	0.480	0.046
BV-PSSM (4)	0.963	0.850	0.026	0.941	**0.481**	0.043

The parameter associated with each kernel corresponds to the width of the *w*mer used to define the kernel. The ROC50 of the best performing value of *w* for each kernel is shown in bold, and the overall best ROC50 is also underlined.

Table 2. Comparative performance of the window-based kernel functions that rely on BLOSUM62

Kernel	Superfamily level			Fold level		
	ROC	ROC50	mRFP	ROC	ROC50	mRFP
AF-GSM (1)	0.906	0.403	0.068	0.720	0.093	0.288
AF-GSM (2)	0.921	0.461	0.055	0.739	0.118	0.255
AF-GSM (6)	0.926	0.549	0.048	0.770	**0.197**	0.217
AF-GSM (7)	0.923	**0.557**	0.056	0.777	0.192	0.210
BF-GSM (1)	0.904	0.488	0.071	0.803	0.166	0.177
BF-GSM (2)	0.923	0.584	0.064	0.808	0.189	0.162
BF-GSM (6)	0.934	**0.669**	0.053	0.822	**0.240**	0.157
BF-GSM (7)	0.933	0.665	0.056	0.812	0.236	0.178
BV-GSM (1)	0.906	0.486	0.070	0.808	0.167	0.176
BV-GSM (2)	0.919	0.571	0.064	0.808	0.182	0.166
BV-GSM (6)	0.930	**0.666**	0.052	0.840	0.242	0.140
BV-GSM (7)	0.929	0.658	0.054	0.845	**0.244**	0.133

AF-GSM, BF-GSM and BV-GSM refer to the BLOSUM62-variants of the corresponding window-based kernels (GSM stands for global scoring matrix). The parameter associated with each kernel corresponds to the width of the *w*mer used to define the kernel. The ROC50 of the best performing value of *w* for each kernel is shown in bold, and the overall best ROC50 is also underlined.

the mRFP is the number of false positives scoring as high or better than the median-scoring true positives.

Among these evaluation metrics, owing to the fact that the positive class is substantially smaller than the negative class, the ROC50 is considered to be the most useful measure of performance for real-world applications (Gribskov and Robinson, 1996). For this reason, our discussions in the rest of this section will primary focus on ROC50-based comparisons. Also, the ROC50 values that are being reported for the superfamily- and fold-level evaluations correspond to the average ROC50 values over the 54 families and 23 superfamilies, respectively.

4 RESULTS

4.1 Performance of the window-based kernels

Table 1 summarizes the performance achieved by the window-based kernels for the superfamily- and fold-level classification problems across a range of w values.

These results show that for both the superfamily and fold level classification problems, the BV-PSSM kernel achieves the best results, the AF-PSSM kernel tends to perform the worst, whereas the BF-PSSM kernel's performance is between these two. In the case of superfamily classification, the performance advantage of BV-PSSM over that of BF-PSSM is relatively small, whereas in the case of fold classification, the former has a clear advantage. It achieves an ROC50 value that is on average 16.3% better across the different window lengths.

Comparing the sensitivity of the three schemes based on the value of w, we see that, as expected, their performance is worse for $w = 1$, as they only consider $w$mers of length 3, and their performance improves as the value of w increases. In general, the BV-PSSM kernel performs better for larger windows, whereas the performance of the other kernels tends to degrade more rapidly as the length of the window increases beyond a point. Again, this result is consistent with the design motivation behind the BV-PSSM kernel. Also, the results show that the best value of w is also dependent on the particular classification problem. For most kernels, the best results for fold classification were obtained with longer windows compared with the superfamily classification.

To see the effect of using sequence profiles, we performed a sequence of classification experiments in which we used the same set of window-based kernel functions, but instead of scoring the similarity between two amino acids using the profile-based scheme [Equation (2)], we used the BLOSUM62 position-independent scoring matrix. The results obtained from these experiments, which are summarized in Table 2, illustrate the advantage of using sequence profiles in designing kernel functions for both remote homology detection and fold recognition. The profile-based kernel functions achieve significant improvements over their non-profile counterparts across all different kernel functions, classification problems and metrics.

Comparing the performance of the profile-based kernel functions across the two classification problems, we see that their overall effectiveness in remote homology detection (superfamily level classification) is much higher than that of fold recognition. This result is in line with the underlying complexity of the classification problem, as the sequence-based signals for fold recognition are extremely weak. This is also manifested by the relative improvement achieved by the profile-based kernel functions over their BLOSUM62-based counterparts (Tables 1 and 2). For fold recognition, the ROC50 values of the profile-based kernels are higher than those based on BLOSUM62 by a factor of two, whereas for remote homology prediction, the relative ROC50 values are higher by 25–30%.

In light of the previously published results on LA-Kernels (Saigo *et al*., 2004), the better results achieved by the BF-PSSM and BV-PSSM kernels over those achieved by the AF-PSSM kernel (which also hold for their corresponding BLOSUM62-based instances of these kernels) were surprising. One explanation for this discrepancy

Table 3. Comparative performance of the local alignment-based kernel functions that rely on sequence profiles

Kernel	Superfamily level			Fold level		
	ROC	ROC50	mRFP	ROC	ROC50	mRFP
2.0, 0.125, 0.0	0.972	0.784	0.014	0.867	0.377	0.111
2.0, 0.250, 0.0	0.972	0.791	0.014	0.873	0.334	0.114
3.0, 0.125, 0.0	0.971	0.796	0.013	0.860	0.382	0.133
3.0, 0.250, 0.0	0.960	0.771	0.027	0.852	0.395	0.138
3.0, 0.750, 1.5	0.982	**0.904**	0.015	0.933	0.530	0.052
3.0, 0.750, 2.0	0.979	0.901	0.017	0.936	**0.571**	0.054

The three parameters for each kernel correspond to the values for the gap opening, gap extension and zero-shift parameters, respectively. The ROC50 of the best performing scheme is underlined.

Table 4. Comparative performance of the local alignment-based kernel functions that rely on BLOSUM45 and BLOSUM62

Kernel	Superfamily level			Fold level		
	ROC	ROC50	mRFP	ROC	ROC50	mRFP
B45, 3.0, 0.0	0.944	0.686	0.037	0.809	0.165	0.169
B45, 10.0, 0.0	0.940	0.687	0.042	0.789	0.200	0.185
B62, 3.0, 0.0	0.947	0.686	0.038	0.781	0.188	0.217
B62, 10.0, 0.0	0.912	0.599	0.060	0.781	0.182	0.185
B62, 5.0, 0.5	0.948	**0.711**	0.039	0.826	**0.223**	0.176
B62, 5.0, 1.0	0.946	**0.711**	0.038	0.808	0.214	0.155

The three parameters for each kernel correspond to the particular global scoring matrix (B45 for BLOSUM45 and B62 for BLOSUM62) and the values for the gap opening and zero-shift parameters, respectively. In all cases, the gap extension cost was set to 1.0. The ROC50 of the best performing scheme is underlined.

may be the fact that our window-based kernels consider only short-length ungapped alignments, and the results may be different when longer alignments with gaps are considered as well.

4.2 Performance of the local alignment-based kernels

Table 3 summarizes the performance achieved by the optimal local alignment-based kernel for the superfamily- and fold-level classification problems across a representative set of values for the gap-opening, gap-extension and zero-shift parameters. These parameter values were selected after performing a study in which the impact of a large number of value combinations was experimentally studied, and represent some of the best performing combinations. Owing to space constraints, this parameter study is not included in this paper.

The most striking observation from these results is the major impact that the zero-shift parameter has to the overall classification performance. For both the superfamily- and fold-level classification problems, the best results are obtained by the SW-PSSM kernel for which the zero shift parameter has been optimized (i.e. the results corresponding to the last two rows of Table 3).

Comparing the classification performance of the SW-PSSM kernel against the window-based kernels (Table 1) we see that the zero-shift optimized SW-PSSM kernel leads to better results than those obtained by the window-based kernels. Moreover, the relative performance advantage of SW-PSSM is higher for fold recognition over the superfamily classification problem. However, if the SW-PSSM kernel does not optimize the zero-shift parameter (i.e. $zs = 0.0$), the window-based kernels consistently outperform the SW-PSSM kernel. We also performed a limited number of experiments to see the extent to which the performance of the window-based kernels can be improved by explicitly optimizing the zero-shift parameter for them as well. Our results show that these kernels are not significantly affected by such optimizations.

To also see the impact of sequence profiles in the context of kernels derived from optimal local alignments, we evaluated the classification performance of a set of kernel functions that compute the optimal local sequence alignment using the BLOSUM45 and BLOSUM62 amino acid scoring matrices. Table 4 shows some of the results obtained with these kernel functions for a representative set of values for the gap opening, gap extension and zero-shift parameters.

Comparing the results of Table 4 with those of Table 3 we see that, as was the case with the window-based kernels, incorporating profile information leads to significant improvements in the overall classification performance. In addition, these results show that (1) the widely used value for the gap-opening cost ($go = 10$) is not necessarily the best for either remote homology detection or fold recognition and (2) the classification performance achieved by local alignment kernels derived from the BLOSUM matrices can be further improved by explicitly optimizing the zero-shift parameter as well.

Table 5. Comparison against different schemes for the superfamily-level classification problem

Kernel	ROC	ROC50	mRFP
SVM-Fisher	0.773	0.250	0.204
SVM-Pairwise	0.896	0.464	0.084
LA-eig ($\beta = 0.2$)	0.923	0.661	0.064
LA-eig ($\beta = 0.5$)	0.925	0.649	0.054
SVM-HMMSTR-Ave	—	0.640	0.038
Mismatch	0.872	0.400	0.084
Profile(4,6)	0.974	0.756	0.013
Profile(5,7.5)	0.980	0.794	0.010
AF-PSSM(2)	0.978	0.816	0.013
BF-PSSM(2)	0.980	0.854	0.015
BV-PSSM(2)	0.973	0.855	0.018
SW-PSSM(3.0,0.750,1.50)	0.982	**0.904**	0.015
AF-GSM(6)	0.926	0.549	0.048
BF-GSM(6)	0.934	0.669	0.053
BV-GSM(6)	0.930	0.666	0.052
SW-GSM(B62,5.0,1,0.5)	0.948	0.711	0.039

The SVM-Fisher, SVM-Pairwise, LA-Kernel and Mismatch results were obtained from Saigo *et al*. (2004). The SVM-HMMSTR results were obtained from Hou *et al*. (2004) and correspond to the best-performing scheme (the authors did not report ROC values). The Profile results were obtained locally by running the publicly available implementation of the scheme obtained from the authors. The ROC50 value of the best performing scheme has been underlined.

4.3 Comparisons with other schemes

Tables 5 and 6 compare the performance of the various kernel functions developed in this paper against that achieved by a number

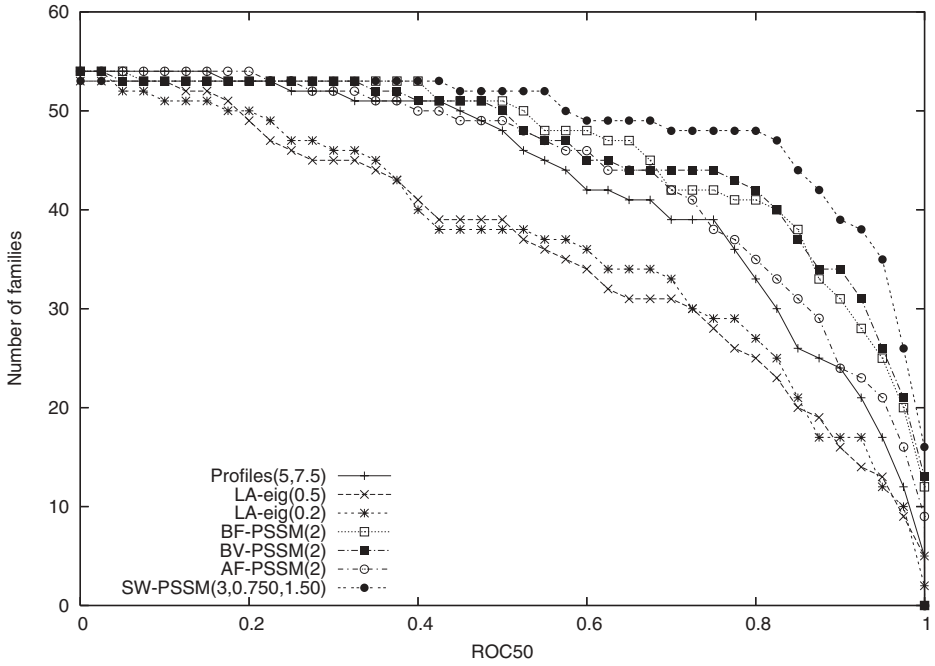

Fig. 1. Comparison of the different SVM-based methods for remote homology detection on the SCOP 1.53 benchmark dataset. The graph plots the total number of families for which a given method exceeds an ROC-50 score threshold.

Table 6. Comparison against different schemes for the fold-level classification problem

Kernel	ROC	ROC50	mRFP
LA-eig($\beta = 0.2$)	0.847	0.212	0.129
LA-eig($\beta = 0.5$)	0.771	0.172	0.193
Profile(4,6)	0.912	0.305	0.071
Profile(5,7.5)	0.924	0.314	0.069
AF-PSSM(4)	0.911	0.374	0.067
BF-PSSM(4)	0.918	0.414	0.060
BV-PSSM(4)	0.941	0.481	0.043
SW-PSSM(3.0,0.750,2.0)	0.936	**0.571**	0.054
AF-GSM(6)	0.770	0.197	0.217
BF-GSM(6)	0.822	0.240	0.157
BV-GSM(7)	0.845	0.244	0.133
SW-GSM(B62,5,1.0,0.5)	0.826	0.223	0.176

The results for the LA-Kernel were obtained using the publicly available kernel matrices that are available at the author's website. The Profile results were obtained locally by running the publicly available implementation of the scheme obtained from the authors. The ROC50 value of the best performing scheme has been underlined.

of previously developed schemes for the superfamily and fold level classification problems, respectively. In the case of the superfamily level classification problem, the performance is compared against SVM-Fisher (Jaakkola *et al*., 2000), SVM-Pairwise (Liao and Noble, 2002) and different instances of the LA-Kernel (Saigo *et al*., 2004), SVM-HMMSTR (Hou *et al*., 2004), Mismatch (Leslie *et al*., 2003) and Profile (Kuang *et al*., 2005). In the case of the fold level classification problem, we only include results for the LA-Kernel and Profile schemes, as these results could be easily obtained from the publicly available data and programs for these schemes.

The results in these tables show that both the window- and local alignment-based kernels derived from sequence profiles (i.e. AF-PSSM, BF-PSSM, BV-PSSM and SW-PSSM) lead to results that are in general better than those obtained by existing schemes. Comparing the ROC50 values obtained by our schemes, we see that each one of them outperforms all existing schemes. The performance advantage of these kernels is greater over existing schemes that rely on sequence information alone (e.g. SVM-Pairwise, LA-Kernels), but still remains significant when compared against schemes that either directly take into account profile information (e.g. SVM-Fisher, Profile) or utilize higher-level features derived by analyzing sequence-structure information (e.g. SVM-HMMSTR). Also, the relative advantage of our profile-based methods over existing schemes is greater for the much harder fold level classification problem over the superfamily-level classification problem. For example, the SW-PSSM scheme achieves ROC50 values that are 13.8 and 81.8% better than the best values achieved by existing schemes for the superfamily- and fold-level classification problems, respectively.

To get a better understanding of the relative performance of the various schemes across the different classes, Figures 1 and 2 plot the number of classes whose ROC50 was greater than a given threshold that ranges from 0 to 1. Specifically, Figure 1 shows the results for the remote homology detection problem, whereas Figure 2 shows the results for the fold detection problem. (Note that these figures contain only results for the schemes that we were able to run locally.) These results show that our profile-based methods lead to higher ROC50 values for a greater number of classes than either the Profile or LA-kernels, especially for larger ROC50 values (e.g. in the range of 0.6–0.95). Also, the SW-PSSM tends to

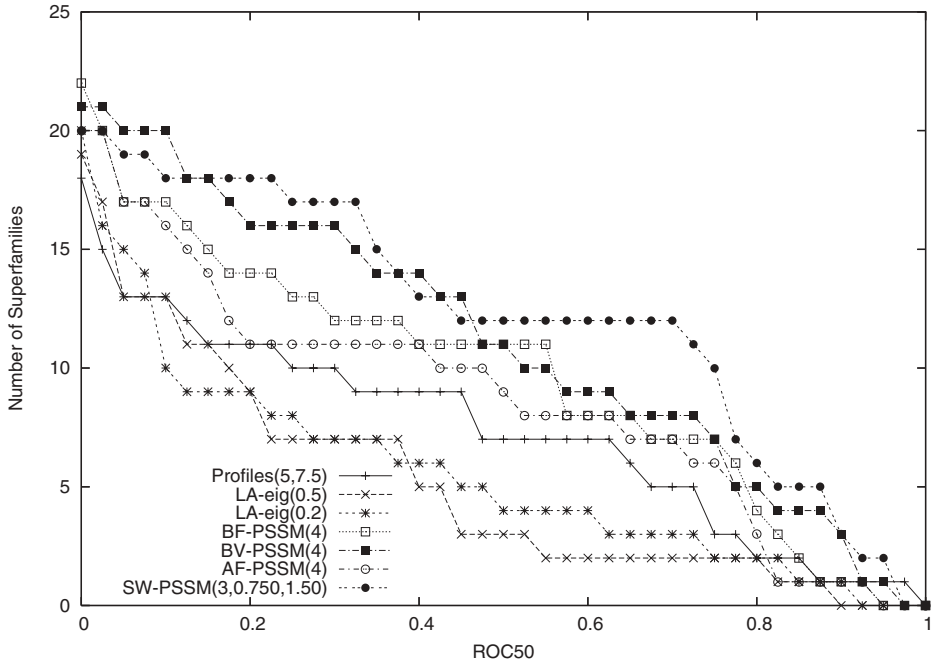

Fig. 2. Comparison of the different SVM-based methods for fold detection on the SCOP 1.53 benchmark dataset. The graph plots the total number of superfamilies for which a given method exceeds an ROC-50 score threshold.

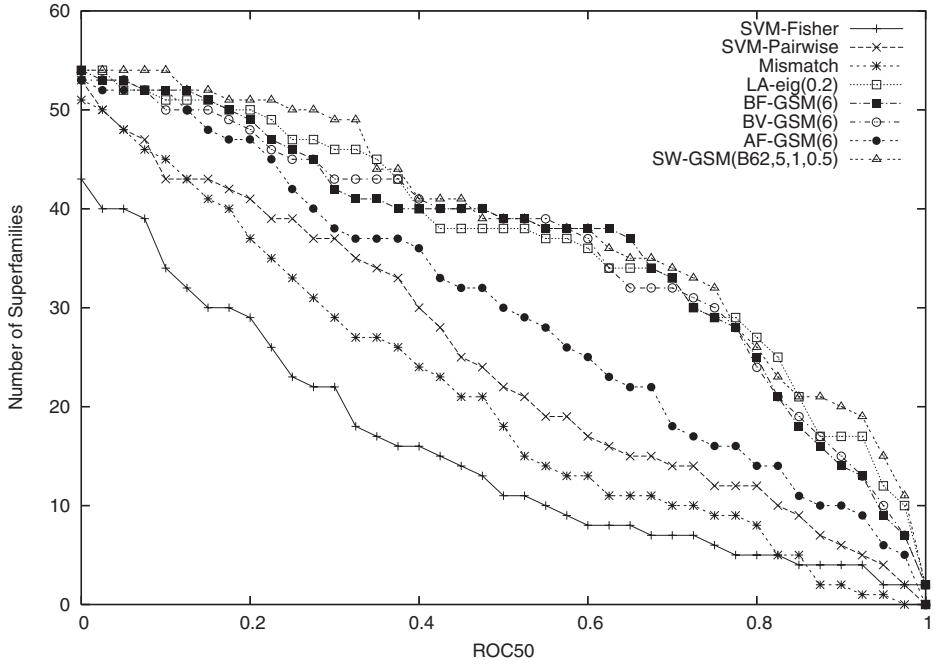

Fig. 3. Comparison of the different non-profile-based SVM methods for remote homology detection on the SCOP 1.53 benchmark dataset. The graph plots the total number of families for which a given method exceeds an ROC-50 score threshold.

consistently outperform the rest of the profile-based direct kernel methods.

In addition, the results for the BF-GSM, BV-GSM and SW-GSM kernels that rely on the BLOSUM scoring matrices show that these kernel functions are capable of producing results that are superior to all of the existing non-profile-based schemes. In particular, the properly optimized SW-GSM scheme is able to achieve significant improvements over the best LA-Kernel-based scheme (7.6% higher ROC50 value) and the best SVM-HMMSTR-based scheme (15.1% higher ROC50 value). This relative performance of BF-GSM,

BV-GSM and SW-GSM kernels over existing non-profile-based schemes is further illustrated in Figure 3, which plots the number of classes whose ROC50 was greater than a given threshold. For almost all threshold values, these three new kernels outperform all previous schemes. Note that this plot does not include results for the SVM-HMMSTR scheme as this information was not reported.

5 DISCUSSION AND CONCLUSION

This paper presented and experimentally evaluated a number of kernel functions for protein sequence classification that were derived by considering explicit measures of profile-to-profile sequence similarity. The experimental evaluation in the context of a remote homology prediction problem and a fold recognition problem show that these kernels are capable of producing superior classification performance over that produced by earlier schemes.

Three major observations can be made by analyzing the performance achieved by the various kernel functions presented in this paper. First, as was the case with a number of studies on the accuracy of protein sequence alignment (Mittelman *et al*., 2003; Wang and Dunbrack Jr, 2004; Marti-Renom *et al*., 2004), the proper use of sequence profiles lead to dramatic improvements in the overall ability to detect remote homologs and identify proteins that share the same structural fold. Second, kernel functions that are constructed by directly taking into account the similarity between the various protein sequences tend to outperform schemes that are based on a feature-space representation [where each dimension of the space is constructed as one of *k*-possibilities in a *k*-residue long subsequence or using structural motifs (Isites) in the case of SVM-HMMSTR]. This is especially evident by comparing the relative advantage of the window-based kernels over the Profile kernel. Third, time-tested methods for comparing protein sequences based on optimal local alignments (as well as global and local-global alignments), when properly optimized for the classification problem at hand, lead to kernel functions that are in general superior to those based on either short subsequences (e.g. spectrum, mismatch, profile or window-based kernel functions) or local structural motifs (e.g. SVM-HMMSTR). The fact that these widely used methods produce good results in the context of SVM-based classification is reassuring as to the validity of these approaches and their ability to capture biologically relevant information.

ACKNOWLEDGEMENT

This work was supported by NSF EIA-9986042; ACI-9982274, ACI-0133464, ACI-0312828, IIS-0431135, the Army High Performance Computing Research Center contract number DAAD19-01-2-0014, and by the Digital Technology Center at the University of Minnesota. Funding to pay the Open Access publication charges for this article was provided by the National Science Foundation.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1997) Gapped blast and PSI-blast: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Baldi,P. *et al*. (1994) Hidden Markov models of biological primary sequence information. *Proc. Natl Acad. Sci. USA*, **91**, 1053–1063.

Ben-Hur,A. and Brutlag,D. (2003) Remote homology detection: a motif based approach. *Bioinformatics*, **19**, i26–i33.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol*., **162**, 705–708.

Gribskov,M. *et al*. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 355–4358.

Gribskov,M. and Robinson,N. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem*., **20**, 25–33.

Gusfield,D. (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, NY.

Heger,A. and Holm,L. (2001) Picasso: generating a covering set of protein family profiles. *Bioinformatics*, **17**, 272–279.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid subsitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Hou,Y. *et al*. (2003) Efficient remote homology detection using local structure. *Bioinformatics*, **19**, 2294–2301.

Hou,Y. *et al*. (2004) Remote homolog detection using local sequence–structure correlations. *Proteins*, **57**, 518–530.

Jaakkola,T. *et al*. (2000) A discriminative framework for detecting remote protein homologies. *J. Comput. Biol*., **7**, 95–114.

Joachims,T. (1999) Making large-scale svm learning practical. In Schölkopf,B., Burges,C. and Smola,A. (eds), *Advances in Kernel Methods—Support Vector Learning*. MIT Press, Cambridge, MA.

Karplus,K. *et al*. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.

Krogh,A. *et al*. (1994) Hidden Markov models in computational biology: applications to protein modeling. *J. Mol. Biol*., **235**, 1501–1531.

Kuang,R. *et al*. (2005) Profile-based string kernels for remote homology detection and motif extraction. *J. Bioinform. Comput. Biol*., **3**, 527–550.

Leslie,C. *et al*. (2002) The spectrum kernel: a string kernel for SVM protein classification. *Pac. Symp. Biocomput*., 564–575.

Leslie,C. *et al*. (2003) Mismatch string kernels for svm protein classification. *Adv. Neural Inf. Process. Syst*., **20**, 467–476.

Liao,L. and Noble,W.S. (2002) Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. In: *Proceedings of the International Conference on Research in Computational Molecular Biology*, Washington DC, pp. 225–232.

Marti-Renom,M. *et al*. (2004) Alignment of protein sequences by their profiles. *Protein Sci*., **13**, 1071–1087.

Mittelman,D. *et al*. (2003) Probabilistic scoring measures for profile–profile comparison yield more accurate short seed alignments. *Bioinformatics*, **19**, 1531–1539.

Murzin,A.G. *et al*. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol*., **247**, 536–540.

Papadimitriou,C.H. and Steiglitz,K. (1982) *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ.

Saigo,H. *et al*. (2004) Protein homology detection using string alignment kernels. *Bioinformatics*, **20**, 1682–1689.

Scholkopf,B. and Smola,A. (2002) *Learning with Kernels*. MIT Press, Boston, MA.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.

Vapnik,V. (1998) *Statistical Learning Theory*. John Wiley, New York.

Wang,G. and Dunbrack,R.L.,Jr. (2004) Scoring profile-to-profile sequence alignments. *Protein Sci*., **13**, 1612–1626.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4248–4254
doi:10.1093/bioinformatics/bti702

Structural bioinformatics

Pcons5: combining consensus, structural evaluation and fold recognition scores

Björn Wallner* and Arne Elofsson

Stockholm Bioinformatics Center, Stockholm University, SE-106 91 Stockholm, Sweden

Received on July 20, 2005; revised on September 5, 2005; accepted on October 1, 2005
Advance Access publication October 4, 2005

ABSTRACT

Motivation: The success of the consensus approach to the protein structure prediction problem has led to development of several different consensus methods. Most of them only rely on a structural comparison of a number of different models. However, there are other types of information that might be useful such as the score from the server and structural evaluation.

Results: Pcons5 is a new and improved version of the consensus predictor Pcons. Pcons5 integrates information from three different sources: the consensus analysis, structural evaluation and the score from the fold recognition servers. We show that Pcons5 is better than the previous version of Pcons and that it performs better than using only the consensus analysis. In addition, we also present a version of Pmodeller based on Pcons5, which performs significantly better than Pcons5.

Availability: Pcons5 is the first Pcons version available as a standalone program from http://www.sbc.su.se/~bjorn/Pcons5. It should be easy to implement in local meta-servers.

Contact: bjorn@sbc.su.se

INTRODUCTION

The use of many different methods to predict the structure of a protein is now state-of-the-art in protein structure prediction. This is facilitated by the different meta- or consensus predictors that are available, e.g. through the meta-server at http://bioinfo.pl/Meta/ (Bujnicki *et al*., 2001). The consensus predictors use the result from different prediction methods to select the best protein model. In principle they are all based on the simple approach of selecting the most abundant representative among the set of high scoring models. Pcons (Lundström *et al*., 2001) was the first fully automated consensus predictor, followed by several others (Fischer, 2003; Ginalski *et al*., 2003). All benchmarking results obtained in the last two years, both at CASP (Moult *et al*., 2003) and in LiveBench (Rychlewski *et al*., 2003) indicate that consensus prediction methods are more accurate than the best of the independent fold recognition methods (Ginalski *et al*., 2005), e.g. in CAFASP3 the performance was 30% higher than that of the best independent fold recognition methods and comparable to the 2–3 best human CASP predictors (Fischer, 2003). The main strength of the consensus analysis is coupled to the structural comparison. However, there are also other factors that can be used in the selection process. The score from the fold recognition method and a structural evaluation of the model are such parameters. A problem with using the score from the fold recognition methods directly is that the scoring scheme might change at any time, leading to a frequent re-optimization of the parameters. In this paper we describe the newest version of Pcons, Pcons5. It consists of three parts: consensus analysis, structural evaluation and a final part dependent on the score from the fold recognition server. In addition, we also present an improved version of Pmodeller (Wallner *et al*., 2003) based on Pcons5 and ProQ (Wallner and Elofsson, 2003).

*To whom correspondence should be addressed.

METHODS

Datasets

All datasets used in the development of Pcons5 were constructed from different versions of LiveBench (Bujnicki *et al*., 2001). These sets contain models that are possible to be obtained for unknown targets and that show a range of quality differences. LiveBench is continuously measuring the performance of different fold recognition web servers by submitting the sequence of recently solved proteins structures, with no obvious close homolog [10^{-3} BLAST cutoff (Altschul *et al*., 1997)] to a protein in the Protein Data Bank (Berman *et al*., 2000).

The structural evaluation module was trained on the same dataset as used in ProQ (Wallner and Elofsson, 2003) (LiveBench-2 data). The parameters for the consensus analysis and score evaluation as well as the final combination were performed on a dataset constructed from LiveBench-4. The LiveBench-4 dataset was collected during the period November 7, 2001 to April 25, 2002 and contains protein structure predictions for 107 targets from 14 individual servers and 3 consensus servers. In total 10 974 protein models for these 11 servers were used: PDB–BLAST, FFAS (Rychlewski *et al*., 2000), Sam-T99 (Karplus *et al*., 1998), mGenTHREADER (Jones, 1999), INBGU (Fischer, 2000), three FUGUE servers (Shi *et al*., 2001), 3D-PSSM (Kelley *et al*., 2000), Orfeus (Ginalski *et al*., 2003) and Superfamily (Gough and Chothia, 2002). The models used were simple backbone copies of the aligned residues from the template.

METHOD DEVELOPMENT

Pcons5 consists of three different modules: consensus analysis, structural evaluation and score evaluation. It has been developed with the goal to be independent of the use of a fixed set of methods/servers, i.e. it should work with any number of methods and with any number of models. Each of the modules in Pcons5 produce two scores reflecting different aspects of model quality. These scores are combined to produce the final score using a weighted sum. In the following subsections the three different modules will be described.

Consensus analysis

The consensus analysis is performed in a similar way as in 3D-Jury (Ginalski *et al*., 2003), with the only difference being that LGscore (Cristobal *et al*., 2001) is used to compare the models. The comparison is done for all and for the first ranked models only, as in the different versions of 3D-Jury. This results in two scores: one reflecting the average similarity to all other models [Equation (1)] and the other reflecting the similarity to all first ranked models [Equation (2)]

$$C_i^{\text{all}} = \frac{1}{N} \sum_{\forall j \notin \text{method(i)}} \text{sim}(i,j) \tag{1}$$

$$C_i^{\text{first}} = \frac{1}{M} \sum_{\substack{\forall j \in \text{rank(j)}=1 \\ j \neq i}} \text{sim}(i,j), \tag{2}$$

where N is the number of comparisons to other methods, M the number of comparisons to first ranked models and sim(i, j) is the similarity between models i and j. Here, we used LGscore but any structural similarity measure should most likely work.

Structural evaluation

The structural evaluation is done using a backbone version of ProQ (Wallner and Elofsson, 2003). ProQ uses distribution of atom–atom contacts, residue–residue contacts, secondary structure information and surface accessibility for different amino acids to assess the quality of protein models. The original version was developed for protein models with all atoms. The version used in Pcons5 uses only the backbone atoms, usually obtained by copying the aligned coordinates from the template. This version of ProQ is not as accurate as the original ProQ version, but since there is no need to build all-atom models, the overall method is considerably faster.

Since the structural evaluation is performed on backbone models it is not possible to use exactly the same structural information as in the all-atom version of ProQ, e.g. using contacts between different types of atoms was no longer possible. However, the same six residue types as in ProQ were used, but the residue–residue contact cutoff had to be increased to 14 Å to compensate for the non-existing side-chains. The cutoff was chosen by trying different cutoffs in the range 6–20 Å. The calculation of surface accessibility also had to be changed. We chose to use a reduced representation defining buried and exposed residues based on number of neighboring CA atoms. Residues with <16 atoms within 10 Å were defined as exposed and residues with >20 atoms within 10 Å as buried. These definitions showed the largest agreement with Naccess (Lee and Richards, 1971). The secondary structure information was the fraction of agreement between predicted secondary structure using PSIPRED (Jones, 1999) and the actual secondary structure in the model. As in the original ProQ version, neural nets were trained to predict LGscore (Cristobal *et al*., 2001) and MaxSub (Siew *et al*., 2000), based on the structural features. A final correlation coefficient of 0.65 was obtained, in comparison with 0.76 for the all-atom ProQ.

Score evaluation

A good indicator of model quality is the score from the fold recognition method or server, a high score is usually connected to a good model. However, since Pcons5 should be independent of a fixed number of servers it is impossible to include the raw score from the server directly in Pcons5. Instead, each raw score is scaled to a common scale based on the reliability of the score. If the reliability is not known Pcons5 will not use the score to compute the final score.

The score evaluation was designed to be easy to update and facilitate the inclusion of new methods, without the need to re-optimize all parameters. Further, if the scoring of a method suddenly changes it should not impact the result too much. To limit the impact on the final score, all scores were scaled using two levels, 'good' and 'very good'. In principle it works as follows: if the score is good the model will obtain one extra point and if the score is even better (very good) the model will have the possibility to get one additional point. Thus, even an extremely high score could only yield two additional points.

The reliability of each server score was assessed by correlate fold recognition score with model quality from LiveBench-4. For each server two cutoffs were used to define 'good' models and two were used to define 'very good' models. These cutoffs were decided by analyzing the quality of models associated with a certain score. In more detail, the first cutoff was set to the score for which 50% of all models had an LGscore > 1.5 and the second to the score for which 90% of all models had LGscore > 1.5. The 'very good' models were defined in a similar manner but with LGscore > 3. For scores falling between these cutoffs a linear fit was used. The process is exemplified here by the PSI-BLAST method, which has a familiar E-value score (Figures 1 and 2). Fifty percent of all models with PSI-BLAST E-value < 10^2 have LGscore > 1.5 and 90% of the models with a score > $10^{-6.2}$ have LGscore > 1.5. And for LGscore > 3 the cutoffs are $10^{-20.9}$ and $10^{-124.3}$ respectively. For scores in between the cutoff values a linear fit is used, e.g. a score of 10^{-5} would yield a scaled score of 0.81 on the 'good' scale and 0 on the 'very good' (Fig. 1). In a way this is a reflection of the reliability of a hit in the database with a certain score. An E-value of 10^{-5} is mostly likely correct but the alignment is probably not optimal. If however the E-value is $10^{-72.6}$ the model is very likely to be of high quality and this is also reflected by a significantly higher scaled score of 1 on the 'good' scale and 0.5 on the 'very good' scale (Fig. 2).

Compiling the final Pcons5 score

The final Pcons5 score is a combination of the six different scores from the three different modules described above. They were combined using multiple linear regression to fit the LGscore quality measure, with the following coefficients

$$\begin{aligned}\text{Pcons5} = {} & 0.53C^{\text{all}} + 0.20C^{\text{first}} + 0.64\text{ProQ}^{\text{MX}} + 0.27\text{ProQ}^{\text{LG}} \\ & + 0.32\text{Score}^{\text{good}} + 0.13\text{Score}^{\text{very good}},\end{aligned} \tag{3}$$

where C^{all} and C^{first} are the scores from the consensus analysis, ProQ^{MX} and ProQ^{LG} the predicted MaxSub and LGscore scores, respectively and $\text{Score}^{\text{good}}$ and $\text{Score}^{\text{very good}}$ are the scaled fold recognition scores for good and very good, respectively.

The fit is rather good explaining 86% of the variance in the data (Table 1). If the range of the parameters are known their influence on the final score can be assessed directly from the size of the coefficients. The range of C^{all}, C^{first} and ProQ^{LG} are all comparable in size (as they are trained to predict the LGscore of the model), ProQ^{MX} needs to be multiplied by ten to put it on the same scale and

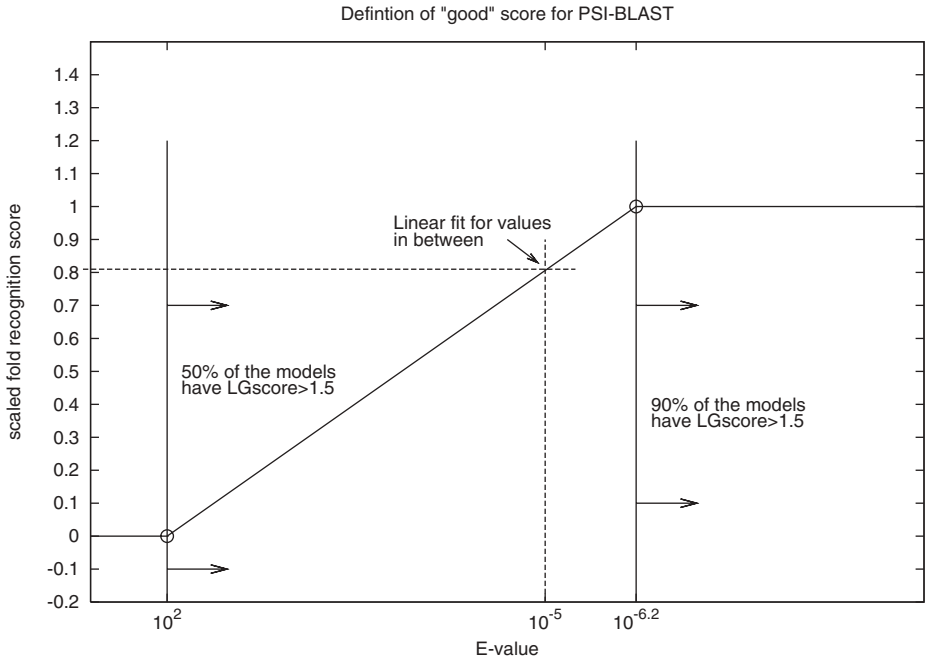

Fig. 1. Scaling of PSI-BLAST scores for the definition of 'good' models.

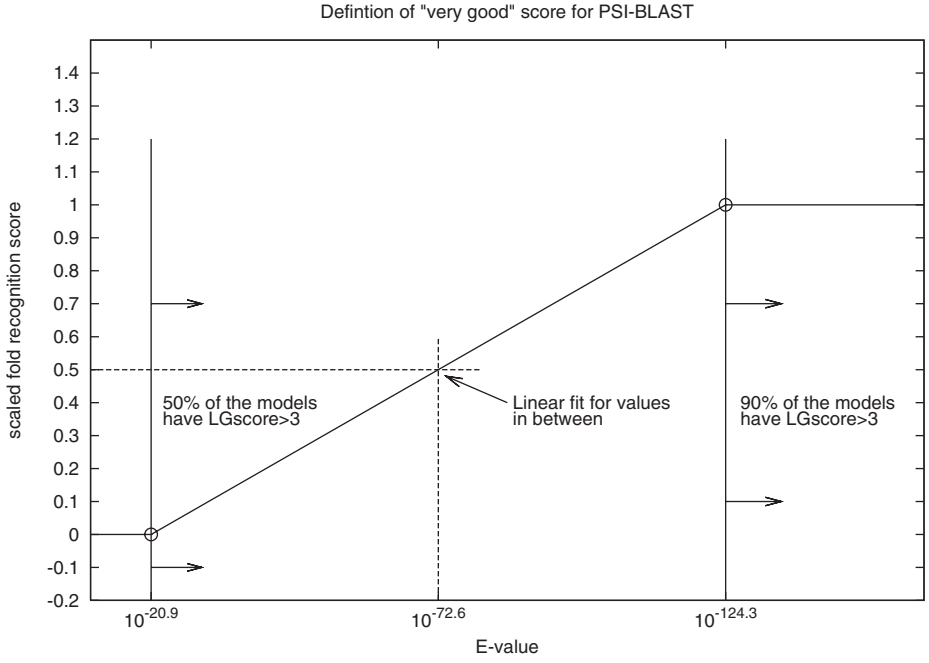

Fig. 2. Scaling of PSI-BLAST scores for the definition of 'very good' models.

$Score^{good}$ and $Score^{very\ good}$ are roughly one-third of the three first parameters. Thus, the consensus analysis is the most important factor, followed by the structural evaluation using $ProQ^{LG}$, while $ProQ^{MX}$ and the two server-specific scores influence the final score to a lesser degree. This is also in agreement with the R^2 values in Table 1.

Pmodeller5

For the previous Pcons version we have developed a corresponding Pmodeller version (Wallner *et al*., 2003). Pmodeller uses Modeller (Sali and Blundell, 1993) to build all-atom models which are assessed using ProQ and the final Pmodeller score is a combination

Table 1. Performance of the individual modules in Pcons5

Score	R^2
C^{all}	0.81
C^{first}	0.77
$ProQ^{MX}$	0.44
$ProQ^{LG}$	0.52
$Score^{good}$	0.55
$Score^{very\ good}$	0.18
Pcons5	**0.86**

The performance was measured by the squared correlation coefficient, R^2, also called the coefficient of determination. The highest marked in boldface. The difference between the second highest is significant with a P-value $<10^{-5}$.

of the Pcons score and ProQ score. In CASP5 it was shown that Pmodeller performs better than its corresponding Pcons version.

In Pmodeller5, we have used a slightly different approach. Instead of a linear combination of the Pcons and ProQ scores, the combination is done in two steps. First Pcons5 is used to find the best scoring models, then all-atom models are built using Modeller6v2 (Sali and Blundell, 1993) for all models with a score within 10% from the highest. These models are then subjected to a re-ranking using the original all-atom version of ProQ (Wallner and Elofsson, 2003), which is significantly better than the backbone ProQ module used in Pcons5. The final Pmodeller score consists only of the ProQ score. The use of only the top 10% scoring models will ensure that only the best models are included in the final ranking. At the same time the algorithm gets significantly faster, since there is no need to build all-atom models for low scoring models.

RESULTS AND DISCUSSION

It is important that new methods are benchmarked and compared with existing methods. Pcons5 has been thoroughly benchmarked in LiveBench and both Pcons5 and Pmodeller5 participated in CASP6.

Performance in LiveBench

ROC analysis from LiveBench-8 is seen in Figure 3. With the exception of two incorrect hits, Pcons5 consistently performs better than the previous versions of Pcons. In addition, it also performs better than the 3D-Jury method that uses consensus from eight selected servers (more or less the same servers that Pcons5 was using), i.e. this 3D-Jury method corresponds to the consensus analysis module in Pcons5. Consequently, the structural evaluation using ProQ and the score from the server is the reason for the 10% improvement relative to the 3D-Jury method. The performance of the best independent server, SAM-T02 (Karplus *et al*., 2003), used by Pcons5 in LiveBench-8 is remarkable. Pcons5 is only marginally better (for >2 incorrect) and it outperforms all previous Pcons versions and even 3D-Jury which also uses SAM-T02 in its consensus analysis. However, one problem with Pcons5 on LiveBench is that we have no control over servers that are used. Sometimes the results from certain independent servers are missing, e.g. if the pressure of the servers is high. Therefore, the comparison with the independent servers is better done on the CASP6 data, where it was guaranteed that the results from as many servers as possible were used as input to Pcons5.

Performance in CASP6

Pcons5 uses a number of independent servers as its input, normally these are submitted through the meta-server (http://bioinfo.pl/meta/). During CASP6 there was a problem running Pcons5 as an automatic server using the meta-server. Instead two different versions of Pcons5 participated in CASP ('Pcons5' and 'SBC-Pcons5'). 'Pcons5' used a limited number of independent servers and was run through the genesilico.pl meta-server (http://genesilico.pl/meta/), while 'SBC-Pcons5' used more (and better) servers from the bioinfo.pl meta-server once these data were available. Unfortunately these data were not always ready within the time limit to participate as a server in CASP. This forced us to have 'SBC-Pcons5' registered as a manual group, even though it was run without any human intervention. For each of the Pcons version a corresponding Pmodeller version also participated in CASP ('Pmodeller5' and 'SBC-Pmodeller5').

As expected the 'SBC-*' versions of Pcons and Pmodeller using more and better servers performed significantly better with 10% higher sum of GDT_TS than the versions using fewer servers. This shows that the success of the consensus approach is dependent on a good set of individual servers. It can be used on a limited number of servers, but the performance can only be expected to be as good as the models it can choose between. The following analysis will be on the 'SBC-*' versions of Pcons and Pmodeller. The relative performance of Pcons and Pmodeller is similar for the versions using fewer servers (data not shown).

To compare the performance of Pmodeller5, Pcons5, with the server they are using and with the other groups participating in CASP6, the GDT_TS (Zemla *et al*., 1999) score for the first ranked models was used. Other scoring schemes like MaxSub (Siew *et al*., 2000) or TM-score (Zhang and Skolnick, 2004) produce similar results (data not shown). Identical to our previous analysis of CASP5 results (Wallner *et al*., 2003) we made two assumptions in our analysis. First, insignificant differences in performances were ignored, by considering two models with a difference <0.05 GDT_TS score to be of similar quality. Second, models where none of the compared methods made a 'correct' prediction were also ignored. This was done by ignoring all targets where none of the compared methods could align >30 residues, i.e. where the GDT_TS multiplied by the length of the target is <0.30. Targets were also divided into comparative modeling targets (CM) by concatenating CM easy and CM hard and to fold recognition/new fold (FR/NF) by combining FR-homologous, FR-analogous and new fold as defined by the CASP assessors (see http://predictioncenter.org). This resulted in a total of 59 domains, divided into 41 CM targets and 18 FR targets, after filtering out models where no predictor made a 'correct' prediction.

Pcons5 versus Pmodeller5

Pmodeller5 performed significantly better than Pcons5 for 10 targets and only significantly worse for 3, Table 2. For the FR&NF models it did not make any model worse than the corresponding Pcons5 model. Since Pmodeller5 uses the result from Pcons5 it is possible that the predictions are based on the same alignment. This is the case for 20% of the Pmodeller5 models, but in none of these cases is the model significantly improved by the homology modeling procedure. Thus the main reason for the increase in performance is the re-ranking of the models using ProQ (Wallner and

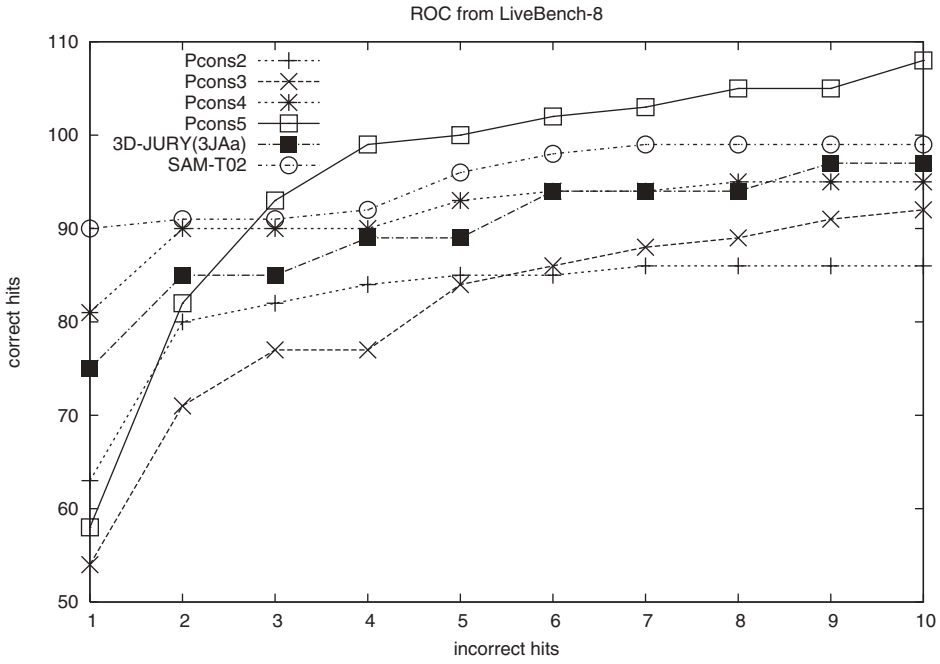

Fig. 3. Receiver operator characteristics on all targets from LiveBench-8 for the different Pcons methods and a 3D-Jury method (3D-JuryA-all). The evaluation was done using MaxSub with 0.1 as cutoff for correct model. As a reference the best performing independent server is also included (data taken from http://bioinfo.pl/LiveBench/).

Table 2. Comparison between Pcons5 and Pmodeller5 in CASP6

	Pcons5	Pmodeller5
All (59)	3	10
Comparative modeling (41)	3	7
Fold recognition and new fold (18)	0	3

The numbers represent the number of times the method is significantly better than the other, as measured by a difference >0.05 in GDT_TS.

Elofsson, 2003). This was also the main conclusion from comparisons of the previous versions of Pcons and Pmodeller (Wallner *et al*., 2003).

Comparison to servers used by Pcons5 and Pmodeller5

The servers used by Pcons5 and Pmodeller5 are listed in Table 3 together with the number of times it was selected by either Pcons5 and Pmodeller5. RAPTOR (Xu *et al*., 2003) is the most frequently selected method both by Pcons5 and Pmodeller5. The main differences in server preference are that Pmodeller5 selects Eidogen-SFST (http://www.eidogen.com) and INBGU (Fischer, 2000) models more frequently than Pcons5, while Pcons5 seems to prefer mGenTHREADER (Jones, 1999) and BasC (Ginalski *et al*., 2004) models. In fact, all four INBGU models selected by Pmodeller5 significantly increase the model quality compared with the Pcons5 model for the same targets. Four of the selected Eidogen-SFST models are also better than the corresponding Pcons5 model.

Table 3. Servers used by Pcons5 and Pmodeller5

Server	Pcons5	Pmodeller5
SAM-T02	—	1
FUGUE3	6	4
SUPFAM_PP	1	—
FFAS03	5	7
BasC	7 (2)	2 (1)
PDB–BLAST	—	2
ORFEUS	5 (1)	4
mGenTHREADER	12	2
BLAST	2	2
3D-PSSM	1	2
RAPTOR	16	15 (1)
PROSPECT2	3	1
Eidogen-SFST	2	12 (4)
INBGU	1	5 (4)
ARBY	—	1
total	61 (3)	60 (10)

The individual servers used as input to Pcons5 in CASP6 and the number of times a particular server was ranked highest by Pcons5 and Pmodeller5, respectively. There were 64 official targets; Pcons5 produced results for 61 and Pmodeller5 for 60. The values within parentheses is the number of models that are significantly improved.

For each group an 'average rank' was also calculated using the following formula:

$$\frac{N}{\sum 1/\mathrm{Rank}},$$

where N is the number of targets and Rank is the number of predictions that are >0.05 GDT_TS units better than the current model.

Table 4. Results from CASP6—average rank

Group	Average rank	Top (%)	Better (%)	Worse (%)
Ginalski	1.4	62	83	1
Skolnick-Zhang	1.8	45	75	2
KOLINSKI-BUJNICKI	2.0	39	80	1
GeneSilico-Group	2.2	37	76	3
BAKER	2.2	33	82	2
CBRC-3D	2.3	33	64	6
CHIMERA	2.4	28	70	2
TOME	2.5	30	60	11
BAKER-ROBETTA_04	2.7	24	64	3
SAM-T04-hand	2.7	25	70	2
SBC-Pmodeller5	2.8	24	69	5
Jones-UCL	2.8	23	71	2
MCon	2.8	24	66	7
ACE	3.0	23	61	8
SBC	3.0	22	67	5
CMM-CIT-NIH	3.0	26	53	37
honiglab	3.1	25	52	32
ZHOUSPARKS2	3.2	22	57	6
BAKER-ROBETTA	3.2	22	62	6
LTB-Warsaw	3.2	21	56	14
Eidogen-EXPM	3.2	20	59	11
VENCLOVAS	3.3	26	39	61
Sternberg	3.3	16	68	1
zhousp3	3.3	20	55	7
CAFASP-Consensus	3.4	17	62	6
FISCHER	3.4	15	68	2
UGA-IBM-PROSPECT	3.5	18	52	6
CaspIta	3.6	16	52	13
Eidogen-SFST	3.7	16	54	17
Shortle	3.7	16	51	24
SAMUDRALA	3.7	17	53	10
Eidogen-BNMX	3.7	16	53	16
WATERLOO	3.9	16	56	14
Bilab	4.1	14	41	28
MacCallum	4.1	11	56	16
3D-JIGSAW	4.1	14	54	2
SBC-Pcons5	4.1	13	66	8
LOOPP_Manual	4.1	16	45	24
Rokko	4.1	14	55	13
mGenTHREADER	4.2	15	39	28
rohl	4.3	16	47	25
RAPTOR	4.3	11	49	8
CBSU	4.4	11	46	13
BioInfo_Kuba	4.5	16	41	34
Pan	4.5	15	40	22
agata	4.7	14	41	28
FUGMOD_SERVER	4.8	11	46	29
fams	4.9	11	39	28
hmmspectr3	4.9	14	39	18
famd	4.9	11	41	28

Servers are marked in boldface. Top, the fraction of prediction that was among the best; Better, the fraction of prediction that was significantly better than average and Worse, the fraction that was significantly worse than the average prediction. All groups with an average rank > 5 are removed. The complete table is available at http://www.sbc.su.se/~bjorn/Pcons5/extras/

Table 5. Results from CASP6—sum of GDT_TS

Group	Sum of GDT_TS for first ranked model (rank)		
	All	CM	FR&NF
Ginalski	50.25 (1)	31.62 (1)	18.62 (1)
KOLINSKI-BUJNICKI	46.58 (2)	30.40 (3)	16.18 (3)
BAKER	46.43 (3)	29.01 (17)	17.42 (2)
Skolnick-Zhang	46.07 (4)	30.92 (2)	15.15 (9)
GeneSilico-Group	45.73 (5)	29.92 (4)	15.81 (7)
CHIMERA	45.55 (6)	29.45 (5)	16.10 (4)
CBRC-3D	44.98 (7)	29.12 (14)	15.86 (6)
SAM-T04-hand	44.83 (8)	28.87 (19)	15.96 (5)
Jones-UCL	44.70 (9)	29.22 (10)	15.48 (8)
FISCHER	44.01 (10)	29.24 (9)	14.78 (11)
Sternberg	43.81 (11)	29.14 (13)	14.67 (12)
BAKER-ROBETTA_04	43.26 (12)	28.21 (25)	15.05 (10)
SBC	42.94 (13)	29.20 (11)	13.75 (20)
TOME	42.92 (14)	29.15 (12)	13.78 (19)
MCon	42.89 (15)	28.68 (21)	14.21 (16)
SBC-Pmodeller5	42.80 (16)	29.26 (7)	13.54 (21)
BAKER-ROBETTA	42.69 (17)	28.11 (26)	14.58 (13)
3D-JIGSAW	42.53 (18)	28.68 (21)	13.85 (17)
CAFASP-Consensus	42.43 (19)	29.05 (16)	13.38 (23)
ACE	42.03 (20)	28.84 (20)	13.19 (24)
zhousp3	41.74 (21)	29.00 (18)	12.74 (28)
SBC-Pcons5	41.53 (22)	28.60 (23)	12.93 (27)
RAPTOR	40.38 (29)	28.21 (25)	12.17 (34)
Eidogen-SFST	39.55 (34)	28.60 (23)	10.96 (53)
mGenTHREADER	36.70 (49)	26.46 (43)	10.24 (65)
FUGUE_SERVER	35.67 (61)	26.35 (44)	9.32 (82)
SAM-T02	34.35 (69)	26.02 (49)	8.33 (99)
Sternberg_3dpssm	33.65 (74)	24.59 (64)	9.06 (88)
Arby	30.24 (88)	20.41 (93)	9.82 (69)
FFAS03	25.15 (104)	17.52 (105)	7.63 (111)

Servers are marked in boldface, servers in italics are used in Pcons5. The sum of the GDT_TS score for the first ranked model from each group, for All, CM and FR&NF.

For a group that always makes (one of) the best predictions this average rank will be 1, whereas if a group always makes the worst prediction the average rank will be identical to the number of groups in the comparison. One advantage with this measure compared with a simple average is that it is less sensitive to one (or a few) bad prediction. In addition, the number of times a certain method did one of the best predictions (i.e. no prediction had a GDT_TS 5 units better) and the number of times a prediction was better or worse than average were also calculated.

In Table 4 the performance of all the groups participating in CASP6 sorted by average rank is shown. Unfortunately not all servers used by Pcons5 participated in CASP6, e.g. the INBGU server and some servers hosted by bioinfo.pl only participated in CAFASP4. However, according to the CAFASP4 evaluation the best independent server was Eidogen-EXPM. Thus, the servers only participating in CAFASP4 could be expected to be ranked slightly below Eidogen-EXPM.

Pmodeller5 shows the highest average rank of all servers. Pcons5 performs significantly worse compared with Pmodeller5. One of the servers used by Pcons5, Eidogen-SFST, is actually ranked slightly higher than Pcons5. Even though it is disappointing that Pcons5 is not ranked higher than Eidogen-SFST, it is still impressive that the ProQ re-ranking managed to make Pmodeller5 the best server.

The sum of the GTD_TS for the first ranked model from each group was also used to measure performance (Table 5). Here,

Pcons5 performs better than the best individual server, in particular for the harder targets. Pmodeller5 is better than Pcons5 for both hard and easy targets, but the real improvement is observed for the easy targets, where it performs almost as well as the best manual groups.

One advantage with a consensus method is that it most often selects a model that is better than the average model and seldom a model that is worse than average. However, even though it usually makes a good choice, it often misses the best possible model, i.e. the best model is ranked high but not at the top. Here, a more specific method or energy function can be used to evaluate the top hits. In our case, by using ProQ we increased the number of top hits produced from 13 to 24% of all targets.

CONCLUSIONS

We have developed a new version of Pcons, Pcons5, that uses structural evaluation and reliability assessment of the server score on top of the consensus analysis. This add-on improves the performance by 10% compared with only using consensus on the LiveBench-8 dataset. The performance compared with previous versions is also slightly higher. The new version is easy to update and works even for 'unseen' methods.

In addition to the development of Pcons5 a new version of Pmodeller has also been developed, Pmodeller5. This method uses ProQ to evaluate the best hits from Pcons5. Pmodeller5 was among the best servers in CASP6 and consistently ranked higher than Pcons5.

Pcons5 is the first Pcons version available as a standalone program from: http://www.sbc.su.se/~bjorn/Pcons5. It should be easy to implement in local meta-servers. The model evalution module in Pmodeller, ProQ, is also available as a standalone program from: http://www.sbc.su.se/~bjorn/ProQ

ACKNOWLEDGEMENTS

This work was supported by grants from the Swedish Natural Sciences Research Council and by a grant from the Graduate Research School in Genomics and Bioinformatics.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI–BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Berman,H.M. *et al*. (2000) The Protein Data Bank. *Nucleic Acids Res*., **28**, 235–242.

Bujnicki,J.M. *et al*. (2001) Livebench-2: large-scale automated evaluation of protein structure prediction servers. *Proteins*, **45** (Suppl. 5), 184–191.

Bujnicki,J.M. *et al*. (2001) Structure prediction meta server. *Bioinformatics*, **17**, 750–751.

Cristobal,S. *et al*. (2001) A study of quality measures for protein threading models. *BMC Bioinformatics*, **2**, 5.

Fischer,D. (2000) Hybrid fold recognition: combining sequence derived properties with evolutionary information. *Pac. Symp. Biocomput*., 119–130.

Fischer,D. (2003) 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins*, **51**, 434–441.

Ginalski,K. *et al*. (2003) 3D-jury: a simple approach to improve protein structure predictions. *Bioinformatics*, **19**, 1015–1018.

Ginalski,K. *et al*. (2005) Practical lessons from protein structure prediction. *Nucleic Acids Res*., **33**, 1874–1891.

Ginalski,K. *et al*. (2003) ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic Acids Res*., **31**, 3804–3807.

Ginalski,K. *et al*. (2004) Detecting distant homology with meta-BASIC. *Nucleic Acids Res*., **32**, (Web Server issue) W576–W581.

Gough,J. and Chothia,C. (2002) SUPERFAMILY: HMMs representing all proteins of known structure. SCOP sequence searches, alignments and genome assignments. *Nucleic Acids Res*., **30**, 268–272.

Jones,D.T. (1999) GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol*., **287**, 797–815.

Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol*., **292**, 195–202.

Karplus,K. *et al*. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.

Karplus,K. *et al*. (2003) Combining local-structure, fold-recognition and new fold methods for protein structure prediction. *Proteins*, **53** (Suppl. 6), 491–496.

Kelley,L.A. *et al*. (2000) Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol*., **299**, 523–544.

Lee,B. and Richards,F.M. (1971) The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol*., **55**, 379–400.

Lundström,J. *et al*. (2001) Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein Sci*., **10**, 2354–2362.

Moult,J. *et al*. (2003) Critical assessment of methods of protein structure prediction (CASP)-round V. *Proteins*, **53** (Suppl. 6), 334–339.

Rychlewski,L. *et al*. (2003) Livebench-6: large-scale automated evaluation of protein structure prediction servers. *Proteins*, **53** (Suppl. 6), 542–547.

Rychlewski,L. *et al*. (2000) Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Sci*., **9**, 232–241.

Sali,A. and Blundell,T.L. (1993) Comparative modelling by statisfaction of spatial restraints. *J. Mol. Biol*., **234**, 779–815.

Shi,J. *et al*. (2001) Fugue: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol*., **310**, 243–257.

Siew,N. *et al*. (2000) Maxsub: an automated measure to assess the quality of protein structure predictions. *Bionformatics*, **16**, 776–785.

Wallner,B. and Elofsson,A. (2003) Can correct protein models be identified? *Protein Sci*., **12**, 1073–1086.

Wallner,B. *et al*. (2003) Automatic consensus-based fold recognition using Pcons, ProQ and Pmodeller. *Proteins*, **53** (Suppl. 6), 534–541.

Xu,J. *et al*. (2003) RAPTOR: optimal protein threading by linear programming. *J. Bioinform. Comput. Biol*., **1**, 95–117.

Zemla,A. *et al*. (1999) Processing and analysis of CASP3 protein structure predictions. *Protein*, (Suppl. 3) pp. 22–29.

Zhang,Y. and Skolnick,J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**, 702–710.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4255–4262
doi:10.1093/bioinformatics/bti684

Gene expression

Noise and rank-dependent geometrical filter improves sensitivity of highly specific discovery by microarrays

Hassan M. Fathallah-Shaykh

Department of Neurological Sciences, Section of Neuro-Oncology, Rush University Medical Center, 1725 West Harrison Street, Chicago, IL 60612, USA

Received on July 27, 2005; revised on September 17, 2005; accepted on September 20, 2005
Advance Access publication September 22, 2005

ABSTRACT

Summary: MASH is a mathematical algorithm that discovers highly specific states of expression from genomic profiling by microarrays. The goal at the outset of this analysis was to improve the sensitivity of MASH. The geometrical representations of microarray datasets in the 3D space are rank-dependent and unique to each dataset. The first filter (F1) of MASH defines a zone of instability whose F1-sensitive ratios have large variations. A new filter (Fs) constructs in the 3D space rank-dependent lower and upper-bound contour surfaces, which are modeled based on the geometry of the unique noise intrinsic to each dataset. As compared with MASH, Fs increases sensitivity significantly without lowering the high specificity of discovery. Fs facilitates studies in functional genomics and systems biology.

Contact: hfathall@rush.edu

Supplementary information: http://www.rushu.rush.edu/neurosci/Fathallah.html

INTRODUCTION

The genomes of several organisms have recently been sequenced and the enthusiasm about the potential of microarrays has been intense (Schena *et al*., 1995; Lockhart *et al*., 1996; DeRisi *et al*., 1997). Microarray studies are increasingly being used to explore biological causes and effects and even to diagnose diseases; however, their data are very noisy and the patterns of expression and molecular signatures of microarrays may not be reproducible (Kothapalli *et al*., 2002; Ntzani and Ioannidis, 2003; Tan *et al*., 2003).

MASH is a mathematical algorithm that yields highly specific states of genetic expression (up or downregulation) from the genome-scale profiling of two samples (Fathallah-Shaykh *et al*., 2004). The term 'highly specific' refers to the high specificity of states of genetic expression discovered by MASH. Specifically, the false discovery rates of MASH and MIDAS in same-to-same comparisons using the 19K microarrays are 1/192 000 and 1347/192 000 measurements, respectively. MASH specificity is significantly better, but its sensitivity is equal to MIDAS. My goals at the outset of this analysis were to better understand the noise and to generate a new algorithm that significantly improves the sensitivity of MASH without lowering its high specificity. Interestingly, sensitivity is not only dependent on the analytical method but also on the quality of the dataset (Fathallah-Shaykh *et al*., 2004).

MATERIALS AND METHODS

Microarrays

Normal brain RNA is obtained by pooling RNA from human occipital lobes harvested from four individuals with no known neurological disease whose brains are frozen <3 h post mortem. Tumor RNAs, isolated from 35 surgical gliomas, 10 surgical meningiomas and 6 cultured glial cell lines, are profiled as compared with aliquots from the same normal brain RNA (Fathallah-Shaykh *et al*., 2003, 2002, 2004; Fathallah-Shaykh, 2005a). The quality of RNA is assayed by gel electrophoresis, only high quality RNA is processed. Microarray experiments use 1.7K (1920 genes) and 19K (19 200 genes) microarrays containing cDNAs spotted in duplicates (Ontario Cancer Institute, Ontario, Canada). The design, which includes dye swapping as described elsewhere, generates four replicate measurements per gene and sample (Fathallah-Shaykh *et al*., 2003, 2002). Each slide contains two replicate adjacent spots. The Cy3/Cy5 design generates two ratios. The Cy5/Cy3 design generates two additional ratios. The total is four replicate ratios with dye-swapping. RNA used in spike-in experiments is transcribed from the same *Arabidopsis* cDNA spotted on microarray slides.

Analysis

The mathematical analysis is performed using functions written in Matlab (Mathworks, Natick, MA). Fs is freely available to academics for non-commercial use. To obtain executable software, please send a request by e-mail to Fs_request@rush.edu.

RESULTS

Definitions

The state of genetic expression of a spot in sample A versus sample B assayed by cDNA arrays is measured by the ratio of the background-subtracted intensities of sample A/background-subtracted intensities of sample B. A ratio >1 (log2 > 0) implies upregulation of the gene in sample A as compared with B. The terms 'genes', 'spots', 'symmetrical', 'rank', and 'spot order' are defined using the 1.7K arrays; there terms are also applicable to other microarrays. The 1.7K microarray contains 1920 cDNAs or controls, here referred to as *genes*, spotted in duplicated to a total of 3840 *spots*. The term *symmetrical* refers to the two images, corresponding to the Cy3 and Cy5 fluorescent dyes, generated from a single microarray slide. Background-subtracted spot intensities are sorted in ascending order to assign a *rank* to every spot. For instance, a spot whose rank is 3000 has a higher background-subtracted spot intensity than all spots whose ranks are <3000. A microarray *Spot Order* (SO) is a listing of its spots sorted by

their ranks. A cDNA spotted slide generates two spot orders, SO1 and SO2, which correspond to Cy3 and Cy5, respectively.

Dye swapping refers to experiments where the Cy3 and Cy5 dyes are reversed between the two samples; they are performed to annul confounding variables introduced by heterogeneous fluorescence of the Cy5 and Cy3 molecules. Each microarray slide yields of a set of symmetrical Cy3/Cy5 images that generate two replicate ratios. Each dye swapping dataset generates four replicate ratios.

The datasets and rationale

The true negative datasets compare the same pool of brain RNA with itself (same-to-same). The goal of the same-to-same comparisons is to collect experimental noise (technical artifacts) independent of biological heterogeneity. In this design, normalized expression ratios $\neq 1$ ($\log 2 \neq 0$) are false positive (noise) because the Cy3/Cy5 symmetrical images contain identical genetic information. The artifactual measurements may be caused by several factors including slide-to-slide differences, variations in the reverse transcription reactions, hybridization, labeling and laser. The same-to-same comparisons include 18 and 20 experiments that generate a total of 9 and 10 dye swapping datasets using the human 1.7K and 19K microarrays, respectively. The experiments are paired by consecutive order. The goal of the new algorithm is to filter the largest number of same-to-same expression ratios originating from technical noise. Ideally, as compared with MASH the new algorithm should discover a smaller number of genes as being differentially expressed in the same-to-same design.

The 1.7K microarray includes 64 genes of *Arabidopsis* cDNA. The true positive datasets include four sets of spike-in dye swapping experiments using 1.7K microarrays, where 1 ng of *Arabidopsis* RNA is added to one sample but not the other. In this design, all 64 genes of *Arabidopsis* cDNA serve as true positives. The sensitivity of MASH is 26/64 [41%, (Fathallah-Shaykh *et al*., 2004)]. Ideally, the new algorithm is expected to discover all 64 *Arabidopsis* genes as being differentially expressed.

MASH summary

MASH includes two filters, F1 and F2. A spot is sensitive to F1 if *both* its symmetrical ranks in SO1 and SO2 are less than the Cutoff Rank (CR). To be resistant to F1, either Cy3 or Cy5 images of the spot must contain enough signals such that at least one of the symmetrical ranks is larger than the CR. The latter is computed empirically from the slopes of the ranking curves (Fathallah-Shaykh *et al*., 2004).

The second filter (F2) of MASH consists of two rules. The first Rule (F2a or f_4) necessitates that all four replicate ratios consistently show up or downregulation; i.e. all four replicate log2(ratios) > 0 or all four < 0. The second Rule of F2 (F2b) necessitates that all four replicate F2b-resistant log2(ratios) must be outside the interval of $\pm 3 *$ the largest standard deviations of all F1-resistant log2(ratios). Genes sensitive to either f_4 or F2b are filtered by transforming their mean log2(ratio) to 0.

Heterogeneous geometrical distributions of noise

The same-to-same datasets consist of errors in measurement generated by technical noise. Figure 1a and d plot the distributions of same-to-same 19K and 1.7K microarray datasets in the 3D space defined by (1) ranks in SO1 (x-axis), (2) ranks in SO2 (y-axis), and (3) log2(ratios) (z-axis), respectively. If the experimental system generates *no* noise, the z-axis coordinates would all be equal to 0. Large positive or negative log2(ratios) reflect large errors (red arrows). The findings reveal that the distributions of noise in the 3D space are heterogeneous because each microarray dataset generates its unique geometrical structure, which differs between datasets. Specifically, the z-axis variations of log2(ratios) about 0 are rank-dependent and unique to a specific dataset (Fig. 1, black arrows).

Zone of instability

Figure 1a and 1d also reveal that the distributions in the 3D space include zones of instability where the log2(ratios) have large absolute values (red arrows). Figure 2a plots the projections of the distribution of noise of Figure 1a onto the 2D space defined by ranks in SO1 (x-axis) and log2(ratios) (y-axis). Figure 2b plots the projections of the distribution of noise of Figure 1a onto the 2D space defined by ranks in SO2 (x-axis) and log2(ratios) (y-axis), respectively. To visualize the rank-dependent behavior of noise, the spots are colored by their ascending ranks (Fig. 2). As in Figure 1a, Figure 2a and b also reveal that the distributions of noise include zones of instability, where log2(ratios) have large values (large errors; red arrows).

Figures 1 and 2 suggest that the zone of instability is dependent on the ranks in both SO1 and SO2. For example, the spots of Figure 2a and b whose ranks in SO1 and SO2 are *both* <10 000 generate unstable or large log2(ratios). The number 10 000 is unique to the dataset plotted in Figure 2; other datasets may be associated with different ranks. Since the first filter of MASH (F1) excludes spots whose ranks in SO1 and SO2 are both smaller than the CR, the question arises whether F1 defines the zone of instability.

A spot is resistant to F1 if either one of its ranks in either SO1 or SO2 is larger than the CR. To study the effects of F1 on the zone of instability, it is applied to filter the data shown in Figures 1a, d, 2a and b. Figure 1b and e plot the log2(ratios) of F1-resistant spots of Figure 1a and d, respectively. Figure 2c and d plot the log2(ratios) of F1-resistant spots of Figure 2a and b, respectively. Figure 2e reveals that the standard deviations of F1-sensitive log2(ratios) (blue; spots filtered by F1) are 5–10 folds larger than F1-resistant log2(ratios) (green; spots not filtered by F1). The findings support the conclusions that (1) spots whose symmetrical ranks are both less than the CR generate a zone of instability containing large errors of measurement [large absolute log2(ratios)], and (2) F1 filters the zone of instability.

Mathematical modeling of noise

Next, I study the effects of F1 on the distributions of different-to-different datasets. As in the geometrical distributions of same-to-same datasets, the distributions of the different-to-different datasets (1) are heterogeneous between datasets and rank-dependent, and (2) include F1-sensitive zones of instability (Fig. 3). However, unlike the same-to-same design, where any log2(ratio) different than 0 is false positive, the distinction between true and false positive ratios in different-to-different comparisons is not evident.

Figures 2 and 3 reveal that F1 deletes the zone of instability, which includes a large portion of the data. The goal of the next section is to devise a method that models the noise intrinsic within each dataset in such a way that the method (1) is applicable to all datasets despite their heterogeneity (Fig. 1), and (2) contours the zone of instability instead of deleting it.

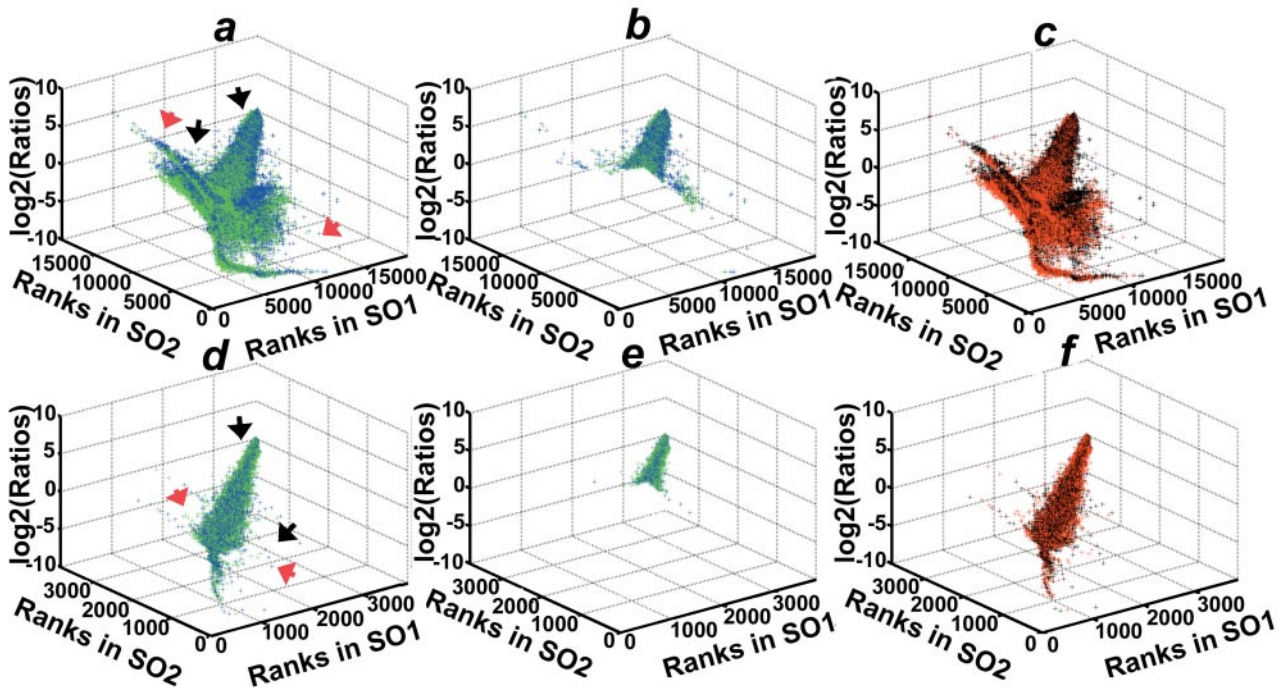

Fig. 1. Dataset-specific geometry and the geometrical distributions of f_4-senstive spots replicate the distributions of all the spots. (**a**) and (**d**) plot the log2(ratios) of all spots of 19K same-to-same and 1.7K same-to-same datasets, respectively. The space is defined by the ranks in SO1, *x*-axis, ranks in SO2, *y*-axis, and log2(ratios), *z*-axis. (**b**) and (**e**) plot the F1-resistant spots of (a) and (d), respectively. Black arrows point to the variability of log2(ratios) at different ranks. Red arrows point to large errors in measurement generated by log2(ratios), whose absolute values are large. Blue and green correspond to the data of the dye swapping experiments. (**c**) and (**f**) plot the log2(ratios) of the f_4-sensitive spots of (a) and (d), respectively. Black and red correspond to the data of the dye swapping experiments. The geometrical distributions of (c) and (f) replicate those of (a) and (d), respectively. The 3D Matlab Figures are presented as Supplementary information.

Geometrical filter Fs

The filter f_4 (F2a) was devised in the glioma study (Fathallah-Shaykh *et al*., 2002). A gene is resistant to f_4 if its four replicate log2(ratios) are all positive or all negative (consistently showing up or downregulation). A gene is sensitive to f_4 if *all* four replicate log2(ratios) are *not* of the same sign. Because the false negative rate of f_4 is only 1.6%, the predominant majority of f_4-sensitive spots are false positive or noise.

Figures 1c, 1f and 3c plot the f_4-sensitive spots of the datasets shown in Figures 1a, 1d and 3a. Interestingly, the findings reveal that the geometrical distribution of f_4-sensitive spots (noise) replicates the unique geometrical distribution of all the spots in the dataset. This is not surprising considering that (1) in same-to-same experiments any log2(ratio) different than 0 is false positive, and (2) only a small fraction of different-to-different datasets is truly differentially expressed. Most importantly, because the geometrical structures/distributions created by f_4-sensitive noise are independent of the spots that are truly differentially expressed, they will serve as a platform for constructing a new filter.

Each dataset generates two geometrical structures in the 3D spaces generated by the log2(ratios) of (1) all the spots (Figs 1a, d and 3a), and (2) the f_4-sensitive spots (spots filtered by f_4; Figs 1c, f and 3c). The distribution of the latter represents noise intrinsic to each dataset. In reality, the two distributions are interwoven in the same space. However, for practical purposes, the spaces/geometrical distributions of all spots in the dataset and f_4-sensitive spots will be referred to as *G* and *G*4, respectively (Fig. 4a and b).

The rationale of the new filter (Fs) is based on the idea that a method that filters all f_4-sensitive spots in *G*4 (*G*4, see Fig. 4b) will lead to a high degree of certainty that the unfiltered spots of *G* are true (Fig. 4a). Fs consists of upper and lower bound contour surfaces that are patterned based on the geometrical structure of *G*4. These contour surfaces will set the upper and lower bound *z*-axis limits at specific ranks such that any spot of *G* that maps outside these bounds is true to a high degree of certainty.

The geometrical structures in *G* and *G*4 consist of spots whose *x*-, *y*- and *z*-axis coordinates are the ranks in SO1, SO2 and log2(ratio), respectively (Figs 1–4). For every spot *k* of coordinates (x_k, y_k, z_k) in *G* (Fig. 4a), the algorithm applies a square-column (C_k) within *G*4 such that (1) the column is parallel to the *z*-axis, and (2) the center of the square maps at the coordinates (x_k, y_k, z_k) (Fig. 4b). Since the log2(ratios) of the spots present within each column reflect the local variability of noise at ranks x_k and y_k, they are *isolated* and their standard deviation and the means of the positive and negative log2(ratios) are computed (Fig. 4c and d). The algorithm increases the width of the square column until it includes a minimum of 100 spots. The optimal number of spots isolated by C_k is varied and computed empirically when the algorithm is completed; 100 spots optimize the false discovery rate.

Let sd be the standard deviation of all log2(ratios) isolated by column C_k. Let μ_p and μ_n be the means of their positive and

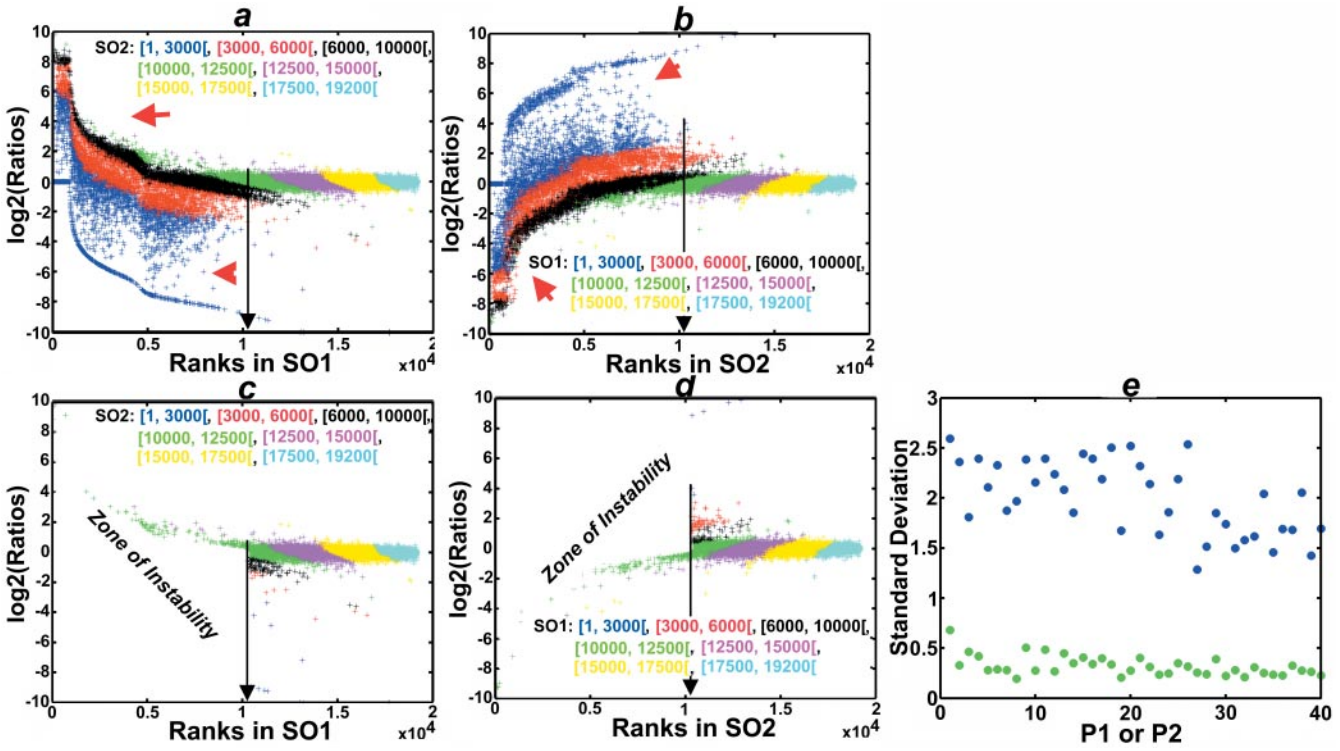

Fig. 2. Zone of instability. The dataset is the same as the one shown in Figure 1a–c. (**a**) is a plot of the log2(ratios), *y*-axis, versus ranks in SO1. In (a) the spots whose ranks in SO2 are within the following intervals, [1, 3000[, [3000, 6000[, [6000, 10000[, [10000, 12500[, [12500, 15000[, [15000, 17500[, and [17500, 19200[, are colored in blue, red, black, green, magenta, yellow, and cyan, respectively. (**b**) is a plot of the log2(ratios), *y*-axis, versus ranks in SO2. In (b) the spots whose ranks in SO1 are within the following intervals, [1, 3000[, [3000, 6000[, [6000, 10000[, [10000, 12500[, [12500, 15000[, [15000, 17500[, and [17500, 19200[, are colored in blue, red, black, green, magenta, yellow, and cyan, respectively. (a) and (b) reveal two zones, a zone of instability whose log2(ratios) have wide variations from the true log2(ratio) = 0. Black arrows point to the CR. Red arrows point to log2(ratios) whose absolute values are large (large errors). (**c**) and (**d**) plot the F1-resistant spots of (a) and (b), respectively. Each 19K array includes two slides, P1 and P2. (**e**) plots the standard deviations of F1-resistant (green) and F1-sensitive (blue) log2(ratios) of the datasets of either P1 or P2.

negative log2(ratios), respectively (Fig. 4d). At every spot of coordinates (x_k, y_k, z_k) in G, the upper and lower limits are set at spots in G having the following coordinates:

(1) An upper-bound limit at $(x_k, y_k, \mu_p + n * \text{sd})$.

(2) A lower-bound limit at $(x_k, y_k, \mu_n - n * \text{sd})$.

The term n is a variable (Fig. 4e). The upper and lower bound limits computed from all the spots of G assemble the upper and lower bound surfaces (Fig. 5). Fs applies the following rules:

- A spot is filtered if its log2(ratio) maps within the 3D space bound by the upper and lower contour surfaces. Alternatively, a spot is resistant if its log2(ratio) maps above the upper-bound surface or below the lower-bound surface.
- A gene is resistant if *all* of its four replicate spots are resistant to the rule above. The log2(ratio) of a sensitive gene is transformed to 0.

At this stage, I am making the assumption of the existence of contour surfaces such that (1) the 3D space, limited by the upper- and lower-bound surfaces includes the overwhelming majority of noise, and (2) the log2(ratios) that map above or below the upper- and lower-bound surfaces, respectively, are true. The *z*-axis positions of the contour surfaces and the 3D space between them are dependent on (1) n, (2) μ_p, μ_n and sd. Interestingly, as expected, the *z*-axis coordinates and 3D space between the contour surfaces are larger over the zone of instability (Fig. 5, arrows) because of the large standard deviations of its log2(ratios) (Fig. 2e).

In theory, the variable n determines both specificity and sensitivity. For example, if n is 'very large', one expects sensitivity to be low because (1) the *z*-axis limits of the contour surfaces will also be large, and (2) the space between the contour surfaces will include all log2(ratios). However, if n is 'small', specificity could be low because the contour surfaces may not include all the noise. The goal is to find a value of n that yields optimal specificity and sensitivity. Specificity will be assayed as 1− the false discovery rate of Fs in same-to-same comparisons. Ideally, Fs should filter all same-to-same log2(ratios). Sensitivity will be assayed by percent discovery of *Arabidopsis* genes in different-to-different spike-in experiments (see above). Ideally, Fs should discover all the *Arabidopsis* genes.

Optimizing n and comparing the sensitivity and specificity of Fs to MASH

The goal is to compare the specificity of MASH to Fs while varying n within the interval [2,6]. MASH consists of F1 + f_4 (F2a) + F2b. The false discovery rate is computed from nine 1.7K and ten 19K

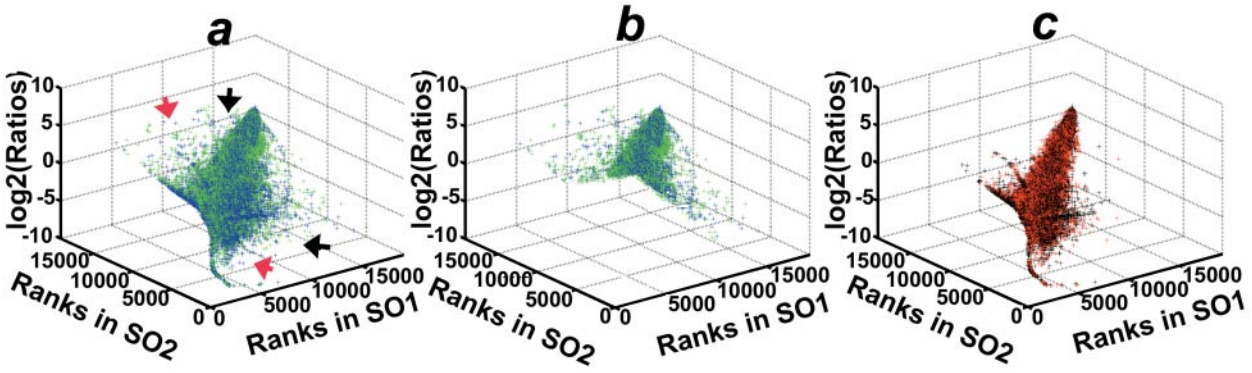

Fig. 3. Geometrical distribution of different-to-different datasets. (**a**) plots the log2(ratios) of all spots from the profiling of a meningioma RNA versus normal brain using the 19K microarrays. The space is defined by the ranks in SO1, *x*-axis, ranks in SO2, *y*-axis, and log2(ratios), *z*-axis. Black arrows point to the variability of log2(ratios) at different ranks. Red arrows point to large measurements generated by log2(ratios), whose absolute values are large. Blue and green correspond to the data of the dye swapping experiments. (**b**) plots the F1-resistant spots of (a). (**c**) plots the log2(ratios) of the f_4-sensitive spots of (a). Black and red correspond to the data of the dye swapping experiments. The geometrical distribution of (c) replicates that of (a).

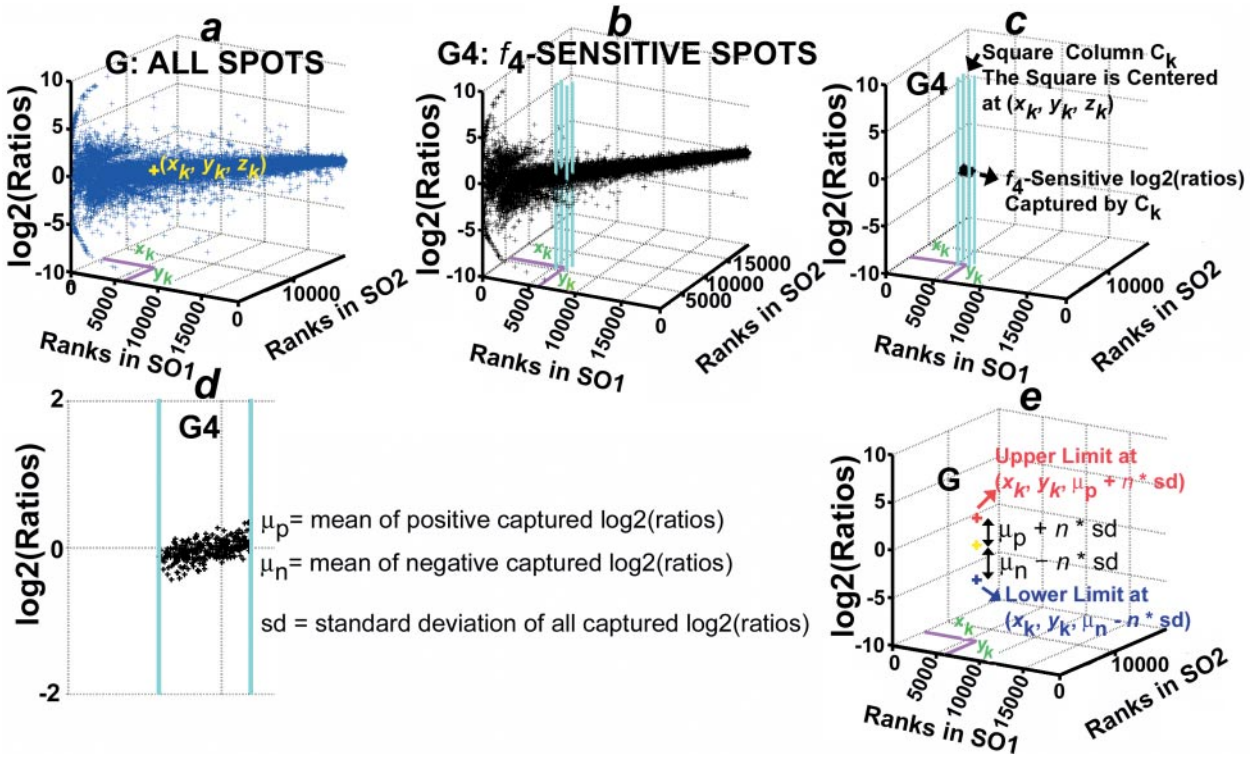

Fig. 4. Computing the upper and lower limits of noise at every spot in the datasets. (**a**)–(**e**) are cartoons that illustrate the construction of upper and lower bound surfaces. (a) is the same dataset shown in Figure 3a; (b) plots the f_4-sensitive noise of (a) (Fig. 3c). For every spot k of coordinates (x_k, y_k, z_k) in the space G, defined by all the spots of the dataset (a, yellow), the algorithm constructs a square column, C_k, in the space $G4$ of its f_4-sensitive spots (b–d) such that (1) the columns are parallel to the z-axis, and (2) the square is centered at the spot (x_k, y_k, z_k). The cyan lines of (b) and (c) are located at the four corners of the square column. The algorithm isolates the log2(ratios) within the square column C_k to compute the μ_p, μ_n, and sd (d). The coordinates of the upper and lower limits at spot k are $(x_p, y_p, \mu_p + n * \text{sd})$ and $(x_k, y_k, \mu_n - n * \text{sd})$, respectively. The term n is a variable.

same-to-same experiments (Fig. 6a and b). The results reveal that the specificity of Fs alone is as high as MASH for $n \geq 3$ (Fig. 6a and b and Table 1).

Sensitivity is assayed by the percentage of *Arabidopsis* genes discovered from the best of four replicate spike-in experiments, where 1 ng Arabidopsis RNA is added to one RNA sample but not the other (Fig. 6c and d). The following filter combinations are applied: (1) Fs alone, and (2) F1 and Fs (Fig. 6c). As compared with MASH, Fs improves the best sensitivity from 41 to 91% at $n = 3$. However, adding F1 and f_4 to Fs lowers the sensitivity to 86% at $n = 3$. In addition, Figure 6d demonstrates that, as compared with MASH, the increase in sensitivity of Fs at $n = 3$ is statistically

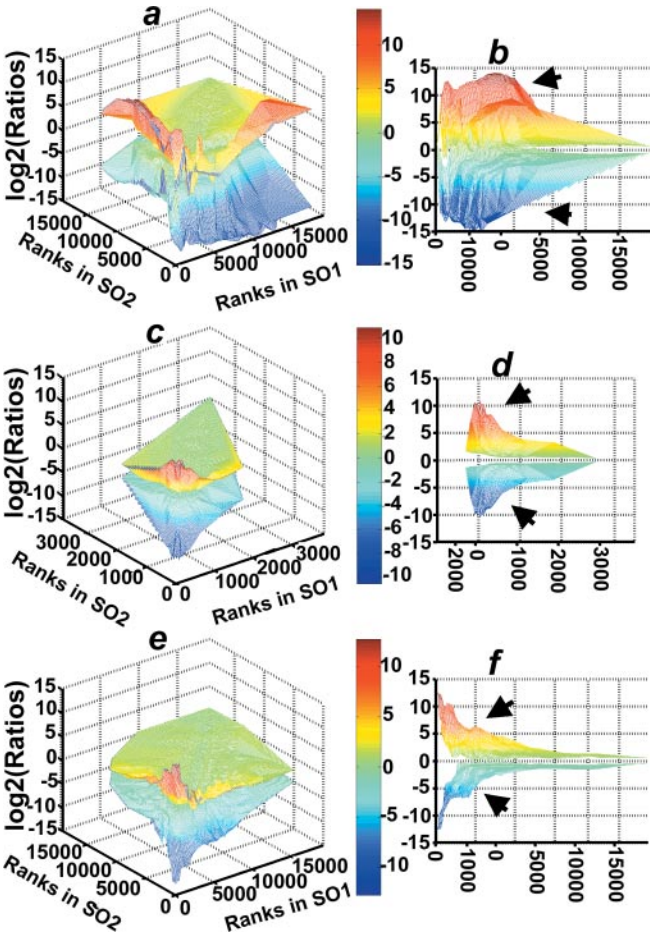

Fig. 5. Fs constructs noise- and rank-dependent contour surfaces. The upper bound contour surface (green to red) is constructed by connecting all the upper bound limits generated by Fs (see cartoon of Fig. 4e). The lower bound contour surface (green to blue) is constructed by connecting all the lower bound limits generated by Fs (see cartoon of Fig. 4e). (**a**), (**c**) and (**e**) are wireframe mesh plots of the lower (green to blue) and upper-bound (green to red) contour surfaces of the datasets of Figures 1a, 1d and 3a, respectively. (**b**), (**d**), and (**f**) show different views of (a), (c), and (e), respectively. Arrows point to large-axis coordinates of the contour surfaces over the zone of instability.

significant in all four replicate spike-in experiments ($P = 0$). These findings support the conclusion that Fs at $n = 3$ significantly improves sensitivity without lowering the high specificity of MASH. Receiver Operating Characteristics (ROC) is the standard approach to evaluate the sensitivity and specificity of diagnostic procedures (Swets and Pickett, 1992). MASH and Fs at $n = 2$, 2.5 and 3 generate the empiric ROC areas of 0.703, 0.96, 0.96 and 0.95, respectively. The accuracy rates are 99.8%, 99.9%, 100% and 100%, respectively (Table 1).

Fs is also applied to analyze four same-to-same datasets of Rosenzweig *et al*. (2004). Each dataset includes 710 'genes' spotted in duplicates to a total of 1420 spots. The false discovery rates per 2840 genes are 4, 2 and 0 at $n = 2$, 4 and 3–6, respectively. The findings demonstrate that Fs is also effective in filtering the noise of datasets acquired in independent laboratories.

DISCUSSION

The findings reveal that microarray datasets are heterogeneous (Figs 1–3). This heterogeneity is reflected by their geometrical structured in the 3D space, whose axes are the ranks in SO1 and SO2 and the log2(ratios). Specifically, this geometry/distribution (1) is unique to each dataset (Figs 1 and 3), (2) includes a zone of instability, whose F1-senstitive spots generate large errors (Fig. 1–3), and (3) displays rank-dependent variability of log2(ratios). Interestingly, the f_4-sensitive spots intrinsic to each dataset (1) replicate the geometry/distribution of all spots in the dataset (Figs 1 and 3) and (2) are independent of the genes that are differentially expressed. This new algorithm constructs rank-dependent upper- and lower-bound contour surfaces that are patterned based on the geometrical structure of f_4-sensitive spots (Fig. 4).

The zone of instability is generated by ratios whose ranks are both less than the CR. This finding is consistent with the results of Baggerly *et al*. (2001) who report that ratios computed from spots containing a small amount of total signal are highly variable, whereas ratios derived from spots containing large amount of total signal are fairly stable. The zone of instability (Figs 1–3) may also explain the results of Tan *et al*. (2003) who demonstrate poor reproducibility of states of genetic expression across different platforms.

Sensitivity is a function of measurable quality parameters; specifically, it is negatively correlated with the Noise Factor (Fathallah-Shaykh *et al*., 2004). In addition, poor data quality has a negative impact on the efficient detection of low-level regulated genes (Raffelsberger *et al*., 2003). Specifically, the distributions of false positive ratios vary between datasets; poor quality datasets contain large false positive ratios (Fathallah-Shaykh *et al*., 2004). Therefore, the degree of confidence that a low- or moderate-level expression ratio is true is dependent, not only on the analytical methods, but also on the unique distribution of noise in that specific dataset. Thus, to annul the confounding effects of data quality on sensitivity, the true positives of this study are designed to represent large differentials generated by adding *Arabidopsis* RNA to one sample but not the other. The specificity and sensitivity of the algorithm are optimal at $n = 3$. Values of $n > 3$ yield lower sensitivity (Fig. 6c) and values of $n < 3$ yield lower specificity (Fig 6a and b). The z-axis positions of the contour surfaces are dependent on the standard deviation of the local (neighborhood) noise isolated by the square columns at specific ranks. The theory developed in this paper leads to a test (Fs) that compares the geometrical structures of distributions in the 3D space. Specifically, Fs divides the space into small subspaces (neighborhoods) and constructs contour surfaces whose z-axis variance is based on the local distributions at specific ranks (Fig. 4).

The first filter of MASH, F1, is stochastic; in addition, the position of the CR is computed empirically (Fathallah-Shaykh *et al*., 2004). However, the analysis detailed in this paper generates the mathematical basis for F1; specifically, F1 filters the zone of instability (Figs 1–3). It is not surprising that Fs is more sensitive than F1 (Fig. 6c); specifically, instead of deleting the zone of instability, Fs generates upper- and lower-bound contour surfaces around it (Figs 1, 2 and 5). Fs, MASH and MIDAS were applied to analyze the same datasets. The specificity of Fs at $n = 3$ is similar to MASH (Table 1 and Fig. 6), whose specificity is significantly better than MIDAS (Fathallah-Shaykh *et al*., 2004). MIDAS includes the Locfit

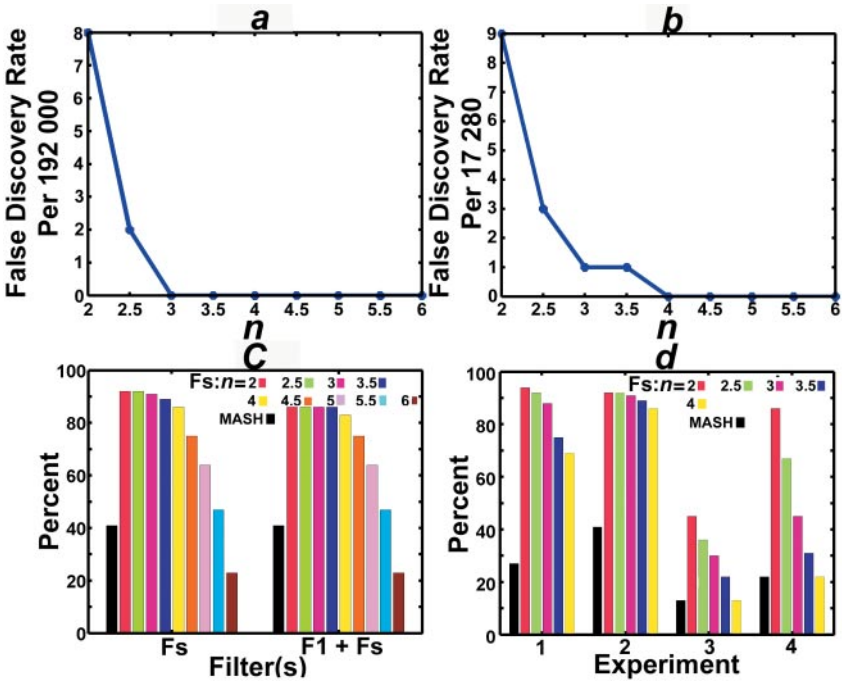

Fig. 6. Fs significantly enhances sensitivity without lowering the high specificity of MASH. (**a**) and (**b**) show the effects of varying n on the false discovery rates of Fs in the analysis of 10 same-to-same 19K and 9 same-to-same 1.7K comparisons, respectively. (**c**) shows the effects of varying n and adding F1 to Fs on the percent discovery of *Arabidopsis* genes from the best of four 1.7K spike-in experiments, where 1 ng *Arabidopsis* RNA is added to one sample but not the other. In (c) n is varied in the interval [2,6] and either Fs or F1 + Fs are applied and compared with MASH (black). (**d**) shows the sensitivity of MASH (black) and Fs at $n =$ 2–4 in all four 1.7K spike-in experiments, where 1 ng *Arabidopsis* RNA is added to one sample but not the other. As compared with MASH, the increase in sensitivity of Fs at $n = 3$ is statistically significant (balanced one-way ANOVA, $P = 0$).

Table 1. Fs at $n = 3$ significantly improves sensitivity without lowering the high specificity of MASH

	Same-to-same false discovery rate		Sensitivity (%)	Empiric ROC Area	Accuracy (%)
	19K	1.7K	1.7K Spike-in	1.7K	1.7K
MASH	1/192 000	1/17 280	41	0.703	99.8
Fs: $n = 2$	8/192 000	9/17 280	92	0.96	99.9
Fs: $n = 2.5$	2/192 000	3/17 280	92	0.96	100
Fs: $n = 3$	0/192 000	1/17 280	91	0.95	100

n is described in the definition of Fs. The same-to-same false discovery rates are computed from 10 dye-swapping 19K and 9 dye swapping 1.7K datasets that compare brain RNA to itself. The false discovery rate of the same-to-same design equals the number of false positive ratios. Thus specificity is measured as 1 – same-to-same false discovery rate. The false discovery rate of Fs at $n = 3$ is similar to MASH. Percent sensitivity refers to the best sensitivity of four replicate spike-in dye swapping 1.7K datasets, where 1 ng of *Arabidopsis* RNA is added to one sample but not the other. Each 1.7K microarray includes 64 *Arabidopsis* genes. The sensitivity of Fs at $n = 3$ is 91%, more than double the sensitivity of MASH. ROC estimates a curve, which describes the inherent tradeoff between sensitivity and specificity of a diagnostic test. The area under the ROC curve is important for evaluating diagnostic procedures because it is the average sensitivity over all possible specificities (Swets, 1979; Metz, 1986; Obuchowski, 2003). Eng, J. (n.d.). ROC analysis: web-based calculator for ROC curves. Retrieved (05/23/05), from http://www.rad.jhmi.edu/roc.

(LOWESS) normalization (Quackenbush, 2002; Yang IV *et al*., 2002), standard deviation regularization (Yang YH *et al*., 2002), iterative linear regression normalization (Quackenbush, 2002), iterative log mean centering normalization (Causton *et al*., 2003), ratio statistics normalization and confidence interval checking (confidence range at 99%) (Chen *et al*., 1997), standard deviation regularization, low intensity filter, slice analysis (Quackenbush, 2002; Yang IV *et al*., 2002), and flip dye consistency checking (Yang YH *et al*., 2002; Quackenbush, 2002).

Unlike other methods this analysis does *not* (1) assume linearity in the error model, (2) correlate levels of transcripts to signal levels, or (3) address the question of accuracy of fold-changes in gene expression (Newton *et al*., 2001; Theilhaber *et al*., 2001; Yang *et al*., 2002; Huber *et al*., 2002; Goryachev *et al*., 2001; Bolstad *et al*., 2003; Irizarry *et al*., 2003). The goal is to discover the genes that are up or downregulated between the samples to a high degree of certainty. The results reveal that the geometrical distributions of f_4-sensitive spots (noise) in the 3D space are non-linear (Figs 1–3). Nonetheless, because the distribution of f_4-sensitive spots models the distribution of all spots in the dataset (Figs 1 and 3), the algorithm builds contour upper- and lower-bound surfaces based on the distributions of f_4-sensitive spots (Figs 4 and 5). Herein, the datasets are normalized by the non-linear method described elsewhere (Fathallah-Shaykh *et al*., 2004). Colantuoni *et al*. (2002) have also described methods for local normalization by non-linear transformations.

Fs is applicable to 2-color (2-channel) microarray data with dye-swapping replicates. Highly specific discovery of states of genetic expression has immediate and numerous applications; specifically, it generates testable hypotheses in biology and medicine that uncover molecular systems behind biological phenotypes. Examples include the phenotypes of resistance to oxidative stress

and motility in cultured glioma and ectopic calcification in meningioma (Fathallah-Shaykh *et al*., 2003; Fathallah-Shaykh, 2005a,b).

Conflict of Interest: None declared.

REFERENCES

Baggerly,K. *et al*. (2001) Identifying differentially expressed genes in cDNA microarray experiments. *J Comput. Biol*., **8**, 639–659.

Bolstad,B.M. *et al*. (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**, 185–193.

Causton,H.C., Quackenbush,J. and Brazma,A. (2003) *Microarray Gene Expression Data Analysis: A Beginner's Guide*. Blackwell Publishing, pp. 55–56.

Chen,Y. *et al*. (1997) Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J. Biomed. Optics*, **2**, 364–374.

Colantuoni,C. *et al*. (2002) SNOMAD (Standardization and NOrmalization of MicroArray Data): web-accessible gene expression data analysis. *Bioinformatics*, **18**, 1540–1541.

DeRisi,J.L. *et al*. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, 680–686.

Fathallah-Shaykh,H.M. (2005a) Genomic discovery reveals a molecular system for resistance to ER and oxidative stress in cultured glioma. *Arch. Neurol*., **62**, 233–236.

Fathallah-Shaykh,H.M. (2005b) Logical networks inferred from highly specific discovery of transcriptionally regulated genes predict protein states in cultured gliomas. *Biochem. Biophys. Res. Comm*., **336**, 1278–1284.

Fathallah-Shaykh,H.M. *et al*. (2002) Mathematical modeling of noise and discovery of genetic expression classes in gliomas. *Oncogene*, **21**, 7164–7174.

Fathallah-Shaykh,H.M. *et al*. (2003) Genomic expression discovery predicts pathways and opposing functions behind phenotypes. *J. Biol. Chem*., **278**, 23830–23833.

Fathallah-Shaykh,H.M. *et al*. (2004) Mathematical algorithm for discovering states of expression from direct genetic comparison by microarrays. *Nucleic Acids Res*., **32**, 3807–3814.

Goryachev,A.B. *et al*. (2001) Unfolding of microarray data. *J. Comp. Biol*., **8**, 443–461.

Huber,W. *et al*. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18**, S96–S104.

Irizarry,R.A. *et al*. (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, 249–265.

Kothapalli,R. *et al*. (2002) Microarray results: how accurate are they? *BMC Bioinformatics*, **3**, 22.

Lockhart,D.J. *et al*. (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat. Biotechnol*., **14**, 1675–1680.

Metz,C.E. (1986) Methodology in radiologic imaging. *Invest. Radiol*., **21**, 720–733.

Newton,M. *et al*. (2001) On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *J. Comp. Biol*., **8**, 37–52.

Ntzani,E.E. and Ioannidis,J.P. (2003) Predictive ability of DNA microarrays for cancer outcomes and correlates: an empirical assessment. *Lancet*, **362**, 1439–1444.

Obuchowski,N.A. (2003) Receiver operating characteristic curves and their use in radiology. *Radiology*, **229**, 3–8.

Quackenbush,J. (2002) Microarray data normalization and transformation. *Nat. Genetics*, **32** (Suppl.), 496–501.

Raffelsberger,W. *et al*. (2003) Quality indicators increase the reliability of microarray data. *Genomics*, **80**, 385–394.

Rosenzweig,B.A. *et al*. (2004) Dye bias correction in dual-labeled cDNA microarray gene expression measurements. *Environ. Health Perspect*., **112**, 480–487.

Schena,M. *et al*. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467–470.

Swets,J.A. (1979) ROC analysis applied to the evaluation of medical imaging techniques. *Invest. Radiol*., **14**, 109–121.

Swets,J.A. and Pickett,R.M. (1992) *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*. New York: Academic Press.

Tan,P.K. *et al*. (2003) Evaluation of gene expression measurements from commercial microarray platforms. *Nucleic Acids Res*., **31**, 5676–5684.

Theilhaber,J. *et al*. (2001) Bayesian estimation of fold-chnages in the analysis of gene expression: the PFOLD algorithm. *J. Comp. Biol*., **8**, 585–614.

Yang,I.V. *et al*. (2002) Within the fold: assessing differential expression measures and reproducibility in microarray assays. *Genome Biol*., **3**, research0062.

Yang,Y.H. *et al*. (2002) Normalization of cDNA microarray data; a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*., **30**, e15.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4263–4271
doi:10.1093/bioinformatics/bti699

Gene expression

Sample size determination for the false discovery rate

Stan Pounds* and Cheng Cheng

Department of Biostatistics, St Jude Children's Research Hospital, 332 N. Lauderdale Street, Memphis, TN 38135, USA

Received on July 14, 2005; revised on September 8, 2005; accepted on September 27, 2005
Advance Access publication October 4, 2005

ABSTRACT

Motivation: There is not a widely applicable method to determine the sample size for experiments basing statistical significance on the false discovery rate (FDR).

Results: We propose and develop the anticipated FDR (aFDR) as a conceptual tool for determining sample size. We derive mathematical expressions for the aFDR and anticipated average statistical power. These expressions are used to develop a general algorithm to determine sample size. We provide specific details on how to implement the algorithm for a *k*-group ($k \geqslant 2$) comparisons. The algorithm performs well for *k*-group comparisons in a series of traditional simulations and in a real-data simulation conducted by resampling from a large, publicly available dataset.

Availability: Documented S-plus and R code libraries are freely available from www.stjuderesearch.org/depts/biostats

Contact: stanley.pounds@stjude.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

The false discovery rate (FDR; Benjamini and Hochberg, 1995), positive FDR (pFDR; Storey, 2002) and conditional FDR (cFDR; Tsai *et al*., 2003) are now recommended measures of statistical significance in the analysis of gene expression data (Storey and Tibshirani, 2003). Each of these measures can roughly be interpreted as the proportion of significant results that are expected to actually be false discoveries. There are now several useful approaches to control or estimate these measures (Benjamini and Hochberg, 1995, 2000; Benjamini and Yekutieli, 2001; Storey, 2002; Allison *et al*., 2002; Pounds and Morris, 2003; Reiner *et al*., 2003; Tsai *et al*., 2003; Pounds and Cheng, 2004; Liao *et al*., 2004; Cheng *et al*., 2004). However, at present, there is very limited guidance on determining how many replicates a planned microarray study needs to detect a given proportion of truly differentially expressed genes when the final statistical analysis uses of these procedures to determine statistical significance.

Several methods for determining sample sizes for microarray studies have been proposed (Pan *et al*., 2002; Lee and Whitmore, 2002; Simon *et al*., 2002; Cui and Churchill, 2003; Mukherjee *et al*., 2003; Gadbury *et al*., 2004; Müller *et al*., 2004; Tsai *et al*., 2005; Jung *et al*., 2005; Jung 2005; Hu *et al*., 2005). However, only a few of these methods determine the sample size when the FDR is the final measure of statistical significance (Gadbury *et al*., 2004; Müller *et al*., 2004; Jung, 2005; Hu *et al*., 2005). Unfortunately, those methods that base sample size calculations on FDR control are very difficult to implement or limited to planning experiments making a simple two-group comparison. Gadbury *et al*. (2004) extrapolate *p*-values obtained from applying a two-sample *t*-test to background data, i.e. data collected in a preliminary or pilot study of the experimental conditions of interest, to a value that might be expected if a larger sample size were used. They do not describe whether or how their algorithm utilizes the non-centrality parameter of the non-central *t*-distribution. Therefore, it is unclear how to generalize their approach to a *k*-group comparison or other types of experiments. Jung (2005) also describes a method for determining the sample size when the two-sample *t*-test is used to perform each hypothesis test but does not discuss how to extend the method to a more complex setting. The method of Müller *et al*. (2004) is very computationally complex and demanding. Hu *et al*. (2005) fit a three-component mixture model to determine the sample size for a two-group comparison using pFDR as the final measure of significance. The remaining methods base sample size determination on other measures of statistical significance or experimental efficiency. There is a definite need to develop a broadly applicable and readily implemented method to determine the sample size for an experiment that uses the FDR, pFDR or cFDR as the ultimate measure of statistical significance.

We have developed a widely applicable and an easily implemented method to determine a sample size that is required to achieve a desired expected discovery rate, which we call average power, while limiting the FDR, pFDR or cFDR below a specified threshold. Based on observations of how the FDR control procedures operate, we propose the anticipated FDR (aFDR) and anticipated average power as conceptual tools to mathematically pose the problem of sample size determination. We then develop an iterative algorithm to solve the sample size problem, as formulated in terms of the aFDR and anticipated average power. We give a detailed description of how to implement the algorithm to determine the sample size for a *k*-group comparison ($k \geq 2$). In simulation studies, the algorithm exhibits desirable properties for choosing the sample size for experiments that perform a *k*-group comparison. Additionally, the algorithm performs well in a 'real-data simulation' of a three-group comparison performed by resampling from a real dataset. Finally, some concluding remarks are offered.

2 APPROACH

2.1 The FDR control and estimation procedures

Suppose that $i = 1, \ldots, m$ hypothesis tests of the form $H_0 : \theta_i = 0$ versus $H_A : \theta_i \neq 0$ are performed. In the context of microarray

*To whom correspondence should be addressed.

studies, i indexes the m features represented on the array, and the null hypothesis $H_0: \theta_i = 0$ typically implies that the expression of feature i is not associated with some phenotype of interest such as clinical response. Moreover, the alternative hypothesis $H_A: \theta \neq 0$ implies that the expression of feature i is associated with the phenotype of interest. For example, θ_i could be the difference between the mean expression of feature i across two experimental groups or the correlation of the expression of feature i with another continuous variable. Benjamini and Hochberg (1995) and others (Allison *et al*., 2002; Storey, 2002; Pounds and Morris, 2003; Tsai *et al*., 2003) note that each of these m hypothesis tests results in one of four distinct outcomes: incorrectly declaring the result significant (i.e. a Type I error, false positive or false discovery), correctly declaring the result significant (i.e. a true positive or true discovery), incorrectly declaring a result to be insignificant (i.e. a false negative or Type II error) or correctly declaring a result to be insignificant (i.e. a true negative). One statistical challenge in this setting is to define a meaningful error metric to address the multiple-testing issue.

The FDR, pFDR and cFDR are now widely regarded as useful multiple-testing error metrics for microarray experiments (Storey and Tibshirani, 2003). Letting V represent the number of false discoveries and R represent the total number of results declared significant (correctly or incorrectly), Benjamini and Hochberg (1995) define the FDR as

$$\mathrm{FDR} = E\left(\frac{V}{R}\,\middle|\, R > 0\right)\Pr(R > 0). \tag{1}$$

The pFDR (Storey, 2002) and cFDR (Tsai *et al*., 2003) are similar measures of statistical significance. The FDR, pFDR and cFDR can be loosely interpreted as the expected proportion of significant results that are false discoveries. Although not explicit in the notation, the FDR, pFDR and cFDR clearly depend on the procedure used to determine which results are significant. Benjamini and Hochberg (1995, 2000) and Benjamini and Yekutieli (2001) have developed procedures to control the FDR at a prespecified level. Storey (2002) has developed a procedure to control the pFDR at a prespecified level. Others (Allison *et al*., 2002; Pounds and Morris, 2003; Pounds and Cheng, 2004; Liao *et al*., 2004; Cheng *et al*., 2004) have developed methods to estimate the FDR, pFDR or cFDR as a function of the threshold α used to determine which p-values will be declared significant.

Some additional notation is now introduced. Let

$$\mathrm{F}_i(\alpha \mid \theta_i, n) = \Pr(p_i \leq \alpha \mid \theta_i, n) \tag{2}$$

represent the probability that the p-value p_i for test i is less than or equal to a fixed α, given a sample size n and a value of the parameter θ_i. Throughout this article, subscripts and arguments that are clear by context or superfluous to the context may be omitted to simplify the notation. For example, we may write $\mathrm{F}_i(\alpha)$ instead of $\mathrm{F}_i(\alpha \mid \theta_i, n)$. Note that for continuously distributed test statistics, $\mathrm{F}_i(\alpha \mid \theta_i = 0, n) = \alpha$ by the definition of a p-value and that $\mathrm{F}_i(\alpha \mid \theta_i, n)$ is the statistical power of the α-level test for $\theta_i \neq 0$. Furthermore, let $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_m\}$, assume a common sample size n is used for all tests and define

$$\mathbf{F}(\alpha \mid \boldsymbol{\theta}, n) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{F}_i(\alpha \mid \theta_i, n). \tag{3}$$

Let $\mathrm{I}(\cdot)$ be the indicator function, i.e. $\mathrm{I}(\cdot) = 1$ if the enclosed statement is true and $\mathrm{I}(\cdot) = 0$ if the enclosed statement is false. Additionally, let $m_0 = \sum_{i=1}^{m} \mathrm{I}(\theta_i = 0)$ represent the number of tests with a true null hypothesis so that

$$\boldsymbol{\pi} = \frac{m_0}{m} \tag{4}$$

is the proportion of tests with a true null hypothesis.

Many of the FDR, pFDR or cFDR estimation or control procedures perform very similar operations on the p-values p_1, $p_2, \ldots, p_m$ computed in the m hypothesis tests. First, the p-values are used to obtain estimates $\hat{\mathbf{F}}(\alpha)$ of $\mathbf{F}(\alpha)$ in (3) and $\hat{\boldsymbol{\pi}}$ of $\boldsymbol{\pi}$ in (4). The methods differ in their approaches to finding $\hat{\boldsymbol{\pi}}$ and $\hat{\mathbf{F}}(\alpha)$. Next, the ordered p-values $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$ are used to compute ratios of the form

$$t_{(i)} = \frac{\hat{\boldsymbol{\pi}} p_{(i)}}{\hat{\mathbf{F}}(p_{(i)})}. \tag{5}$$

The original FDR-control procedure developed by Benjamini and Hochberg (1995) can be expressed in this framework by conservatively using $\hat{\boldsymbol{\pi}} = 1$ in its operations (Benjamini and Hochberg, 2000). Some methods (Allison *et al*., 2002; Pounds and Morris, 2003; Tsai *et al*., 2003; Pounds and Cheng, 2004; Liao *et al*., 2004; Cheng *et al*., 2004) simply report $t_{(i)}$ as an estimate of the proportion of results with a p-value less than or equal to $p_{(i)}$ that are false discoveries. Other methods (Benjamini and Hochberg, 1995, 2000; Benjamini and Yekutieli, 2001; Storey, 2002; Reiner *et al*., 2003) are developed to control the FDR, pFDR or cFDR at a prespecified level τ and perform a few additional operations. These control methods compute

$$q_{(i^*)} = \min_{i \geq i^*} t_{(i)} \tag{6}$$

for $i^* = 1, \ldots, m$. Each result with $q_{(i^*)} \leq \tau$ is then declared significant. Of note, the control procedures use the p-values to determine a threshold $\hat{\boldsymbol{\alpha}}$ in such a manner that declaring all results with a p-value less than or equal to $\hat{\boldsymbol{\alpha}}$ ensures that the desired level τ of FDR, pFDR or cFDR control is maintained (Benjamini and Hochberg, 1995; Storey, 2002). In particular,

$$\hat{\boldsymbol{\alpha}} = \max\{p_{(i)} : q_{(i)} \leq \tau\}. \tag{7}$$

That is, $\hat{\boldsymbol{\alpha}}$ is equal to the largest $p_{(i)}$ such that $q_{(i)} \leq \tau$. If none of the $q_{(i)}$ is less than or equal to τ, then no results are declared significant.

2.2 Power in the multiple-testing setting

Sample size calculations for microarray experiments must be based on a metric of statistical power that is appropriate for the multiple-testing setting. We define the average power of the multiple-testing procedure Ω by

$$\mathbf{G}(\Omega \mid \boldsymbol{\theta}, n) = \frac{1}{m - m_0}\sum_{i=1}^{m} \mathrm{I}(\theta_i \neq 0)\Pr(p_i \leq \hat{\boldsymbol{\alpha}}_\Omega \mid \boldsymbol{\theta}, n), \tag{8}$$

where $\hat{\boldsymbol{\alpha}}_\Omega$ is the p-value threshold determined by the procedure Ω. The average power has been considered in previous work; Gadbury *et al*. (2004) call $\mathbf{G}(\cdot)$ the expected discovery rate and Cheng *et al*. (2004) call $1 - \mathbf{G}(\cdot)$ the false non-discovery proportion. When Ω is the procedure that declares significant all p-values less than a common fixed threshold α, the average power is written $\mathbf{G}(\alpha)$ instead of $\mathbf{G}(\Omega)$.

2.3 The anticipated false discovery ratio

Now, consider planning an experiment in which m hypothesis tests will be performed and a particular procedure Ω will be used to control the FDR, pFDR or cFDR at a desired level τ. Also, suppose that one wants the procedure Ω to have average power δ across the tests examining a false null hypothesis. Assume that all hypotheses will be tested using a common sample size n. Additionally, assume that the power of each statistical test strictly increases with increasing n. Most classical hypothesis-testing procedures satisfy this assumption. Clearly, under these assumptions, the average power increases monotonically with n. However, this relationship must be expressed in a mathematically useful way to guide the sample size selection process. In particular, one needs to be able to describe the anticipated properties of the procedure Ω in terms of the number of hypothesis tests resulting in each of the four distinct outcomes (false positives, true positives, false negatives, true negatives) as a function of the sample size n. This is non-trivial, because the p-value threshold $\hat{\alpha}$ used to declare significance is determined by properties of the observed p-value distribution, hence $\hat{\alpha}$ is a random variable. However, it is feasible to derive expressions for the FDR and power of the fixed α procedure. The expressions for the fixed α procedure may yield useful approximations to determine the sample size when the FDR, pFDR or cFDR will be controlled using (5) and (6). Therefore, we proceed by deriving expressions for the fixed α procedure which will provide useful approximations for purposes of sample size determination.

The computation of an anticipated value for $\hat{\mathbf{F}}(\alpha \,|\, n)$ is straightforward when a power formula $\hat{\mathbf{F}}_i^*(\alpha \,|\, \theta_i, n)$ that represents or approximates $\Pr(p_i \leq \alpha \,|\, \theta_i, n)$ is available for each hypothesis test $i = 1, \ldots, m$. In particular, the power formulas can be used to evaluate (2) for each i, hence (3) can also be evaluated. Therefore, we define

$$\tilde{\mathbf{F}}(\alpha \,|\, \tilde{\boldsymbol{\theta}}, n) = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{F}}_i^*(\alpha \,|\, \tilde{\theta}_i, n) \tag{9}$$

as the anticipated significant proportion at the p-value threshold α, given a value $\boldsymbol{\theta} = \{\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_m\}$ of $\boldsymbol{\theta}$ that is specified for purposes of sample size determination. Assuming that $\tilde{\boldsymbol{\theta}}$ is accurate, the definition in (9) gives a reasonable approximation of $\mathbf{F}(\alpha)$ as a function of n. Thus, (9) is an estimate of (3). Each FDR procedure that operates on p-values uses an estimate $\hat{\mathbf{F}}(\alpha)$ of $\mathbf{F}(\alpha)$ in the denominator of (5). Subsequently, because the different procedures are estimating the same quantity, each procedure should yield a similar value of the denominator. Therefore, for each FDR procedure that operates on p-values, (9) is used to define the anticipated significant proportion at the threshold α for a sample size n.

Additionally, an anticipated value of $\hat{\boldsymbol{\pi}}$ is easily computed, given a power formula for each hypothesis test i. The value of $\hat{\boldsymbol{\pi}}$ used in (5) depends heavily on the FDR procedure Ω to be used in the final analysis. If Ω is the original FDR control procedure proposed by Benjamini and Hochberg (1995), then clearly $\hat{\boldsymbol{\pi}}_{\Omega} = 1$. Other methods base their estimate $\hat{\boldsymbol{\pi}}$ on properties of the observed p-value distribution. We now derive an expression to compute an anticipated value of $\hat{\boldsymbol{\pi}}$ given a sample size n for most of the other methods. For $i = 1, \ldots, m$, let

$$\hat{\mathbf{f}}_i^*(\alpha \,|\, \theta_i, n) = \frac{\mathrm{d}}{\mathrm{d}\alpha} \hat{\mathbf{F}}_i^*(\alpha \,|\, \theta_i, n), \tag{10}$$

and let

$$\hat{\mathbf{f}}^*(\alpha \,|\, \boldsymbol{\theta}, n) = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{f}}_i^*(\alpha \,|\, \boldsymbol{\theta}, n). \tag{11}$$

In particular, several methods rely on the inequality

$$\pi \leq \min_p \mathbf{f}(p), \tag{12}$$

as expressed by Pounds and Morris (2003), where $\mathbf{f}(\cdot)$ is the derivative of $\mathbf{F}(\cdot)$ with respect to α, to motivate using the minimum of $\hat{\mathbf{f}}^*(\cdot)$ as the value $\hat{\boldsymbol{\pi}}$ in (5). It is easy to evaluate (11) given the formulas for the distributions of the test statistics under the null and alternative hypotheses (Section S1, Supplementary materials). Therefore, the anticipated null proportion estimate, given a specified $\tilde{\boldsymbol{\theta}}$ and n, is defined as

$$\tilde{\boldsymbol{\pi}}_{\Omega}(\tilde{\boldsymbol{\theta}}, n) = \min_{0 \leq p \leq 1} \hat{\mathbf{f}}^*(p \,|\, \tilde{\boldsymbol{\theta}}, n) \tag{13}$$

for each procedure Ω that uses inequality (12) to motivate its estimate of the null proportion.

Now, define

$$\mathrm{aFDR}_{\Omega}(\alpha \,|\, \tilde{\boldsymbol{\theta}}, n) = \frac{\tilde{\boldsymbol{\pi}}_{\Omega}(\tilde{\boldsymbol{\theta}}, n)\alpha}{\tilde{\mathbf{F}}(\alpha \,|\, \tilde{\boldsymbol{\theta}}, n)} \tag{14}$$

as the aFDR of the procedure Ω at the threshold α given n and $\tilde{\boldsymbol{\theta}}$. The aFDR represents the anticipated value of the ratios in (5) corresponding to p-values close to α assuming that $\tilde{\boldsymbol{\theta}} \approx \boldsymbol{\theta}$ and the experiment will use a sample size of n. If α is chosen so that $\mathrm{aFDR}(\alpha) = \tau$, then (6) and (7) imply that the threshold $\hat{\alpha}$ determined by the procedure Ω will tend to be greater than or equal to α when $\tilde{\boldsymbol{\theta}} \approx \boldsymbol{\theta}$. Therefore, basing power estimates on this well-chosen α should tend to produce sample size estimates that achieve or exceed a desired level of statistical power.

2.4 The anticipated average power

As previously mentioned, the FDR, pFDR and cFDR control procedures do not prespecify the p-value significance threshold. Hence, it is difficult to derive an analytical expression for $\mathbf{G}(\Omega)$ for those procedures. However, for the fixed α procedure, it is straightforward to evaluate (8) for any n and $\boldsymbol{\theta}$, given a power formula for each hypothesis test $i = 1, \ldots, m$. Therefore, given a specified value $\tilde{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ for purposes of sample size calculations, define

$$\tilde{\mathbf{G}}(\alpha \,|\, \tilde{\boldsymbol{\theta}}, n) = \mathbf{G}(\alpha \,|\, \tilde{\boldsymbol{\theta}}, n) \tag{15}$$

as the anticipated average power of the fixed α procedure with sample size n. When α is chosen so that $\mathrm{aFDR}(\alpha) = \tau$ and $\tilde{\boldsymbol{\theta}} \approx \boldsymbol{\theta}$, then it follows that $\mathbf{G}(\Omega) \approx \tilde{\mathbf{G}}(\Omega) \geq \tilde{\mathbf{G}}(\alpha)$ because $\hat{\alpha}$ tends to be greater than or equal to α, as previously discussed at the end of Section 2.3.

3 METHODS

3.1 The general sample size algorithm

Our objective is to choose a sample size n so that using a particular procedure Ω to control the FDR, pFDR or cFDR at a prespecified level τ has average power δ or greater. An approximate objective is to find n and

$\tilde{\alpha}$ that satisfy

$$\tilde{\mathbf{G}}(\tilde{\alpha} \mid \tilde{\boldsymbol{\theta}}, n) \geq \delta \tag{16}$$

and

$$\text{aFDR}(\tilde{\alpha} \mid \tilde{\boldsymbol{\theta}}, n) \leq \tau, \tag{17}$$

given a value $\tilde{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ specified for purposes of determining the sample size. The value $\tilde{\boldsymbol{\theta}}$ can be specified to correspond to a setting of particular interest or estimated from background data when it is available. In Section 3.3, we discuss how to use background data to specify a value of $\tilde{\boldsymbol{\theta}}$ for the one-way k-group comparison. A simple iterative algorithm can be used to find $\tilde{\alpha}$ and n that satisfy the requirements in (16) and (17).

ALGORITHM 1. Sample Size Determination.

(1) For each hypothesis test to be performed, specify a formula for its statistical power at the α-level.
(2) Determine the procedure Ω to be used for FDR, pFDR or cFDR control.
(3) Specify the desired average power δ and the desired level τ of FDR, pFDR or cFDR control.
(4) Specify $\tilde{\boldsymbol{\theta}}$ for purposes of sample size determination.
(5) Set n to some initial minimum sample size n_0 (e.g. $n_0 = 3$).
(6) Compute $\tilde{\mathbf{G}}(\alpha \mid \tilde{\boldsymbol{\theta}}, n)$.
(7) Let $\tilde{\alpha}_n$ be the smallest value of α such that $\tilde{\mathbf{G}}(\alpha \mid \tilde{\boldsymbol{\theta}}, n) \geq \delta$.
(8) Compute $\tilde{\boldsymbol{\pi}}_\Omega(\tilde{\boldsymbol{\theta}}, n)$.
(9) Compute aFDR $(\tilde{\alpha}_n \mid \tilde{\boldsymbol{\theta}}, n)$.
(10) If aFDR $(\tilde{\alpha}_n) \leq \tau$, stop and report n as the sample size estimate. Otherwise, increase n by 1 and return to Step 6.

In the Supplementary materials (Section S2), we show that Algorithm 1 converges to a solution of the requirements in (16) and (17). Additionally, if the specified $\tilde{\boldsymbol{\theta}}$ accurately reflects $\boldsymbol{\theta}$, it follows that $\mathbf{G}(\Omega)$ should be greater than or equal to $\tilde{\mathbf{G}}(\alpha) \geq \delta$ because the final $\tilde{\alpha}_n$ satisfies the relation $\text{aFDR}(\tilde{\alpha}_n) \leq \tau$ (see the end of Section 2.3). Furthermore, we show that Algorithm 1 avoids several technical difficulties that an alternative algorithm could possibly encounter (Section S3, Supplementary materials).

3.2 The k-group comparison

Algorithm 1 is very general. To implement it in practice, application-specific details must be provided. We now elaborate on how to implement Algorithm 1 for a one-way comparison of k-groups. Suppose that the expression of each of $i = 1, \ldots, m$ features is to be compared across k-groups. Each group will be represented by n independent experimental subjects, and each feature will be measured exactly once per subject. In the context of microarray experiments, this means that n biological replicates per group will be performed with only one technical replicate per biological replicate. We show how to use Algorithm 1 to determine the number of biological replicates n to be performed for each experimental group.

Suppose that the (possibly transformed) expression values of each feature satisfy the assumptions of one-way ANOVA. All measurements of the feature are statistically independent and normally distributed with equal variance and (possibly unequal) group-specific means. For $i = 1, \ldots, m$ and $j = 1, \ldots, k$, let μ_{ij} be the mean (transformed) expression of feature i in group j. Also, for $i = 1, \ldots, m$, let σ_i^2 be the common variance of the (transformed) expression measurements of feature i within each of the k-groups. Then, for $i = 1, \ldots, m$, let

$$\theta_i = \frac{\sum_{j=1}^{k} (\mu_{ij} - \mu_{i.})^2}{2\sigma_i^2}, \tag{18}$$

where $\mu_{i.} = 1/k \sum_{j=1}^{k} \mu_{ij}$ for $i = 1, \ldots, m$. For $i = 1, \ldots, m$, one-way ANOVA is to be used to test $H_0 : \theta_i = 0$ (i.e. mean expression is equal for all k-groups or $\mu_{i1} = \mu_{i2} = \cdots \mu_{ik}$) versus $H_A : \theta_i > 0$ (i.e. mean expression of at least two groups differs or $\mu_{ij} \neq \mu_{ij^*}$ for some j and j^*).

Step 1 of Algorithm 1 requires us to obtain a power formula for each hypothesis test. We can use the standard one-way ANOVA power formula (Scheffe', 1959) for this purpose, because one-way ANOVA will be used to perform each hypothesis test. In the setting described above, the one-way ANOVA F-statistic F_i for feature i follows an F-distribution with $k - 1$ numerator degrees of freedom, $k(n - 1)$ denominator degrees of freedom, and non-centrality parameter

$$\lambda_i = n\theta_i \tag{19}$$

for $i = 1, \ldots, m$. Thus, given $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_m\}$, the power of the ANOVA test for each feature is easily computed.

Steps 2 and 3 of Algorithm 1 require the investigator to choose the procedure used to control the FDR, pFDR or cFDR, the desired level τ of control for the selected error metric and the desired average power δ. Certainly, these choices are application specific. In our simulation studies and the example below, we choose the q-value procedure (Storey, 2002) to control the pFDR and examine various choices of τ and δ.

Step 4 requires one to specify values of $\tilde{\boldsymbol{\theta}}$ to use in later calculations. This is typically done in one of two ways: either to specify $\tilde{\boldsymbol{\theta}}$ that corresponds to values that are of particular interest or to use background data to obtain $\tilde{\boldsymbol{\theta}}$. Clearly, the first approach is application specific and subject to arbitrary determinations of what is 'of particular interest.' The second approach requires careful statistical considerations; our proposed method for using background data to obtain $\tilde{\boldsymbol{\theta}}$ is outlined in Section 3.3.

Step 5 of Algorithm 1 simply requires setting a minimum sample size to start the iterative portion of the algorithm (steps 6 through 10).

Step 6 of Algorithm 1 is easily implemented, given the specified value of $\tilde{\boldsymbol{\theta}}$. Each component of the sum in (15) can be computed using the classical one-way ANOVA power formula. There is a unique solution to the constraint specified in Step 7 of Algorithm 1 because $\tilde{\mathbf{G}}(\alpha)$ satisfies the properties of a cumulative distribution function.

Step 8 of Algorithm 1 is easily implemented for the k-group comparison. We have shown that $e^{-\lambda}$ is the minimum of the probability density function of the ANOVA p-value (Section S4, Supplementary materials), where λ is the non-centrality parameter of the distribution of the one-way ANOVA F-statistic. Therefore, for each procedure Ω that uses inequality (12) to motivate its estimator of π, (19) suggests that

$$\tilde{\boldsymbol{\pi}}_\Omega(\tilde{\boldsymbol{\theta}}, n) = \frac{1}{m} \sum_{i=1}^{m} e^{-n\tilde{\theta}_i} \tag{20}$$

can be used in Step 8 of Algorithm 1. For the original FDR control procedure proposed by Benjamini and Hochberg (1995), $\tilde{\boldsymbol{\pi}}_\Omega = 1$, as previously indicated.

Recall that each component of the sum in (9) is given by the level α of each test i such that $\tilde{\theta}_i = 0$ and the power of the tests for all other i. Therefore, Step 9 of Algorithm 1 is easily computed by substituting (9) and (20) into (14). Finally, Step 10 simply compares the result of Step 9 to the preselected τ to determine whether the algorithm should be terminated or further iterations are necessary.

3.3 Using background data

We propose a simple and effective method that uses background data to obtain $\tilde{\boldsymbol{\theta}}$. To simplify the presentation, assume that the background data have an equal sample size $\acute{n}$ representing each experimental group. The approach can be modified to a more general setting by straightforward modification of the formulas derived below. Suppose that the background data can be used to perform one-way ANOVA for each feature i. For $i = 1, \ldots, m$, let $\acute{F}_i$ and $\acute{p}_i$ be the F-statistics and p-values obtained by applying one-way ANOVA to the background data. For each feature i, an estimate $\acute{\theta}_i$ of θ_i can be obtained by proper scaling of $\acute{F}_i$. Patnaik (1949) notes that the non-central F-distribution with ν_1 and ν_2 degrees of freedom and non-centrality parameter λ can

be approximated by a central F-distribution with $\nu^* = (\nu_1 + 2\lambda)^2/(\nu_1 + 4\lambda)$ and ν_2 degrees of freedom and scaled by a factor $(\nu_1 + 2\lambda)/\nu_1$. The approximation suggests that

$$\mathrm{E}(\acute{F}_i) \approx \left(\frac{\nu_2}{\nu_2 - 2}\right)\left(\frac{\nu_1 + 2\lambda_i}{\nu_1}\right) \tag{21}$$

and motivates

$$\acute{\theta}_i = \max\left(0, \frac{\nu_1}{2\acute{n}}\left(\frac{\nu_2 - 2}{\nu_2}\acute{F}_i - 1\right)\right) \tag{22}$$

as a moment-based estimator of θ_i for $i = 1, \ldots, m$. The estimates produced by (22) must be adjusted for multiplicity; clearly each test i yielding a small p-value $\acute{p}_i$ will also yield large $\acute{\theta}_i$. An FDR, pFDR or cFDR estimation or control procedure can be applied to $\acute{\boldsymbol{p}} = \{\acute{p}_1, \ldots, \acute{p}_m\}$ to perform a useful multiplicity adjustment for the purpose of sample size estimation. Suppose that application of an FDR estimation or control procedure to $\acute{\boldsymbol{p}}_i$ yields an estimate $\acute{\boldsymbol{\pi}}$ of π and values $\acute{q}_1, \ldots, \acute{q}_m$ as in (6). For $i = 1, \ldots, m$, the absence of parentheses in the subscript indicates that $\acute{q}_i$ corresponds to the p-value of the test i, as originally indexed (i.e. prior to ordering the p-values in ascending order). Then, for $i = 1, \ldots, m$, let

$$\tilde{\theta}_i = \begin{cases} 0 & \text{if } \acute{p}_i \geq \acute{p}_{(\lfloor m\acute{\pi} \rfloor)} \\ (1 - \acute{q}_i)\acute{\theta}_i & \text{otherwise.} \end{cases} \tag{23}$$

The adjustment in (23) sets the $\lceil m\acute{\pi} \rceil$ components of $\tilde{\theta}$ with the largest background data p-values equal to 0, consistent with the interpretation of the value of $\acute{\pi}$ that follows from definition (4). The remaining components are scaled in a manner that capitalizes on the Bayesian interpretation of the q-value as the probability that a rejection is a false discovery (Storey, 2003). Given this interpretation, the term $(1 - \acute{q}_i)\acute{\theta}_i$ in (23) is an estimate of the expected value of θ_i in the Bayesian framework. While this property has not been explicitly proven for FDR estimates produced by other procedures, we proceed under the assumption that the values of $\acute{q}_i$ produced by similar methods should work reasonably well for the purpose of computing an estimate $\tilde{\theta}$ for use in sample size calculations.

Our algorithm that uses background data to determine $\tilde{\boldsymbol{\theta}}$ can be summarized as follows:

ALGORITHM 2. Using k-group comparison background data to obtain $\tilde{\boldsymbol{\theta}}$ for Step 4 in Algorithm 1.

(1) Apply one-way ANOVA to the background data to compute an F-statistic $\acute{F}_i$ and p-value $\acute{p}_i$ for each feature $i = 1, \ldots, m$.

(2) Use (22) to obtain $\acute{\boldsymbol{\theta}} = \{\acute{\theta}_i, \acute{\theta}_2, \ldots, \acute{\theta}_m\}$.

(3) Apply an FDR, pFDR or cFDR procedure to $\acute{p}_1, \acute{p}_2, \ldots, \acute{p}_m$ to obtain $\acute{\boldsymbol{\pi}}$ and $\acute{q}_1, \ldots, \acute{q}_m$.

(4) Use (23) to obtain $\tilde{\boldsymbol{\theta}} = \{\tilde{\theta}_1, \ldots, \tilde{\theta}_m\}$.

In our simulation and example below, we use the spacing loess histogram (SPLOSH; Pounds and Cheng, 2004) to implement Step 3 of Algorithm 2. Pounds and Cheng (2004) show that the values of $q_1, \ldots, q_m$ produced by SPLOSH are more stable than those produced by the q-value procedure (Storey, 2002). Additionally, if the values of $q_1, \ldots, q_m$ are interpreted as estimates of the cFDR, Pounds and Cheng (2004) show that those produced by SPLOSH are more accurate than those produced by the q-value procedure. In simulation studies, Pounds and Cheng (2004) note that the values of $\acute{q}_i$ produced by Storey's (2002) procedure tend to underrepresent the proportion of false discoveries among those results deemed significant because he applies the right-sided minimization operation in (6) to unsmoothed values of t_i in (5). Pounds and Cheng (2004) reduce the bias by proposing a method that results in more stable t_i before applying the right-sided minimization operation. Certainly, alternatives to Step 3 of Algorithm 2 should be further explored in future research. Nevertheless, using SPLOSH to implement Step 3 of Algorithm 2 results in desirable performance characteristics in the simulation studies described below.

4 RESULTS

4.1 Traditional simulation studies

Two series of traditional simulation studies were performed to evaluate the performance of Algorithms 1 and 2. The first simulation series considers the setting in which $\tilde{\boldsymbol{\theta}}$ is arbitrarily chosen and happens to equal the actual $\boldsymbol{\theta}$. The second simulation series examines the performance of Algorithm 1 when Algorithm 2 is applied to background data to determine $\tilde{\boldsymbol{\theta}}$. Each simulation series considers settings in which the expressions of $m = 1000$ features are compared across $k = 2$ or $k = 3$ groups. Additionally, for each feature i, the true value of θ_i is either zero or some $\eta > 0$. In each setting, $m\pi$ of the features are not differentially expressed across groups, and the remaining components of $\boldsymbol{\theta}$ equal a common value η. Additionally, in both simulation series, it is assumed that the one-way ANOVA F-tests are statistically independent and that the standard assumptions of one-way ANOVA hold. Each simulation performs 1000 independent repetitions of the assumed setting. In both series, Storey's (2002) q-value is used to control the pFDR in the final analysis of each repetition.

In the first simulation series, $\tilde{\boldsymbol{\theta}}$ is fixed and assumed equal to $\boldsymbol{\theta}$. For each setting studied in the first simulation series, Algorithm 1 was used to determine the per-group sample size n for the planned experiment. In each repetition, the first simulation series generated a collection $\mathcal{F} = \{F_1, \ldots, F_m\}$ of F-statistics according to the assumed setting, obtained the corresponding p-values $\boldsymbol{p} = \{p_1, \ldots, p_m\}$, applied the q-value procedure (Storey, 2002) to $\boldsymbol{p}$ to determine significance, counted the number of each outcome (false positive, false negative, true positive, true negative), calculated the ratio D of true positives to the number of features with $\theta = \eta > 0$ and calculated the ratio $Q = V/R$ of the number V of false positives to the number R of significant results. The results for each setting were summarized across repetitions. For computational efficiency, the simulation generated F-statistics directly without first generating data and applying one-way ANOVA. Under the assumptions of one-way ANOVA, the information in $\tilde{\boldsymbol{\theta}}$ and n provides sufficient information to generate the F-statistics without first generating datasets. Additionally, this assumption and (18) imply that the effect sizes are smaller for a setting with $k = 3$ than for a setting with $k = 2$ when both settings have equal values of η.

Table 1 gives the simulation estimates of the expected value (EV) and standard error (SE) of $\hat{\alpha}$ and D. Table 1 shows that $\hat{\alpha}$ tends to be greater than $\tilde{\alpha}_n$. The simulation estimate of the expected value (EV) of $\hat{\alpha}$ is greater than $\tilde{\alpha}$ in all settings. Additionally, the simulation estimates of the average power (i.e. the expected value of D) exceeded δ in all settings. Furthermore, in all settings, the proportion D of differentially expressed features that were declared significant exceeded δ in each repetition (data not shown). Thus, in the settings considered, Algorithm 1 finds a sample size so that the performed experiment is almost certain (i.e. has probability very near 1) to declare more than δ of the truly differentially expressed features as significant. These results support our conjectures of Sections 2.3 and 2.4 that $\hat{\alpha}$ should tend to be greater than $\tilde{\alpha}_n$ and that $G(\Omega)$ should tend to be greater than $\tilde{\mathbf{G}}(\tilde{\alpha}_n) \geq \delta$. Additionally, simulation estimates of the expected value of Q were very near the specified level τ in all settings (data not shown), in agreement with known properties of the q-value procedure (Storey, 2002).

The second simulation series studies the performance of Algorithm 1 when Algorithm 2 is applied to background data to

Table 1. Results of the first simulation series

Setting	Selection		Calculation		$\hat{\alpha}$	D
(k, π, η)	τ	δ	n	$\tilde{\alpha}_n$	EV (SE)	EV (SE)
(2, 0.7, 0.5)	0.05	0.5	15	0.010	0.020 (0.002)	0.916 (0.018)
(2, 0.7, 0.5)	0.05	0.8	23	0.015	0.021 (0.002)	0.991 (0.006)
(2, 0.7, 1.0)	0.05	0.5	9	0.008	0.021 (0.002)	0.943 (0.016)
(2, 0.7, 1.0)	0.05	0.8	12	0.017	0.021 (0.002)	0.989 (0.006)
(2, 0.7, 0.5)	0.10	0.5	12	0.021	0.043 (0.005)	0.899 (0.021)
(2, 0.7, 0.5)	0.10	0.8	19	0.032	0.046 (0.005)	0.988 (0.006)
(2, 0.7, 1.0)	0.10	0.5	7	0.019	0.044 (0.005)	0.918 (0.020)
(2, 0.7, 1.0)	0.10	0.8	10	0.034	0.046 (0.005)	0.987 (0.007)
(2, 0.9, 0.5)	0.05	0.5	20	0.003	0.005 (0.001)	0.928 (0.027)
(2, 0.9, 0.5)	0.05	0.8	30	0.004	0.005 (0.001)	0.995 (0.007)
(2, 0.9, 1.0)	0.05	0.5	12	0.002	0.005 (0.001)	0.954 (0.022)
(2, 0.9, 1.0)	0.05	0.8	16	0.004	0.005 (0.001)	0.994 (0.007)
(2, 0.9, 0.5)	0.10	0.5	17	0.006	0.011 (0.001)	0.916 (0.032)
(2, 0.9, 0.5)	0.10	0.8	26	0.008	0.011 (0.002)	0.993 (0.009)
(2, 0.9, 1.0)	0.10	0.5	10	0.005	0.011 (0.001)	0.939 (0.026)
(2, 0.9, 1.0)	0.10	0.8	14	0.008	0.011 (0.002)	0.992 (0.009)
(3, 0.7, 0.5)	0.05	0.5	18	0.010	0.021 (0.002)	0.928 (0.016)
(3, 0.7, 0.5)	0.05	0.8	27	0.015	0.021 (0.003)	0.993 (0.005)
(3, 0.7, 1.0)	0.05	0.5	10	0.009	0.021 (0.002)	0.937 (0.017)
(3, 0.7, 1.0)	0.05	0.8	14	0.017	0.021 (0.002)	0.992 (0.005)
(3, 0.7, 0.5)	0.10	0.5	15	0.020	0.044 (0.005)	0.918 (0.019)
(3, 0.7, 0.5)	0.10	0.8	22	0.037	0.046 (0.005)	0.988 (0.006)
(3, 0.7, 1.0)	0.10	0.5	8	0.022	0.043 (0.005)	0.915 (0.020)
(3, 0.7, 1.0)	0.10	0.8	12	0.033	0.046 (0.005)	0.991 (0.006)
(3, 0.9, 0.5)	0.05	0.5	24	0.003	0.005 (0.001)	0.948 (0.022)
(3, 0.9, 0.5)	0.05	0.8	34	0.004	0.005 (0.001)	0.995 (0.007)
(3, 0.9, 1.0)	0.05	0.5	13	0.003	0.005 (0.001)	0.949 (0.023)
(3, 0.9, 1.0)	0.05	0.8	18	0.004	0.005 (0.001)	0.996 (0.006)
(3, 0.9, 0.5)	0.10	0.5	21	0.005	0.011 (0.001)	0.941 (0.025)
(3, 0.9, 0.5)	0.10	0.8	30	0.009	0.011 (0.002)	0.994 (0.008)
(3, 0.9, 1.0)	0.10	0.5	11	0.006	0.011 (0.001)	0.934 (0.026)
(3, 0.9, 1.0)	0.10	0.8	16	0.008	0.011 (0.002)	0.995 (0.008)

Table 2. Results for the second simulation series

Setting	Selection		$n \mid n \leq 50$	$D \mid n \leq 50$	$n \leq 50$
(k, π, η)	τ	δ	EV (SE)	EV (SE)	Pr
(2, 0.7, 0.5)	0.05	0.5	18.8 (3.8)	0.953 (0.044)	1.000
(2, 0.7, 0.5)	0.05	0.8	35.9 (6.9)	0.999 (0.002)	0.628
(2, 0.7, 1.0)	0.05	0.5	13.2 (2.2)	0.991 (0.011)	1.000
(2, 0.7, 1.0)	0.05	0.8	29.8 (7.6)	1.000 (0.000)	0.802
(2, 0.7, 0.5)	0.10	0.5	15.0 (2.9)	0.943 (0.044)	1.000
(2, 0.7, 0.5)	0.10	0.8	31.1 (7.4)	0.999 (0.004)	0.679
(2, 0.7, 1.0)	0.10	0.5	10.4 (1.5)	0.986 (0.013)	1.000
(2, 0.7, 1.0)	0.10	0.8	27.3 (9.0)	1.000 (0.000)	0.995
(2, 0.9, 0.5)	0.05	0.5	21.1 (7.2)	0.872 (0.131)	1.000
(2, 0.9, 0.5)	0.05	0.8	31.7 (7.9)	0.984 (0.032)	0.671
(2, 0.9, 1.0)	0.05	0.5	18.0 (5.7)	0.985 (0.026)	1.000
(2, 0.9, 1.0)	0.05	0.8	30.1 (7.5)	1.000 (0.001)	0.706
(2, 0.9, 0.5)	0.10	0.5	17.3 (5.6)	0.859 (0.129)	1.000
(2, 0.9, 0.5)	0.10	0.8	28.6 (8.2)	0.983 (0.029)	0.685
(2, 0.9, 1.0)	0.10	0.5	14.9 (4.4)	0.981 (0.027)	1.000
(2, 0.9, 1.0)	0.10	0.8	26.0 (7.4)	1.000 (0.001)	0.748
(3, 0.7, 0.5)	0.05	0.5	15.7 (3.4)	0.841 (0.101)	1.000
(3, 0.7, 0.5)	0.05	0.8	28.0 (7.2)	0.984 (0.021)	0.762
(3, 0.7, 1.0)	0.05	0.5	11.9 (2.0)	0.965 (0.029)	1.000
(3, 0.7, 1.0)	0.05	0.8	26.4 (9.3)	1.000 (0.001)	0.957
(3, 0.7, 0.5)	0.10	0.5	12.7 (2.5)	0.825 (0.097)	1.000
(3, 0.7, 0.5)	0.10	0.8	26.6 (9.5)	0.985 (0.020)	0.901
(3, 0.7, 1.0)	0.10	0.5	9.5 (1.4)	0.953 (0.031)	1.000
(3, 0.7, 1.0)	0.10	0.8	22.2 (8.1)	0.999 (0.002)	1.000
(3, 0.9, 0.5)	0.05	0.5	16.5 (5.5)	0.657 (0.225)	1.000
(3, 0.9, 0.5)	0.05	0.8	25.1 (7.7)	0.913 (0.093)	0.796
(3, 0.9, 1.0)	0.05	0.5	14.6 (4.3)	0.934 (0.069)	1.000
(3, 0.9, 1.0)	0.05	0.8	24.3 (8.0)	0.997 (0.008)	0.825
(3, 0.9, 0.5)	0.10	0.5	13.9 (4.2)	0.651 (0.211)	1.000
(3, 0.9, 0.5)	0.10	0.8	22.6 (8.0)	0.909 (0.090)	0.839
(3, 0.9, 1.0)	0.10	0.5	12.3 (3.3)	0.926 (0.073)	1.000
(3, 0.9, 1.0)	0.10	0.8	21.9 (8.2)	0.996 (0.008)	0.863

determine $\tilde{\boldsymbol{\theta}}$. In each repetition, a set of background one-way ANOVA F-statistics $\acute{\mathcal{F}}$ was generated according to the assumed setting and then Algorithm 2 was used to determine $\tilde{\boldsymbol{\theta}}$ and Algorithm 1 used to determine a sample size n^*. Each repetition yielding a feasible sample size ($n^* > 50$) generated a set of F-statistics $\mathcal{F}$, computed the corresponding p-values, applied the q-value procedure and tabulated the number of tests resulting in each of the four outcomes. For each setting, the results were summarized across repetitions with a feasible sample size.

Table 2 gives the results of the second simulation series. In most cases, the simulation estimates of the expected sample size (conditioned on a feasible sample size) were similar to or greater than what would be obtained if the true $\boldsymbol{\theta}$ were used. In all cases considered, the expected value (given a feasible sample size n) of the proportion D of differentially expressed probes declared significant exceeds the desired average power δ. The properties of Q were consistent with previously proven control properties of the q-value procedure (data not shown; Storey, 2002).

4.2 A real-data simulation

The above simulations show the utility of Algorithms 1 and 2 only in the considered settings (Mehta *et al*., 2004). Some aspects of the considered settings are unrealistic, particularly the assumption that the one-way ANOVA F-tests are statistically independent. Therefore, we conducted what we term a real-data simulation to explore the performance of our method in a more realistic setting. We first give the background about the dataset used in the real-data simulation, then explain how the real-data simulation was performed and finally describe the results.

Ross *et al*. (2004) profiled the gene expression of pediatric acute myleoid leukemia in diagnostic bone marrow samples. Numerous objectives were considered by Ross *et al*. (2004); here we use the dataset to explore the utility of Algorithms 1 and 2 for determining the sample size necessary to have at least 50% average power to identify features that are differentially expressed across three disease subtypes (core-binding karyotype, MLL rearrangement and others) while keeping the pFDR at or below 5%. The real data simulation used the log-transformed signals obtained from the Microarray Analysis Software 5.0 normalization algorithm (Affymetrix, 2002, (www.affymetrix.com). Pounds and Cheng (2005) have questioned the value of probe filtering in the analysis of microarray data; thus no probe filtering was applied.

In each repetition, a set of $\acute{n} = 4$ samples were drawn from each class with replacement to serve as a background dataset $\acute{\Delta}$.

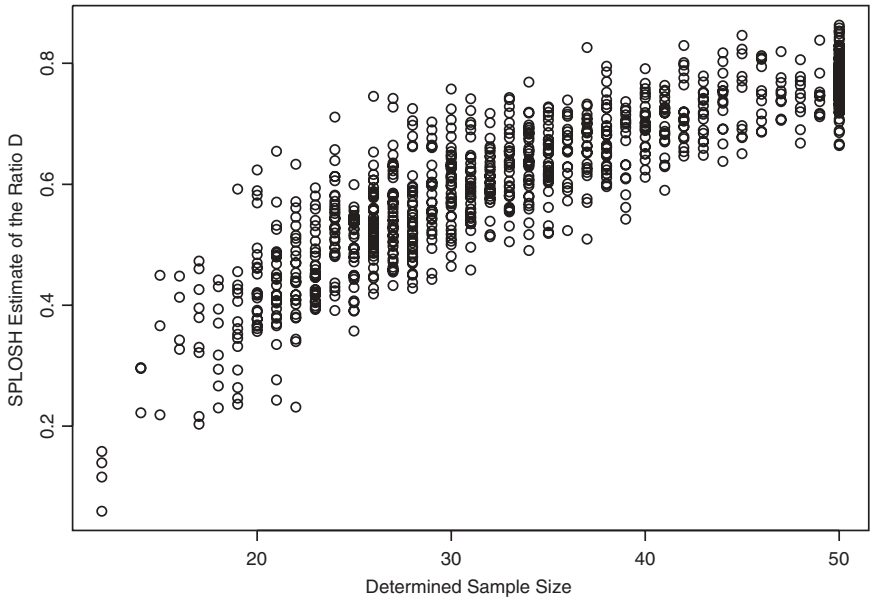

Fig. 1. Real-data simulation results. Each point corresponds to a repetition of the real-data simulation.

Algorithm 2 was then applied to $\acute{\Delta}$ to determine the sample size n^*. A sample of $n = \min(n^*, 50)$ was then drawn from each class with replacement to form a dataset Δ. One-way ANOVA and the a-value procedure were applied to Δ to determine significance and SPLOSH (Pounds and Cheng, 2004) was used to estimate the proportion D of truly differentially expressed features that were declared significant. Results were then summarized across repetitions.

The mean of the SPLOSH estimates of the proportion of truly differentially expressed features declared significant was 0.611. This is greater than the desired $\delta = 0.5$. Furthermore, the SPLOSH estimate of the proportion of truly differentially expressed features declared significant was greater than $\delta = 0.5$ in 794 of the 1000 repetitions. In 148 repetitions out of 1000 (14.8%), Algorithms 1 and 2 found $n^* > 50$, but $n = 50$ was used instead. In each of these 148 repetitions, the SPLOSH estimate of the proportion of differentially expressed features declared significant was greater than the desired average power δ.

Figure 1 shows the results of each replication of the real-data simulation. The figure shows the variability of the sample size estimates produced by Algorithm 2 across generated background datasets and the variability of the SPLOSH estimate of the ratio D across repetitions yielding equal sample size estimates. The figure suggests that the average power estimates increase with sample size, as would be expected under standard statistical theory. These results suggest that Algorithms 1 and 2 perform well in this real-data setting.

5 DISCUSSION

We have introduced the aFDR and anticipated average power as conceptual tools to mathematically frame the problem of sample size determination for an experiment using the FDR, pFDR or cFDR as the final measure of statistical significance. Furthermore, we have developed a general algorithm (Algorithm 1) to solve the sample size problem as posed in terms of the aFDR and average power. This algorithm is very general and can potentially be implemented for any setting in which one can find an existing power formula for each hypothesis test to be performed. We have derived details to implement Algorithm 1 for the specific setting of using one-way ANOVA to identify features that are differentially expressed across k experimental groups. The method could be extended to compute the sample size when non-parametric methods, such as the Kruskal–Wallis test, are used by considering the asymptotic relative efficiencies of those procedures (Hettmansperger, 1984). With minimal efforts, application-specific details could be derived for many other types of experiments. For example, to determine the sample size needed to accurately identify features associated with survival, we could adapt published methods for determining the sample size for a Cox regression analysis (Hsieh and Lavori, 2000) for use in Algorithm 1. Similarly, if expression is to be examined in a linear or logistic regression analysis, we could use power formulas for those methods (Hseih *et al.*, 1998) in Algorithm 1 as well. Finally, the algorithm does not require computationally involved and intensive calculations such as Markov chain Monte Carlo simulations or the bootstrap. The ease of implementation should facilitate the rapid development of application-specific details for many classes of experiments.

We derived application-specific details to implement Algorithm 1 for the purpose of planning an experiment in which the objective is to identify features that are differentially expressed across $k \geq 2$ experimental groups. To our knowledge, Algorithm 1 is the first to be shown useful for planning an experiment other than a simple two-group comparison. Our algorithm performs very well in our traditional simulation studies, when the parameter values $\tilde{\boldsymbol{\theta}}$ are correctly specified by the user. Excellent performance was also noted when Algorithm 2 is applied to background data to determine a value for $\tilde{\boldsymbol{\theta}}$. Additionally, Algorithms 1 and 2 performed well in the real-data simulation; in almost 80% of the repetitions, the SPLOSH estimate of the proportion of differentially expressed features identified as significant exceeded the desired average power of 50%.

These simulation studies suggest that the algorithm will be dependable in practice, at least for planning k-group comparisons.

Additionally, we have considered some components of the sample size problem not explored by previous works proposing methods to determine the sample size for studies using the FDR in the final analysis. We have proposed a simple and an effective way to adjust estimates of effect size for multiplicity (Algorithm 2). We have made some initial progress in mathematically expressing the relationship between statistical power and the properties of the estimate $\hat{\pi}$ of the proportion π of tests with a true null hypothesis. Moreover, we have used a real-data simulation as a technique to examine the effectiveness of a proposed method in practice. Others (Jung, 2005; Hu *et al*., 2005) have used simulations of data with block correlation structures to study the reliability of their methods under those types of dependency. It would be of interest to study the performance of our method under those correlation structures as well. It would also be interesting to compare the performance of the various methods in traditional and real-data simulation studies.

The performance of Algorithm 1 may depend on the choice of the procedure Ω used to control the FDR, pFDR or cFDR in the final analysis. Thus, we included reference to the procedure Ω in our formulas and algorithms. We only explored the performance of our algorithm when Storey's (2002) q-value is used for the final analysis. Nevertheless, we anticipate that the algorithm will perform well for the family of methods that operate on p-values by using Equations (5) and (6), because of the striking similarities of these methods. We have not yet explored the performance of Algorithm 1 in planning an experiment when resampling-based methods, such as that of Yekutieli and Benjamini (1999), are used to control the FDR. These resampling methods are most useful when the results of the individual tests are strongly correlated. However, in the absence of alternative methods developed for these settings, Algorithm 1 may still prove to be a useful tool for purposes of experimental planning. If dependency among test statistics is of special concern, Benjamini and Yekutieli (2001) have shown that a simple modification of the original FDR control procedure (Benjamini and Hochberg, 1995) can control the FDR under any dependence structure. This modification can be stated as scaling the ratios in (5) by the constant $\sum_{i=1}^{m}(i^{-1})$. Therefore, it is quite possible that this constant can be incorporated into Algorithm 1 to reliably determine the sample size for an experiment that will use the Benjamini and Yekutieli (2001) procedure to control the FDR in the final analysis. The utility of Algorithm 1 for planning experiments in such settings is yet to be explored.

The performance of Algorithm 1 may also depend on the choice and basis of the power formula for the individual hypothesis tests. For example, if a large-sample power approximation is used to compute the power of the individual tests then the sample size estimates produced by Algorithm 1 should be considered reliable only when those power approximations hold.

The performance of Algorithm 2 for using background data to determine $\tilde{\boldsymbol{\theta}}$ may depend on the choice of the procedure applied to the background p-values to obtain $\acute{\pi}$ and $\acute{q}_1, \ldots, \acute{q}_m$. In this paper, we selected SPLOSH for this purpose because Pounds and Cheng (2004) have shown that it provides accurate and stable estimation of the cFDR in simulation studies. Although Storey and Tibshirani (2003) have hinted that the q-value can be interpreted as an estimate of the pFDR, Storey (2002) has only shown the q-value to be an effective pFDR control procedure. Pounds and Cheng (2004) have described how the operation defined by (6) introduces bias to the ratios in (5), which Benjamini and Hochberg (2000) suggest are reasonable estimates of the FDR. The bias is clearly downward, therefore, interpreting the q-value [or any other value obtained from a right-sided minimization operation as in (6)] as an estimate of the FDR may tend to understate the actual prevalence of false positives in the set of results declared significant. However, Storey (2002) has proven desirable properties of the q-value when used as a control procedure, and the operation in (6) clearly gives Storey's procedure greater power than SPLOSH when both are applied as control procedures. This insight motivates our choices to use SPLOSH to compute (6) for purposes of adjusting background estimates of the effect size for multiplicity and to use the q-value as the control procedure in the final analysis. Further research is needed to determine the relative value of the various FDR, pFDR and cFDR procedures for Step 3 of Algorithm 2.

The aFDR differs from the realized FDR described by Genovese and Wasserman (2002). Given a fixed p-value significance threshold α and sufficient information to determine the truth of each null hypothesis and the power of each hypothesis test, the aFDR represents the ratio of the expected number of false positives to the expected number of significant results. As such, the aFDR is simply a conceptual tool for performing power and sample size calculations. In this respect, the aFDR differs markedly from the realized FDR, which is the ratio of the number of false positives to the number of significant results for a particular realization of an experiment.

Some simple modifications may improve the performance of Algorithm 1. As presented, Algorithm 1 computes an anticipated null proportion estimate $\tilde{\pi}$, which is a function of the sample size n. Conceptually, however, the null proportion π is a population parameter that does not depend on the sample size. Therefore, it may be worthwhile to consider letting $\tilde{\pi}$ equal the background estimate $\acute{\pi}$ for all n. This modification would simplify the calculations and should lead to slightly smaller sample size estimates because $\tilde{\pi}(n) \geq \acute{\pi}$ for all n (section S5, Supplementary materials). However, the performance of the modified algorithm needs to be evaluated in further research.

ACKNOWLEDGEMENTS

We wish to thank Dr Angela McArthur for editorial assistance. This research was supported in part by the NIH PAAR Group grant U01 GM-061393 (C.C.), the NIH Cancer Center Support Grant CA-21765 (S.P. and C.C.) and the American Lebanese Syrian Associated Charities (S.P. and C.C.).

Conflict of Interest: none declared.

REFERENCES

Affymetrix (2002) Statistical algorithms description document.

Allison,D.B. *et al.* (2002) A mixture model approach for the analysis of microarray gene expression data. *Comput. Stat. Data Anal.*, **39**, 1–20.

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B*, **57**, 289–300.

Benjamini,Y. and Hochberg,Y. (2000) On the adaptive control of the false discovery rate in multiple testing with independent statistics. *J. Edu. Behav. Stat.*, **25**, 60–83.

Benjamini,Y. and Yekutieli,D. (2001) The control of the false discovery rate in multiple testing under dependency. *Ann. Stat.*, **29**, 1165–1188.

Cheng,C. *et al.* (2004) Statistical significance threshold criteria for analysis of microarray gene expression data. *Stat. Appl. Gene. Mol. Biol.*, **3**, e36.

Cui,X.Q. and Churchill,G.A. (2003) Springer, How many mice and how many arrays? Replication in mouse cDNA microarray experiments. In Johnson,K.F. and Lin,S.M. (eds), Kluwer Academic Publishers, Norwell, MA. *Methods of Microarray Data Analysis III*, 139–154.

Gadbury,G.L. *et al*. (2004) Power and sample size estimation in high dimensional biology. *Stat. Meth. Med. Res*., **14**, 325–338.

Genovese,C. and Wasserman,L. (2002) Operating characteristics and extensions of the false discovery rate procedure. *J. R. Stat. Soc. B*, **24**, 499–517.

Hettmansperger,T.P. (1984) John Wiley & Sons, *Statistical Inference Based on Ranks*. New York.

Hseih,F.Y. *et al*. (1998) A simple method of sample size calculation for linear and logistic regression. *Stat. Med*., **17**, 1623–1634.

Hseih,F.Y. and Lavori,P.W. (2000) Sample-size calculations for the Cox proportional hazards regression model with nonbinary covariates. *Controlled Clinical Trials*, **21**, 552–560.

Hu,J. *et al*. (2005) Practical FDR-based sample size calculations in microarray experiments. *Bioinformatics*, **21**, 3264–3272.

Jung,S.-H. (2005) Sample size for FDR-control in microarray data analysis. *Bioinformatics*, **21**, 3097–3104.

Jung,S.-H. *et al*. (2005) Sample size calculation for multiple testing in microarray data analysis. *Biostatistics*, **6**, 157–169.

Lee,M.-L. and Whitmore,G. (2002) Power and sample size for microarray studies. *Stat. Med*., **11**, 3543–3570.

Liao,J.G. *et al*. (2004) A mixture model for estimating the local false discovery rate in DNA microarray analysis. *Bioinformatics*, **20**, 2694–2701.

Mehta,T. *et al*. (2004) Towards sound epistemological foundations of statistical methods for high-dimensional biology. *Nat. Genet*., **36**, 943–947.

Mukherjee,S. *et al*. (2003) Estimating dataset size requirements for classifying DNA microarray data. *J. Comput. Biol*., **10**, 119–142.

Müller,P. *et al*. (2004) Optimal sample size for multiple testing: the case of gene expression microarrays. *J. Am. Stat. Assoc*., **99**, 990–1001.

Pan,W. *et al*. (2002) How many replicates of arrays are required to detect gene expression changes in microarray experiments? A mixture model approach *Genome Biol*., **3**, e5.

Patnaik,P.B. (1949) The noncentral Chi-squared and *F*-distributions and their applications. *Biometrika*, **10**, 445–478.

Pounds,S. and Cheng,C. (2004) Improving false discovery rate estimation. *Bioinformatics*, **20**, 1737–1745.

Pounds,S. and Morris,S.W. (2003) Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of *p*-values. *Bioinformatics*, **19**, 1236–1242.

Reiner,A. *et al*. (2003) Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics*, **19**, 368–375.

Ross,M.B. *et al*. (2004) Gene expression profiling of pediatric acute Myelogenous Leukemia. *Blood*, **104**, 3679–3687.

Scheffe',H. (1959) John Wiley and Sons, *The Analysis of Variance*. New York.

Simon,R. *et al*. (2002) Design of studies using DNA microarrays. *Genet. Epidemiol*., **23**, 21–36.

Storey,J.D. (2002) A direct approach to false discovery rates. *J. R. Stat. Soc. B*, **64**, 479–498.

Storey,J.D. (2003) The positive false discovery rate: a Bayesian interpretation and the q-value. *Ann. Stat*., **31**, 2013–2035.

Storey,J.D. and Tibshirani,R. (2003) Statistical significance for genomewide studies. *Proc. Natl Acad. Sci. USA*, **100**, 9440–9445.

Tsai,C.-A. *et al*. (2003) Estimation of false discovery rates in multiple testing: application to gene microarray data. *Biometrics*, **59**, 1071–1081.

Tsai,C.-A. *et al*. (2005) Sample size for gene expression microarray experiments. *Bioinformatics*, **21**, 1502–1508.

Yekutieli,D. and Benjamini,Y. (1999) Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics. *J. Stat. Plan. Infer*., **82**, 171–196.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4272–4279
doi:10.1093/bioinformatics/bti708

Gene expression

The influence of missing value imputation on detection of differentially expressed genes from microarray data

Ida Scheel[1,*], Magne Aldrin[2], Ingrid K. Glad[1], Ragnhild Sørum[1], Heidi Lyng[3] and Arnoldo Frigessi[2,4]

[1]Department of Mathematics, University of Oslo, PO Box 1053, Blindern, NO-0316 Oslo, Norway, [2]Department of Statistical Analysis, Image Analysis and Pattern Recognition, Norwegian Computing Center, NO-0314 Oslo, Norway, [3]Department of Radiation Biology, The Norwegian Radium Hospital, NO-0310 Oslo, Norway and [4]Department of Biostatistics, University of Oslo, NO-0317 Oslo, Norway

Received on June 9, 2005; revised on September 20, 2005; accepted on October 5, 2005
Advance Access publication October 10, 2005

ABSTRACT

Motivation: Missing values are problematic for the analysis of microarray data. Imputation methods have been compared in terms of the similarity between imputed and true values in simulation experiments and not of their influence on the final analysis. The focus has been on missing at random, while entries are missing also not at random.

Results: We investigate the influence of imputation on the detection of differentially expressed genes from cDNA microarray data. We apply ANOVA for microarrays and SAM and look to the differentially expressed genes that are lost because of imputation. We show that this new measure provides useful information that the traditional root mean squared error cannot capture. We also show that the type of missingness matters: imputing 5% missing not at random has the same effect as imputing 10–30% missing at random. We propose a new method for imputation (LinImp), fitting a simple linear model for each channel separately, and compare it with the widely used KNNimpute method. For 10% missing at random, KNNimpute leads to twice as many lost differentially expressed genes as LinImp.

Availability: The R package for LinImp is available at http://folk.uio.no/idasch/imp

Contact: idasch@math.uio.no

Supplementary information: http://folk.uio.no/idasch/imp

1 INTRODUCTION

Missing values are a predominant problem for the analysis of microarray data, a high throughput technology to evaluate the expression of thousands of genes simultaneously (Lee, 2004). Missing values arise due to technical failure, low signal-to-noise ratio and measurement error (Lee, 2004; Wit and McClure, 2004). Typically ∼1–10% of the data are missing (de Brevern *et al*., 2004), affecting up to 95% of the genes. Many available algorithms for the statistical analysis of microarray data require a full dataset (Wit and McClure, 2004), because the underlying statistical methodology is based on balanced data. This includes for example SAM (Troyanskaya *et al*., 2001, www-stat.stanford.edu/~tibs/SAM), PAM (Tibshirani *et al*., 2001, www-stat.stanford.edu/~tibs/PAM) and ANOVA for microarrays (Kerr *et al*., 2000), implemented in the software MAANOVA (www.jax.org/staff/churchill/labsite/software). Hence all missing values need to be imputed before, e.g. testing for differential gene expression between biological samples. The output of the analysis is seriously influenced by the quality of the applied imputation method: many of the differentially expressed genes are lost, and falsely new differentially expressed genes are generated, compared with the analysis of the true full dataset. Even the robust measures of family wise errors and false discovery rates cannot take this into consideration. In his comment to Sebastiani *et al*. (2003), Gary A. Churchill wrote 'Among the many small problems that have yet to be addressed in microarray analysis, missing data methods stand out in my mind as one of the more pressing'. de Brevern *et al*. (2004) studied the extent of missing values in eight published microarray experiments. There were between 0.8 and 10.6% missing values. Genes with at least one missing value ranged from 3.8 to 94.7%. Kim *et al*. (2005) studied a dataset from Gasch *et al*. (2001) originally containing 6361 genes and 156 experiments. After removing columns that had >8% missing values and removing the genes with one or more missing values, a matrix of dimension 2641×44 remained, a reduction of 88% of the data.

In this paper we concentrate on cDNA microarrays, which are microchips with more than ten thousands of spots each corresponding to a gene. On each spot hybridization of two samples happens, resulting in signals from two channels, one dyed red and the other green. Microarrays are then optically scanned: spots are detected from the background and red and green signal intensities are measured. Missing values originate from imperfections at the level of chip production and treatment, hybridization and scanning. Dust present on the chip, irregularities in the spot production and inhomogeneous hybridization all lead to spots which are manually or automatically flagged, and corresponding signals are then considered as missing. Because probes are printed on spots in random order, without consideration of their expected intensities, spatial noise effects, which are present, cannot be translated into spatially smooth expected intensities. In addition to such signals missing at random, available software flags out signals which cannot be distinguished from the background or have a too irregular form because the signal itself is too low. In these cases, values are missing not at random, the missingness depending on the signal intensity.

*To whom correspondence should be addressed.

As a typical example, the AGILENT feature extraction software G2567AA flags out signals when the intensity is extremely low with respect to signal intensities in other spots on the same array (called 'population outliers') or when the local background is highly irregular. Normally a mixture of missing at random and not at random will be present.

K-nearest neighbors (KNNimpute) (Troyanskaya *et al*., 2001) is the most commonly used imputation method. It is the only imputation method implemented in SAM, PAM and MAANOVA, and is therefore routinely applied. KNNimpute has been shown to impute values in a satisfactory way for up to 20% of missing log ratios if missingness is at random, see Troyanskaya *et al*. (2001). Their paper compared the imputed values with the true values in a simulated experiment, where spot ratios were erased at random. The same simulation and validation setup is used to investigate other competing imputation methods, among which are BPCA (Oba *et al*., 2003), LSimpute (Bø *et al*., 2004), GMCimpute (Ouyang *et al*., 2004) and LLSimpute (Kim *et al*., 2005). Feten *et al*. (2005) also compare imputed values with the true values, though in a more refined way than the common root mean squared error (RMSE). While comparing imputed values with the true values is an important measure of performance, it fails to address the more fundamental question of what is the effect of such imputations on the final output of the statistical analysis. Only Ouyang *et al*. (2004) compared the number of mis-clustered genes for different methods.

In this paper we propose a simple and natural imputation method, LinImp, based on a linear model for each channel separately. We investigate its performance and compare it with KNNimpute when values are missing both at random and not at random. In the last case, we model the missingness depending on the signal. We evaluate the method by comparing the resulting list of differentially expressed genes based on the imputed dataset with the same list based on an analysis of the true full dataset. Hence we count how many of the genes in the list are lost and added when analyzing the imputed dataset. In our experiments 47–97% of the differentially expressed genes are lost if nothing is done when 10% of the data are missing. Up to 90% of this is recovered when imputing. KNNimpute shows up to three times as many lost differentially expressed genes as LinImp.

2 SYSTEMS AND METHODS

2.1 LinImp: linear model-based imputation

The imputation method we propose, LinImp, is based on the linear model for y_{ijkg}, the base 2 logarithm of the intensity in array i, channel (dye) j, variety k and gene g

$$y_{ijkg} = \mu + A_i + D_j + G_g + AD_{ij} + AG_{ig} + DG_{jk} + VG_{kg} + \varepsilon_{ijkg}, \quad (1)$$

where ε_{ijkg} are independent normally distributed error terms with mean zero and variance σ^2. For simplicity we assume that each gene is printed only once on each array, such that one gene is represented by only one spot on each array. The varieties are the experimental conditions under study, such as for example type of tissue. If we have a arrays, 2 channels (dyes) on each array, v varieties and N genes, then $i = 1, \ldots, a$, $j = 1, 2$, $g = 1, \ldots, N$ and $k = 1, \ldots, v$ and there are $2aN$ observations. μ is the overall mean, A_i is the effect of array i, D_j is the effect of dye j, G_g is the overall effect of gene g, AD_{ij} is the interaction between array i and dye j, AG_{ig} is the interaction between array i and gene g, DG_{jg} is the interaction between dye j and gene g and VG_{kg} is the interaction between variety k and gene g. Model (1) was proposed in Kerr *et al*. (2000) to find differentially expressed genes. Written in matrix form the model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (2)$$

where $\boldsymbol{y}$ is a vector of length $2aN$ and $\boldsymbol{X}$ is a matrix of zeros and ones. Denote $\boldsymbol{A}^{\mathrm{T}} = (A_1, \ldots, A_a)^{\mathrm{T}}$, $\boldsymbol{D}^{\mathrm{T}} = (D_1, D_2)^{\mathrm{T}}$ etc., then $\boldsymbol{\beta} = (\mu, \boldsymbol{A}^{\mathrm{T}}, \boldsymbol{D}^{\mathrm{T}}, \boldsymbol{G}^{\mathrm{T}}, \boldsymbol{AD}^{\mathrm{T}}, \boldsymbol{AG}^{\mathrm{T}}, \boldsymbol{DG}^{\mathrm{T}}, \boldsymbol{VG}^{\mathrm{T}})^{\mathrm{T}}$. Each pair of i and j corresponds to only one variety k. Because of this, the effects V and AD are confounded, so it is wise to have only one of the two in the model. We chose (1) with AD instead of V because it saturates the design space (Kerr *et al*., 2002).

LinImp works as follows. Let $\boldsymbol{Y}$ be the $N \times 2a$ observed data matrix with observed and missing values. We initialize the imputation (e.g. we used KNNimpute) with a full data matrix $\boldsymbol{Y}^0$. Then we estimate the parameter vector $\boldsymbol{\beta}$ in model (2) using the dataset $\boldsymbol{Y}^0$. Denote this estimated parameter vector as $\hat{\boldsymbol{\beta}}^0$. Next, we impute the missing values in $\boldsymbol{Y}$ with their expected values using (2) and $\hat{\boldsymbol{\beta}}^0$ to obtain the new full data matrix $\boldsymbol{Y}^1$. We iterate the procedure until convergence, for example until at iteration M in some norm $\|\boldsymbol{Y}^M - \boldsymbol{Y}^{M-l}\| < \delta$, where δ is a fixed small value. The full data matrix $\boldsymbol{Y}^{\mathrm{imp}} = \boldsymbol{Y}^M$ is then the final imputed dataset. The pseudocode of the algorithm is given in Figure 1. Running LinImp requires a couple of minutes for the largest dataset in this paper. LinImp is practically an automatic imputation method. Of course the small value δ must be chosen, but to our experience the results are very robust with respect to this choice. Also, KNNimpute is a reasonable and simple choice for initial imputation. Linear model imputation in general has been proposed before, see for instance Pyle (1999). Notice that estimating $\boldsymbol{\beta}$ in (2) is easy when data are complete because then it is possible to simplify the estimation algorithm. In the case of missing values, and hence unbalanced design, the estimation of $\boldsymbol{\beta}$ becomes a formidable computational task.

1: Linear model: $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$
   $\boldsymbol{Y}$: Observed data matrix of dimension $N \times 2a$ with $2aN - n$ observed values and n missing values.
   k, l: Vectors such that the jth missing value in $\boldsymbol{Y}$ is located in the k_jth row and the l_jth column.
   δ: The convergence criterion, preset in the code.
2: Initialize: $\boldsymbol{Y}^0$ (imputed data set)
   (Initial imputation done by for example KNNimpute)
3: Initialize: $i \leftarrow 1$
4: Initialize: convergence=FALSE
5: **while** convergence=FALSE **do**
6: $\quad \boldsymbol{Y}^i \leftarrow \boldsymbol{Y}$
7: $\quad$ Fit the linear model for $\boldsymbol{Y}^{i-1}$ to obtain $\widehat{\boldsymbol{\beta}}^{i-1}$
8: $\quad$ **for** $j = 1$ to n **do**
9: $\quad\quad \boldsymbol{Y}^i_{k_j l_j} \leftarrow E[\boldsymbol{Y}^{i-1}_{k_j l_j} | \boldsymbol{\beta}^{i-1} = \widehat{\boldsymbol{\beta}}^{i-1}]$
10: $\quad$ **end for**
11: $\quad$ **if** $\|\boldsymbol{Y}^i - \boldsymbol{Y}^{i-1}\| < \delta$ then
12: $\quad\quad$ convergence=TRUE
13: $\quad\quad \boldsymbol{Y}^{\mathrm{imp}} \leftarrow \boldsymbol{Y}^i$
14: $\quad$ **else**
15: $\quad\quad i \leftarrow i + 1$
16: $\quad$ **end if**
17: **end while**

Fig. 1. Pseudocode for the algorithm for LinImp.

2.2 KNNimpute

KNNimpute (Troyanskaya *et al*., 2001) is widely used, for instance it is the only imputation method available in SAM, PAM and MAANOVA. Hence it is important to analyze the effect KNNimpute has on detecting differentially

expressed genes. KNNimpute works as follows. For each row i in the data matrix, corresponding to gene g, with one or more missing values, the k nearest neighbor rows are found. It is necessary that the k nearest neighbors have data in the columns where row g had missing data. To define a neighborhood structure between rows, a metric is necessary. The distance $d_{gg'}$ from row g to row g' is the Euclidean distance of the two vectors omitting the entries for which row g and row g' have missing values. If there are one or more missing entries in row g' in places where row g has non-missing entries, the squared difference for these entries is set to the average of the squared difference for the non-missing entries. When the k nearest neighbors are found, the missing entry in column c in row g is imputed as the weighted average of the values in column c in the k-nearest neighbor rows, the weights being the inverse distances.

There seems to be some confusion in the literature about the KNNimpute algorithm. Some authors (Lee, 2004; Ouyang *et al*., 2004) describe an algorithm where neighbors are not allowed to have any missing values. This can create problems in datasets with a lot of missing values because only a few, or none, neighbors actually are free of missing values and the imputation becomes impossible or poor. Others, for instance Oba *et al*. (2003), describe algorithms where the neighbors are allowed to have missing values, but the corresponding missing differences are not imputed when calculating the distance $d_{gg'}$. This will cause falsely low distances for neighbors with a lot of missing values. These versions of KNNimpute are too simplistic and less efficient than full KNNimpute, which is used in our comparisons. In this paper we use the KNNimpute implementation available in the R package `impute` (by Hastie, T., Tibshirani, R., Narasimhan, B. and Chu, G., available at http://cran.r-project.org/). This implementation does not suffer from any of the abovementioned weaknesses.

While traditionally KNNimpute is applied to the $N \times a$ data matrix of log ratios of intensities, in this paper we apply KNNimpute to the $N \times 2a$ data matrix of log intensities.

2.3 Detecting differentially expressed genes

There are various ways of detecting differentially expressed genes. In this paper we use two such common approaches to evaluate imputation. When using the linear model (1), the quantity of interest is $VG_{1g} - VG_{2g}$ for determining if gene g is differentially expressed between varieties 1 and 2. For example, $VG_{1g} - VG_{2g} = 1$ equals a 2-fold change between the two tissues, because y_{ijkg} is the base 2 logarithm of the intensity. The linear model (1) is presented in Kerr *et al*. (2000) as ANOVA for microarrays, and it is implemented in MAANOVA. An alternative approach is based on hypothesis testing directly on the matrix of log ratios, as done by Significance Analysis of Microarrays (SAM) (Tusher *et al*., 2001).

2.4 Data

We have used two spotted cDNA microarray datasets for exploring the new imputation method and the overall recovery rate of missing differentially expressed genes. LinImp imputes missing values separately in each channel, and hence uses the channel data directly, not their log ratios. Such data are more rarely published. The first dataset is based on a study of human cell lines and the other dataset is typical for clinical studies on primary tumors. The intensities and log ratios are generally higher in experimental studies on cell lines than in clinical studies based on primary tumors.

The dataset based on human cell lines is composed of three dye-swaps, thus six arrays. The data are from the NIEHS experiment comparing treated and control human cell lines, as described in Kerr *et al*. (2002). It is publicly available at http://www.jax.org/staff/churchill/labsite/datasets/expression/niehs. The dataset based on primary tumors is based on samples from cervical tumors before and after radiotherapy and is composed of 16 dye-swaps and thus 32 arrays and is available from our Supplementary information web page. In the NIEHS dataset there were 1907 genes and no missing values, thus a full intensity data matrix of dimension 1907×12. In the original cervical cancer dataset 22% of the data were missing, affecting 70% of the 14 229 genes. We have removed the genes with one or more missing values, leaving data from 4246 genes. The resulting intensity data matrix is of dimension 4246×64. These truths are then used as bases for simulating missing values.

When genes with at least one missing value are removed from the analysis, the effect can be dramatic. When missing at random, the percentage of lost differentially expressed genes will be approximately the same as the percentage of genes with one or more missing values. When missing not at random, the percentage of lost differentially expressed genes can be even higher since genes that are differentially expressed are more likely than others to have missing values.

2.5 More realistic models of the missingness

A value in a data matrix is missing at random, MAR, or missing completely at random (MCAR), if the probability of it being missing does not depend on the value that is missing. A value is missing not at random, MNAR, if the probability of it being missing is dependent on the value that is missing. When missing at random, usually both channel signals from the spot are missing, which means that we are in the MAR situation. A basic reason for microarray data to be missing not at random is that the foreground intensity is lower than the background intensity. Another reason is that low intensities are per se sometimes considered too noisy and conservatively flagged out. Missing not at random happens most often for just one of the two channel signals in a spot. Of course this means that when analyzing log ratios the whole spot is missing. We have separated the analysis of imputation of values missing at random and not at random in order to see differences in the effects of imputation for the two types of missing.

When simulating datasets with values missing at random, we have assumed both channels from the spot to be missing. Therefore, to simulate a total of r% of the entries missing at random, we have drawn $r/2$% of the spots at random and made both channel signals missing. For both the cervical cancer data and the NIEHS data we have chosen the missing percentages r to be 1, 5, 10, 15, 20, 25, 30, 35 and 40. We have simulated 50 independent missing datasets for each percentage missing.

Our mechanism creating values missing not at random favors missingness of low intensities. We proceed as follows. For each spot the lowest of the two signals is considered. These lowest signals are ordered and the s% percentile is found, say 5%. We then produce a dataset with r% of the total number of entries missing (say 1%) by drawing at random exclusively from below the s% percentile and making the lowest channel signal from these spots missing. A histogram of the lowest base 2 log intensity for each spot for both datasets can be seen in Figure 2. On the basis of this we have chosen the threshold $s = 25$ for the NIEHS dataset and for the cervical cancer dataset the threshold $s = 5$. For the NIEHS dataset the missing percentages r run over 1, 2.5, 4, 5.5, 7, 8.5, 10, 11.5 and 13. For the cervical cancer dataset the missing percentages r run over 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 and 5.

2.6 Measures of performance of imputation methods

Imputation methods for microarray data are discussed in the literature in terms of the RMSE, where the error is the difference between the imputed value and the true one. Troyanskaya *et al*. (2001) normalize the RMSE by dividing it by the average value over all observations in the true full dataset, which is useful because it enables comparisons across different datasets. This is denoted by NRMSE and is adopted in this paper. Oba *et al*. (2003) and Kim *et al*. (2005) normalize the RMSE by dividing it by the standard deviation of the values in the true full dataset. This measure is denoted here by NRMSE2. Ouyang *et al*. (2004) normalize the RMSE by dividing it by the root mean square of all the observations in the true full dataset. This measure is denoted here by NRMSE3. Bø *et al*. (2004) do not normalize.

Unfortunately none of the various RMSE measures describe the real effect of imputation on the final analysis. We are interested in evaluating the effect imputation has on the final output of the statistical analysis in question, and a different measure is needed. A typical end-product of a statistical analysis is a list of interesting genes. How is such a list affected by the errors of imputation? A way to produce such a list is by hypothesis

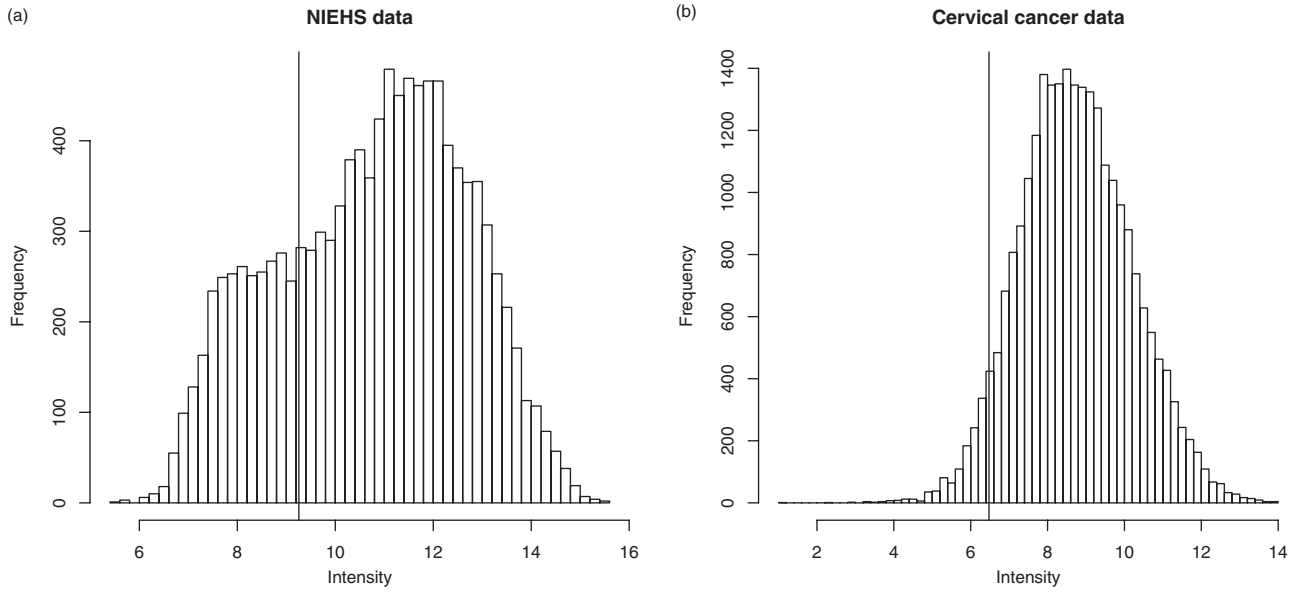

Fig. 2. A histogram of the lowest log2 intensity for each spot for the NIEHS dataset with a vertical line indicating the 25% quantile (**a**) and a histogram of the lowest log2 intensity for each spot for the cervical cancer dataset with a vertical line indicating the 5% quantile (**b**).

testing, using for example the linear model (1) as in Kerr *et al*. (2000). Here we measure the success of imputation by looking to lost and added differentially expressed genes compared with the list of differentially expressed genes from an analysis of the true full dataset. That is we look for genes which would be on the list if we knew the true full dataset but are lost due to imputation errors and genes which enter the list by mistake again as an effect of imputation errors.

When using the linear model (1) to analyze a dataset with two different varieties, we test for each gene g if $VG_{1g} - VG_{2g}$ significantly differs from 0. To facilitate comparison of methods, we fixed the length of the list of differentially expressed genes for both datasets to 100. When the list length is fixed, the numbers of lost and added differentially expressed genes are the same.

In addition we evaluate the effect of intensity-based imputation when analyzing ratio datasets. For that we used the methods of Tusher *et al*. (2001) implemented in SAM. When using SAM for testing on one-class data each gene is assigned a score, the average log ratios for that gene divided by a sum of the standard deviation for that gene and a small positive constant. We also here fixed the length of the lists to 100. This gave an estimated FDR of 1% for the NIEHS dataset.

The lists of length 100 of differentially expressed genes for the true full NIEHS dataset when analyzing using the linear model and using SAM agree for 97 genes.

3 RESULTS

In Figure 3 we compare the percentage of lost differentially expressed genes when analyzing the datasets using the linear model (1). The figure is based on the average of the 50 runs, and results for both LinImp and KNNimpute are shown. The ratios between the average percentage lost genes for KNNimpute and LinImp can be seen at the top of the plots. The averages of the percentages of potentially lost differentially expressed genes are also shown at the top of the plots. That is the differentially expressed genes based on the true full dataset that have one or more missing values in the simulated dataset and thus are lost if genes with one or more missing values were deleted.

Imputing missing values clearly makes a vast improvement on identifying differentially expressed genes with the linear model (1). For the NIEHS dataset when 10% of the data are missing at random, at least 47% of the differentially expressed genes would be missing if instead of imputing, genes with one or more missing values were deleted. By imputing with KNNimpute 80.8% of these genes are recovered and 89.4% by imputing with LinImp. When 10% of the NIEHS data are missing not at random, at least 48.3% of the differentially expressed genes would be missing if genes with one or more missing values were deleted. By imputing with KNNimpute 75.1% of these genes are recovered and 83.4% by imputing with LinImp. For the cervical cancer dataset when 10% of the data are missing at random, at least 97.1% of the differentially expressed genes would be missing if genes with one or more missing values were deleted. By imputing with KNNimpute 89.7% of these genes are recovered and 88.7% by imputing with LinImp. When 5% of the cervical cancer data are missing not at random, at least 62% of the differentially expressed genes would be missing if genes with one or more missing values were deleted. By imputing with KNNimpute 62.9% of these genes are recovered and 74.2% by imputing with LinImp.

For the NIEHS data, LinImp outperforms KNNimpute for all missing percentages, for both missing at random and missing not at random. KNNimpute shows 50–100% more lost differentially expressed genes than LinImp. For the cervical cancer data, the results are quite similar for LinImp and KNNimpute for missing at random, but for missing not at random KNNimpute shows 20–40% more lost differentially expressed genes than LinImp. Of course, LinImp might have an advantage with respect to KNNimpute since analysis is done by the same model used for imputation. Still, if the linear model is to be used for the analysis, the comparison is fair. Because of the possible advantage LinImp has when analyzing by the linear model, we also did an analysis using SAM, which is done on log ratio data. As an example of alternatives to KNNimpute, here we also imputed using LSimpute

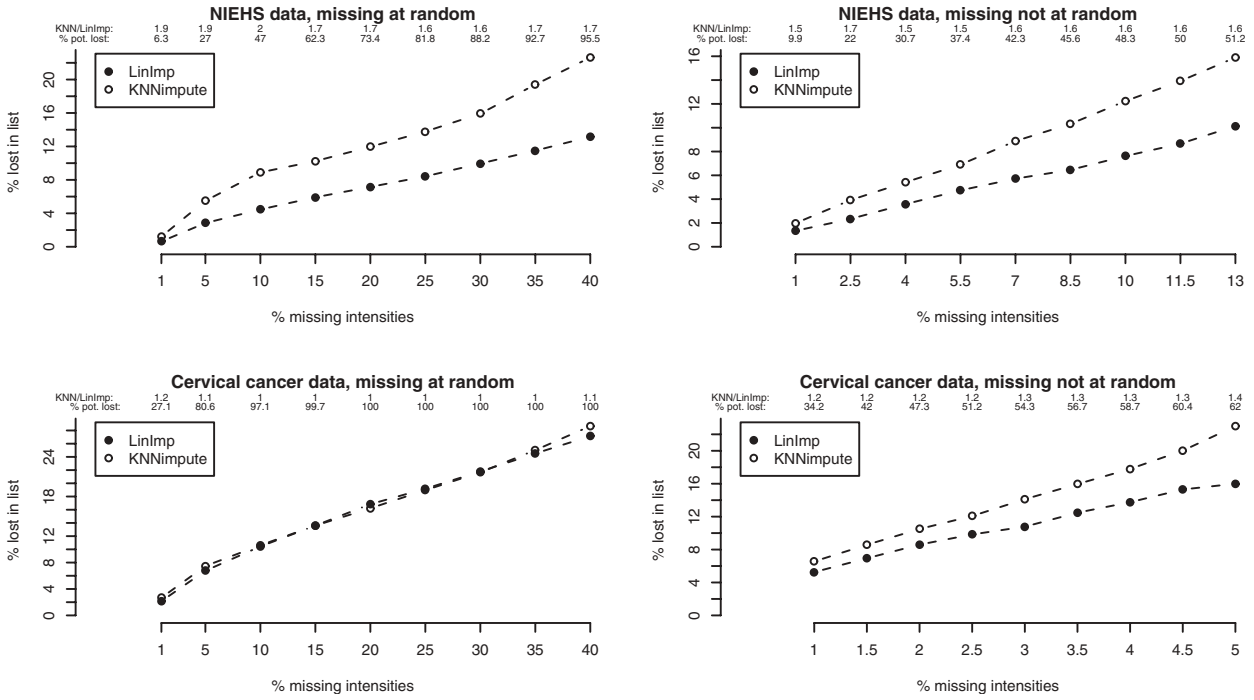

Fig. 3. Percentage lost differentially expressed genes when analyzing the datasets by using the linear model, the averages of the 50 runs. At the top of each plot ratios between the average percentage lost for KNNimpute and LinImp are shown, as well as the average percentage potentially lost differentially expressed genes when imputation is not done.

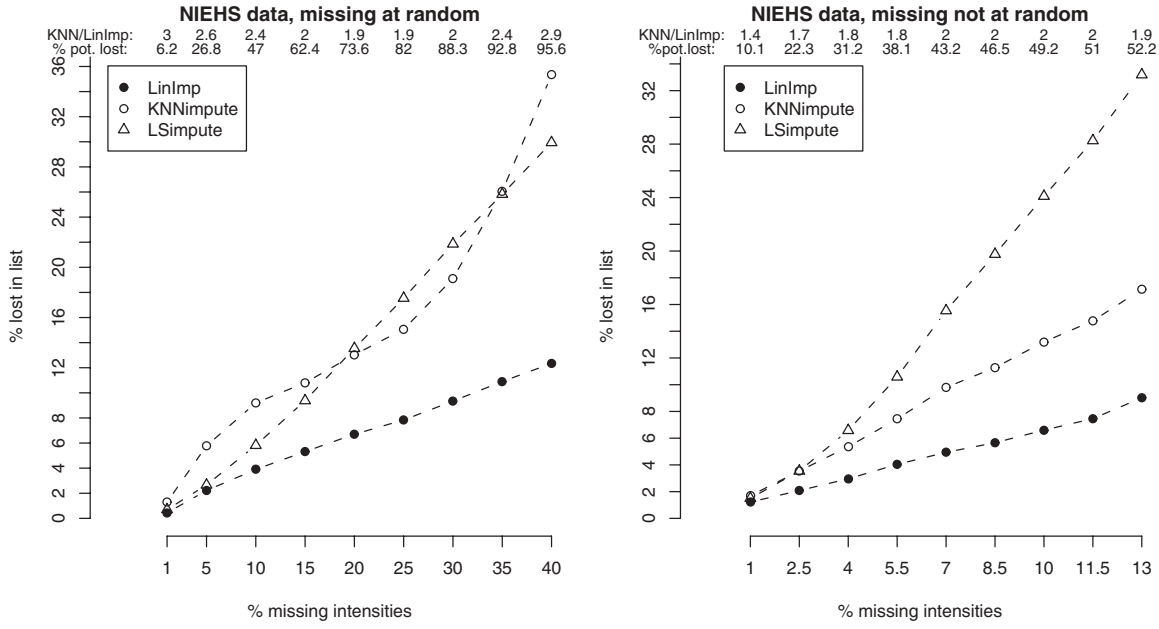

Fig. 4. Percentage lost differentially expressed genes when analyzing the NIEHS data by using SAM, the averages of the 50 runs. At the top of the plots ratios between the average percentage lost for KNNimpute and LinImp are shown, as well as the average percentage potentially lost differentially expressed genes when imputation is not done.

(Bø *et al*., 2004). LSimpute is an imputation method for log ratio data which utilizes the correlation structure. It is based on regression with non-missing log ratios as explanatory variables. Note that LinImp is based on a different type of regression model with the explanatory variables describing the experimental conditions. In Figure 4 we compare the percentage of lost differentially expressed genes when analyzing the NIEHS dataset using SAM. LinImp performs better than KNNimpute for the NIEHS data also when

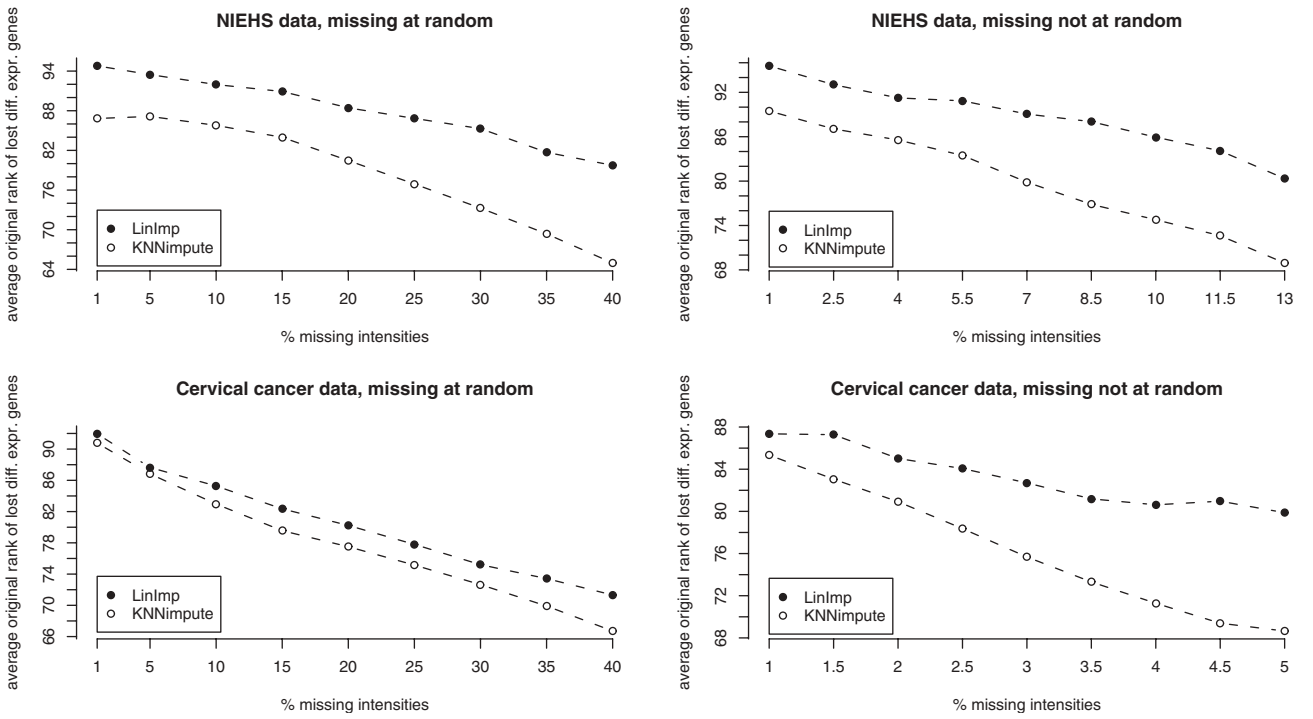

Fig. 5. The average of the rank the genes that are lost have in the list based on the true full dataset (original rank), when analyzing by using the linear model. The plot shows the average of the 50 runs.

analyzing with SAM instead of the linear model. LSimpute shows better results than KNNimpute for low percentages missing at random, but worse for missing not at random.

For the same percentage of missing, the percent lost differentially expressed genes is higher for missing not at random than for missing at random. This is the case for both analysis methods and both imputation methods. For the NIEHS data the results for 10% missing not at random are approximately the same as the results for 20% missing at random, for both imputation methods. For the cervical cancer data the results for 5% missing not at random are approximately the same as the results for 10% missing at random when imputing with LinImp. When imputing with KNNimpute the results for 5% missing not at random are approximately the same as the results for 30% missing at random. The reason for the difference between missing at random and not at random is that in simulating missing not at random genes that have low values have a higher probability of having missing values than other genes. When a gene that is differentially expressed when comparing two varieties is very low expressed for one of the two varieties, all the intensities of that particular gene for that variety, that is half of all the intensities for that gene, are likely to be very low and thus missing at the same time. This results in a lot of imputed values for that gene and makes it more vulnerable in the analysis. It also indicates that imputing low values is difficult for both imputation methods, even though LinImp does a better job than KNNimpute for missing not at random for both analysis methods.

How serious is the loss? Where are the genes that are lost located in the list of differentially expressed genes based on the true full dataset? In Figure 5 we have plotted the position of the lost genes in the list based on the true full dataset when analyzing using the linear model. Specifically we plot the average position the lost genes would have had in the list of differentially expressed genes if there were no missing values. The figure shows the average of the 50 runs for each percent missing. Rank 1 means top of the list and most significant and rank 100 means bottom of the list and least significant, thus the lower the number the more serious the loss is. As expected the curves decrease with increasing percentage. The fact that the curves of LinImp are always above those of KNNimpute shows that LinImp performs better than KNNimpute, which confirms Figure 3. Figure 5 also shows that the loss is more serious for missing not at random than missing at random for the same percentage missing. There is a dependency between Figures 5 and 3, because the more genes lost the higher the average position in the original list, but Figure 5 provides additional information. For example, for 1% missing at random and missing not at random in the NIEHS dataset, there is a clear difference between LinImp and KNNimpute in favor of LinImp, whereas in Figure 3 there is no difference. This means that the genes that are lost when imputing with KNNimpute are more significant in the original list than those that are lost when imputing with LinImp. The loss is less serious using LinImp than KNNimpute.

Since most studies use RMSE values in the evaluation of imputation performance, we have plotted the average of the NRMSE values for the 50 runs in Figure 6. Results for both LinImp and KNNimpute are shown. Also here LinImp outperforms KNNimpute. For the same percentage of missing, the NRMSE values for the cervical cancer data are better for missing at random than missing not at random, which coincides with the results from the lost differentially expressed genes. Still, the conclusions from Figures 3 and 4 are somewhat different from those of Figure 6. For the NIEHS

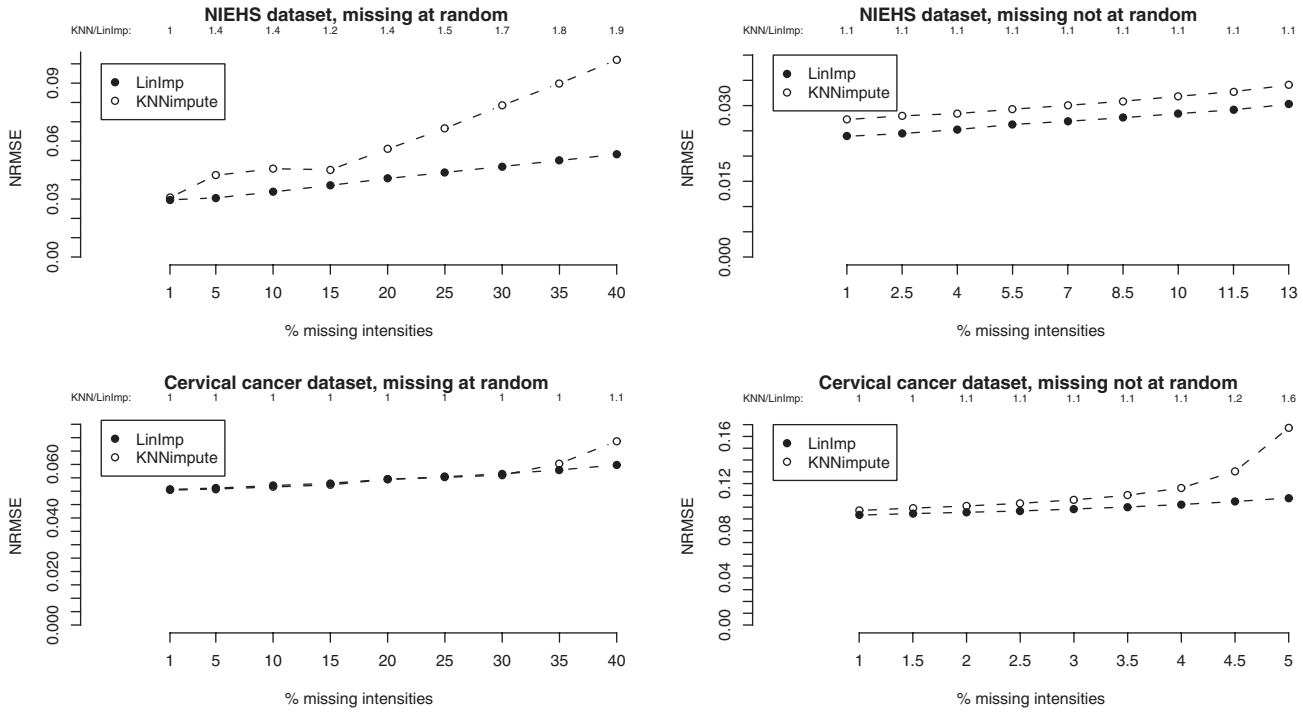

Fig. 6. Normalized RMSE, averages of the 50 runs.

data the NRMSE values are actually a bit better for missing not at random than missing at random. Whereas NRMSE is quite stable for both LinImp and KNNimpute for both datasets for missing not at random and for the cervical cancer data for missing at random, the increase in percentage lost is more dramatic. Also, the NRMSE values are 10% higher for KNNimpute than LinImp for the NIEHS data for missing not at random, whereas Figure 3 using the linear model implies that KNNimpute leads to up to 60% more lost differentially expressed genes than LinImp and Figure 4 using SAM implies that KNNimpute leads to up to twice as many lost differentially expressed genes than LinImp. The new way of evaluating imputation performance thus seems to provide useful information that NRMSE cannot capture.

Almost all the differences between the imputation methods in Figures 3, 4 and 6 are significantly larger than 0, even when correcting for multiple testing. This means that LinImp is significantly better than KNNimpute most of the time. The only exception is 1% missing and some of the other percentages missing at random for the cervical cancer data, which is not surprising in light of the figures. KNNimpute is never significantly better than LinImp. The average of the differences between the results of the imputation methods together with their estimated standard errors can be seen in Tables 1–3 available on our Supplementary information web page.

4 DISCUSSION AND CONCLUSION

Though KNNimpute is the most used imputation method several alternatives have been proposed, all for log ratio datasets. Oba *et al*. (2003) present a Bayesian principal component analysis approach, BPCA, based on an EM-like algorithm. Datasets with 1–20% entries missing at random are simulated from original full datasets and the performance is evaluated by computing the NRMSE2. BPCA shows better NRSME2 values than KNNimpute when the number of samples is large, though possibly a suboptimal version of KNNimpute was used. When the number of samples is <40 KNNimpute performs equally well or better than BPCA. Also, for one dataset the NRMSE2 for the BPCA is tripled from 1 to 20% missing values, while KNNimpute is much more stable. Zhou *et al*. (2003) investigate imputation based on linear and non-linear regression with Bayesian gene selection. The results are better for both versions compared with KNNimpute, though only 1 and 5% of missing data are investigated. Bø *et al*. (2004) show 15–20% smaller RMSE values for LSimpute than for KNNimpute for 10% entries missing at random. Ouyang *et al*. (2004) impute with GMCimpute, modeling data with a Gaussian mixture and using the EM algorithm. The number of mixture components is determined empirically. The simulation includes only very low missing probabilities in the range 0.003–0.04. The performance is evaluated by computing NRMSE3 and the number of mis-clustered genes. GMCimpute shows better results than KNNimpute, but the version of KNNimpute utilized requires the neighbors to be complete. It is possible to improve on this, so that KNNimpute could have performed better. Nguyen *et al*. (2004) compare KNNimpute to imputation via OLS and PLS regression with other genes as explanatory variables. The methods are evaluated by looking to the relative estimation error as a function of the true expression value. KNNimpute performs best near the median of the true expression values, while PLS seems best for the more extreme expression values. Kim *et al*. (2005) introduce a local least squares imputation method, LLSimpute, imputing a missing value for a gene by a linear combination of similar genes. It is called local because it uses only the most similar genes.

The method differs from LSimpute in that they use also the L_2-norm for determining similarity between genes, while LSimpute uses only the Pearson correlation. LLSimpute shows lower NRMSE2 values than KNNimpute and BPCA with 1–20% entries missing. Specifications on KNNimpute do not allow to understand whether KNNimpute has been implemented at best. Feten *et al*. (2005) investigate six imputation methods, four based on regression with other genes as explanatory variables and KNNimpute both with genes and observations as neighbors. The conclusion is that for datasets with strong correlation structure, KNNimpute with genes as neighbors performs best. LinImp outperforms KNNimpute when the linear model captures linear relationships within the log intensities that KNNimpute cannot capture. As Feten *et al*. (2005) concluded KNNimpute works well for highly correlated data. A possible improvement on LinImp for such datasets is to exploit the correlation between genes. When assuming uncorrelated data, as in LinImp, the expectation of the error term conditioned on information from other genes is 0. If the correlation structure would be considered, the multinomial distribution would give a conditional expectation for the error term different from 0. Conditioning should be done on information from other genes that are highly correlated with the gene which value is to be imputed. For computational reasons one cannot condition on information from all the other genes, thus it would not be completely automatic, since the number of correlated genes to consider when imputing for another gene would need to be decided.

In addition to erasing data at random Nguyen *et al*. (2004) also have an experiment where the probability of a gene having a missing value depends on the expression level. The conclusion is that the results are similar to the missing at random case. We have seen that imputing values that are missing not at random has a more serious effect on the final analysis than imputing values that are missing at random. Of course, in reality the missingness in a microarray dataset is a mixture of missing at random and missing not at random. We have introduced a new imputation method, LinImp. In most of our experiments we have found that LinImp performs better than the widely used KNNimpute, in particular when comparing resulting lists of differentially expressed genes. Finally, we conclude that looking to the actual effect imputation has on the final analysis gives valuable information in addition to the traditional RMSE.

Conflict of Interest: none declared.

REFERENCES

de Brevern,A.G. *et al*. (2004) Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering. *BMC Bioinformatics*, **5**, 114.

Bø,T.H. *et al*. (2004) LSimpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Res*., **32**, e34.

Feten,G. *et al*. (2005) Prediction of missing values in microarray and use of mixed models to evaluate the predictors. *Stat. Appl. Genet. Mol. Biol*., **4**, 10.

Gasch,A.P. *et al*. (2001) Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol. Biol. Cell*, **12**, 2987–3003.

Kerr,M.K. *et al*. (2000) Analysis of variance for gene expression microarray data. *J. Comput. Biol*., **7**, 819–837.

Kerr,M.K. *et al*. (2002) Statistical analysis of a gene expression microarray experiment with replication. *Stat. Sinica*, **12**, 203–217.

Kim,H. *et al*. (2005) Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, **21**, 187–198.

Lee,M.-L.T. (2004) *Analysis of Microarray Gene Expression Data*. Kluwer Academic Publishers, MA.

Nguyen,D.V. *et al*. (2004) Evaluation of missing value estimation for microarray data. *J. Data Sci*., **2**, 347–370.

Oba,S. *et al*. (2003) A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, **19**, 2088–2096.

Ouyang,M. *et al*. (2004) Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, **20**, 917–923.

Pyle,D. (1999) *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 275–297.

Sebastiani,P. *et al*. (2003) Statistical challenges in functional genomics. *Stat. Sci*., **18**, 33–70.

Tibshirani,R. *et al*. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl Acad. Sci. USA*, **99**, 6567–6572.

Troyanskaya,O. *et al*. (2001) Missing value estimation methods for cDNA microarrays. *Bioinformatics*, **17**, 520–525.

Tusher,V.G. *et al*. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl Acad. Sci. USA*, **98**, 5116–5121.

Wit,E. and McClure,J. (2004) *Statistics for Microarrays: Design, Analysis and Inference*. John Wiley and Sons Ltd, West Sussex, England, pp. 65–69.

Zhou,X. *et al*. (2003) Missing-value estimation using linear and non-linear regression with Bayesian gene selection. *Bioinformatics*, **19**, 2302–2307.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4280–4288
doi:10.1093/bioinformatics/bti685

Gene expression

A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data

Yang Xie[1,*], Wei Pan[1] and Arkady B. Khodursky[2]

[1]Division of Biostatistics, School of Public Health, University of Minnesota, Minneapolis, MN 55455, USA and
[2]Department of Biochemistry, Molecular Biology and Biophysics, University of Minnesota, St Paul, MN 55108, USA

Received on June 30, 2005; revised on September 2, 2005; accepted on September 20, 2005
Advance Access publication September 27, 2005

ABSTRACT

Motivation: False discovery rate (FDR) is defined as the expected percentage of false positives among all the claimed positives. In practice, with the true FDR unknown, an estimated FDR can serve as a criterion to evaluate the performance of various statistical methods under the condition that the estimated FDR approximates the true FDR well, or at least, it does not improperly favor or disfavor any particular method. Permutation methods have become popular to estimate FDR in genomic studies. The purpose of this paper is 2-fold. First, we investigate theoretically and empirically whether the standard permutation-based FDR estimator is biased, and if so, whether the bias inappropriately favors or disfavors any method. Second, we propose a simple modification of the standard permutation to yield a better FDR estimator, which can in turn serve as a more fair criterion to evaluate various statistical methods.

Results: Both simulated and real data examples are used for illustration and comparison. Three commonly used test statistics, the sample mean, SAM statistic and Student's *t*-statistic, are considered. The results show that the standard permutation method overestimates FDR. The overestimation is the most severe for the sample mean statistic while the least for the *t*-statistic with the SAM-statistic lying between the two extremes, suggesting that one has to be cautious when using the standard permutation-based FDR estimates to evaluate various statistical methods. In addition, our proposed FDR estimation method is simple and outperforms the standard method.

Contact: yangxie@biostat.umn.ed

1 INTRODUCTION

DNA microarrays are biotechnologies that allow highly parallel and simultaneous monitoring of the whole genome (Brown and Botstein, 1999). Increasingly, they are used to detect genes expressed differentially under different conditions (Spellman *et al*., 1998). Typically, two steps are used to declare differentially expressed (DE) genes: first, one computes a summary or test statistic (e.g. the sample mean) for each gene and rank the genes in order of their test statistics; second, one chooses a threshold for the test statistics and call genes whose statistics are above the threshold 'significant' ones (Smyth *et al*., 2003). False discovery rate (FDR) introduced by Benjamini and Hochberg (1995) has become a popular way to formally assess the statistical significance level in microarray data analysis. FDR is defined as the expected percentage of false positives among the claimed positives. If we claim that r top ranked genes are significant DE genes, the expected percentage of equally expressed (EE) genes among these r genes is the FDR.

FDR can be used for several purposes in statistical analysis. First, FDR is related to the choice of cut-off for 'significance' to control the error rate in multiple tests. Benjamini and Hochberg (1995) introduced FDR as an error measure for multiple-hypothesis testing and proposed a sequential method based on *P*-values to control FDR. Storey (2002, 2003) proposed directly estimating FDR for a fixed rejection region, largely increasing the popularity of FDR in practice. Later, many authors (Tsai *et al*., 2003; Pounds and Cheng, 2004; Dalmasso *et al*., 2005) studied various issues related to FDR estimation, especially for microarray gene expression data. When FDR is used to provide an upper bound on the error one can tolerate, the conservativeness of FDR estimation is not an issue. Actually, Storey (2002, 2004) showed the conservative property of their FDR estimator. Second, some recent literature pointed out some connections between FDR and variable selection (Abramovich *et al*., 2000; Ghosh *et al*., 2004; Devlin *et al*., 2003; Bunea *et al*., 2003). Third, FDR can be used as a criterion to evaluate new statistical methods or compare different procedures: when claiming the same number of total positives, the method with the lowest FDR is regarded as the best. If the truths are known, such as in simulation studies or some calibration datasets derived from spike-in experiments, the use of FDR as a criterion to compare different methods is analogous to using sensitivity and specificity as criteria and is very straightforward. In typical biological experiments, the truth is unknown and an estimated FDR instead can be used. Tibshirani and Bair (2003) used both true and estimated FDR to evaluate the use of eigenarray in microarray data analysis (http://www-stat.stanford.edu/~tibs/research.html). Shedden *et al*. (2005) used estimated FDR to compare seven methods for producing expression summary statistics for Affymetrix arrays. Other authors (Broberg, 2003; Pan, 2003; Xie *et al*., 2004; Wu, 2005) also used estimated FDR to compare different methods in microarray data analysis. It is reasonable and fair only when the estimated FDR approximates the true FDR well, or at least, the estimated FDRs for various methods being compared reflect the same trend of the true FDRs; that is, even if an FDR estimator is biased, it should not improperly favor or disfavor any particular statistical method being compared. We emphasize that the 'fairness' of FDR estimation is a necessary property when it is used as a criterion; this paper will focus on this aspect of FDR estimation.

Knowing the distribution of a test statistic under the null hypothesis (called null distribution) is important for FDR estimation.

*To whom correspondence should be addressed.

Some regularized statistics, such as the SAM-statistic (Tusher *et al*., 2001; Efron *et al*., 2001; Pan *et al*., 2003), perform well for microarray data, but their null distributions are in general unknown; permutation methods have become popular to estimate null distributions owing to their flexibility and generality. However, there are some problems when using permutation to estimate null distributions for microarray data. Pollard and Van der Laan (2003, 2004) pointed out that when the number of replicates in two groups is different, the permutation test for two-sample comparison may not be valid. Other authors (Efron *et al*., 2001; Pan, 2003; Zhao and Pan, 2003) have noticed this problem and addressed it by modifying the test statistic so that the standard permutation can still estimate the null distribution well. Guo and Pan (2004) addressed the problem by using weighted permutation scores that down weight the influence of (predicted) DE genes on estimating the null distribution. Nevertheless, to our knowledge, there has been no consideration on whether the bias of the FDR estimator introduced by the standard permutation, if it exists, may depend on the test statistic being used; if true, it implies that the resulting FDR estimates cannot be used as a criterion to fairly compare various statistical methods. The purposes of this paper are (1) to investigate both theoretically and empirically whether the standard permutation-based FDR estimation method is biased, and if yes, whether this bias favors or disfavors any particular statistic; (2) to propose a new FDR estimator that can serve as a better criterion to evaluate various statistical methods.

2 METHODS

2.1 Test statistics

For the purpose of clarity, we only consider one-sample comparisons here, though extensions to two-sample comparisons and other more general settings are straightforward (Tusher *et al*., 2001; Broet *et al*., 2004). Suppose after preprocessing the data, we have observed gene expression levels (e.g. log ratios of the two channel intensities in cDNA arrays) $X_{i1},\ldots,X_{ik}$ for gene i, $i=1,\ldots,G$ from k arrays. The goal is to test H_i0: $E(X_{ij})=0$ for $i=1,\ldots,G$. We will consider three commonly used test statistics. The first one is the SAM-statistic (Tusher *et al*., 2001), shortened as S-statistic,

$$S_i=\frac{\bar{X}_i}{(V_i+V_0)/\sqrt{k}}, \tag{1}$$

where $\bar{X}_i=\sum_{j=1}^{k}X_{ij}/k$ and $V_i^2=\sum_{j=1}^{k}(X_{ij}-\bar{X}_i)^2/(k-1)$ are the sample mean and sample variance of the expression levels for gene i, and V_0 is a constant used to stabilize the denominator of the test statistic. V_0 can be chosen in different ways; one is $V_0=\text{median}(V_1,\ldots,V_G)$.

The second is the mean statistic, $M_i=\bar{X}_i$, which corresponds to the early practice of simply using fold changes as a significance indicator (e.g. Broet *et al*., 2002). The third one is the Student's t-statistic, $t_i=\bar{X}_i/V_i$, which is a standardized mean statistic.

2.2 A standard method for FDR estimation

For a fixed cut-off value d for a test statistic Z_i, we can obtain the true or realized FDR and its estimate as (Storey and Tibshirani, 2003)

$$\text{FDR}(d)=\pi_0\text{FP}(d)/\widehat{\text{TP}}(d),\quad \widehat{\text{FDR}}(d)=\widehat{\pi_0}\widehat{\text{FP}}(d)/\widehat{\text{TP}}(d), \tag{2}$$

where π_0 is the proportion of EE genes among all genes, and $\widehat{\pi_0}$ is its estimator. FP is the number of true false positive genes, i.e., the number of genes which are EE genes but claimed as DE genes, $\widehat{\text{FP}}$ is the estimated number of false positive genes. $\widehat{\text{TP}}(d)$ is the total number of genes claimed as DE genes when the cut-off value is d.

2.2.1 Standard permutation method In order to obtain $\widehat{\text{FP}}$, we need to estimate the distribution of the test statistic Z_i under the null hypothesis H_{i0} (that gene i is an EE gene). Rather than assuming a parametric distribution for the null distribution of Z_i, a class of non-parametric methods have been proposed to estimate it empirically (Efron *et al*., 2001; Tusher *et al*., 2001; Xu *et al*., 2002; Pan *et al*., 2003). The idea is to permute the data and calculate the null statistic z_i, in the same way as calculating z_i, but based on the permuted data. Under the null hypothesis, the empirical distribution of the null statistics can be used to approximate the null distribution. In the current context of the one-sample test, under H_{i0}, we can permute the data by randomly keeping or flipping the sign of each of $X_{i1},\ldots,X_{ik}$. When k is small, we can consider all possible permutations; otherwise, a large number of random permutations, say B, can be used. Calculating the same test statistic from the b-th permuted data results in the null statistic $z_i^{(b)}$ for $b=1,\ldots,B$ and $i=1,\ldots,G$. For any given $d>0$, if we claim any gene i satisfying $|Z_i|>d$ to be significant, we estimate the true positive (TP) numbers and false positive (FP) numbers as

$$\widehat{\text{TP}}(d)=\#\{i:|Z_i|>d\},\quad \widehat{\text{FP}}(d)=\sum_{b=1}^{B}\#\{i:|z_i|^{(b)}>d\}/B. \tag{3}$$

We plug $\widehat{\text{TP}}(d)$ and $\widehat{\text{FP}}(d)$ into Equation (2) to calculate FDR(d) and $\widehat{\text{FDR}}(d)$. Other more sophisticated methods, such as SAM (Tusher *et al*., 2001) or mixture model (Pan *et al*., 2003; McLachlan and Peel, 2000) can be equally applied.

2.2.2 Proportion of EE genes Based on expression (2), we need to estimate π_0, the proportion of EE genes, to calculate FDR. Many authors have studied the issue based on the distribution of P-values (Storey, 2002; Allison *et al*., 2002; Pounds and Morris, 2003; Pounds and Cheng, 2004; Guan *et al*., 2004, http://www.biostat.umn.edu./rrs.php; Wu *et al*., 2004, http://www.biostat.umn.edu./rrs.php). However, owing to the difficulty of assigning P-values, the estimation of π_0 remains challenging. In fact, if the standard permutation method is used to estimate P-values, the same argument as to be discussed next implies that the P-values will be overestimated, leading to overestimation of π_0 (Guo and Pan, 2004). Other non-parametric approaches can only estimate an upper bound of π_0 (Dalmasso *et al*., 2005). Because estimation of π_0 is itself an unsettled research question, and more relevantly here, is not the focus of our current work, we bypass it in simulations: for simulated data, we use true π_0 in expression (2), which represents the ideal (but not practical) performance of the standard method. For real data, however, we use an estimated π_0.

2.2.3 Problem with the standard permutation: statistical theory The idea of using null statistics of all genes to construct the null distribution is based on the assumption that the null statistics of all genes are identically distributed. However, as shown next, the null statistic of a DE gene does not have the same distribution as that of EE genes. Hence, the empirical distribution of the null statistics of all genes may not approximate the true null distribution well.

Suppose for gene i, its observed gene expression level X_{ij} on array j has mean μi and variance σ_i^2; $\mu_i=0$ if it is an EE gene, and $\mu_i\neq 0$ otherwise. Define Bernoulli random variable Y_{ij} as: $Y_{ij}=1$ (corresponding to keeping the sign of X_{ij}) with probability $\pi=0.5$ and $Y_{ij}=-1$ (corresponding to flipping the sign of X_{ij}) with probability $1-\pi=0.5$, and assume that Y_{ij} and X_{ij} are independent. Then the random variable $W_{ij}=Y_{ij}X_{ij}$ represents the permuted gene expression level in the standard permutation method. It is simple to verify that $E(Y_{ij})=2\pi-1=0$, and we have

$$\begin{aligned}E(W_{ij})&=E(Y_{ij}E(X_{ij}))=(2\pi-1)\mu_i=0\\ \text{Var}(W_{ij})&=E(\text{Var}(Y_{ij}X_{ij}|Y_{ij}))+\text{Var}(E(Y_{ij}X_{ij}|Y_{ij}))\\ &=E(Y_{ij}^2\sigma_i^2)+\text{Var}(Y_{ij}\mu_i)\\ &=\sigma_i^2+\mu_i^2.\end{aligned}$$

If gene i is an EE gene, $\mu_i = 0$, and thus $\mathrm{Var}(W_{ij}) = \sigma_i^2 = \mathrm{Var}(X_{ij})$; otherwise, $\mu_i \neq 0$, and $\mathrm{Var}(W_{ij}) > \mathrm{Var}(X_{ij})$. The consequence is that permuted expression levels of DE genes inflate the variation of the distribution of all null statistics, as pointed out by previous authors (e.g. Pan, 2003). Although the heuristic argument is intuitively reasonable, it may not be equally transparent to everyone. Below we provide a more detailed, and hence more convincing discussion on this for each test statistic.

To facilitate discussion, we suppose that gene i is a DE gene throughout this section, and rewrite $X_{ij} = X_{ij}^* + \mu_i$; X_{ij}^* can be regarded as the expression level of gene i if gene i were equally expressed.

The mean statistic The null statistic for DE gene i is

$$m_i = \sum_{j=1}^{k} \frac{W_{ij}}{k} = \sum_{j=1}^{k} \frac{Y_{ij}X_{ij}^*}{k} + \mu_i \sum_{j=1}^{k} \frac{Y_{ij}}{k},$$

while if gene i were an EE gene, its null statistic would be

$$m_i^* = \sum_{j=1}^{k} \frac{Y_{ij}X_{ij}^*}{k}.$$

Because $\mu_i \neq 0$, it can be shown that $\mathrm{Var}(m_i) = \mathrm{Var}(m_i^*) + \mu_i^2/k$. Therefore, the distribution of the null statistic of a DE gene has heavier tails than that of an EE gene. In other words, because of the presence of both DE and EE genes, the distribution of the null statistics of all genes, as adopted in the standard permutation method, has heavier tails than that of only EE genes. Note that the difference between $\mathrm{Var}(m_i)$ and $\mathrm{Var}(m_i^*)$ depends on both μ_i and k, the difference will get smaller when k increases.

The t-statistic The null statistic for DE gene i is

$$t_i = \frac{\sum_{j=1}^{k} Y_{ij}X_{ij}^*/k + \mu_i \sum_{j=1}^{k} Y_{ij}/k}{V(Y_{ij}X_{ij}^* + \mu_i Y_{ij})/\sqrt{k}}.$$

In contrast, if gene i were an EE gene, its null statistic would be

$$t_i^* = \frac{\sum_{j=1}^{k} Y_{ij}X_{ij}^*/k}{V(Y_{ij}X_{ij}^*)/\sqrt{k}},$$

where $V(R_{ij})$ is the sample standard deviation of $\{R_{i1}, \ldots, R_{ik}\}$. Although, as shown earlier, the variance of the numerator of t_i is larger than that of t_i^*, the variance of the denominator of t_i may be also larger than that of t_i^*. Hence, we cannot simply conclude that $\mathrm{Var}(t_i) > \mathrm{Var}(t_i^*)$. Although it seems non-trivial to establish analytically, we use simulation to compare the variances of t_i, t_i^*, m_i and m_i^* under the assumption that Xij has a normal distribution.

We simulated X_{ij}^* from a standard normal distribution (i.e. with mean 0 and variance 1), and Y_{ij} from a Bernoulli distribution specified earlier, with $i = 1, \ldots, 100\,000$ and $j = 1, \ldots, k$. With $\mu_i = 2$ and $\mu_i = 0.5$, we calculated each m_i, m_i^*, t_i and t_i^*. Table 1 gives the sample variances of the four statistics with $k = 3, \ldots, 6$. It can be seen that m_i has a larger variance than m_i^*s; and the difference between the two is larger for a smaller k. In most cases, t_i has a larger variance than t_i^*s, but there is an exception when $\mu_i = 0.5$ and $k = 3$. So we cannot get a simple conclusion that variance of t_i is always bigger than variance of t_i^*, which is different from the situation of mean statistic. Of course, by the central limit theorem, the asymptotic distribution of z_i is the same as that of z_i^* as k tends to infinity for both the mean statistic and t-statistic. To compare the impact of DE genes on different test statistics, we calculated the relative difference for the mean statistic, $[\mathrm{var}(m_i) - \mathrm{var}(m_i^*)]/\mathrm{var}(m_i^*)$ and similarly that for the t-statistic. Table 1 shows that the relative difference of mean statistic is larger than that of t-statistic, so the discrepancy between the distribution of m_i and m_i^* is larger than that of t_i and t_i^*.

The SAM statistic As a modified t-statistic with a constant V_0 being added to the denominator, the behavior of the SAM statistic lies between the mean statistic and the t-statistic: if $V_0 = 0$, the SAM statistic is the same as the t-statistic; as V_0 tends to infinity, the SAM statistic reduces to the mean statistic (Efron *et al*., 2001). Therefore, we expect that the discrepancy between the distribution of the null statistic of EE genes and that of DE genes lies between that for the mean statistic and that for the t-statistic.

Table 1. Variances of the null mean statistics for a DE gene (m_i) and a corresponding EE gene (m_i^*), variances of the null t-statistics for a DE gene (t_i) and a corresponding EE gene (t_i^*) with various numbers of replicates k and true difference of the means between an EE gene and a DE gene (μ_i)

μ_i	k	3	4	5	6
2	$\mathrm{Var(m_i)}$	1.67	1.26	1.00	0.84
	$\mathrm{Var}(m_i^*)$	0.33	0.25	0.20	0.17
	Relative difference	4.00	4.02	3.98	4.07
	$\mathrm{Var(t_i)}$	32.97	7.19	3.66	2.49
	$\mathrm{Var}(t_i^*)$	12.4	2.93	1.98	1.64
	Relative difference	1.66	1.42	0.84	0.52
0.5	$\mathrm{Var(m_i)}$	0.42	0.31	0.25	0.21
	$\mathrm{Var}(m_i^*)$	0.33	0.25	0.20	0.17
	Relative difference	0.25	0.26	0.25	0.27
	$\mathrm{Var(t_i)}$	9.63	2.98	1.99	1.68
	$\mathrm{Var}(t_i^*)$	12.40	2.93	1.98	1.64
	Relative difference	–0.22	0.01	0.01	0.02

Relative difference for the mean statistic is defined as $[\mathrm{Var}(m_i) - \mathrm{Var}(m_i^*)]/\mathrm{Var}(m_i^*)$, and that for the t-statistic is $[\mathrm{Var}(t_i) - \mathrm{Var}(t_i^*)]/\mathrm{Var}(t_i^*)$.

In summary, by permuting expression levels of all the genes, both EE and DE genes, the standard permutation tends to overestimate the tails of the null distribution, leading to conservative inference, e.g. overestimating P-values, FP and FDR.

2.3 A new method for FDR estimation

We propose a new permutation based FDR estimation method. If we know which genes are EE genes, we only use these EE genes alone to construct the null distribution without using DE genes and thus avoid the trouble of the standard permutation method. In practice, we never know for sure which genes are EE genes, whose identification may be in fact the purpose of the whole analysis. However, we can use the predicted EE genes to do the permutation and construct the null distribution. First, we predict DE genes based on a summary statistic, then remove the predicted DE genes, and use the remaining genes in permutation. To do so, we have to address first which statistic to use to predict DE genes. A simple and natural way is to use the same statistic as the test statistic to predict DE genes. But if the performance of the test statistic itself is not good, this method may not work well. So an alternative way is to use a statistic that in general has a good performance; the SAM statistic seems to be a reasonable candidate (Tusher, 2001; Lonnstedt and Speed, 2002; Qin and Kerr, 2003; Xie *et al*., 2004). Based on our limited experience, we decided to use the S-statistic.

Another question is how many genes should be removed. Because predicting the number of DE genes is quite challenging, we propose removing the same number of genes as that of claimed significant DE genes. For example, if we identify top 50 genes as significant DE genes, we remove 50 most significant genes based on the S-statistic from the gene list, and then use the remaining genes to do the permutation, construct the null distribution and, therefore, estimate the FP and FDR. More specifically, the new FDR estimation procedure works as follows. Suppose z_i is our test statistic. For any given $d > 0$, we claim any gene i satisfying $|Z_i| > d$ to be significant, and we estimate TP as

$$\widehat{\mathbf{TP}}(d) = \#\{i : |Z_i| > d\}.$$

We define a set of non-significant genes $D(d)$ as $D(d) = \{i : |S_i| \leq d'\}$, where d' is chosen so that the number of genes not in set $D(d)$ is the same

as $\widehat{\mathbf{TP}}(d)$. As before, we permute observed expression levels B times; for each permuted dataset b, we calculate the null statistic $z_i^{(b)}$. Then, we use only the genes in $D(d)$ to estimate FP:

$$\widehat{\mathbf{FP}}(d) = \sum_{b=1}^{B} \#\{i \in D(d) : |z_i|^{(b)} > d\}/B.$$

Finally, FDR is estimated as $\widehat{\mathbf{FDR}}(d) = \widehat{\mathbf{FP}}(d)/\widehat{\mathbf{TP}}(d)$. Note that we do not use π_0 (or its estimate) in $\widehat{\mathbf{FDR}}(d)$ because we only use the genes in $D(d)$ to count false positives, which is equivalent to estimating π_0 as $1 - \widehat{\mathbf{TP}}(d)/G$.

3 RESULTS

3.1 Simulated data

To evaluate the performance of the standard FDR estimation method for different test statistics and whether our proposed FDR estimation method works, we used different simulation set-ups. For each simulation set-up, we simulated data 50 times, and used the mean FDR from these 50 replicates for comparisons. Because the variances of the results from these simulations were quite small, the Monte Carlo errors were negligible. In simulation set-up 1, a simulated dataset had $G = 4000$ genes, among which $G_1 = 400$ were DE genes and the other 3600 were EE genes on $k = 5$ arrays, so the proportion of EE genes was $\pi_0 = 0.9$. For EE gene i, its observed intensity log-ratios followed a normal distribution: $X_{ij} \sim N(0, 4)$ for $j = 1, \ldots, 5$; for DE gene i, $\mu_i \sim N(0, 16)$ and $X_{ij} \sim N(\mu_i, 4)$ for $j = 1, \ldots, 5$. Simulation set-up 2 was similar to set-up 1, but the standard deviation of gene i's expression level was not a constant; instead, it followed a continuous uniform distribution between 0 and 5. In simulation set-up 3, each simulated dataset was generated to mimic a real study (Tani *et al*., 2002), the purpose of which was to comprehensively define a family of genes whose transcription depends on the activity of leucine-responsive regulatory protein, or Lrp, in *Escherichia coli*. There were 4281 genes, 6 replicates and 800 DE genes randomly chosen (corresponding to $\pi_0 = 0.81$). For DE genes, the sample mean of each gene in real data was used as the true mean to generate simulated data; for EE genes, their means were all set at 0. For each gene, the sample variance was used as the true variance, and the expression level of each gene followed a normal distribution. Simulation set-up 4 was the same as set-up 3 except that the number of DE genes was increased to 2000 (leading to $\pi_0 = 0.53$); the purpose was to investigate how a small π_0 influences FDR estimation. Simulation set-up 5 was similar to set-up 1 but the number of DE genes was decreased to 200 ($\pi_0 = 0.95$). We applied the standard and new permutation methods to estimate FDRs using the mean (M), t and S test statistics. As mentioned earlier, when estimating FDR in the standard permutation method, we used the true π_0, an ideal but not practical case providing the best possible performance for the method; in contrast, we do not use the true π_0 for our new method.

Figure 1 compares the performance of the standard permutation and our new method when using the mean statistic as the test statistic under simulation set-ups 1–4. It shows that the standard permutation method largely over-estimates FDRs and the new method performs much better with its FDR estimates closer to the true ones. In simulation 4, after removing the DE genes predicted by the S-statistic, the FDR estimates based on our new method, though much better than that of the standard permutation, are still higher than the true ones. The reason is that there are a large number of DE genes (2000) in this set-up; because only relatively few DE genes are removed, the presence of many other remaining DE genes still affects the null distribution estimation.

Figures 2 and 3 present the results for the S-statistic and t-statistic, respectively. Again the standard permutation overestimates FDR. In general, the new method works better than the standard permutation, especially for the S-statistic. For the t-statistic, the new method gives larger biases than that of the standard method for simulation set-ups 3 and 4. The reason is that the standard method is implemented here using the true π_0 to estimate FDR, which is not possible in practice; in contrast, the new method always overestimates π_0 with $\hat{\pi}_0 = 1 - \widehat{\mathbf{TP}}/G$ when the number of removed genes ($\widehat{\mathbf{TP}}$) is fewer than the true number of DE genes, which is the case for the two plots in Figure 3. Nevertheless, the new method still works better than the standard permutation when a small number of genes are claimed to be significant, which often is of practical interest.

More importantly, by comparing Figures 1, 2 and 3 we can see why estimated FDRs based on the standard permutation method cannot be used as a fair criterion to evaluate the performance of the test statistics. In simulation set-up 1, the mean statistic gives the lowest true FDR while the t-statistic gives the highest; we can draw the same conclusion when using the proposed new FDR estimates, however, we would incorrectly conclude that the mean statistic gives the highest FDR if the standard permutation method is used. In simulation set-up 2, the S-statistic and the t-statistic give lower true FDRs than the mean statistic; the standard FDR estimators give the same conclusion, but the degree of bias for the mean statistic is much higher than that for the other two statistics. Simulations 3 and 4 give the similar conclusion that the bias of the standard FDR estimator depends on the test statistic, favoring the t-statistic and the S-statistic. Our new proposed FDR estimator provide a more fair criterion to compare the various statistics.

The choices on which and how many genes should be removed in the new FDR estimation method will affect its performance. As shown for simulation set-up 4, removing far fewer genes than the true number of DE genes may still result in overestimating FDR, though often to a lesser degree than that of the standard permutation. As an extreme in the other direction, we consider simulation set-up 5 (with $\pi_0 = 0.95$). Here we consider removing top 50 genes, 100 genes, 200 genes and 400 genes, respectively. To facilitate comparisons, in addition, we include results based on permuting only true EE genes, which is ideal but not practical, providing the best scenario. As shown in Table 2 as expected, permuting only true EE genes leads to excellent estimates of FDR while permuting all genes overestimates FDR, especially for the mean statistic. The more genes we remove, the lower the FDR we estimate. If we remove 200 genes, the same number as the true number of DE genes, the FDR estimates are very close to the true FDRs. As expected, if we remove 400 genes, the FDRs are slightly underestimated. Ideally, if the estimated π_0 is close to the truth, we can remove the same number of estimated DE genes. But as discussed earlier, most current methods overestimate π_0. For example, we used Storey and Tibshirani's (2003) method to estimate π_0 in this simulation and obtained $\hat{\pi}_0 = 0.975$, which corresponds to about 100 DE genes; removing only 100 predicted DE genes still results in various biases of the FDR estimates in the standard permutation for the three statistics. On the other hand, we can also see from Table 2 that our proposed simple procedure (removing the same number of genes as TP genes) can work well; see the final section for a further discussion on this issue.

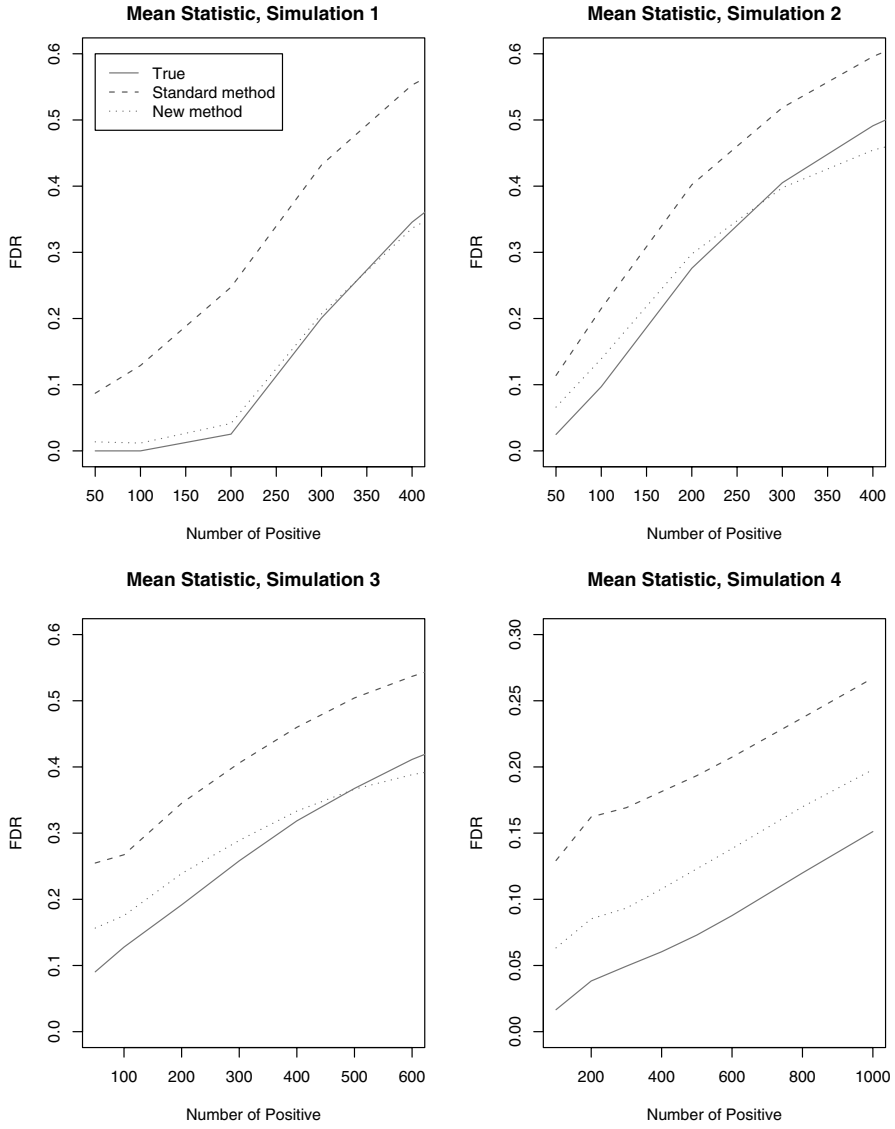

Fig. 1. FDR curves when using the sample mean as the test statistic under different simulation set-ups. Simulation 1, $X_{ij} \sim N(\mu_i, 4)$, the proportion of EE genes is $\pi_0 = 0.9$; Simulation 2, $X_{ij} \sim N(\mu_i, \sigma_i)$ and σ_i follows a uniform distribution, $\pi_0 = 0.9$; Simulation 3, mimicking the Lrp data, $\pi_0 = 0.81$; Simulation 4, mimicking the Lrp data, $\pi_0 = 0.53$.

As suggested by a referee, we also compared the performance of the method that downweights the influence of DE genes (Guo and Pan, 2004). From Table 2, we can see that the weighted method improves results over the standard permutation with less biased FDR estimates, especially for the *S*- and *t*-statistics, but may give a slightly larger bias of the FDR estimate for the mean statistic, thus slightly disfavoring the mean statistic. Larger studies are needed to draw a firm conclusion.

3.2 Chromosomal evolution data

A cDNA microarray experiment with three replications was used to compare the standard and the new FDR estimation methods. The purpose of the experiment was to identify duplications and deletions in genomic DNA (gDNA) of *E.coli*; more details can be found in Zhong *et al*. (2004).

We used Storey and Tibshirani's (2003) method to estimate π_0 and obtained $\hat{\pi}_0 = 1.002$; hence, we decided to use $\widehat{\pi_0} = 1$ for the standard method. Table 3 shows that the *S*-statistic performs best compared to the mean and *t*-statistics in terms of giving the lowest false positive numbers based on both the standard and new methods; though the standard permutation method gives higher false positive numbers than that of the new method, and these differences are especially large for the mean statistic, these observations are in agreement with that of the simulations.

In this experiment, 63 genes have been confirmed to be duplications or deletion genes (i.e. true positives) by real-time PCR and Southern blots. Based on these 63 genes, we can calculate an upper bound for the true false positive number as the number of genes identified by the test statistic but not in the list of 63 true positive genes. Because the follow-up experiment mainly targeted the genes

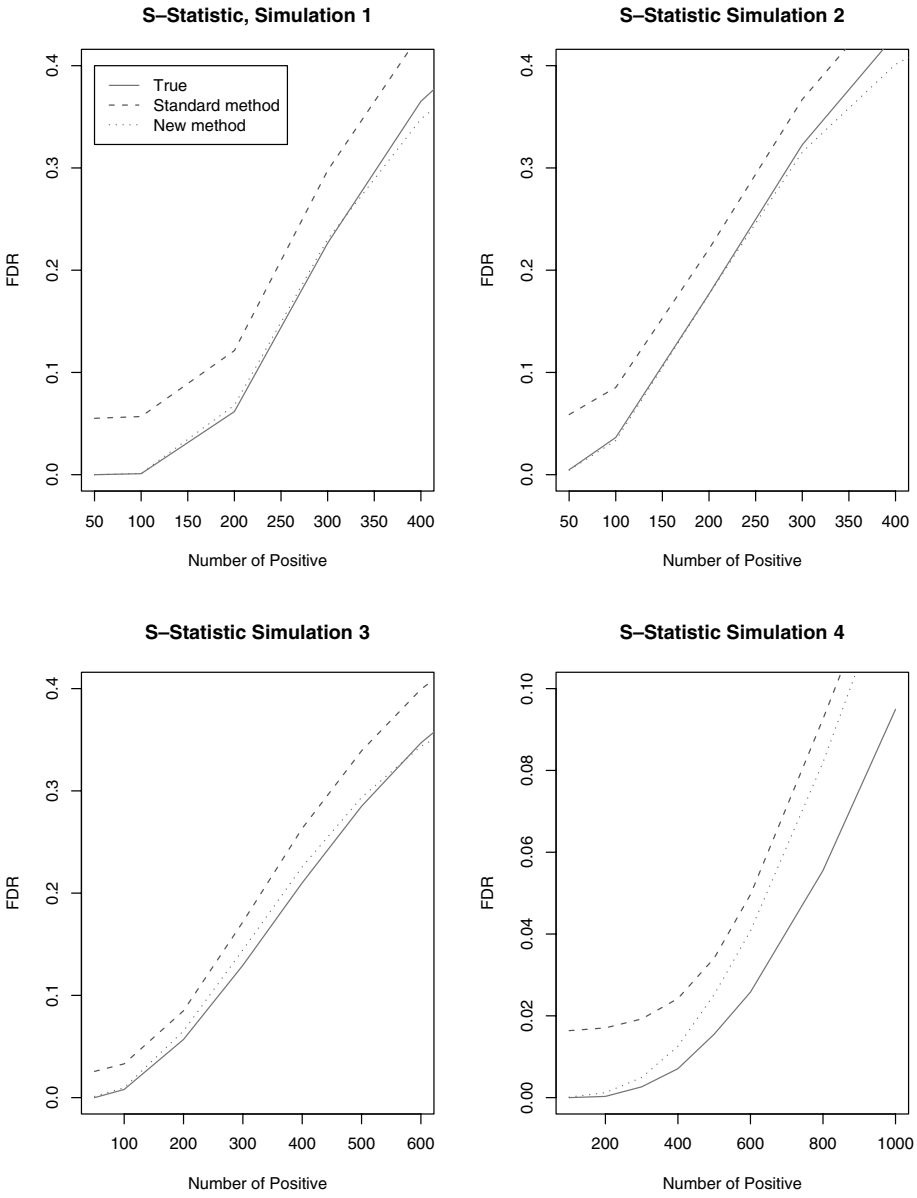

Fig. 2. FDR curves when using S as the test statistic under different simulation set-ups. Simulation 1, $X_{ij} \sim N(\mu_i, 4)$, $\pi_0 = 0.9$; Simulation 2, $X_{ij} \sim N(\mu_i, \sigma_i)$ and σ_i follows a uniform distribution, $\pi_0 = 0.9$; Simulation 3, mimicking the Lrp data, $\pi_0 = 0.81$; Simulation 4, mimicking the Lrp data, $\pi_0 = 0.53$.

with large absolute values of the mean statistics, the upper bound of the true false positive number should be most accurate for the mean statistic. Table 3 shows that if we use the mean statistic to identify 100 significant genes, there should be at most 39 false positive genes; the standard permutation estimates 84 genes as false positives out of 100 significant ones, while the new method gives 38. Hence, the standard permutation largely overestimates the FDR and the new method provides a better estimator. On the other hand, because many top genes ranked by the S-statistic or the t-statistic were not examined in follow-up, the upper bounds of the true false positive numbers for them are likely to be too loose, as evidenced by that the estimated FPs are all well under the bounds using either the standard or the new method.

4 DISCUSSION

This paper investigates the performance of permutation based FDR estimators for the mean, S- and t-statistics. As predicted by our theoretical analysis, our simulation study has confirmed that the standard permutation method overestimates FDR, even when we assume that the proportion of true DE genes is known. The degree of overestimation is especially serious when using the sample mean as the test statistic, less so for the S-statistic, and the least for the t-statistic. Because the magnitude of the bias depends on the test statistic being used, we should be cautious when using estimated FDR as a criterion to evaluate the performance of various test statistics. Our proposed method can estimate the true FDR

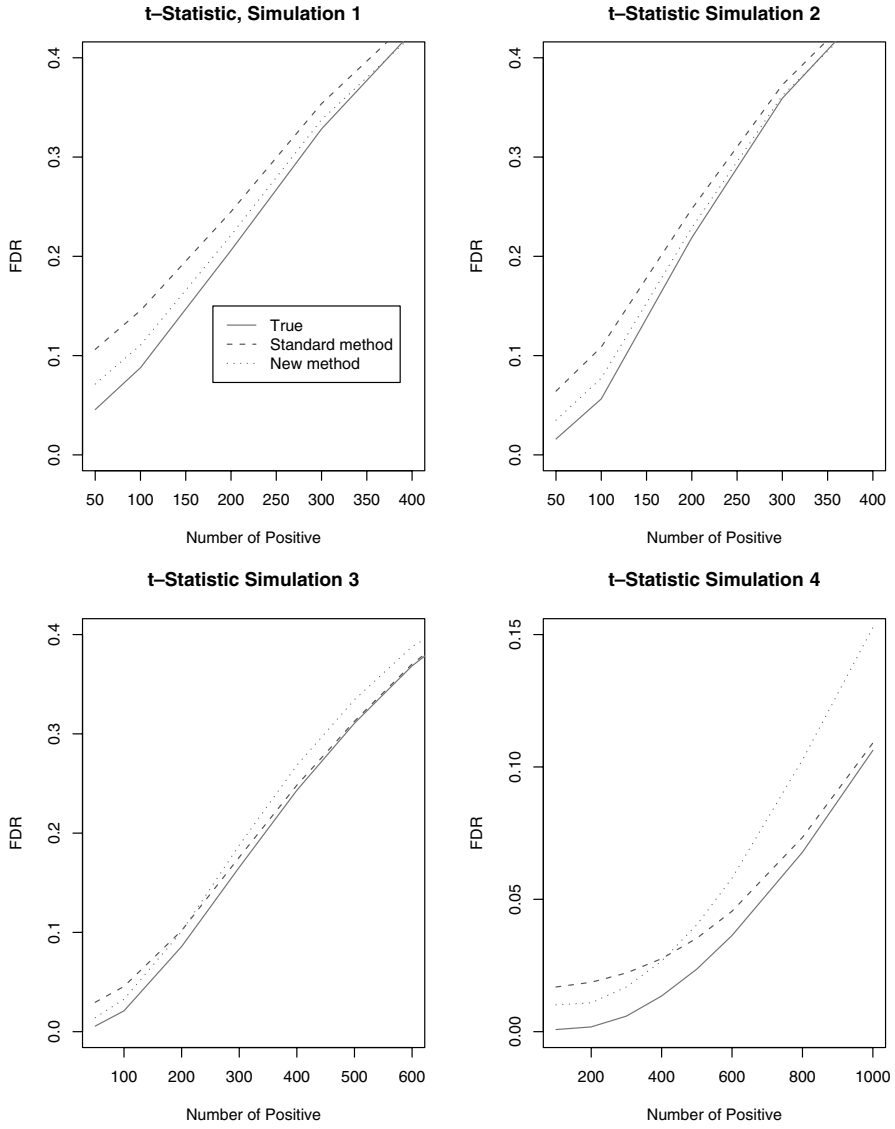

Fig. 3. FDR curves when using t as the test statistic under different simulation set-ups. Simulation 1, $X_{ij} \sim N(\mu_i, 4)$, $\pi_0 = 0.9$; Simulation 2, $X_{ij} \sim N(\mu_i, \sigma_i)$ and σ_i follows a uniform distribution, $\pi_0 = 0.9$; Simulation 3, mimicking the Lrp data, $\pi_0 = 0.81$; Simulation 4, mimicking the Lrp data, $\pi_0 = 0.53$.

better, hence providing a better means to evaluate various test statistics.

The basic idea underlying the new method is quite simple: because it is DE genes that cause the problem, removing the DE genes should improve the performance of the resulting FDR estimator. Our simulation and real data example show that the FDR estimation can be improved by permuting only predicted EE genes. We demonstrate that using the S-statistic to predict EE genes in the new method works well, though any other methods for detecting DE genes (Lonnstedt and Speed, 2002; Efron *et al*., 2001; Kendziorski *et al*., 2002; Newton and Kendziorski, 2003) that have proved useful can be also used.

An important parameter in our proposed method is the number of genes to be removed. In the current work, we have proposed removing the same number of genes as the number of identified significant DE genes. This method is simple and performs well in most cases. A justification is that FDR estimation depends more critically on the tails of the null distribution; Table 2 shows that removing a small number of the extreme genes effectively eliminates most of the bias. Nevertheless, if the number of DE genes is high, the current proposal may still overestimate FDR, although the degree of the bias is much less than that of the standard permutation method. On the other hand, when the true number of DE genes is smaller than that of claimed significant DE genes, the current proposal may underestimate FDR, which however is not really a serious issue. First, the biologists generally have a rough idea about the proportion of DE genes for the experiments. It is rare for one to try to identify more significant genes than the true ones because, with a smaller number of replicates and thus quite limited statistical power, the resulting FDR should be too high for the list of the identified genes to be

Table 2. True FDR (column T) and its estimates using the standard permutation (column S), the new method (column New) with various numbers of genes removed, permuting only true EE genes (column N) and weighted method (column W) for the mean-, *S*- and *t*-statistics for simulation set-up 5: there are $G = 4000$ genes, among which 200 are DE genes. The highlighted numbers are FDR estimates based on our proposed method: removing the same number of genes as the number of identified significant DE genes

						New (#genes removed)			
	$\widehat{\text{TP}}$	T	S	N	W	(50)	(100)	(200)	(400)
M	50	0.00	0.14	0.00	0.02	**0.01**	0.00	0.00	0.00
	100	0.05	0.28	0.05	0.11	0.12	**0.06**	0.05	0.04
	200	0.39	0.60	0.38	0.44	0.48	0.42	**0.37**	0.34
	300	0.56	0.74	0.56	0.60	0.65	0.60	0.55	0.49
	400	0.65	0.81	0.65	0.68	0.73	0.69	0.64	**0.58**
S	50	0.00	0.06	0.00	0.01	**0.00**	0.00	0.00	0.00
	100	0.10	0.17	0.11	0.12	0.14	**0.11**	0.10	0.10
	200	0.41	0.50	0.42	0.43	0.47	0.44	**0.40**	0.38
	300	0.57	0.66	0.58	0.57	0.62	0.60	0.56	0.52
	400	0.66	0.73	0.66	0.65	0.70	0.68	0.65	**0.59**
t	50	0.16	0.23	0.18	0.18	**0.18**	0.18	0.17	0.17
	100	0.30	0.37	0.32	0.32	0.33	**0.32**	0.31	0.30
	200	0.50	0.55	0.51	0.50	0.54	0.52	**0.50**	0.48
	300	0.62	0.66	0.62	0.60	0.65	0.64	0.61	0.58
	400	0.69	0.72	0.69	0.66	0.72	0.70	0.68	**0.64**

Table 3. The confirmed upper bound of false positive number, estimates based on the standard permutation and new method for the chromosomal evolution data

Statistic	$\widehat{\text{TP}}$	Upper bound	Standard	New
Mean	20	1	5	2
	40	1	10	2
	60	1	22	7
	80	19	67	28
	100	39	84	38
S	20	1	5	0
	40	7	10	0
	60	19	18	3
	80	28	24	4
	100	44	37	13
t	20	9	6	3
	40	23	14	9
	60	39	26	17
	80	56	41	30
	100	72	59	46

useful. (Note that, as discussed earlier, if the number of replicates is high, the overestimation problem with the standard permutation method will largely diminish, and thus it is no longer compelling to correct the standard permutation.) Second, if the biologists have no idea about the number of DE genes, and to be conservative, we recommend the following procedure: first estimating π_0, and then only using the current proposal if the proportion of claimed significant genes is smaller than $1 - \hat{\pi}_0$, and using the standard permutation method otherwise. Because, using the same argument as before (and based on our experience with simulated data), the permutation method (with all genes) will tend to overestimate π_0 (see also Wu *et al*., 2004), this conservative approach is in general still no worse than the standard permutation method.

ACKNOWLEDGEMENTS

This work was supported by NIH grants HL65462 and GM066098 and a UM AHC Development grant.

Conflict of Interest: none declared.

REFERENCES

Abramovich,F., Benjamini,Y., Donoho,D. and Johnstone,I. (2000) Adapting to unknown sparsity by controlling the false discovery rate. *Technical Report*. Department of Statistics, Stanford University.

Albers,W. *et al*. (1976) Asymptotic expansions for the power of distribution free tests in the one-sample problem. *Annal. Stat*., **4**, 108–156.

Allison,D.B. *et al*. (2002) A mixture model approach for the analysis of microarray gene expression data. *Comput. Stat. Data. An*., **39**, 1–20.

Baldi,P. and Long,A.D. (2001) A Bayesian framework for the analysis of microarray expression data: regularized *t*-test and statistical inferences of gene changes. *Bioinformatics*, **17**, 509–519.

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc*., **57**, 289–300.

Broet,P. *et al*. (2004) A mixture model-based strategy for selecting sets of genes in multiclass response microarray experiments. *Bioinformatics*, **20**, 2562–2571.

Broet,P. *et al*. (2002) Bayesian hierarchical model for identifying changes in gene expression from microarray experiments. *J. Comput. Biol*., **9**, 671–683.

Brown,P. and Botstein,D. (1999) Exploring the new world of the genome with DNA microarrays. *Nat. Genet*., **21**(suppl.), 33–37.

Broberg,P. (2003) Statistical methods for ranking differentially expressed genes. *Genome Biol*., **4**, R41.

Bunea,F. *et al*. (2003) The consistency of the FDR estimator. *Technical Report*. Department of Statistics, Florida State University, .

Cui,X. *et al*. (2005) Improved statistical tests for differential gene expression by shrinking variance components estimates. *Biostatistics*, **6**, 59–75.

Dalmasso,C. *et al*. (2005) A simple procedure for estimating the false discovery rate. *Bioinformatics*, **21**, 660–668.

Devlin,B. *et al*. (2003) Analysis of multilocus models of association. *Gen. Epidemiol*, **25**, 36–47.

Efron,B. *et al*. (2001) Empirical Bayes analysis of a microarray experiment. *J. Am. Stat. Assoc*., **96**, 1151–1160.

Iyer,V. *et al*. (2001) Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, **409**, 533–538.

Ghosh,D. *et al*. (2004) The false discovery rate: a variable selection perspective. *J. Stat. Plan. Infer*. (in press).

Guan,Z. *et al*. (2004) 'Model-Based Approach to FDR Estimation' *Research Report 2004-016*, Division of Biostatistics, University of Minnesota.

Guo,X. and Pan,W. (2004) Using weighted permutation scores to detect differential gene expression with microarray data. *J. Bioinformatics Comput. Biol*., **3**, 989–1006.

Hampel,F.R., Ronchetti,E.M., Rousseeuw,P.J. and Stahel,W.A. (1986) *Robust Statistics: The Approach Based on Influence Function*. John Wiley, NY.

Kendziorski,C. *et al*. (2002) On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statist. Med*., **22**, 3899–3914.

Khodursky,A.B. *et al*. (2000) DNA microarray analysis of gene expression in response to physiological and genetic changes that affect tryptophan metabolism in *Escherichia coli*. *Proc. Natl Acad. Sci. USA*, **97**, 12170–12175.

Khodursky,A.B. *et al*. (2003) *Escherichia coli* spotted double-strand DNA microarrays: RNA extraction, labeling, hybridization, quality control, and data management. *Methods Mol. Biol. USA*, **224**, 61–78.

Lambert,D. (1990) Robust two-sample permutation tests. *Ann. Stat*., **13**, 606–625.

Lehmann,E.L. and Stein,C. (1949) On the theory of some nonparametric hypotheses. *Ann. Math. Stat*., **20**, 28–45.

Lonnstedt,I. and Speed,T. (2002) Replicated microarray data. *Statistica Sinica*, **12**, 31–46.

McLachlan,G.J. and Peel,D. (2000) *Finite Mixture Models*. Wiley, NY.

Newton,M. and Kendziorski,C. (2003) Parametric empirical Bayes method for microarrays. *The Analysis of Gene Expression Data: Methods and Software,* Springer, New York.

Pan,W. (2003) On the use of permutation in and the performance of a class of nonparametric methods to detect differential gene expression. *Bioinformatics*, **19**, 1333–1340.

Pan,W. *et al*. (2003) A mixture model approach to detecting differentially expressed genes with microarray data. *Funct. Integr. Genomics*, **3**, 117–124.

Pollard,K.S. and Van der laan,M.J. (2003) Multiple testing for gene expression data: an investigation of null distributions with consequences for the permutation test. In *Proceedings of the 2003 International MultiConference in Computer Science and Engineering (METMBS'03)* Los Vegas, USA, pp. 3–9.

Pollard,K.S. and Van der Laan,M.J. (2004) Choice of null distribution in resampling based multiple testing. *J. Stat. Plan. Inf.*, **125**, 85–101.

Pounds,S. and Cheng,C. (2004) Improving false discovery rate estimation. *Bioinformatics*, **20**, 1–9.

Pounds,S. and Morris,S.W. (2003) Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of *P*-values. *Bioinformatics*, **19**, 1236–1242.

Qin,L. and Kerr,K. (2004) Empirical evaluation of methodologies for microarray data analysis. *Nucleic Acids Res.*, **32**, 5471–5479.

Ren,B. *et al*. (2000) Genome-wide location and function of DNA binding proteins. *Science*, **290**, 2306–2309.

Shedden,K. *et al*. (2005) Comparison of seven methods for producing Affymetrix expression scores based on false discovery rates in disease profiling data. *BMC Bioinformatics*, **6**, 26.

Smyth,G.K. *et al*. (2003) Statistical issues in cDNA microarray data analysis. *Functional Genomics: Methods and protocols*, **224**, 111–136.

Spellman,P. *et al*. (1998) Comprehensive identification of cell cycle-regulated genes of the Yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Cell Biol.*, **9**, 3273–3297.

Spino,C. and Pagano,M. (1991) Efficient calculation of the permutation distribution of trimmed means. *J. Am. Stat. Assoc.*, **86**, 729–737.

Storey,J.D. (2002) A direct approach to false discovery rates. *J. R. Stat. Soc. B*, **64**, 479–498.

Storey,J.D. and Tibshirani,R. (2003) Statistical significance for genome-wide experiments. *Proc. Natl Acad. Sci. USA*, **100**, 9440–9445.

Storey,J.D. *et al*. (2004) Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *J. R. Stat. Soc. B*, **66**, 187–205.

Tani,T. *et al*. (2002) Adaptation to famine: A family of stationary-phase genes revealed by microarray analysis. *Proc. Natl Acad. Sci. USA*, **99**, 13471–13476.

Tibshirani,R. and Bair,E. (2003) Improved detection of differential gene expression through the singular value decomposition.

Tsai,C. *et al*. (2003) Estimation of false discovery rates in multiple testing: application to gene microarray data. *Biometrics*, **59**, 1071–1081.

Tusher,V.G. *et al*. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl Acad. Sci. USA*, **98**, 5116–5121.

Wright,G.W. and Simon,R.M. (2003) A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics*, **19**, 2448–2455.

Wu,B. *et al*. (2004) Parametric and nonparametric FDR estimation revisited *Research Report 2004-015*, Division of Biostatistics, University of Minnesota.

Wu,B. (2005) Differential gene expression detection using penalized linear regression models: the improved SAM statistics. *Bioinformatics*, **21**, 1565–1571.

Xie,Y. *et al*. (2004) A case study on choosing normalization methods and test statistics for two-channel microarray data. *Comp. Funct. Genom.*, **5**, 432–444.

Xu,X.L. *et al*. (2002) A regression-based method to identify differentially expressed genes in microarray time course studies and its application in an inducible Huntington's disease transgenic model. *Hum. Mol. Genet.*, **11**, 1977–1985.

Zhao,Y. and Pan,W. (2003) Modified nonparametric approaches to detecting differentially expressed genes in replicated microarray experiments. *Bioinformatics*, **19**, 1046–1054.

Zhong,S. *et al*. (2004) Evolutionary genomics of ecological specialization. *Proc. Natl Acad. Sci. USA*, **101**, 11719–11724.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4289–4296
doi:10.1093/bioinformatics/bti703

Systems biology

Improved approach for proteochemometrics modeling: application to organic compound—amine G protein-coupled receptor interactions

Maris Lapinsh, Peteris Prusis, Staffan Uhlén and Jarl E. S. Wikberg*

Department of Pharmaceutical Biosciences, Uppsala University, Box 591 BMC, SE-751 24 Uppsala, Sweden

Received on June 23, 2005; revised on September 20, 2005; accepted on October 3, 2005
Advance Access publication October 4, 2005

ABSTRACT

Motivation: Proteochemometrics is a novel technology for the analysis of interactions of series of proteins with series of ligands. We have here customized it for analysis of large datasets and evaluated it for the modeling of the interaction of psychoactive organic amines with all the five known families of amine G protein-coupled receptors (GPCRs).

Results: The model exploited data for the binding of 22 compounds to 31 amine GPCRs, correlating chemical descriptions and cross-descriptions of compounds and receptors to binding affinity using a novel strategy. A highly valid model ($q^2 = 0.76$) was obtained which was further validated by external predictions using data for 10 other entirely independent compounds, yielding the high $q^2\text{ext} = 0.67$. Interpretation of the model reveals molecular interactions that govern psychoactive organic amines overall affinity for amine GPCRs, as well as their selectivity for particular amine GPCRs. The new modeling procedure allows us to obtain fully interpretable proteochemometrics models using essentially unlimited number of ligand and protein descriptors.

Contact: jarl.wikberg@farmbio.uu.se

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Drug discovery relies essentially on combinatorial chemistry and high throughput screening (HTS). Computations (e.g. docking) using the three-dimensional (3D) structure of the target and quantitative structure–activity relationships (QSAR) are also used. However, neither QSAR nor docking can assure that a drug candidate will interact only with the target and not with other members of the proteome. Deriving high-resolution 3D structures is also often problematic.

Ligands bind often to series of proteins and approaches that focus on the differences in the molecular recognition mechanisms and which are able to predict selective interaction partners are warranted. To this end we recently introduced a bioinformatics approach for drug-design termed proteochemometrics (Prusis *et al*., 2001; Wikberg *et al*., 2003, 2004).

In proteochemometrics one analyses the experimentally determined interaction strength of series of ligands with series of proteins. Proteochemometrics is based on quantitative descriptions derived from structural and physicochemical properties of interacting ligands and proteins, which are correlated to interaction affinity using mathematical modeling. In this way, proteochemometrics models the so-called ligand–receptor interaction space (Wikberg *et al*., 2004).

The first proteochemometric studies modeled peptide interactions with chimeric and wild-type melanocortin G protein-coupled receptors (GPCRs) (Prusis *et al*., 2001, 2002) and organic compound interactions with wild-type and chimeric α_1-adrenergic receptors (Lapinsh *et al*., 2001). More recent studies analyzed the binding of organic compounds to multi-chimeric melanocortin receptors (Lapinsh *et al*., 2005) and the interactions of organic amines to a series of 21 different amine GPCRs (Lapinsh *et al*., 2002b). The latter study represented four out of the five biogenic amine GPCR families, namely, serotonin, dopamine, histamine and adrenergic receptors. However, 10 of the receptors were serotonin receptor subtypes, while only one was a histamine receptor and none was a muscarinic acetylcholine receptor. The dataset was thus unbalanced and it also suffered from a large fraction of missing affinity values. Applying proteochemometrics onto it still yielded a statistically valid model. However, the modeling required a very complex description of the data, which involved more than 12 000 cross-terms and higher order cross-terms, which made it very difficult to comprehend the physical meaning of the model (see Lapinsh *et al*., 2002b for details).

The current study was undertaken to derive a more simple and sturdy proteochemometric modeling approach and apply it to the five families of amine GPCRs. To achieve this the modeling algorithms were altered to make the analysis of large-scale datasets affordable, while improving modeling quality. These improvements made the interpretation of the model straightforward, revealing particular molecular interactions that govern the studied compounds' overall affinity for amine GPCRs, as well as each particular compound's selectivity for each particular amine GPCR.

*To whom correspondence should be addressed.

2 METHODS

2.1 Interaction data

Data for 32 organic amine interaction with 31 amine GPCRs were taken from the Psychoactive Drug Screening Program (PDSP) database (http://pdsp.cwru.edu/pdsp.asp) (see Supplementary data for details). The receptors represented five amine GPCR families and included ten serotonin, seven adrenergic, five dopamine, five muscarinic acetylcholine and four histamine receptor subtypes. Most of the organic amines were tricyclic and/or piperidine/piperazine ring containing compounds; the series included approved and candidate drugs (antipsychotics, antidepressants, antiparkinson agents, antihistamines, etc.) as well as some other psychoactive amines.

The affinity values covered a range of more than five logarithmic units. Particular receptor subtypes (e.g. DRD2, ADA1A, 5HT2A and HRH1) showed high average affinity for the compound series, whereas the compounds preferred none of the receptor families as a whole, when compared to any other family.

The large number of observations allowed us to divide the dataset into a work-set comprising 22 compounds that were used for model creation and a prediction set comprising 10 compounds set aside and used after the completion of the proteochemometric model to assess the model's predictive ability. (For further details see Supplementary data).

2.2 Description of organic compounds

Structures of compounds were drawn using ISIS/Draw and converted to 3D by the Corina unit of the Tsar 3.3 (Accelryc Inc., http://www.accelrys.com) software package. Partial atomic charges were derived using the Charge2 unit of Tsar 3.3 and the geometry was optimized by performing energy minimization using the Cosmic utility of Tsar 3.3.

The thus obtained 3D structures were described by grid independent descriptors (GRINDs) (Pastor *et al*., 2000) calculated by Almond 3.1 (Multivariate Infometric Analysis S.r.l., http://miasrl.com) software. GRINDs are alignment independent descriptors that relate to the ability of a compound to form favorable interactions with independent pharmacophoric groups. The generation of these descriptors involves several steps. First, molecular interaction fields (MIFs) are calculated by placing probe groups on grid points surrounding the molecule. Grid nodes that show the energetically most favorable interactions with the molecule and concomitantly are situated as far as possible from each other are then extracted from the MIFs. The distances between each of any two extracted nodes and the products of their energy values are then calculated. Finally, the maxima of products falling within specified distance ranges (smoothing windows) for node pairs of the same MIF (auto-correlograms) and different MIFs (cross-correlograms) are used as descriptors for the molecules.

The Almond software allows the use of up to four MIFs, i.e. it provides four auto-correlograms and six cross-correlograms. The MIFs used herein were obtained using the following probes: DRY (hydrophobic probe), O (sp2 carbonyl oxygen), N1 (neutral flat NH) and Cl (chlorine). Default parameters were selected for the distance between grid points (0.5 Å) and the number of extracted nodes (100 for each MIF). Moreover, the default width was used for the smoothing window (0.8 grid units, i.e. 0.4 Å), resulting in 67 GRINDs in each of 10 correlograms.

We also created four additional sets of descriptors, using in each set three of the four abovementioned MIFs and substituting the fourth by a newly developed molecular shape field (Fontaine *et al*., 2004). Molecular shape was described by using N1 field nodes at a repulsion energy of 1 kcal/mol to outline the surface of the molecule. The local curvature of the surface was then calculated at each node, as described by Fontaine *et al*. (2004). Convex regions were considered to be more important for the shape than concave. This is because the former may form complementary interactions in the receptors' ligand-binding pockets or cause steric hindrances. Here we selected 100 of the most convex nodes (these actually outline the most protruded regions of the molecule and are referred to as TIPs). The TIP–TIP auto-correlogram was generated using curvature–curvature products and cross-correlograms with MIFs were generated using curvature-energy products.

2.3 Description of receptors

Previous 3D modeling and mutagenesis studies indicate that the GPCR ligand-binding pockets for endogenous amines and low molecular weight organic compounds are located in a cavity formed between the receptors' transmembrane regions (Bikker *et al*., 1998; Jacoby *et al*., 1999). We accordingly derived the receptor descriptions from the differences in the physicochemical properties of the seven cell membrane-spanning alpha-helical regions in the receptor series. Receptor amino acid sequences were retrieved from the Swiss-Prot database (http://www.ebi.ac.uk/swissprot) and aligned according to the conserved amino acid positions (Baldwin *et al*., 1997). The amino acids of TM1–TM7 were as follows (using the numbering of the ADA1A human): 25–49, 62–86, 98–122, 145–166, 185–206, 273–292 and 309–328. Of the 159 sequence residues selected 16 were conserved in all receptors. The non-conserved residues were subsequently coded using the three z-scale descriptors, $z1$–$z3$, derived by Sandberg *et al*. (1998). Thus, the physicochemical differences in the ligand-binding region of the amine GPCRs were accordingly encoded by a total of $143 * 3 = 429$ descriptors.

2.4 Principal component analysis of compound and receptor descriptors

Prior to further computations descriptors were preprocessed. First, the number of descriptors was reduced by applying principal component analysis (PCA).

PCA is a multivariate projection method that can be used to compress datasets containing large numbers of variables. Contrary to the original variables, the so-termed principal components (PCs) are orthogonal to each other (Wold *et al*., 1987; Eriksson and Johansson, 1996). After calculating A PCs, the $\mathbf{X}$ matrix with size N rows (objects) and K columns (variables) is decomposed into two smaller matrices, the score matrix $\mathbf{T}$ of size N by A and loading matrix $\mathbf{P}$ of size K by A according to the following equation:

$$\mathbf{X} = \mathbf{T}\mathbf{P}' + \mathbf{E} = \mathbf{t}_1\mathbf{p}'_1 + \mathbf{t}_2\mathbf{p}'_2 + \cdots + \mathbf{t}_A\mathbf{p}'_A + \mathbf{E}, \quad (1)$$

where $\mathbf{P}'$ is the transpose of the loading matrix, $\mathbf{t}$ the score vector, $\mathbf{p}'$ the transpose of the loading vector and $\mathbf{E}$ the matrix of residuals (unexplained part of the data). The majority of the variation within the original data can often be represented by a small number of components. Extracting $N - 1$ components explains all the variation of the original data. PCA was performed using SIMCA-P 9.0 software (Umetrics AB, http://www.umetrics.com).

Since the descriptors of the ligands are not correlated to the descriptors of the receptors we performed the PCA separately on the z-scales of the 31 receptors and on the GRINDs of the 22 work-set compounds (the 10 test-set compounds were not included in the PCA; the PC scores for them were calculated by summing the products of value of each GRIND descriptor for the compound with the loading of the respective descriptor). Prior to PCA all descriptors were mean centered and scaled to unit variance. Moreover, in order to fully preserve interpretability of models all components were extracted. Thus, having at hand 22 organic compounds the variance of GRIND descriptors was compressed into 21 components (GRIND-PCs), while the z-scale descriptors of 31 receptors were compressed into 30 components (ZSCALE-PCs).

2.5 Calculation of ligand–receptor cross-terms

Ligand–receptor recognition can evidently only partially be explained by linear combinations of ligand and receptor descriptors. For example if the ligands by virtue of some feature (property) interact with non-varied receptor residues, a simple assumption would be that the binding affinity relates linearly with the intensity of this given property. In reality, however, binding is governed by complex processes that depend on the complementarity of the properties of the interacting entities. In proteochemometrics this may

be accounted for by computation of ligand–receptor cross-terms (see e.g. Lapinsh *et al*., 2005 and references therein). Cross-terms were here formed by multiplying the principal components of descriptors of compounds (GRIND-PCs) and receptors (ZSCALE-PCs). In this way one additional descriptor block was obtained comprising $21 * 30 = 630$ descriptors. The total number of descriptors obtained thus became $21 + 30 + 630 = 681$. In fact, for any proteochemometrics dataset this number would be equal to the number of possible ligand–receptor combinations minus one with the use of the present approach for data preprocessing.

2.6 Block scaling of descriptors

In the PCA step the components became scaled relatively to each other. The first component of each block, which encapsulates the major differences between ligands and receptors, obtained the largest variance. The second component obtained the second largest variance, etc. Any further scaling (e.g. to unit variance) would hide the major patterns in the initial data and exaggerate minor non-systematic variations. Furthermore, when cross-terms are computed from the PCA space each cross-term obtains a variance that is proportional to the product of the variances of its originators. Therefore scaling of the cross-term descriptors reflects the significance of the underlying ligand and receptor properties, and accordingly no re-scaling is required.

However, the use of three descriptor blocks for which the descriptors are not directly comparable prompts the need for block scaling. Accordingly, while the mutual scaling of descriptors within each of the three blocks was frozen, each block was scaled to unit variance (i.e. the sum of variances of all GRIND-PCs, all ZSCALE-PCs and all cross-terms being set to unity). Block scaling was then afforded by systematically changing the variance of the blocks in 0.25 variance unit intervals until an optimal model was obtained.

2.7 Partial least-squares projections to latent structures

Correlation of descriptions to ligand–receptor affinity was performed by partial least-squares projections to latent structures (PLS) (for an in-depth review of the PLS see Geladi and Kowalski, 1986).

PLS derives a regression equation in which the regression coefficients reveal the direction and magnitude of the influence of x-variables on the response. For a proteochemometric model, including L ligand descriptors (i.e. GRIND-PCs), R receptor descriptors (ZSCALE-PCs) and cross-terms thereof, the equation derived for the response variable (i.e. ligand–protein interaction affinity) is expressed as follows:

$$y = \bar{y} + \sum_{l=1}^{L}(\text{coeff}_l * x_l) + \sum_{r=1}^{R}(\text{coeff}_r * x_r) + \sum_{1=1,r=1}^{L*R}(\text{coeff}_{l,r} * x_l * x_r). \quad (2)$$

The goodness-of-fit of the PLS models was characterized by the fraction of explained variation of **Y** (r^2). The predictive ability was characterized by the fraction of the predicted **Y**-variation (q^2), assessed by cross-validation, as previously described (Wold, 1995; Baroni *et al*., 1993). The q^2 computed using five randomly formed groups was used to adjust the variance of descriptor blocks and to determine the optimal number of PLS components.

Along with estimation of the conventional q^2 parameter we introduced several additional estimates to assess a model's predictive ability. Thus, in order to assess its ability to predict the affinity of novel receptors we repeatedly formed cross-validation groups by excluding one-fifth of the receptors and used the models based on the remaining receptors to compute affinities for the excluded ones, yielding q^2rec. Similarly, to assess the capacity of the model to predict the affinity of new ligands we repeatedly formed cross-validation groups by excluding one-fifth of the ligands, yielding q^2lig. Along with these validations we also performed predictions for the 10 compounds that had not been used in the model creation and thus could not have influenced the scaling and complexity of the PLS model. (The predictive ability for these compounds is here termed q^2ext.) We also performed validation by response permutation as described by Eriksson and Johansson (1996). In short, models were re-calculated 100 times for randomly re-ordered y-data and q^2 values were plotted as a function of the correlation coefficient between the original y and permuted y. The intercept of the regression line (i.e. the correlation coefficient being zero) indicates whether or not the original q^2 value could have been obtained by pure chance.

PLS modeling was performed using SIMCA-P 9.0 and Q2 (Multivariate infometric analysis S.r.l., www.miasrl.com) software. (Q2 was used for repeatedly performed cross-validations using randomly formed groups.)

2.8 Contribution of ligand properties for binding affinity and selectivity

The contributions of the x-variables were assessed from the PLS regression coefficients. Since the predictor variables are correlated to the y-data by means of the PLS regression equation, the regression coefficients reveal the significance of ligand and receptor properties for the interaction affinity. Thus, the regression coefficient of a compound descriptor represents the direction and magnitude that the underlying property influences the affinity for 'an average' amine GPCR. Furthermore, the coefficients for the cross-terms involving this descriptor summarize the importance of the underlying property for the compound's receptor selectivities.

Since the model included principal components rather than the original GRIND descriptors, regression coefficients were multiplied by the loadings of the original descriptor in each principal component, thereby allowing interpretations of the particular ligand properties represented by each GRIND. In this way, the regression coefficient of a GRIND descriptor could be assessed according to the following equation:

$$\text{coeff}_{\text{GRIND}} = \sum_{a=1}^{21}(\text{coeff}_{\text{GRIND-PC}_a} * p_{\text{GRIND},a}), \quad (3)$$

where $\text{coeff}_{\text{GRIND-PC}}$ is the regression coefficients for GRIND-PCs, and $p_{\text{GRIND},a}$ the loading of a given GRIND descriptor in principal component a. As $\text{coeff}_{\text{GRIND}}$ represents the change in the calculated average affinity of a compound when the GRIND value increases by 1 SD, it will be further referred to as Δy_{GRIND}. Moreover, the contribution of a GRIND descriptor to the affinity for a particular receptor R could be assessed according to the equation:

$$\Delta y_{\text{GRIND,R}} = \sum_{a=1}^{21}\left(\left(\text{coeff}_{\text{GRIND-PC}_a} + \sum_{b=1}^{30}(\text{coeff}_{\text{CROSS}_{a,b}} * x_{\text{ZSCALE-PC}_{b,R}})\right) * p_{\text{GRIND},a}\right), \quad (4)$$

where $\Delta y_{\text{GRIND},R}$ is the change in calculated affinity of a compound for the particular receptor R when the GRIND value increases by 1 SD, and $x_{\text{ZSCALE-PC}_{b,R}}$ the score for receptor R in principal component b.

3 RESULTS AND DISCUSSION

3.1 Proteochemometrics modeling

Descriptors of ligands, receptors and their cross-terms were correlated to ligand–receptor interaction affinity using PLS. Several models were created, using descriptors of compounds formed from different combinations of TIP and MIFs (Table 1).

As shown in Table 1, all models based on descriptors derived from different combinations of four TIP/MIFs (i.e. models 1–5) were highly predictive, the q^2s being in the range 0.75–0.77. However, the models differed in their ability to afford predictions for new compounds, as assessed by the q^2lig and q^2ext parameters. Model 2 thus showed the lowest q^2lig value, while model 1 showed the highest (0.51 versus 0.61). Model 2, which lacks the N1 field (i.e. the field resulting from the H-bond donor probe) also showed a lower

Table 1. Results of PLS modeling using GRINDs from different combinations of MIFs

No.	MIFs used	r^2	q^2	q^2lig	q^2ext
1	DRY, N1, Cl, TIP	0.87	0.75	0.61	0.63
2	DRY, O, Cl, TIP	0.90	0.76	0.51	0.57
3	DRY, O, N1, TIP	0.90	0.76	0.55	0.66
4	DRY, O, N1, Cl	0.89	0.76	0.51	0.62
5	O, N1, Cl, TIP	0.90	0.77	0.52	0.65
6	DRY, N1, TIP	0.88	0.76	0.59	0.67
7	DRY, N1	0.86	0.71	0.55	0.61
8	DRY, TIP	0.88	0.73	0.59	0.58
9	N1, TIP	0.88	0.74	0.56	0.40

q^2ext value (0.57) compared with the four other models (0.62–0.67). These results thus indicate that electron-attracting properties (i.e. the location and strength of electronegative atoms and groups in the compounds) are important for the receptor recognition.

By contrast, the q^2lig values were only 0.51–0.55 for the models including the O field (i.e. H-bond acceptor probe field) compared with 0.61 for model 1 which did not use this MIF. These results indicate that the differences in H-bond donating properties of the compounds, such as presence and location of hydroxy groups, yield mainly only chance correlations to the receptor affinity. (In interpreting this, one should keep in mind that all compounds contain an amine group that presumably interact with the conserved aspartic acid residue in TM3 of amine GPCRs. However, the aspartic acid residues and amine groups are invariant in the dataset. Proteochemometrics modeling can therefore not assess their importance.)

Inclusion or omission of the Cl field GRINDs only marginally influenced the predictive ability of the model (c.f. models 1 and 6). By contrast, removing any of the TIP, DRY or N1 fields (models 7–9) significantly reduced the q^2lig and q^2ext values. These latter three fields characterize the shape of the molecule, the strength and location of hydrophobic and H-bond donor moieties and seem to form a minimum set of MIFs required for modeling amine GPCR interactions. The analysis thus showed that model 6 was the best. Accordingly model 6 was used in all subsequent analysis, unless otherwise stated. (In the following sections this model will be referred to as 'the model').

3.2 Assessment of predictive ability of the model

The predictive ability of a proteochemometric model can be evaluated in different ways. The conventional q^2 parameter characterizes the predictions of combinations of ligands and receptors already present in the dataset, but tested in other combinations. However, one of the purposes of our study was to assess the capacity of proteochemometrics to afford predictions for novel yet pharmacologically uncharacterized organic compounds. Therefore, along with the conventional q^2 parameter we estimated q^2lig and q^2ext, which were found to be 0.59 and 0.67 for the model, respectively (Table 1).

The model was further subjected to repeatedly performed cross-validation with two random groups (i.e. validation with half of all observations excluded). This very harsh validation mode certified model sturdiness, the q^2 of 100 repeats being 0.68 with SD 0.02. Thus, a whole half of all data points could be omitted without endangering model validity. Finally, the predictive ability of the model for new receptors was also assessed; the q^2rec being 0.62 (thus revealing the potential use of proteochemometrics in finding ligands for yet biologically/pharmacologically uncharacterized GPCRs). Moreover, the model was also validated by response permutations. The negative q^2 intercepts (–0.34 using five cross-validation groups and –0.48 using two groups) obtained from this analysis show that randomized data produce non-predictive models.

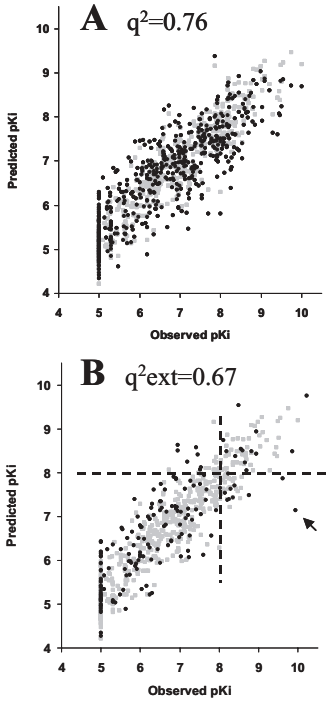

Fig. 1. Relation of predicted versus observed pK_i values derived from the PLS model. (**A**) shows predictions from conventional cross-validation using five random groups (black symbols). (**B**) shows results from external validation, i.e. black symbols are the predictions of the 10 compounds that had not been used during model creation (for details see text). Goodness-of-fit of the model (gray symbols, calculated versus observed pK_i values) is shown in both (A) and (B).

Results of validations are graphically shown in Figure 1. Results for cross-validation with five groups are shown in Figure 1A and Figure 1B shows results for external predictions. As can be seen from Fig. 1A, the predictive ability for compound–receptor combinations is very good, the average prediction error being $<0.5\ pK_i$ units. As seen from Fig. 1B, for only one test-set compound–receptor interaction a misprediction by $>2\ pK_i$ units occurred, while for the remaining observations the average prediction error was 0.55. The misprediction $>2\ pK_i$ units occurred for roxindole for the 5HT1A receptor. In fact this misprediction can be explained by the absence of any compounds with high affinity for the 5HT1A receptor in the work-set. Despite this the model still correctly predicted that roxindole has the highest affinity for the 5HT1A receptor among all test-set compounds, thus showing that an experimentalist would get proper guidance on the direction of the affinity also in this case.

Another way of viewing the predictive ability of the model is to set an arbitrary cutoff limit, such as a $pK_i > 8$, which in a real setting might be a selection criteria for a candidate compound. As is shown

in Figure 1B, on 20 occasions a $pK_i > 8$ is predicted for test-set compound–receptor combinations. Of these, 13 combinations have indeed an experimentally determined $pK_i > 8$, while for the remaining 7 the measured pK_i value was 6.95 or higher. Furthermore, the model predicts high affinity for a number of ligand–receptor interactions for which measured data are not available results. These indicate the potential use of the model in screening of compound databases for high affinity binders to particular amine GPCRs.

3.3 Interpretation of the model

Contribution of ligand properties for binding affinity and selectivity was assessed by estimating the contributions of GRINDs according to the Δy_{GRIND} and $\Delta y_{GRIND,R}$ measures. A Δy_{GRIND} is a regression coefficient of a GRIND descriptor and shows the descriptor's influence on the compounds' overall (average) affinity for amine GPCRs. A $\Delta y_{GRIND,R}$ value can be considered as a regression coefficient of a GRIND for a particular receptor. Comparing the latter for different receptors thus allows one to assess the influence of particular GRINDs for a compound's selectivity for any receptor pair. In Figure 2A are plotted Δy_{GRIND} values for DRY–DRY, N1–N1, TIP–TIP auto-correlograms, and DRY–N1, DRY–TIP and N1–TIP cross-correlograms. The GRINDs are for each correlogram arranged in order of increasing distance between node pairs, the vertical separators representing a distance range from 0 to 26.8 Å. Similarly, in Figure 2B–F are given examples for $\Delta y_{GRIND,R}$ for particular receptors.

Inspecting Figure 2 reveals that DRY–DRY correlogram descriptors (representing the ability of a molecule to form hydrophobic interactions) are the most important for all receptors. On one hand the positive Δy_{GRIND} and $\Delta y_{GRIND,R}$ values for the DRY–DRY descriptors at node distances from 0 up to 4 Å indicate that the presence of a hydrophobic group is in general of high importance for ligand binding to the amine GPCRs. Moreover, the presence of an additional DRY field at a distance 6–8 Å from the first one gives a further positive contribution to the binding. On the other hand, Δy_{GRIND} values for DRY–DRY descriptors at distances between 8 and 20 Å are close to zero, showing that distantly located hydrophobic groups have only minor impact on the average affinity of the compounds for the amine receptors. However, such interactions may still be important for the selectivity of the compounds for particular receptors. For example, for the DRD2 receptor they yield a positive influence on the affinity (Fig. 2C).

Positive values are also given to Δy_{GRIND}s of DRY–N1 cross-correlogram descriptors; the optimal distance between hydrophobic and H-bond acceptor MIFs being 6–10 Å. Moreover, inspection of Figure 2B–F reveals that DRY–N1 descriptors have highly positive $\Delta y_{GRIND,R}$ values for 5HT2A and DRD2 receptors (at distances up to 14 Å), whereas these descriptors contribute only marginally for other receptors. Thus, the mutual location of hydrophobic and H-bond acceptor moieties not only determines the average affinity of compounds for amine GPCRs but also is important for receptor subtype selectivity.

By contrast, most N1–N1 descriptors show slightly negative Δy_{GRIND} values. Thus, the mutual location of several H-bond acceptor groups appears to have low contribution to the ligands' average affinity. However, as revealed in Figure 2B–F N1–N1 descriptors show very negative $\Delta y_{GRIND,R}$ values for particular receptors at several distance ranges, such as for the 5HT2A and ACM1. Negative Δy_{GRIND} and $\Delta y_{GRIND,R}$ values are also given to N1–TIP cross-correlogram descriptors at distances from 4 to 16 Å (an exception is the ACM1 receptor), whereas at larger distances $\Delta y_{GRIND,R}$ values for 5HT2A and DRD2 obtain positive values. Negative values are also assigned to short distances (up to 4 Å) of the DRY–TIP cross-correlogram descriptors, suggesting that very protruded hydrophobic moieties do not contribute favorably to the binding of ligands to the amine GPCRs. Comparisons of all three correlograms including the TIP field reveal the importance of the overall shape of a molecule for receptor selectivity. Thus, interactions over large distance ranges yield positive coefficients for 5HT2A and DRD2 (and somewhat lower for ADA1A) but not for HRH1 and ACM1.

In a further analysis we linked the patterns of Figure 2 to the MIFs of particular compounds showing HRH1 or DRD2 selectivity.

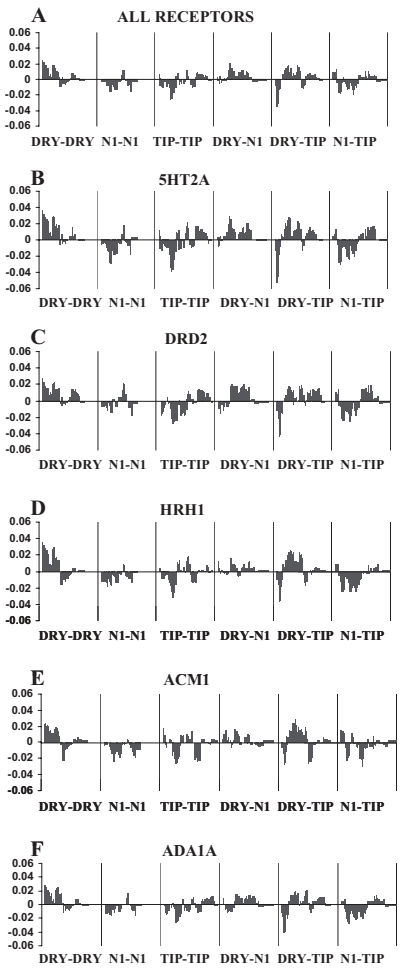

Fig. 2. Contribution of GRIND descriptors for explaining the binding affinity of organic compounds for amine GPCRs. (**A**) shows the PLS regression coefficients of GRINDs computed according to Equation (3). (**B**–**F**) show regression coefficients of GRINDs for 5HT2A, DRD2, HRH1, ACM1 and ADA1A receptors, respectively, computed according to Equation (4). Increments on the *Y*-axes indicate the change of affinity in pK_i units when a GRIND value is increased by 1 SD. The interval between the vertical separators represents the distance range 0–26.8 Å for each particular GRIND (see text for further details).

We selected these two receptors as a demonstration case because their affinity profiles are distinct. In particular, clozapine, chlorpromazine, olanzapine and several other compounds show significantly higher affinity for HRH1, while haloperidol, aripiprazole, risperidone, fluphenazine and some other compounds prefer DRD2. Inspection of the $\Delta y_{GRIND,R}$ values for these two receptors (Fig. 2C and D) reveals several patterns. Firstly, the DRY–DRY correlograms reveal that hydrophobic interactions influence ligand affinity for DRD2 and HH1R differently. For distances between DRY nodes of up to 8 Å the $\Delta y_{GRIND,HH1R}$s show larger positive values than the $\Delta y_{GRIND,DRD2}$s, while at distances >8 Å the $\Delta y_{GRIND,HRH1}$s, but not the $\Delta y_{GRIND,DRD2}$s, are negative. The presence of one, or two closely located, strong hydrophobic groups is thus needed to render a compound HRH1 selective, while several distantly located hydrophobic moieties are needed to create a DRD2 selective one. Secondly, the TIP–TIP correlograms reveal that the overall shape of the molecule is important for selectivity. Thirdly, it can be seen from the DRY–N1 and N1–TIP correlograms that a strong N1 field situated at certain distances from the DRY and TIP nodes may improve the affinity for the DRD2 with-out affecting the affinity for the HRH1.

The patterns of the foregoing paragraph are further visualized in Figure 3 by showing the MIFs around some HRH1/DRD2 and DRD2/HRH1 selective compounds. Compounds are there arranged in the order of their relative preference for the two receptors, with the MIFs represented as follows: DRY in beige, N1 in red and TIP in green. Inspections of Figure 3 reveal that distantly located DRY fields are present only for the two most DR2D selective compounds, namely haloperidol and risperidone. For haloperidol these fields, which appertain to the chlorophenyl and fluorophenyl moieties of the compound, are much weaker than the fields for the HRH1 selective compounds (clozapine, olanzapine and chlorpromazine). Moreover, also for the two remaining DRD2 selective compounds (fluphenazine and risperidone), the DRY field descriptors computed at distances from 3 to 6 Å show lower values than for any of the three HRH1 selective compounds.

Comparisons of structures also reveal systematic changes in the overall shape of the molecules, which are altered from rounded for the more HRH1 selective ones to elongated for the DRD2 selective ones.

Finally, inspection of the relative location of the N1 and TIP fields shows that only for the DRD2 selective compounds the N1-TIP nodes exist with locations of >16 Å from each other. Over shorter distances the values for the GRINDs overlap for both the HRH1 and DRD2 selective compounds.

4 DISCUSSION

In proteochemometrics the strength of ligand–protein interactions is correlated to chemical descriptions and cross-description ('cross-terms') of the interacting moieties. Cross-terms would not be needed if the ligands interacted with the invariant parts of the proteins only (e.g. with the 16 entirely conserved amino acids of the amine GPCRs transmembrane regions). However, differences in the binding affinity profiles of the ligand series arise since recognition is governed by complementary properties of receptors and ligands. Supplementing the descriptions by cross-terms is then used to reveal how ligand and receptor property combinations affect the interaction strengths.

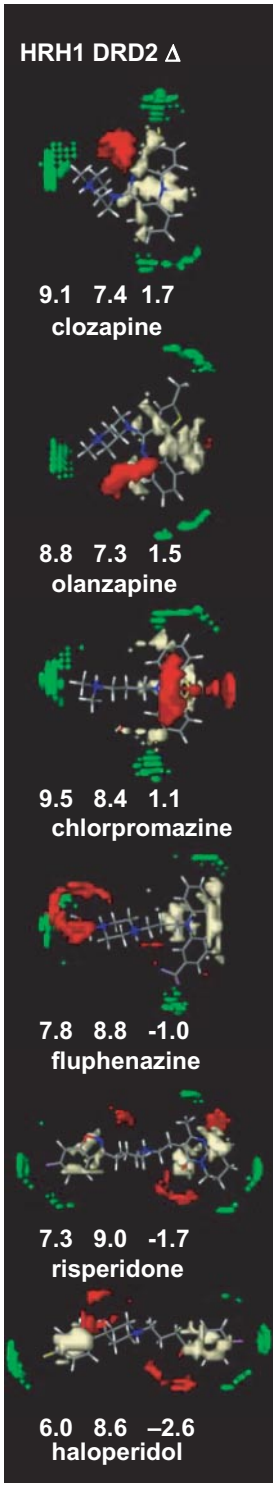

Fig. 3. Graphical representation of MIFs for some HRH1/DRD2 and some DRD2/HRH1 selective compounds. Shown are (from top to bottom) clozapine, olanzapine, chlorpromazine, fluphenazine, risperidone and haloperidol. The DRY MIFs are represented by beige, N1 by red and TIP by green. Indicated are also the pK_i values for each respective structure's binding to the HRH1 and DRD2 receptors calculated from the proteochemometric model. The computed HRH1/DRD2 selectivity ($\Delta = pK_{iHRH1} - pK_{iDRD2}$) of the compounds is also indicated.

Cross-terms may be obtained by multiplication of each ligand descriptor with each receptor descriptor. Correlation by PLS allows one to use a multitude of descriptors while tolerating their mutual colinearity. However, computing cross-terms directly from the descriptors of the current dataset would have resulted in almost 300 000 new variables, which would make further analysis highly resource consuming and, in fact, to our knowledge impossible with currently available academic or commercial software. We elected therefore to use PCA preprocessing, which allows one to keep the number of variables lower than the number of objects in a dataset, without compromising the information content of it.

We here applied PCA separately to ligand and receptor descriptors, which was followed by computations of cross-terms between ligand and receptor PCs. In fact, preliminary modeling using smaller subsets of descriptors revealed that the scores and calculated/predicted *Y* values of PLS models resulting from this type of preprocessing were identical to those of the models based on original descriptors and cross-terms thereof. However, differences would appear if one scaled the PCs relatively to each other, scaled cross-terms relatively to each other or used higher order cross-terms. Such procedures would give a risk for chance-correlations and subsequent deteriorations of models and were therefore avoided.

In earlier proteochemometrics studies the astronomic number of cross-terms was avoided by using simple descriptors (e.g. a limited number of binary descriptors), which is possible in simplified cases (i.e. for simple datasets). However, conventional QSAR has proven that the use of numerous descriptors is required for a thorough representation of the structures/properties of organic compounds. This is mandatory when using, e.g. GRID, CoMFA and GRIND. A proper description of complex biological macromolecules is obviously not an easier task and would require numerous structural descriptors or a large number of descriptors derived from sequence monomers (Kastenholz *et al*., 2000; Lapinsh *et al*., 2002a).

In our present study we aimed at representing compounds and receptors with descriptors that relate to major determinants for receptor–ligand interactions. The structures of organic compounds were characterized by GRIND descriptors. These describe the ability of compounds to interact with different MIF probes mutually located at varying distance ranges. An advantage of using GRIND descriptors is that they do not require the molecules to be spatially aligned with each other when creating a data matrix from series of compounds. In order to overcome some earlier shortages of the GRIND descriptors a recently developed molecular shape field (TIP) was used along with the MIFs (see Fontaine *et al*., 2004). The usefulness of the new molecular shape descriptors was indeed confirmed since the TIP–DRY and N1–TIP correlograms were among the most important for explaining ligand–receptor interactions. However, creation of several models based on different combinations of MIFs showed that not all MIFs were relevant, thus allowing us to find molecular interactions of importance for organic amine GPCR interaction affinity.

3D structures of compounds were created by the Corina unit and the geometry was optimized by the cosmic utility of the Tsar 3.3 software package. The conversion and optimization process was very fast, taking only a few seconds per molecule. This contrasted to our previous approach (Lapinsh *et al*., 2002b) where large conformational ensembles of each molecule were obtained by a time-consuming simulated annealing procedure. Accordingly the current approach could potentially be used to apply on large combinatorial libraries or to screen large compound databases. Moreover, Corina seems to provide some advantage since in our previous study the hydrophobic moieties tended to bunch together in the 3D structures for flexible molecules, while Corina creates extended low energy conformations that are close to the X-ray determined structures (c.f. also Sadowski and Gasteiger, 1994). In fact, for the present dataset simulated annealing produced inferior models compared with Corina generated structures (Lapinsh, M. and Wikberg, J.E.S., unpublished data). However, further studies on the dependence of proteochemometrics modeling to the approach for 3D modeling of compounds using broader datasets are warranted, to allow generalizations and to pinpoint the best method to be used for particular datasets.

The GPCRs in the present study were encoded by three *z*-scales of the amino acids of the transmembrane regions of the receptors. These *z*-scales represent the major differences in physicochemical properties of amino acids and would be the ones that primarily determine the ligand interactions with the receptors. Thus, overall using the present approaches for ligand and protein descriptions our technology affords predictive models without the need for ligand docking and accurate protein 3D structures.

The present dataset included 31 receptor subtypes representing five amine GPCR families. An advantage of the present data matrix was that it had quite few missing values. Otherwise affinities for weak binders are often omitted in scientific reports. For sake of mathematical modeling 'positive' and 'negative' interaction data are equally important. Thus, also for this reason the present model was improved compared with the earlier model where the interaction matrix contained about 30 percent values missing in a systematic fashion (Lapinsh *et al*., 2002b).

The validity of our model was assessed not only by the conventional q^2 estimate, generally used in QSAR, but also by predicting affinity for ligands and receptors entirely excluded from the model. The high values of the q^2lig, q^2rec and q^2ext estimates obtained herein indicate clearly the validity of our present modeling approach.

A clear advantage of the present modeling approach compared with the earlier one is that it gives fully interpretable models. In the previous study only a rough summary of the importance of different types of MIFs was possible to obtain (Lapinsh *et al*., 2002b), while the current approach allows unambiguous assessment of the importance of each descriptor for the interaction affinities. We here used this feature to assess the contribution of each GRIND for the compounds' overall affinity for the amine GPCRs, as well as for receptor subtype selectivity. Such analysis of a proteochemometric model may provide an experimentalist with suggestions how to modify a compound chemically in order to improve its selectivity and to afford a new compound with a desirable affinity profile.

In conclusion we have here shown how proteochemometrics can be adapted for the analysis of large-scale datasets yielding models which are straightforward to interpret in a chemical sense.

ACKNOWLEDGEMENTS

This work was financially supported by the Swedish VR (621-2002-4711).

Conflict of Interest: Wikberg declares that he holds stocks in Genetta Soft, a vendor of bioinformatics software.

REFERENCES

Baldwin,J.M. *et al*. (1997) An alpha-carbon template for the transmembrane helices in the rhodopsin family of G-protein-coupled receptors. *J. Mol. Biol*., **272**, 144–164.

Baroni,M. *et al*. (1993) Generating optimal linear PLS estimations (GOLPE): an advanced chemometric tool for handling 3D-QSAR problems. *Quant. Struct.-Act. Relat*., **12**, 9–20.

Bikker,J.A. *et al*. (1998) G-Protein coupled receptors: models, mutagenesis, and drug design. *J. Med. Chem*., **41**, 2911–2927.

Eriksson,L. and Johansson,E. (1996) Multivariate design and modeling in QSAR. *Chemom. Intell. Lab*., **34**, 1–19.

Fontaine,F. *et al*. (2004) Incorporating molecular shape into the alignment-free GRid-INdependent descriptors. *J. Med. Chem*., **47**, 2805–2825.

Geladi,P. and Kowalski,B.R. (1986) Partial least-squares regression: a tutorial. *Anal. Chim. Acta*, **185**, 1–17.

Jacoby,E. *et al*. (1999) A three binding site hypothesis for the interaction of ligands with monoamine G protein-coupled receptors: implications for combinatorial ligand design. *Quant. Struct.-Act. Relat*., **18**, 561–572.

Kastenholz,M.A. *et al*. (2000) GRID/CPCA: a new computational tool to design selective ligands. *J. Med. Chem*., **43**, 3033–3044.

Lapinsh,M. *et al*. (2001) Development of proteo-chemometrics: a novel technology for the analysis of drug-receptor interactions. *Biochim. Biophys. Acta*, **1525**, 180–190.

Lapinsh,M. *et al*. (2002a) Classification of G-protein coupled receptors by alignment-independent extraction of principal chemical properties of primary amino acid sequences. *Protein Sci*., **11**, 795–805.

Lapinsh,M. *et al*. (2002b) Proteo-chemometrics modeling of the interaction of amine G-protein coupled receptors with a diverse set of ligands. *Mol. Pharm*., **61**, 1465–1475.

Lapinsh,M. *et al*. (2005) Proteochemometric mapping of the interaction of organic compounds with melanocortin receptor subtypes. *Mol. Pharm*., **67**, 50–59.

Pastor,M. *et al*. (2000) GRid-INdependent descriptors (GRIND): a novel class of alignment-independent three-dimensional molecular descriptors. *J. Med. Chem*., **43**, 3233–3243.

Prusis,P. *et al*. (2001) PLS modeling of chimeric MS04/MSH-peptide and MC1/MC3-receptor interactions reveals a novel method for the analysis of ligand–receptor interactions. *Biochim. Biophys. Acta*., **1544**, 350–357.

Prusis,P. *et al*. (2002) Proteo-chemometrics analysis of MSH peptide binding to melanocortin receptors. *Protein Eng*., **15**, 305–311.

Sadowski,J. and Gasteiger,J. (1994) Comparison of automatic three-dimensional model builders using 639 X-ray structures. *J. Chem. Inf. Comput. Sci*., **34**, 1000–1008.

Sandberg,M. *et al*. (1998) New chemical descriptors relevant for the design of biologically active peptides. A multivariate characterization of 87 amino acids. *J. Med. Chem*., **41**, 2481–2491.

Wikberg,J.E.S. *et al*. (2003) Melanocortin receptors: ligands and proteochemometrics modeling. *Ann. N. Y. Acad. Sci*., **994**, 21–26.

Wikberg,J.E.S., Lapinsh,M. and Prusis,P. (2004) Proteochemometrics: a tool for modeling the molecular interaction space. In Kubinyi,H. and Müller,G. (eds), *Chemogenomics in Drug Discovery—A Medicinal Chemistry Perspective*. Wiley-VCH, Weinheim, pp. 289–309.

Wold,S. *et al*. (1987) Principal component analysis. *Chemom. Intell. Lab*., **2**, 37–52.

Wold,S. (1995) PLS for multivariate linear modeling. In van de Waterbeemd,H. (ed.), *Chemometric Methods in Molecular Design*. VCH, Weinheim, Germany, pp. 195–218.

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 23 2005, pages 4297–4301
doi:10.1093/bioinformatics/bti694

Databases and ontologies

Mapping PDB chains to UniProtKB entries

Andrew C. R. Martin

Department of Biochemistry and Molecular Biology, University College London, Gower Street, London WC1E 6BT, UK

Received on July 29, 2005; revised on September 22, 2005; accepted on September 24, 2005
Advance Access publication September 27, 2005

ABSTRACT

Motivation: UniProtKB/SwissProt is the main resource for detailed annotations of protein sequences. This database provides a jumping-off point to many other resources through the links it provides. Among others, these include other primary databases, secondary databases, the Gene Ontology and OMIM. While a large number of links are provided to Protein Data Bank (PDB) files, obtaining a regularly updated mapping between UniProtKB entries and PDB entries at the chain or residue level is not straightforward. In particular, there is no regularly updated resource which allows a UniProtKB/SwissProt entry to be identified for a given residue of a PDB file.

Results: We have created a completely automatically maintained database which maps PDB residues to residues in UniProtKB/SwissProt and UniProtKB/trEMBL entries. The protocol uses links from PDB to UniProtKB, from UniProtKB to PDB and a brute-force sequence scan to resolve PDB chains for which no annotated link is available. Finally the sequences from PDB and UniProtKB are aligned to obtain a residue-level mapping.

Availability: The resource may be queried interactively or downloaded from http://www.bioinf.org.uk/pdbsws/

Contact: andrew@bioinf.org.uk

1 INTRODUCTION

The Protein Data Bank (PDB) (Berman *et al*., 2000) is the primary resource in which protein structure data are deposited while SwissProt and trEMBL (Boeckmann *et al*., 2003) are the major resources containing protein sequence data. trEMBL is an automatic translation from the EMBL DNA databank while SwissProt contains a huge number of carefully maintained manual annotations. A recent effort has seen the integration of the SwissProt and trEMBL data with the Protein Information Resource (PIR) to create the UniProt Knowledgebase (UniProtKB) (Bairoch *et al*., 2005), designed as the central access point for extensive curated protein information, including function, classification and cross-references.

One might expect that mapping between the PDB and UniProtKB would be a trivial exercise. However, this is not the case. In some cases, PDB entries provide cross-links to UniProtKB/SwissProt, or UniProtKB/SwissProt provides links to PDB. In other cases, no link between the resources is supplied. In the case of cross-links provided from the PDB, this may be to the UniProtKB/SwissProt accession code or the identifier. These may become outdated and are rarely corrected in PDB entries. Clearly, a mapping between the PDB and UniProtKB/SwissProt is extremely valuable. The detailed annotations and cross-links to other resources available in UniProtKB can then be applied to PDB chains. Recent changes to UniProtKB/SwissProt have improved the mapping and have added PDB chain information and the range of residues in the UniProtKB/SwissProt entry which corresponds to a given PDB chain.

Previously, we developed a mapping between PDB chains and UniProtKB/SwissProt codes as part of an analysis of protein fold distributions for different enzyme classes (Martin *et al*., 1998). In order to account for supplied links from PDB chain to UniProtKB/SwissProt, UniProtKB/SwissProt entry to PDB file and unlinked files, the process was surprisingly complex. More recently, we developed a system for mapping between PDB protein chains and enzyme classification (EC) numbers available on http://www.bioinf.org.uk/pdbsprotec/ (Martin, 2004) and a system for mapping SNP data onto protein structures (Cavallo and Martin, 2005). In these cases we made use of a mapping between PDB chains and UniProtKB/SwissProt codes made available to us by the EBI. However, the downloadable version of this resource is not regularly updated leading to a need for us to develop an automatically updated version.

Links in the PDB to a UniProtKB/SwissProt entry may contain either the UniProtKB/SwissProt identifier (ID) or the accession code (AC) and are presented at the chain level. Links in the other direction (from UniProtKB/SwissProt to the PDB) are updated more frequently, but until Release 45 of UniProtKB/SwissProt (October, 2004) were at the whole PDB file level. In our previous implementation, `fasta33` (Pearson and Lipman, 1988) was used to resolve which chain was involved and a brute-force FASTA search against UniProtKB/SwissProt was used for any unassigned chains. In 1998, the whole process took some 3–4 days to run and unfortunately was not designed in a manner that could allow fast and easy updates as new PDB or UniProtKB/SwissProt entries became available.

With the exponential growth in the size of the PDB and the sequence databanks, a complete run now takes of the order of 10 days on a single Athlon XP 2800+ processor. Since the data resources (UniProtKB/SwissProt, UniProtKB/trEMBL and the PDB) are typically updated within this period, it becomes essential either to parallelize the procedure or, more efficiently, to move to a system which is automatically updateable and does not need to be re-generated from scratch.

The problem of PDB chain to UniProtKB/SwissProt mapping has been partially addressed by the Research Collaboratory for Structural Bioinformatics (RCSB). Their beta site provides a mapping of PDB files to associated UniProtKB/SwissProt entries (ftp://beta.rcsb.org/pub/pdb/uniformity/derived_data/pdb2sp.txt). At the time

of writing, this file had been updated approximately a month earlier providing UniProtKB/SwissProt accession codes for most PDB chains (some entries, e.g. 1a7m, are missing). Chimeric chains such as chain A of PDB entry 1gk5 (Chamberlin *et al*., 2001) are also correctly handled, though there is no indication of which residues map to which UniProtKB/SwissProt entry. However, a number of cross-links appear to be to other databases even when entries appear in UniProtKB/SwissProt. For example, chain N of PDB entry 1a02 is given a cross-link to an entry simply termed '1353774' (presumably an EMBL or Genbank identifier), but this entry should map to UniProtKB/SwissProt entry Q13469. Prior to recent updates, the file had not been updated for almost two years when it had appeared in a different format listing the entries without chain information and providing both UniProtKB/SwissProt identifier and accession for most entries (although the accession was missing from ∼20% of entries). Clearly the file format is in a state of flux and contains a number of inconsistencies and missing entries where mappings should be possible.

The new XML format files containing PDB data (available on ftp://beta.rcsb.org/pub/pdb/uniformity/data/XML/all/) now provide a mapping between 'entities' and UniProtKB/SwissProt entries where an entity corresponds to a unique gene segment. However, this is a beta release. Over the course of this work, the format of these files has been in a state of flux and there have been relevant changes. Like the original PDB files, the XML files still suffer from inconsistencies in the use of UniProtKB/SwissProt identifiers and accession codes.

In addition to including cross-links provided in the standard PDB files, the RCSB mapping (in both the flat file and the XML files) includes per-entity mappings derived from cross-links in UniProtKB/SwissProt entries. Thus, while a large amount of information is available, extracting the UniProtKB/SwissProt accession code together with the associated PDB chain and residue range on a regular automated basis remains a complex task. Either one must parse all the XML files or one must use the flat file (assuming it is regularly updated) and resolve cases which are not correct UniProtKB/SwissProt cross-links.

The problem has also been addressed by the macromolecular structure database (MSD) group at the European Bioinformatics Institute (EBI). They have created a mapping between PDB chain and UniProtKB/SwissProt accession code at the individual residue level which is used in the MSDlite search system (http://www.ebi.ac.uk/msd-srv/msdlite/). The mapping is not restricted to cross-links found in the two source databases and internally their mapping correctly handles chimeric chains and provides a complete per-residue mapping. However, this information is not available on the web server version. The data are also available for download (complete with information on chimeras), but the downloadable version is currently only updated on an occasional basis, or on request. While more complete than the mappings provided by the RCSB, it is still incomplete and a number of mappings (particularly those related to viral proteins and antibodies) are absent.

2 METHODS

Here we describe a new, completely automated mapping which we have performed and which is updated automatically as new data become available from the PDB or UniProtKB (containing both UniProtKB/SwissProt and UniProtKB/trEMBL).

The derivation of the mapping occurs in several stages each of which is described below. The mapping is created within a PostgreSQL relational database which may be queried via our web site and is dumped to a flat file which may be downloaded. Each stage of the mapping procedure is designed to be 'updateable'. In other words, re-running the procedure will only perform the necessary updates. All processing is performed using a set of Perl scripts which interface to PostgreSQL via Perl/DBI.

2.1 Mirroring the data

The UniProtKB data ('full' data files incorporating updates on top of official releases) are mirrored from the Expasy FTP site (ftp://ftp.expasy.org/databases/uniprot/knowledgebase/) while the PDB files are mirrored from the EBI (ftp://ftp.ebi.ac.uk/pub/databases/rcsb/pdb/data/structures/all/pdb). A modified version of the well-known Perl mirror script is used which is capable of storing a local de-compressed version of a remote compressed archive (http://www.bioinf.org.uk/software/mirror/).

2.2 Extracting data from UniProt

PostgreSQL (http://www.postgresql.org/) is used to store all the data during processing. The overall schema is illustrated in Figure 1. The first processing stage is to extract data from UniProt. A Perl script is run on the UniProtKB/SwissProt data and again on the UniProtKB/trEMBL data. The 'sprot' table is populated with the accession code, the sequence and the date of the latest modification. A second 'acac' table is populated with mappings between primary and secondary accession codes. A third 'idac' table is populated with mappings between identifiers and accession codes. A fourth 'pdbac' table is populated with mappings between an accession and a PDB code where these are provided. Since development was started before UniProtKB/SwissProt introduced chain information, these data are currently not stored. For updating, the database table is then scanned for any entries which contain a primary accession code which is now present as a secondary accession code in the 'acac' table. These are deprecated entries and are deleted from the database as the data have now been replaced by the new entry.

To allow updating, before an entry is placed in the database, we check whether it exists already. If it does not exist, we proceed as above. If it does exist, then we compare the stored date with the date read from the file. If the date is the same then we simply skip this entry. If the date in the file is newer, then we update the sequence and the date and replace all associated entries in the 'acac' and 'pdbac' tables.

2.3 Extracting chain information and mappings from PDB files

Each PDB file is processed in turn to extract a list of the chains it contains. For each chain, we store the PDB code, the chain, an accession code of '?', a flag to say the entry has not yet been validated, the source of the information ('pdbchain'—simply an indication that this is a chain in the PDB which has not yet been mapped to UniProtKB/SwissProt), the date at which the entry was processed, a flag to say that the entry has not yet been aligned and four zero values which will later be replaced by percentage identity, overlap, length and fraction-overlap when the chain is mapped to a UniProt entry. All these data are stored in a single 'pdbsws' table. This table is the main table which will provide all the mappings from PDB chains to UniProtKB/SwissProt entries.

Cross-references to UniProtKB/SwissProt are also extracted from `DBREF` records if they are present. These are generally indicated by a database identifier of 'SWS'; however a few cases use 'UNP' to indicate UniProt. If no `DBREF` records are found then the procedure looks for `REMARK 999` records. Where present, these cross-references generally provide one UniProtKB code (ID or accession), but in the case of chimeric chains there may be more than one code. Thus if there is only one code, we update the existing record in the 'pdbsws' table; if there are more entries, then additional records are inserted into 'pdbsws'. Where an accession code was provided in the PDB file, we change the source of information from

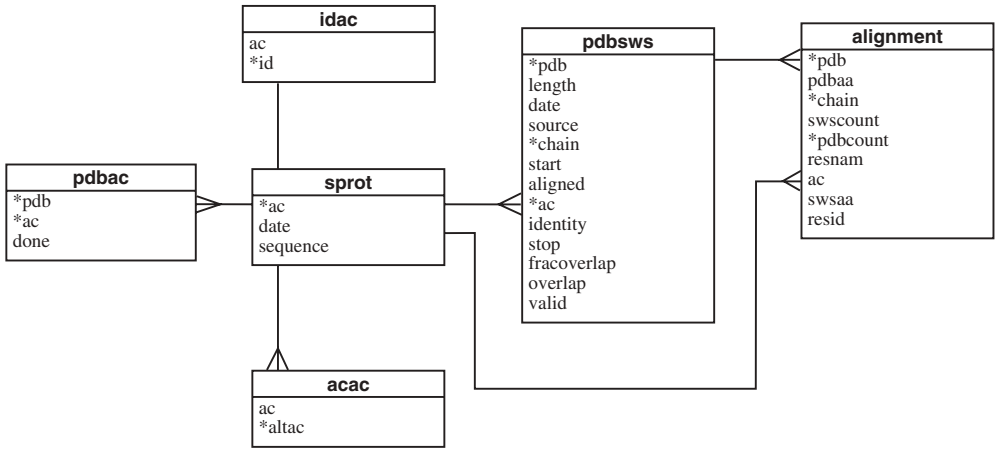

Fig. 1. Entity-relationship diagram for the database used for processing the mapping data. A simple line represents a one-to-one relationship while a 'crow's foot' represents a one-to-many relationship. For example, one UniProtKB accession in the 'sprot' table can link to several secondary accessions in the 'acac' table. Primary keys are indicated with asterisks. The 'pdbsws' table is the main table linking PDB chains to UniProtKB entries while links at the residue level are in the 'alignment' table.

'pdbchain' to 'pdb' indicating that an accession code has been assigned from information in the PDB file.

Currently, we assume that any changes made to a PDB file will not affect the data in which we are interested. Updates to PDB files are rare and tend to be minor in nature. More significant changes such as corrections to the sequence are accommodated by removing a PDB file (making it obsolete), replacing it by a new entry. Therefore updating is handled simply by not re-processing any entry which already exists in the database.

2.4 Fixing PDB cross-references and patching information from UniProtKB for single-chain entries

At this stage we implement a number of corrections to the information extracted from PDB files. First some entries will have provided UniProtKB/SwissProt IDs rather than accession codes. IDs always contain an underscore character, so we replace these entries in the 'pdbsws' table with the accession obtained from the 'idac' table.

A further problem is that while outdated accession codes are maintained in UniProtKB/SwissProt entries as 'secondary accessions', some UniProtKB/SwissProt IDs also change, but no reference to these is maintained in the downloadable UniProtKB/SwissProt files. For example, `LYS_CHICK` recently changed to `LYSC_CHICK`. The UniProtKB/SwissProt cross-references in the PDB files are therefore invalid. Such invalid entries are identified and the cross-reference is replaced by an accession of '?'.

During testing, one PDB file was identified which contained both the ID and the accession as separate `DBREF` records. Since the 'pdbsws' table enforces the constraint that the pdb code, chain and accession must form a unique triplet (a compound primary key), the updates will have failed and we can simply drop the entries for which the code appearing in the 'pdbsws' table is an ID in the 'idac' table and the ID is not the same as the accession. In a few cases, the PDB code appears in the `DBREF` cross-reference where one expects to find the UniProtKB/SwissProt code. These entries are replaced by accessions of '?'. All such 'problem' entries will be fixed during the brute-force scan.

At this stage, the 'pdbsws' table will contain UniProtKB accession codes for all chains that have cross-references provided in the PDB file even if these had been provided as identifiers. However, some of the accessions may be deprecated secondary accession codes. We therefore replace these with the primary accession code using the data stored in the 'acac' table. If a mapping between a PDB chain and a UniProt primary accession exists, it is possible that the entry with this primary accession will be removed from UniProtKB in a future release. This accession will now become a secondary accession code and a new accession will become the correct primary accession for this PDB chain. The update procedure will automatically correct the accession should this situation arise. It will also ensure that the entry is marked as unaligned, so that the alignment between a PDB chain and the entry will be updated later in the pipeline.

We then validate the accession codes to ensure that they are all valid UniProtKB primary accessions. Once validated, the 'valid' flag in 'pdbsws' table is set.

Some mappings will appear both in PDB files and in UniProtKB/SwissProt entries, so we now mark entries in the 'pdbac' table as done wherever we already have a mapping from the PDB. Thus PDB takes precedent over UniProtKB/SwissProt because, when development was started, only the mapping in the PDB files was provided at the chain level.

2.5 Adding cross-links from UniProt

In the next stage, we add cross-references supplied by UniProtKB which were not supplied in the PDB files. In the case of single-chain PDB files, this is easy since we can simply transfer the data from the 'pdbac' table to the 'pdbsws' table. The source field is updated to indicate that the information came from UniProtKB/SwissProt.

Since development of this method was started before Release 45 of UniProtKB/SwissProt (the first one to have per-chain mappings to the PDB), the process is more complex in the case of multi-chain PDB files. In this case, we create a file in FASTA format containing the sequences from the PDB file. The UniProtKB sequence is then extracted from the 'sprot' table into a FASTA file and aligned with each sequence in the PDB-derived FASTA file using the Smith–Waterman algorithm as implemented in `ssearch33` (Pearson and Lipman, 1988). The best sequence identity is recorded and all chains having this sequence identity are found. Entries in the 'pdbsws' table are updated to link the chain to the UniProt accession and, again, the source field is updated to indicate that the information came from UniProt. Future development of the software may make use of the per-chain information from UniProtKB/SwissProt.

For updating, we only transfer accessions from UniProtKB if they do not already exist as valid accessions in the 'pdbsws' table. In addition, we mark entries in 'pdbac' as used once the accession codes have been transferred to the 'pdbsws' table. Thus when the processing code is run again, only un-used entries will be considered.

2.6 Brute-force scan

At this stage, all cross-links between PDB files and UniProtKB entries that are available in the source data files have been inserted into the main 'pdbsws' table.

Remaining entries in the table will fall into one of the following categories: (1) Chains with entries in UniProtKB which do not have cross-links listed in the source files; (2) Chains which do not have corresponding entries in UniProt and (3) non-protein or short-peptide chains.

The next stage, therefore, is to perform a 'brute-force' scan of remaining PDB chains against the combined UniProtKB (SwissProt and trEMBL) database. First a combined UniProtKB FASTA file is created if either the UniProtKB/SwissProt or the UniProtKB/trEMBL FASTA file has been modified since a concatenated version of the two files was last created.

Any PDB chains which have not yet been assigned a valid accession are scanned against the combined UniProtKB data using `fasta33`. The sequence used for the PDB chains is that found in the ATOM records. Any non-standard amino acids are simply ignored (see, e.g. the MSE residue at I113, I116 and I182 in PDB entry 487d). While it could be argued that the sequence from the SEQRES records would be more appropriate for this scan, this decision was made for consistency and simplicity since it is the sequence from the ATOM records which must be used in the later alignment stage. The best match is found and is accepted if either (1) the overlap is at least 30 residues and the identity is at least 90%, (2) the overlap is at least 15 residues with at least 93% identity (i.e. 14 out of 15 residues) or (3) the whole of the PDB chain is matched with 100% identity. The 'pdbsws' table is updated with the accession code for the best match, together with the percentage identity, the overlap, the length of the PDB chain and the fractional overlap.

Whether or not a match was found, the 'pdbsws' table is updated to indicate that the entry has been processed with a brute-force scan and the date on which this was performed is recorded. When running an update, we need to find out whether PDB chains, which previously did not match a UniProtKB entry, match any new or updated entries. Using the date stored for the last processing of an entry, we create a FASTA database containing only those UniProtKB entries that have been added or updated since the PDB chain was last processed and scan using `fasta33` against only those entries.

When updating, we also check PDB chains where the sequence identity was <100% or where <90% of the PDB chain was matched. This is done to see if there are more recent entries with a better match. Since this is somewhat more time-consuming, it is only done on entries that were last processed at least one month ago.

2.7 Perform alignments

The final stage of the processing is to align PDB chains with UniProt entries. This is slightly inefficient in that alignments may well have been performed already (during the brute-force scan) without storing the alignment data, but it makes the processing much simpler if it is performed as an isolated step. The UniProtKB entry is extracted from the 'sprot' table and written in FASTA format. The sequence of the PDB chain is extracted from the ATOM records of the file and also written in FASTA format and the sequences are aligned using `ssearch33`. The alignment is then mapped onto the residue identifiers from the PDB file consisting of the residue number and optional insert code.

Finally the alignments can be dumped to a flat file containing the residue-level mappings between PDB chains and UniProtKB codes.

3 RESULTS AND DISCUSSION

Initial population of the database takes ~10 days indicating why it is very important that the system is able to be updated. An update corresponding to a new full release of UniProtKB/SwissProt takes <17 h. Approximate timings for populating the database and updating it are shown in Table 1.

Table 1. Approximate times required to populate and update the database shown in hours

Processing stage	Approximate wall-clock time (h)	
	Initial population	Updating
Processing SwissProt	0.5	0.5
Processing trEMBL	1.5	1.5
Processing PDB files	2.0	0.1
Fixing cross-references, etc	0.5	0.2
Brute-force scan	216.0	13.0
Performing alignments	13.5	0.6
Dumping results	0.3	0.3
Database data analysis	0.5	0.5
Total	234.8	16.7

Timings were on a system using an Athlon XP 2800+ processor, but are highly dependent on other parameters such as disk and network access speeds and, most importantly, the size of the database. 'Database data analysis' represents the time taken for PostgreSQL `analyze` steps to update the indexes—see text.

Table 2. Sources of link information in the complete mapping

Source of mapping data	Number of chains mapped
PDB entry	40 664
UniProtKB	15 057[a]
Brute-force scan	10 324[b]
DNA	6261
Short peptides	1647
`fasta33` failed	111
Unmatched	1063

[a]Since links from PDB to UniProtKB take priority over links in the other direction, this figure considers only those links from UniProtKB to PDB where links in the other direction are absent.

[b]While 10 324 chains were assigned by the brute-force scan, 815 of these were chains in multi-chain PDB files linked from UniProtKB/SwissProt but which were not identified as matching because other chains matched with a higher sequence identity. The true number of additional chains found by the brute-force scan is therefore 9509.

The PostgreSQL database is easily able to cope with the rather large tables. The 'sprot', 'idac' and 'acac' tables have more than 2 million rows each, while the 'alignment' table contains nearly 8 million rows. However, we found it was important to run the PostgreSQL `analyze` command at regular intervals while populating the database. This updates the statistics on the database contents and allows indexes to work with maximum efficiency. If this was not done, the main 'postmaster' process could start to crawl using lots of CPU time and achieving very little.

Table 2 shows the number of chains mapped to UniProt entries from each of the sources of information. The vast majority of entries mapped using a link in the PDB entry will also have a link from UniProt. However, since links from the PDB currently take priority over links from UniProtKB, this information is not recorded.

3.1 Comparison with the EBI mapping

As a validation of the mapping we have created, we have made some comparisons with the mapping produced and kindly provided to us by the EBI.

We have identified one case in which a protein from the wrong species has been identified by our method. PDB entry 1rbf (blank chain name) is an exact match to UniProtKB/SwissProt entry P61824 from *Bison bison*. However 1rbf is a structure of part of the chain from *Bos taurus* (P61823). Over the 104 residues of the sequence included in the structure, these two sequences are 100% identical. Chain A of PDB file 1aby (Looker *et al*., 1992) consists of two copies of the haemoglobin alpha chain (UniProtKB/SwissProt entry P69907) spliced together. Currently our mapping and the EBI MSDLite mapping both match only one of these in the alignment. Thus far, we have identified no other anomalies in our data.

We did, however, find a small number of minor problems in the EBI mapping. PDB entry 1dsj corresponds to UniProtKB/SwissProt entry P12520 and the chain begins with a HETATM 'ACE' group (an N-terminal acetylation) and ends with an additional HETATM 'NH2' group. The most recent downloadable EBI mapping, dated September 21, 2004, maps both of these to real amino acids (Thr49 and Cys76 in the UniProtKB/SwissProt entry, respectively). However, the new mapping from UniProtKB/SwissProt to residue ranges within chains has corrected this error.

We also identified an error in the EBI's downloadable mapping for 5azu which contains four identical chains (A–D). All these match UniProtKB/SwissProt entry P00282. However, in their mapping residues 28–30 of the B chain were erroneously identified as coming from Q51325 (this is a secondary accession code for P19919). Again this error does not occur in the mapping from UniProtKB/SwissProt residue ranges to PDB chains.

The mapping provided in the UniProtKB/SwissProt file provides a PDB chain and then specifies the range of residues within the UniProtKB/SwissProt entry that matches that chain. This scheme is unable to address chimeric sequences such as that found in PDB file 1a7m (Hinds *et al*., 1998). In this PDB file residues 1–47 and 82–180 come from UniProtKB/SwissProt entry P09056 while residues 48–81 come from P15018. In these two UniProtKB/SwissProt entries, a cross-reference to PDB file 1a7m is provided, but the residue range is not given. Our system correctly addresses chimeric chains from the PDB (providing DBREF records are present describing the chimeric construction). The exception to correct processing of chimeric chains is the 'self-chimera', 1aby chain A, described above.

While the downloadable mapping from the EBI is not regularly updated, the MSDLite web server also contains mapping data. We have noted some anomalies in these data as well. For example, while the downloadable mapping for PDB entry 487d adopts the same strategy as ours of simply ignoring non-standard amino acids (MSE at I113, I116 and I182), the MSDLite server correctly identifies the UniProtKB entries, but does not include an alignment at all. Similarly for PDB entry 1val, the MSDLite identifies the same UniProtKB entries as our server, but provides no alignment.

At the time of writing, we have identified 115 chimeric chains in the PDB for which residue range mappings are not present in UniProtKB/SwissProt. As shown in Table 2, the brute-force scan of our method identifies approximately 9500 additional chain mappings (representing ~12.5% of chains in the PDB) for which cross-links were not present in either the PDB or UniProtKB/SwissProt. After accounting for DNA chains, short peptides and cases where `fasta33` failed, only around 1050 chains (1.5% of chains in the PDB) were unassigned to UniProt sequences. Some chains, such as antibodies, are only partial assignments. The constant domain is assigned, but the variable domain is not because antibody variable domains do not appear in UniProt.

The procedure also identified a number of errors in the residue ranges specified in `DBREF` records of PDB files. For example, PDB file 1qsn (Rojas *et al*., 1999) contains a `DBREF` record which indicates that residues 9–19 of chain B should match residues 9–19 of UniProtKB/SwissProt entry P02303 (a secondary accession which has been replaced by P61830). However, the residues in chain B are numbered from 309, so this range should be 309–319. The `DBREF` record in PDB entry 1cxx gives a residue range of 81–193 for the A chain matching Q05158, but the `ATOM` records start from residue 117 and the `SEQRES` records appear to start from 82. Similar problems were identified in PDB entries 1a45, 1dj8, 1dox, 1doy, 1fo7, 1fv2, 1g50, 1g50, 1g6w, 1g6w, 1g6y, 1gd2, 1hgx, 1hqo, 1hqo, 1hr8, 1hr8, 1hr8, 1jid, 1b10, 1k0a, 1k0a, 1k0b, 1k0b, 1ltj, 1m1d, 1kna, 1kne, 4cat, 2pgk, 1bpl.

3.2 Search interface and availability

The complete mapping is available for download via the author's web site at http://www.bioinf.org.uk/pdbsws/. The site also provides a search interface allowing searches on the basis of PDB code (optionally with chain label), UniProtKB accession or UniProtKB/SwissProt identifier, all optionally with residue numbers. The results provide links to the PDB and full UniProtKB entries. The web interface also provides a REST-style API (representational state transfer)—an option to return results in plain text making it easy to parse. This allows simple queries to be made from Perl scripts using the Perl LWP package avoiding the necessity for 'screen scraping' of HTML. This is invaluable for users wishing to employ the results in automated scripts and provides an easy alternative to a SOAP interface. Full instructions are provided on the web site.

ACKNOWLEDGEMENTS

The author wishes to thank members of the MSD and SwissProt groups at the EBI (in particular, Sameer Valenka, Virginie Mittard, Phil McNeil, Rolf Apweiler and Kim Henrick) for making their PDB/SwissProt mapping available. This work was funded by a grant from the Wellcome Trust.

Conflict of Interest: none declared.

REFERENCES

Bairoch,A. *et al*. (2005) The Universal Protein Resource (UniProt). *Nucleic Acids Res*., **33**, D154–D159.

Berman,H.M. *et al*. (2000) The Protein data bank. *Nucleic Acids Res*., **28**, 235–242.

Boeckmann,B. *et al*. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nuclec Acids Res*., **31**, 365–370.

Cavallo,A. and Martin,A.C.R. (2005) Mapping SNPs to protein sequence and structure data. *Bioinformatics*, **21**, 1443–1450.

Chamberlin,S.G. *et al*. (2001) Solution structure of the mEGF/TGFalpha44-50 chimeric growth factor. *Eur. J. Biochem*., **268**, 6247–6255.

Hinds,M.G. *et al*. (1998) Solution structure of leukemia inhibitory factor. *J. Biol. Chem*., **273**, 13738–13745.

Looker,D. *et al*. (1992) A human recombinant haemoglobin designed for use as a blood substitute. *Nature (London)*, **356**, 258–260.

Martin,A.C. *et al*. (1998) Protein folds and functions. *Structure*, **6**, 875–884.

Martin,A.C.R. (2004) PDBSprotEC: a Web-accessible database linking PDB chains to EC numbers via SwissProt. *Bioinformatics*, **20**, 986–988.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Rojas,J.R. *et al*. (1999) Structure of Tetrahymena GCN5 bound to coenzyme A and a histone H3 peptide. *Nature (London)*, **401**, 93–98.

BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 23 2005, pages 4302–4303
doi:10.1093/bioinformatics/bti705

Genome analysis

GARSA: genomic analysis resources for sequence annotation

Alberto M. R. Dávila[1,6,*], Daniel M. Lorenzini[2], Pablo N. Mendes[3], Thiago S. Satake[4], Gabriel R. Sousa[1,6], Linair M. Campos[3], Camila J. Mazzoni[1], Glauber Wagner[1,6], Paulo F. Pires[3], Edmundo C. Grisard[2], Maria C. R. Cavalcanti[5] and Maria Luiza M. Campos[3]

[1]DBBM, Instituto Oswaldo Cruz, Fiocruz, [2]Laboratório de Bioinformática and Laboratório de Protozoologia—MIP/CCB, Universidade Federal de Santa Catarina, [3]Departamento de Ciência da Computação/Núcleo de Computação Eletrônica—UFRJ, [4]Engenharia de Bioprocessos e Biotecnologia, Universidade Federal do Paraná, [5]Instituto Militar de Engenharia and [6]Bioinformatics and Molecular Evolutionary Genetics Group, Brazil

Received on July 10, 2005; revised on September 7, 2005; accepted on October 5, 2005
Advance Access publication October 6, 2005

ABSTRACT

Summary: Growth of genome data and analysis possibilities have brought new levels of difficulty for scientists to understand, integrate and deal with all this ever-increasing information. In this scenario, GARSA has been conceived aiming to facilitate the tasks of integrating, analyzing and presenting genomic information from several bioinformatics tools and genomic databases, in a flexible way. GARSA is a user-friendly web-based system designed to analyze genomic data in the context of a pipeline. EST and GGS data can be analyzed using the system since it accepts (1) chromatograms, (2) download of sequences from GenBank, (3) Fasta files stored locally or (4) a combination of all three. Quality evaluation of chromatograms, vector removing and clusterization are easily performed as part of the pipeline. A number of local and customizable Blast and CDD analyses can be performed as well as Interpro, complemented with phylogeny analyses. GARSA is being used for the analyses of *Trypanosoma vivax* (GSS and EST), *Trypanosoma rangeli* (GSS, EST and ORESTES), *Bothrops jararaca* (EST), *Piaractus mesopotamicus* (EST) and *Lutzomyia longipalpis* (EST).

Availability: The GARSA system is freely available under GPL license (http://www.biowebdb.org/garsa/). For download requests visit http://www.biowebdb.org/garsa/ or contact Dr Alberto Dávila.

Contact: davila@fiocruz.br

The increasing amount of genome data and the consequent possibilities for genome analyses has raised new levels of difficulty for scientists to understand, integrate and deal with all this ever-increasing information. One of the main problems is to manipulate and process different file formats, using a number of tools that usually do not easily communicate with each other. Researchers have to deal with dozens of sequence formats (Rice *et al*., 2000) and several different software packages to analyze nucleotide sequences within a typical bioinformatics pipeline. As a consequence, to overcome heterogeneity, redundancy and low productivity, biologists use alternative strategies such as scripts or adaptation/reuse of some available modules (e.g. Bioperl). Although effective, such approaches are far from being ideal since the intermediate files generated throughout the process are usually not properly stored and organized, generating a large number of files and versions that can potentially lead to processing errors, wrong analyses and/or inferences. The use of database management systems adds facilities such as integrity constraints, transaction management and query languages (SQL), amongst others. Despite the description of several analysis pipelines in the literature, such as the EST pipeline system (Xu *et al*., 2003), the ESTAP (Mao *et al*., 2003) and the ESTWeb (Paquola *et al*., 2003), as far as we know, none of them was specifically designed for GSS analyses or a combination of GSS with transcriptome projects.

Considering the above mentioned problems and the increasing number of network-based projects, in which laboratories can be geographically dispersed, we have conceived a web-based environment named GARSA (genomic analyses resources for sequence annotation), aimed to facilitate the analysis, integration and presentation of genomic information, concatenating several bioinformatics tools and sequence databases, using a flexible and user-friendly approach.

GARSA system is specially designed to analyze genomic data, presenting a pipeline, also called workflow, composed of selected bioinformatics software packages, and an intuitive web-based interface. Its underlying platform includes Perl, Bioperl, CGI, Apache and MySQL, as well as several Linux-based bioinformatics packages. In the current version, the system can analyze EST, Orestes and GSS data, accepting as inputs (1) chromatograms, (2) downloads from GenBank, (3) Fasta files stored locally or (4) a combination of all of these inputs. GARSA uses the Phred/Phrap package (http://www.phrap.org/phredphrapconsed.html) to process chromatograms, evaluate the quality of traces and remove vector contamination. CAP3 program (Huang and Madan, 1999) is used for clustering, while for gene finding, the system employs the Yacop metatool (Tech and Merkl, 2003), which includes programs such as Critica, Glimmer and ZCURVE. Selected programs of the EMBOSS package (Transeq, Geecee, Cusp) (Rice *et al*., 2000) are used to translate, estimate G + C content and codon usage, from both predicted ORFs and clusters. Clusters are submitted to (standalone) Blast similarity searches (http://www.ncbi.nlm.nih.gov/BLAST/) against NR, NT, UniProt and any other custom

*To whom correspondence should be addressed.

database built by the user. Conserved domain searches are also performed using the NCBI's CDD tool and Interpro.

Similarity search results are stored in the corresponding GARSA database tables and then users can select them and build multiple alignments using ClustalW (Thompson *et al*., 1994). Alignments are presented in ClustalW, Phylip and WebLogo (Crooks *et al*., 2004) formats for download and then users can do further analyses with them. Phylogenetic trees are built using SeqBoot, Dnadist/Protdist, Neighbor and Consense programs of the Phylip package (Felsenstein, 1989). A feature for registered users to enter comments or annotations on any clusters has also been added. Several gene discovery projects can easily be included in the system, as GARSA can simultaneously deal with multiple projects.

The Library table identifies each genomic library used in the project. Sequences from each library are uploaded to the Reads table. These sequences may come from local experiments or GenBank downloads. CAP3 results are stored in the Clustering, Clusters and Clusters_Fasta tables. Predicted ORFs identified by Yacop or Glimmer are stored in the ORF_Predicted table. Each similarity search execution is registered in the Similarity_Search table, and its hits are stored in the Blast_Hit and Interpro_Hit tables. The Taxonomy table contains classification data of organisms, which are referred by annotations stored in the Annotation table. Along with the possibility to define parameters, to run and to check the analyses' pipeline, there is also an option to query the results stored in the database and then users can retrieve and manually check the clusters with 'hits' and 'no-hits', using '*E*-value', 'score', 'query_frame', 'query_start', 'query_end' and/or 'description' as parameters.

GARSA is being used for the sequence analyses of *Trypanosoma vivax* (GSS and EST) (Guerreiro *et al*., 2005), *Trypanosoma rangeli* (GSS, EST and ORESTES), *Bothrops jararaca* (EST), *Piaractus mesopotamicus* (EST) and *Lutzomyia longipalpis* (EST). Installation and usage documentation are available at http://www.biowebdb.org/garsa/documentation.html

Furthermore, the GARSA system is unique on integrating (1) gene finders, (2) phylogeny software, (3) multiproject environment and (4) user-based authenticated access. A new version towards comparative genomics analyses is being developed to integrate more software packages in the pipeline, such as GO tools (http://www.geneontology.org/GO.tools.shtml), RepeatMasker (http://repeatmasker.org) and Eukaryotes gene finders, and to provide a self-extract installer for local installation.

ACKNOWLEDGEMENTS

We would like to thank Dr José Marcos Ribeiro (NIAID/NIH) for suggestions and for sharing his experience on EST analysis, João Setubal (VBI and LBI/IC/UNICAMP) for allowing us to modify the algorithm for processing EST chromatograms and MCT/CNPq, IAEA, CIRAD and FAPESP for financial support.

Conflict of Interest: none declared.

REFERENCES

Crooks,G.E. *et al*. (2004) WebLogo: a sequence logo generator. *Genome Res*., **14**, 1188–1190.

Felsenstein,J. (1989) PHYLIP—Phylogeny Inference Package (Version 3.2). *Cladistics*, **5**, 164–166.

Guerreiro,L.T. *et al*. (2005) Exploring the genome of *Trypanosoma vivax* through GSS and in silico comparative analysis. *OMICS* **9**, 116–128.

Huang,X. and Madan,A. (1999) CAP3: a DNA sequence assembly program. *Genome Res*., **9**, 868–877.

Mao,C. *et al*. (2003) ESTAP—an automated system for the analysis of EST data. *Bioinformatics*, **19**, 1720–1722.

Paquola,A.C. *et al*. (2003) ESTWeb: bioinformatics services for EST sequencing projects. *Bioinformatics*, **19**, 1587–1588.

Rice,P. *et al*. (2000) EMBOSS: the european molecular biology open software suite. *Trends Genet*., **16**, 276–277.

Tech,M. and Merkl,R. (2003) YACOP: enhanced gene prediction obtained by a combination of existing methods. *In Silico Biol*., **3**, 441–451.

Thompson,J.D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.

Xu,H. *et al*. (2003) EST pipeline system: detailed and automated EST data processing and mining. *Genomics Proteomics Bioinformatics*, **1**, 236–242.

BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 23 2005, pages 4304–4306
doi:10.1093/bioinformatics/bti691

Sequence analysis

CREDO: a web-based tool for computational detection of conserved sequence motifs in noncoding sequences

Tobias Hindemitt and Klaus F. X. Mayer*

MIPS/Institute for Bioinformatics, GSF Research Centre for Environment and Health, Ingolstaedter Landstrasse 1, 85758 Neuherberg, Germany

Received on May 13, 2005; revised on September 19, 2005; accepted on September 23, 2005
Advance Access publication October 4, 2005

ABSTRACT

Summary: CREDO is a user-friendly, web-based tool that integrates the analysis and results of different algorithms widely used for the computational detection of conserved sequence motifs in noncoding sequences. It enables easy comparison of the individual results. CREDO offers intuitive interfaces for easy and rapid configuration of the applied algorithms and convenient views on the results in graphical and tabular formats.

Availability: http://mips.gsf.de/proj/regulomips/credo.htm

Contact: kmayer@gsf.de

Supplementary information: A detailed help file for CREDO is available on http://mips.gsf.de/proj/regulomips/help.htm. Further supplementary material is available on *Bioinformatics* online.

In higher eukaryotes gene expression is regulated under a variety of constraints such as tissue specificity, developmental or environmental conditions. Understanding the mechanisms that orchestrate this tight regulation is a major challenge in modern biology. RNA-polymerase II-mediated transcription is activated or repressed by transcription factors (TFs). Transcription factor binding sites, also called *cis*-regulatory elements (CREs), constitute a gene's regulatory regions, in particular its promoter. CREs are typically short (6–12 bp) and often degenerate in sequence. Experimental detection and characterization of CREs are feasible but time-consuming and only few examples for experimental CRE detection on genome level have been reported (Lee *et al*., 2002).

Computational methods are powerful, cost effective and can support experimental approaches to detect CREs. The latest approaches are based on cross-species comparison of orthologous sequences (i.e. phylogenetic footprinting) (Gumucio *et al*., 1992; Duret and Bucher, 1997) and the comparative analysis of noncoding sequences from co-expressed genes (van Helden *et al*., 1998; Hughes *et al*., 2000). These approaches are based on the assumption that functional elements are conserved, which allows distinguishing them from non-functional regions in their vicinity. A number of computational tools have been developed to make use of this opportunity. Global alignment tools (e.g. DIALIGN; Morgenstern, 1999) represent the most widely used approach for phylogenetic footprinting. In addition, several motif detection procedures which do not rely on co-linearity have been published. They have either been specifically developed for phylogenetic footprinting (e.g. FootPrinter; Blanchette and Tompa, 2003) more generally for the detection of conserved motifs in the upstream regions of functionally related or co-expressed genes (e.g. AlignACE and MotifSampler; Hughes *et al*., 2000; Thijs *et al*., 2001) or for the identification of sequence conservation in biopolymers (e.g. MEME; Bailey and Elkan, 1994).

It has been suggested to compare and integrate results derived from different methods and to use complementary tools in combination rather than rely on a single method (Tompa *et al*., 2005). Such an approach is time-consuming and difficult owing to varying output formats and different graphical representation of the respective results. CREDO (*Cis*-Regulatory Element Detection Online) integrates, combines and visualizes the analyses of AlignACE, DIALIGN, FootPrinter, MEME and MotifSampler and therefore facilitates the comparison of their results. In contrast to AlignACE, MEME and MotifSampler—which are designed for a more general detection of conserved sequence motifs—FootPrinter is focused on the detection of conserved sequence motifs in noncoding sequences of orthologous genes and should not be used for applications like the identification of sequence conservation in the upstream regions of co-expressed genes. CREDO enables to run each of the algorithms simultaneously on a given dataset and summarizes the outputs of all programs graphically, in tables and within an XML file. In order to ensure complete platform independence and to avoid installation of additional software on the user side, CREDO interfaces are exclusively based on standard web technologies. Details on the integration of the different algorithms are provided as Supplementary material.

Input sequences can be pasted into the CREDO web form. Regions of special interest (e.g. known regulatory elements) can be indicated by capital letters. These regions will be highlighted in the graphical output (Fig. 1A).

Almost all parameters of the algorithms applied can be adapted. The CREDO web interface provides a structured parameter selection form. Parameters are subdivided into two groups: basic parameters (e.g. motif size or motif number) and advanced parameters. By default advanced parameters are hidden and preset values are being used. For expert users the opportunity to change these parameters and refine the analysis is provided. As a starting point, CREDO provides three different and widely applied presettings. The first presetting has been designed for users who aim to carry out phylogenetic footprinting with closely related species. The second presetting has been designed for phylogenetic footprinting with more distantly related species and finally the third presetting for users who set out to search for conserved sequence motifs in co-expressed

*To whom correspondence should be addressed.

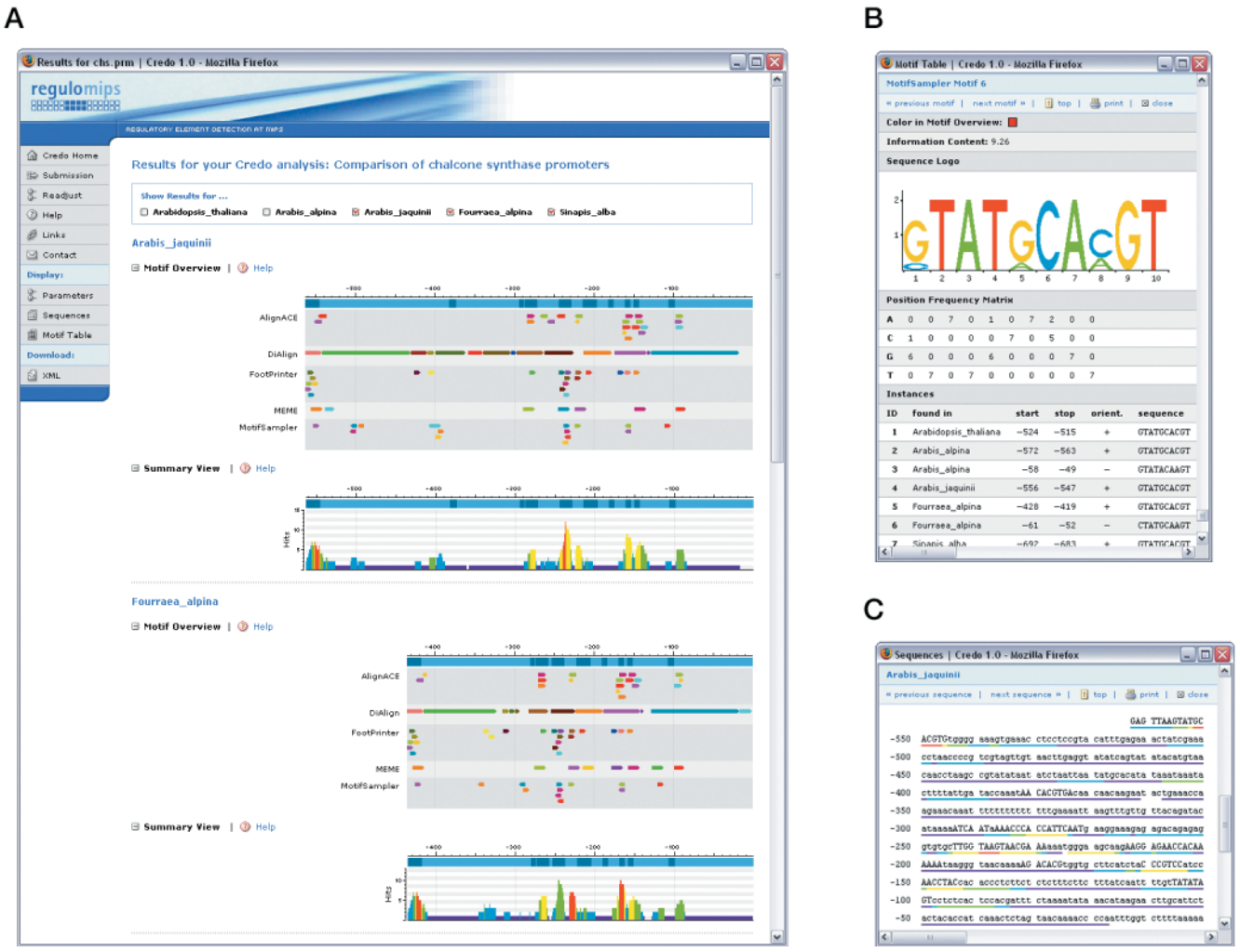

Fig. 1. CREDO result page. (**A**) Input sequences are represented as blue bars. Sequence regions marked within the sequence input are highlighted in darker blue. Within the motif overview occurrences of predicted motifs are depicted as coloured arrows and are linked to the respective motif data (see B). The summary view graphically summarizes the motifs found by all programs for each base pair. The number of total motif hits is indicated by the height of the bar and the number of different programs that report a motif is colour coded (purple, one; light blue, two; green, three; yellow, four; red, five). The panel can be hidden by clicking on the '−' symbol. (**B**) The motif table provides all relevant data and a sequence logo for each predicted motif. (**C**) Input sequences are displayed. Positions hit by one or several motif predictions are underlined. Colours correspond to those within the summary view (see A).

genes is provided. It is important to emphasize that these presettings provide only a starting point for the analysis. The parameter selection should be refined in subsequent analyse since optimized parametrization is a prerequisite for significant results.

After completion of the analysis the user is notified by e-mail and a graphical overview of the results is made available (Fig. 1A). For each input sequence the motifs detected by the individual algorithms along with a summary view are displayed. This view summarizes the motifs found by all programs and hence facilitates the identification of sequence regions where results of the different algorithms are coincident.

The graphical representations of motif occurrences are linked to underlying analytical data. The result pages include links to three pop-up windows that contain the table of found motif data, input sequences and chosen parameters, respectively. The motif table (Fig. 1B) provides all important motif data and includes a sequence logo (Schneider and Stephens, 1990; Lenhard and Wasserman, 2002)—a highly intuitive graphical representation—for each motif detected. To display motifs in their sequence environment, the respective positions are underlined in the pop-up window depicting the input sequences (Fig. 1C).

All relevant data, including input sequences and parameters used, as well as the analytical results can be downloaded as XML file. An example that illustrates the effectiveness of CREDO is provided within the Supplementary material and in more detail on the CREDO homepage (http://mips.gsf.de/proj/regulomips/credo.htm).

ACKNOWLEDGEMENTS

The authors would like to acknowledge Daniela Fölsl, Stefanie Maisel and Heiko Schoof for help during the web server implementation of CREDO and Thomas Rattei for providing additional

infrastructure. The authors would also like to thank Claudia Englbrecht for critical reading of the manuscript. This work has in part been founded by the GABI programme of German Ministry of Education and Research (BMBF).

Conflict of Interest: none declared.

REFERENCES

Bailey,T.L. and Elkan,C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **2**, 28–36.

Blanchette,M. and Tompa,M. (2003) FootPrinter: a program designed for phylogenetic footprinting. *Nucleic Acids Res.*, **31**, 3840–3842.

Duret,L. and Bucher,P. (1997) Searching for regulatory elements in human noncoding sequences. *Curr. Opin. Struct. Biol.*, **7**, 399–406.

Gumucio,D.L. *et al.* (1992) Phylogenetic footprinting reveals a nuclear protein which binds to silencer sequences in the human gamma and epsilon globin genes. *Mol. Cell. Biol.*, **12**, 4919–4929.

Hughes,J.D. *et al.* (2000) Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.*, **296**, 1205–1214.

Lee,T.I. *et al.* (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.

Lenhard,B. and Wasserman,W.W. (2002) TFBS: computational framework for transcription factor binding site analysis. *Bioinformatics*, **18**, 1135–1136.

Morgenstern,B. (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, **15**, 211–218.

Schneider,T.D. and Stephens,R.M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.*, **18**, 6097–6100.

Thijs,G. *et al.* (2001) A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, **17**, 1113–1122.

Tompa,M. *et al.* (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.*, **23**, 137–144.

van Helden,J. *et al.* (1998) Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.*, **281**, 827–842.

BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 23 2005, pages 4307–4308
doi:10.1093/bioinformatics/bti698

Genetics and population analysis

OxfordGrid: a web interface for pairwise comparative map views

Hongyu Yang and Alan R. Gingle*

Center for Applied Genetic Technologies, University of Georgia, 111 Riverbend Road, Athens, GA 30602, USA

Received on August 5, 2005; revised and accepted on September 27, 2005
Advance Access publication October 4, 2005

ABSTRACT

Summary: OxfordGrid is a web application and database schema for storing and interactively displaying genetic map data in a comparative, dot-plot, fashion. Its display is composed of a matrix of cells, each representing a pairwise comparison of mapped probe data for two linkage groups or chromosomes. These are arranged along the axes with one forming grid columns and the other grid rows with the degree and pattern of synteny/colinearity between the two linkage groups manifested in the cell's dot density and structure. A mouse click over the selected grid cell launches an image map-based display for the selected cell. Both individual and linear groups of mapped probes can be selected and displayed. Also, configurable links can be used to access other web resources for mapped probe information.

Availability: OxfordGrid is implemented in C#/ASP.NET and the package, including MySQL schema creation scripts, is available at ftp://cggc.agtec.uga.edu/OxfordGrid/

Contact: agingle@uga.edu

INTRODUCTION

OxfordGrid was initially developed to provide comparative views of orthologous loci for our cotton (http://cotton.agtec.uga.edu/OxfordGrid/index.aspx) and CGGC/sorghum (http://cggc.agtec.uga.edu/oxfordgrid/images.aspx) database resources. Its development, motivated by the need to provide real-time Oxford grid generation and relational database (RDBMS) compatibility, benefited from collaboration with practical genetic mappers developing high-density genetic maps for cotton (Rong *et al*., 2004) and sorghum (Bowers *et al*., 2003). The dynamic image map-based interface facilitates real-time interactive features like 'linear selection mode' that allows users to identify syntenic regions and list their associated probes. Other applications that generate Oxford grids for genetic map or similar dot-plots for genomic sequence data have been developed (e.g. Brodie *et al*., 2004; Edwards, 1991; Huang and Zhang, 2004); however, they are not compatible with our strategy of implementing real-time interactive features that are compatible with a broad range of web browsers and do not require client-side software additions. Our approach has led to the development of a MySQL compatible web application that provides a compact and interactive Oxford Grid interface and database. Like its CGGC counterpart, the application can also be configured to serve as a portal to other web resources with the configuration details described in the installation document (ftp://cggc.agtec.uga.edu/OxfordGrid/readme.pdf).

*To whom correspondence should be addressed.

DESCRIPTION

OxfordGrid is a web application that generates an interactive Oxford grid (Edwards, 1991) display for the study of genome organization from comparative genetic map data. It is composed of a matrix of grid cells, each representing a pairwise comparison of map data for linkage group pairs. The linkage groups are arranged along the axes with one forming cell columns and the other cell rows. A dot inside a cell represents a molecular marker/probe that has been mapped to both linkage groups with its coordinates being the associated map locations. The degree and the pattern of synteny between the two linkage groups are manifested in the dot density and pattern within the cell. This is especially true for high-density maps where inserts can be detected and orientation established based on the presence and slope of linear segments in the dot pattern. Of course, chromosomal duplications are apparent as well.

Using OxfordGrid to compare two linkage groups is a multistep process. Combinations of maps/species are selected from a menu, which launches the multicell display (Fig. 1A), providing a comparative overview for all linkage group pairs. A detailed view of a particular linkage group pair (Fig. 1B) can be launched with a mouse click over its associated cell as indicated by the arrow. This view provides two types of queries. In the point mode, each dot is an interactive link that can be configured to navigate either directly to a detailed probe information page or via a list of probes when more than one are associated with a single dot's map coordinates. All navigation is automatic with the configuration details described in the installation package's readme file. In the linear mode, users can, with mouse clicks, select endpoints of an identified linear segment (Fig. 1B). The interface then superimposes a line segment between the endpoints, highlights the dots on or near the linear segment and displays a list of the associated probe hyperlinks, which are also configurable (Fig. 1C).

OxfordGrid is an ASP.NET (Microsoft) web application and is compatible with the Windows Internet Information Server (IIS). The Oxford grid images are generated, on the fly, as bitmap objects using the .NET framework's drawing classes and the interactive features of the interface are implemented using classic HTML image maps. It is a data-driven application and the downloadable version is compatible with a MySQL RDBMS back-end. The associated schema contains tables to accommodate map, synteny and

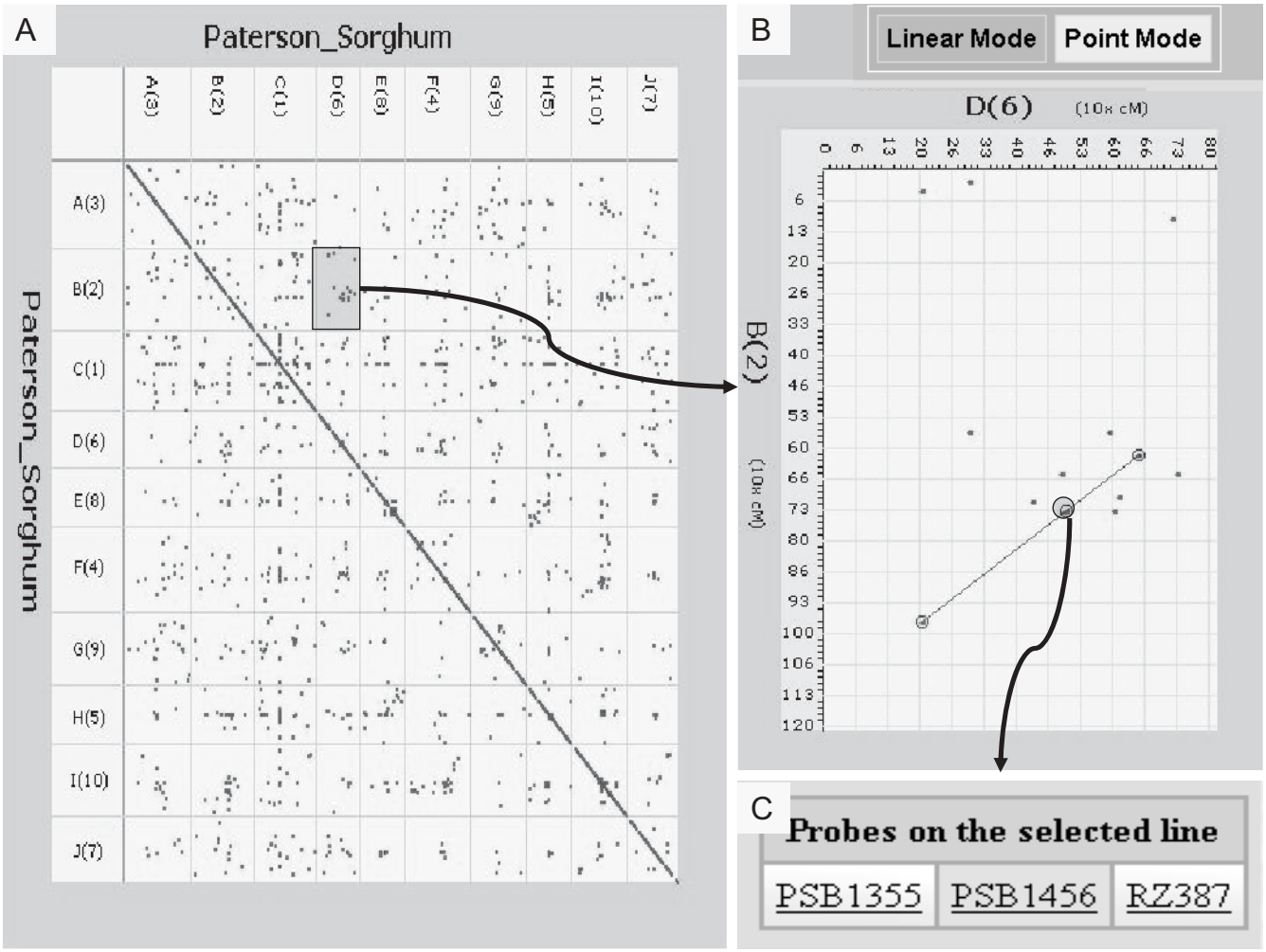

Fig. 1. Views providing an illustration of the OxfordGrid's functionality are shown. The multiple cells view (**A**) contains a grid of cells representing pairwise comparisons for each of the possible linkage group combinations and provides an overview of the comparative map. A mouse click over a selected cell launches a detailed view (**B**) where, e.g. linear groups of probes can be identified and selected to obtain probe lists/links (**C**).

taxonomy-related data as well as configurable URL/navigation information. The table for synteny data can become quite large as the number of possible map pairs increases and, therefore, indexes are employed for species and linkage group. Detailed information on the schema is available in the installation package readme file. Also, to extend OxfordGrid's application range, we have developed scripts for importing relevant data from the GMOD/CMap (http://www.gmod.org/cmap/index.shtml) database schema and these are included in the installation package along with instructions on their use and integration into data processing pipelines. Information is also included on the data mappings between the two schemas.

ACKNOWLEDGEMENTS

The authors wish to thank the collaborating laboratories for providing data and advice. We are grateful to the National Science Foundation, the United States Department of Agriculture, the Georgia Research Alliance, the National Grain Sorghum Producers and the University of Georgia Research Foundation for financial support.

Conflict of Interest: none declared.

REFERENCES

Bowers,J.E. *et al.* (2003) A high-density genetic recombination map of sequence-tagged sites for sorghum, as a framework for comparative structural and evolutionary genomics of tropical grains and grasses. *Genetics*, **165**, 367–386.

Brodie,R. *et al.* (2004) Jdotter: a Java interface to multiple dotplots generated by dotter. *Bioinformatics*, **20**, 279–281.

Edwards,J.H. (1991) The Oxford grid. *Ann. Hum. Genet.*, **55**, 17–31.

Huang,Y. and Zhang,L. (2004) Rapid and sensitive dot-matrix methods for genome analysis. *Bioinformatics*, **20**, 460–466.

Rong,J. *et al.* (2004) A 3347-locus genetic recombination map of sequence-tagged sites reveals features of genome organization, transmission and evolution of cotton (Gossypium). *Genetics*, **166**, 389–417.

BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 23 2005, pages 4309–4311
doi:10.1093/bioinformatics/bti689

Genetics and population analysis

HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients

Giovanni Montana

Department of Mathematics, Imperial College, London, UK

Received on August 1, 2005; revised on September 21, 2005; accepted on September 22, 2005
Advance Access publication September 27, 2005

ABSTRACT

We have developed a simulation tool `HapSim` for the generation of haplotype data. The simulated haplotypes are such that their allele frequencies and linkage disequilibrium coefficients match exactly those estimated in a real sample.

Availability: The program is available as an R package and can be downloaded from http://cran.r-project.org/

Contact: g.montana@imperial.ac.uk

INTRODUCTION

With the widespread availability of molecular data, novel computational methods for gene mapping are being developed at an increasing rate. It is often the case that the statistical properties and behavior of these methods need to be assessed and tested by simulation, especially when no analytical derivations are available. A number of simulation programs have been developed and are currently used to this end; see, for instance, Hudson (2002) and Excoffier *et al*. (2000).

Once a mechanism for the generation of DNA polymorphism data is in place, entire case–control association studies can also be simulated, e.g. through the random allocation of phenotypes conditional on genotypes at one or more loci, and according to a variety of different genetic models. Among other applications, such a simulation framework allows for comparisons and performance evaluations of multi-locus gene mapping methods to be made.

We suggest a simple haplotype simulation approach whose main feature lies in its ability to produce haplotype data with pre-specified allele frequencies and pairwise linkage disequilibria as measured by the r^2 index. Assuming that linkage disequilibria of order greater than two can be ignored, a haplotype is modeled as a multivariate random variable possessing known marginal distributions and pairwise correlation coefficients. We have implemented an algorithm that can efficiently generate large samples of such variables.

SIMULATION ALGORITHM

Suppose we have collected a sample of N haplotypes comprising L binary markers coded as 0 and 1, and call p_i the estimated allele frequency at locus i. Further assume that, for all pairs of markers i and j, the linkage disequilibrium coefficient r_{ij}^2 has been calculated as $(p_{ij} - p_i p_j)^2 [p_i p_j (1 - p_i)(1 - p_j)]^{-1}$. The joint probability p_{ij} of observing the same allele at both loci is then easily derived.

We now wish to simulate a multivariate Bernoulli variable $S = (S_1, S_2, \ldots, S_m)$ such that each variable S_i has a marginal distribution $p_i = \Pr(S_i = 1)$, and the correlation between each pair of variables, say S_i and S_j, is exactly r_{ij}. The following algorithm draws a random sample from the distribution of S:

(1) Compute a covariance matrix $\mathbf{C} = \{c_{ij}\}$ by solving the equations $\Phi(z(p_i), z(p_j); c_{ij}) = p_{ij}$ uniquely for c_{ij} $(1 \leq i < j \leq L)$. Here the notation $\Phi(z(x), z(y); \rho)$ refers to the c.d.f. of a standard bivariate normal distribution with correlation coefficient ρ, whereas $z(p)$ denotes the p-th quantile of the univariate standard normal distribution.

(2) Simulate a random vector $m = (m_1, \ldots, m_L)$ from a multivariate normal distribution with mean vector $(0, \ldots, 0)$ and covariance matrix $\mathbf{C}$. The required random sample $s = (s_1, s_2, \ldots, s_L)$ is then obtained by thresholding the components of m according to the following rule: $s_i = 1$ if $m_i \leq z(p_i)$ else $s_i = 0$.

The underlying simulation strategy we adopted follows a well-known general recipe taken from the stochastic simulation cookbook, i.e. transform a random variable to a normally distributed variate via the c.d.f. and then map back. Despite its appealing simplicity, this algorithm had not been implemented before in the context of simulating binary genetic markers.

A few computational remarks are in order. The $L(L-1)/2$ equations that appear in Step (1) are solved with high precision by an iterative bisection method; the function implementing the equation solver has been written in C in order to achieve a substantial speed-up at run-time. It is important to notice that the covariance matrix $\mathbf{C}$ obtained from all the pairwise calculations of Step (1) is not guaranteed to be positive defined. When this is indeed the case, the non-positive definitiveness poses a problem, as a working covariance matrix is needed for sampling from a multivariate normal distribution[1]. Accordingly, the algorithm executes an intermediate step as needed:

(1a) Replace the non-positive definite covariance matrix $\mathbf{C}$ by its approximated positive definite version.

The approximation above, sometimes called bending (Hayes and Hill, 1981), requires performing an initial spectral decomposition of $\mathbf{C}$; all the eigenvalues smaller than a minimum tolerance threshold are then replaced by this minimum value.

In all cases when the covariance matrix approximation applies, the mean square error between the target r^2 and the corresponding

[1]Multivariate normal variates are drawn by the *mvrnorm* function in the MASS library, also available from the CRAN repository.

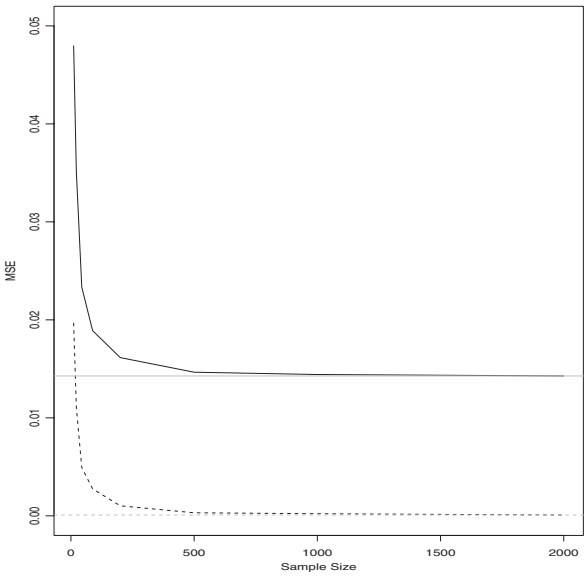

Fig. 1. ACE data: mean square error of marginal probabilities (dashed line) and pairwise correlations (solid line) against sample size.

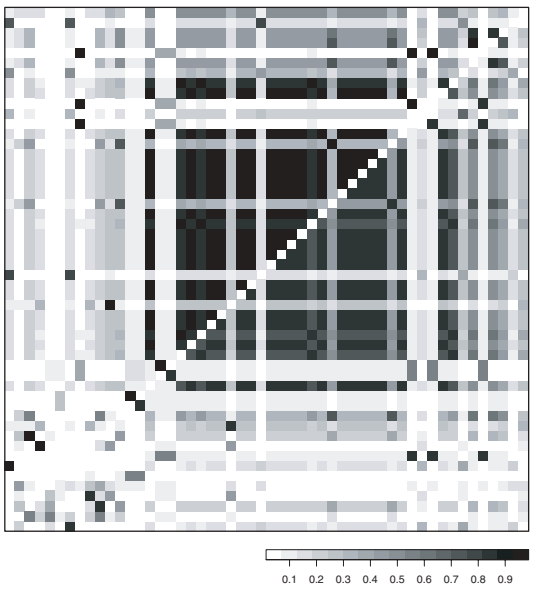

Fig. 2. ACE data: hybrid LD matrix obtained from real and artificial samples of 11 individuals each.

LD coefficient estimated from a simulated sample will stay above a positive constant, even when the sample size increases. However, as we briefly illustrate in the sequel, the approximation error is indeed small, at times almost negligible, and the estimated LD plot is generally in very good agreement with its target. On the other hand, the estimated allele frequencies are not affected by this approximation, and are unbiased.

ILLUSTRATION

Let us assume that a sample dataset X has been loaded into R. The generation of virtual haplotypes is a 2-fold process: first, the covariance matrix $\mathbf{C}$ is computed by creating a haplotype object $H \leftarrow$ `haplodata`(X); then, a sample Y of n haplotypes is obtained by the command $Y \leftarrow$ `haplosim`(n, H). Other summary statistics (e.g. allele frequencies and a locus-specific genetic diversity index) can also be easily extracted from both the H and Y objects.

We briefly illustrate the use of `HapSim` on the ACE (angiotensin I converting enzyme) data, as analyzed and made available by Lin and Altman (2004). This dataset contains 52 SNPs typed on DCP1 in 11 individuals. All the 22 phased chromosomes were used as a target sample. We then simulated haplotype samples of sizes ranging from 12 to 2000. For each sample size, 100 replicated samples were generated, and the average mean square errors of both marginal probabilities and correlation coefficients were obtained.

Figure 1 clearly shows that, as the sample size increases, the marginal probability estimates quickly converge to the target values; on the other hand, the error in the estimation of pairwise correlations does not go down to zero due to the covariance matrix approximation. The minimum bias obtained at large sample sizes is marked in Figure 1 by a horizontal line.

Figure 2 reassures that this error is not much of a concern. The upper triangular matrix represents the LD coefficients as estimated from the real data, whereas the lower triangular matrix displays the corresponding LD in a small sample of 22 simulated haplotypes; it clearly emerges that the block-like LD structure of the real data is reproduced with high accuracy.

DISCUSSION

We have described a simple simulation procedure for generating artificial haplotypes matching pre-specified first and second order data summaries. An important assumption of our model is that the pairwise LD coefficients describe sufficiently well the real association structure among markers in the dataset at hand, whereas disequilibria of order greater than two (e.g. LD involving three or more markers) are close to zero, and can be ignored. When this assumption is violated, and in regions with very low LD, the simulated haplotype patterns may not always be in very good agreement with the observed ones.

Our implementation only relies on simulating a multivariate normal distribution, and therefore allows the user to efficiently generate many large samples in a matter of a few seconds. Even the initial and most computationally intensive task involving the solution of a large number of equations is quickly performed on a common desktop PC.

The R package provides a set of functions for haplotype simulation which is intended to be used as a component of a broader simulation study coded in R. Two potential applications are in order. The LD coefficient r^2 arises naturally in the context of association mapping studies and is related to their power (Pritchard and Przeworski, 2001); therefore `HapSim`, being able to generate large samples of chromosomes that closely mimic observed r^2 patterns, can be used as a building block of a simulation program for power and sample size estimation. We are currently employing it to characterize the properties of selected SNP tagging algorithms that only rely on pairwise LD. Other possible applications are under evaluation in a collaboration with GlaxoSmithKline.

Conflict of Interest: none declared.

REFERENCES

Excoffier,L. *et al*. (2000) SIMCOAL: a general coalescent program for the simulation of molecular data in interconnected populations with arbitrary demography. *J. Hered*., **91**, 506–509.

Hayes,J.F. and Hill,W.G. (1981) Modification of estimates of parameters in the construction of genetic selection indices ('bending'). *Biometrics*, **37**, 483–493.

Hudson,R.R. (2002) Generating samples under a Wright–Fisher neutral model of genetic variation. *Bioinformatics*, **18**, 337–338.

Lin,Z. and Altman,R.B. (2004) Finding haplotype tagging SNPs by use of principal components analysis. *Am. J. Hum. Genet*., **75**, 850–861.

Pritchard,J.K. and Przeworski,M. (2001) Linkage disequilibrium in humans: models and data. *Am. J. Hum. Genet*., **69**, 1–14.

BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 23 2005, pages 4312–4314
doi:10.1093/bioinformatics/bti704

Databases and ontologies

ATID: a web-oriented database for collection of publicly available alternative translational initiation events

Jun Cai*, Jing Zhang, Ying Huang† and Yanda Li

MOE Key Laboratory of Bioinformatics and Department of Automation, Tsinghua University, Beijing 100084, China

Received on June 5, 2005; revised on September 15, 2005; accepted on October 3, 2005
Advance Access publication October 10, 2005

ABSTRACT

Summary: Alternative translational initiation is an important cellular mechanism contributing to the diversity of protein products and functions. We develop a database that provides a comprehensive collection of alternative translational initiation events. The purpose of this alternative translational initiation database (ATID) is to facilitate the systematic study of alternative translational initiation of genes. The current version of database contains 300 genes from *Homo sapiens*, *Mus musculus* and other species. Each of the genes has two or more isoforms due to alternative translational initiation. Resources in ATID, including gene information, alternative products of genes and domain structures of isoforms, are provided through a user-friendly web interface.

Availability: The ATID database is available for public use at http://bioinfo.au.tsinghua.edu.cn/atie/

Contact: caijun99@mails.tsinghua.edu.cn

Supplementary information: Supplementary instructions about this database and further statistical analyses can be found on the web page (http://bioinfo.au.tsinghua.edu.cn/atie/)

The mechanisms to generate protein diversity by alternative gene expression allow an organism to increase its level of complexity. These mechanisms include alternative use of promoters, splice sites and translational initiation codons. Specifically, the use of alternative translational initiation codons in a single mRNA contributes to the generation of protein diversity. The genes produce two or more versions of the proteins and the shorter version initiated from a downstream in-frame start codon lacks the N-terminal amino acids of the full-length isoform version (Kozak, 2002).

Since the first discovery of alternative translational initiation, a small, yet growing, number of mRNAs initiating translation from alternative start codons have been reported (Kozak, 2002). Alternative translational initiation is an important cellular mechanism in the expression process of mRNAs. Experimental studies of individual genes revealed cases of functional differences (attributed to insertions, deletions or substitutions of functional protein domains and selection of the cellular localization) among alternatively initiated isoforms (Lock *et al*., 1991; Luque *et al*., 1998; Perry *et al*., 2000). However, a systematic collection of existing observations on alternative translational initiation events is still lacking, which limits the progress in the functional analysis of this important mechanism. The need for such a resource is clear with respect to the important fact that translational polymorphism acts as a potential source of proteins variety (Kochetov and Sarai, 2004; Kochetov *et al*., 2005). In this paper, we establish a database named ATID (alternative translational initiation database) as a systematic collection of all publicly available genes with alternative translational initiation, including their alternative translational isoforms and detailed annotations.

ATID is a web-oriented database that uses a WWW-browser interface to access data under the SQL framework. The records of alternative translational events are converted and stored in a MySQL database program (http://www.mysql.com). The web page allows interaction between the users and the data application. Indexing key identifiers of the database optimizes batch queries from the HTTP web page interface. When a HTTP request is triggered, we can import items of SQL database into the application. Then, the results are sent in the format of HTML to the website.

The database contains 650 alternatively translated protein products belonging to a total of 300 genes. These records of alternative translational initiation were mainly collected from the Swiss-Prot protein database (Bairoch and Apweiler, 2000) and the Entrez protein database on NCBI (http://www.ncbi.nlm.nih.gov/Entrez/). Some other data, the information of 89 alternative translational initiation events was extracted from published research literature. The genes involve many species including *Homo sapiens*, *Mus musculus*, *Bos taurus*, *Saccharomyces cerevisiae*, Virus, etc.

An overview is created to illustrate the database interface on web pages. A tree structure of the species classifying the database entries is constructed and all the entries can be browsed according to the species they belong to. Each entry includes detailed information of the gene including the accession identifier, species, gene name, publication or database references and other information on the isoform products. By clicking on isoform ID, the information on alternative translational products will be displayed. Elements such as isoelectric point (pI) value, molecular weight and sequence information are designated to annotate the isoform products of the gene. The distributions of the domain content in the amino acids sequence, which concerns with the protein function, are scanned by the family matching system Pfam 16.0 (http://pfam.wustl.edu) (Bateman *et al*., 2002) with a given threshold of the *E*-value and are displayed for each isoform product. Also, a topological structure graph is given to help users understand the process of alternative translational initiation from an mRNA to different isoforms. Figure 1A shows the topological structure of the FGF2 gene in

*To whom correspondence should be addressed.
†Present address: Center of Excellence in Bioinformatics, University at Buffalo, 901 Washington Street, Buffalo, NY 14203, USA.
The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

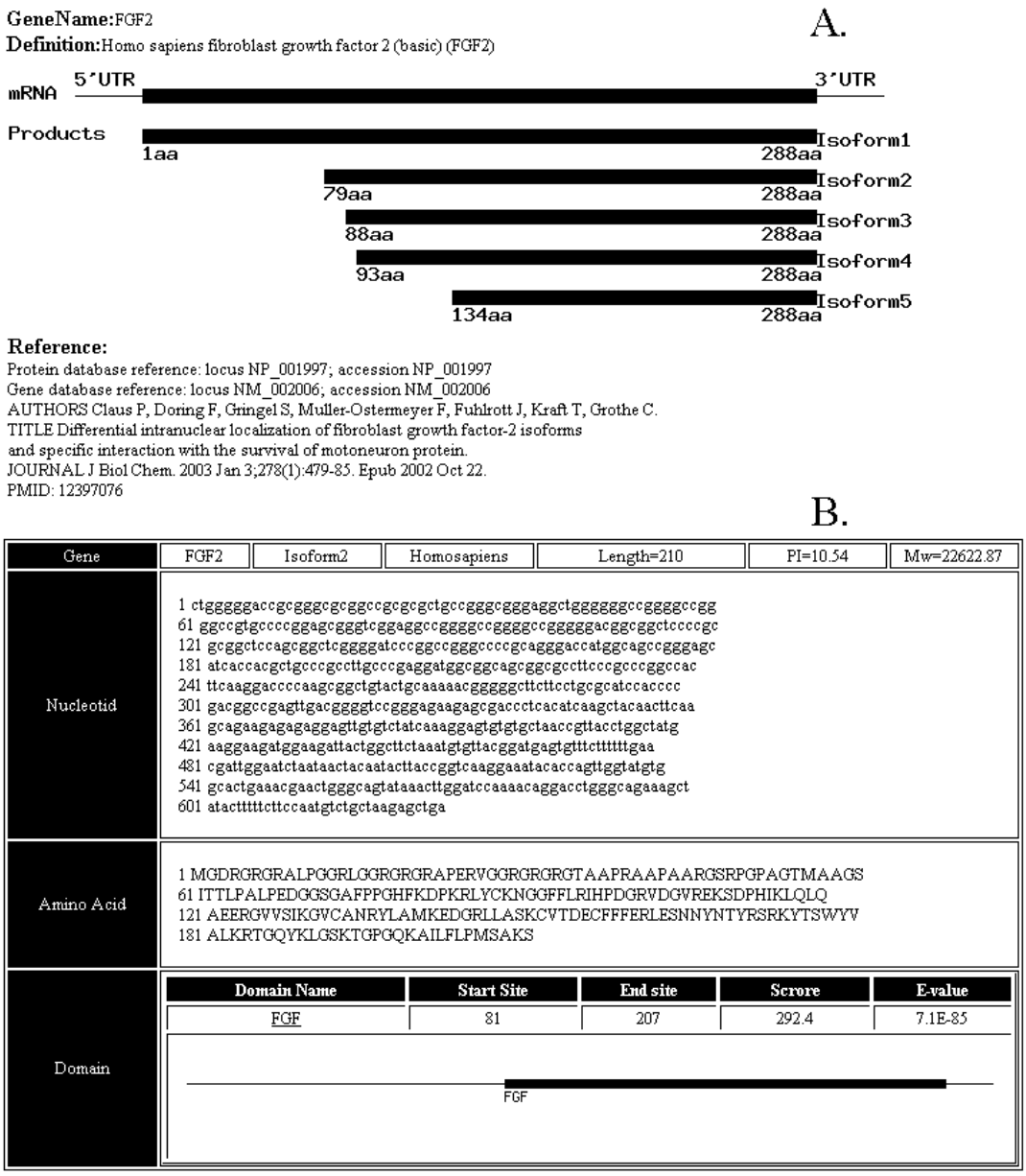

Figure 1. ATID screenshots. (**A**) Topological structure of five alternative isoforms initiated from FGF2 gene and its references of NCBI database and publication. (**B**) Annotation page of 22.6 kDa isoform encoded from FGF2 gene, including its sequence and domain content information.

H.sapiens as an example. In Figure 1A, there are four shorter isoforms lacking the N-terminal amino acids fragments from the original 30.8 kDa full-length isoform. The annotation page for the 22.6 kDa isoform of FGF2 is shown in Figure 1B as an example. The database supports two other ways of querying the entries besides directly accessing the entries on the species tree. One is by keyword query and the other is by query of sequence similarity. In the former way, keywords such as the accession number and the gene name can be submitted. In the latter case, the nucleotide or amino acid sequences can be submitted in FASTA format through the web interface. The submitted sequence will be compared with the sequences in the database by the BLAST program (Altschul *et al*., 1997). The cluster of database entries that contains the most similar sequences will be returned in a table format.

The ATID database provides a resource for future biological analyses of alternative translational initiation, including the development of computational methods for the functional analysis of alternative translational initiation. It presents the most comprehensive collection of alternative translational events to date, and it will be updated timely by continuous collection of new records appearing in the literature and other databases. More features like the prediction of genes with putative alternative translational initiation and their functions will be developed in future updates of the database.

ACKNOWLEDGEMENTS

We thank Dr X. Zhang for his help in the writing. This work was partially supported by the NSFC grants (60171038 and 60234020) and the PhD student grant of Tsinghua University.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*., **28**, 45–48.

Bateman,A. *et al*. (2002) The Pfam protein families database. *Nucleic Acids Res*., **30**, 276–280.

Kochetov,A.V. and Sarai,A. (2004) Translational polymorphism as a potential source of plant proteins variety in *Arabidopsis thaliana*. *Bioinformatics*, **20**, 445–447.

Kochetov,A.V. *et al*. (2005) The role of alternative translation start sites in the generation of human protein diversity. *Mol. Genet. Genomics*, **273**, 491–496.

Kozak,M. (2002) Pushing the limits of the scanning mechanism for initiation of translation. *Gene*, **299**, 1–34.

Lock,P. *et al*. (1991) Two isoforms of murine hck, generated by utilization of alternative translational initiation codons, exhibit different patterns of subcellular localization. *Mol. Cell. Biol*., **11**, 4363–4370.

Luque,C.M. *et al*. (1998) An alternative domain determines nuclear localization in multifunctional protein 4.1. *J. Biol. Chem*., **273**, 11643–11649.

Perry,M.E. *et al*. (2000) P76MDM2 inhibits the ability of p90MDM2 to destabilize p53. *J. Biol. Chem*., **275**, 5733–5738.

BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 23 2005, pages 4315–4316
doi:10.1093/bioinformatics/bti693

Databases and ontologies

MACiE: a database of enzyme reaction mechanisms

Gemma L. Holliday[1], Gail J. Bartlett[2,†], Daniel E. Almonacid[1], Noel M. O'Boyle[1], Peter Murray-Rust[1], Janet M. Thornton[2] and John B. O. Mitchell[1,*]

[1]Unilever Centre for Molecular Science Informatics, Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge, CB2 1EW, UK and [2]EMBL-EBI, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, UK

Received on July 21, 2005; revised on September 22, 2005; accepted on September 23, 2005
Advance Access publication September 27, 2005

ABSTRACT

Summary: MACiE (mechanism, annotation and classification in enzymes) is a publicly available web-based database, held in CMLReact (an XML application), that aims to help our understanding of the evolution of enzyme catalytic mechanisms and also to create a classification system which reflects the actual chemical mechanism (catalytic steps) of an enzyme reaction, not only the overall reaction.
Availability: http://www-mitchell.ch.cam.ac.uk/macie/
Contact: jbom1@cam.ac.uk

A great deal of knowledge about enzymes, including structures, gene sequences, mechanisms, metabolic pathways and kinetic data, now exists. However, it is spread between many different databases and throughout the literature. Here we announce the completion of the initial version of MACiE, a unique database of the chemical mechanisms of enzymatic reactions.

Web resources such as BRENDA (Schomburg *et al*., 2004), KEGG (Kanehisa *et al*., 2004) and the International Union of Biochemistry and Molecular Biology (IUBMB) Enzyme Nomenclature website (IUBMB, 2005, http://www.chem.qmul.ac.uk/iubmb/enzyme/) contain descriptions of the overall reactions performed by enzymes, accompanied in some cases by a textual or graphical description of the mechansim. MACiE is unique in combining detailed stepwise mechanistic information (including 2D animations), a wide coverage of both chemical space and the protein structure universe, and the chemical intelligence of CMLReact (Holliday,C.L., Murray-Rust,P., and Rzepa,H.S., 2005, manuscript submitted to *J. Chem. Inf. Modeling*). MACiE usefully complements both the mechanistic detail of the Structure–Function Linkage Database (SFLD) for a small number of enzyme superfamilies (Pegg *et al*., 2005) and the wider coverage with less chemical detail provided by EzCatDB (Nagano, 2005) which also contains a limited number of 3D animations.

*To whom correspondence should be addressed.
†Present Address: Bioinformatics Support Service (Biochemistry Building), Centre for Bioinformatics, Division of Molecular Biosciences, Faculty of Life Sciences, Imperial College London, London, SW7 2AZ, UK

DESIGN

The MACiE dataset evolved from that published in the Catalytic Site Atlas (CSA) (Bartlett *et al*., 2002; Porter *et al*., 2004), and each entry is selected so that it fulfils the following criteria:

(1) There is a 3D crystal structure of the enzyme deposited in the Protein Databank (PDB) (Berman *et al*., 2000).
(2) There is a relatively well-understood mechanism available. Taken from the literature, these cover a variety of methodologies, including chemical and biochemical studies, quantum mechanical calculations and structual biology reports.
(3) The enzyme is unique at the H level of the CATH classification—a hierarchical classification system of protein domain structures (Orengo *et al*., 1997)—unless there is a homologue with a significantly different chemical mechanism.
(4) Where there are a number of possible PDB codes available the entry should be, if possible, a wild-type enzyme.

All MACiE enzymes are also contained in the Enzyme Commission (EC) classification system (IUBMB, 2005, http://www.chem.qmul.ac.uk/iubmb/enzyme/), that is, they all have four number codes describing their overall reaction. The first level (Class) describes the basic reaction type. The second and third levels (subclass and sub-subclass, respectively) describe the reaction in further detail and the final level (serial number) describes substrate specificity. For example, the β-lactamases (Fig. 1) are assigned the EC number 3.5.2.6, i.e. a hydrolase (3) acting on a C–N bond (5) in a cyclic amide (2) with a β-lactam as the substrate (6).

In MACiE, the data centre on the catalytic steps involved in the chemical mechanism as well as the overall reaction. Each entry includes the following steps:

- Enzyme name and EC number
- PDB code and CATH codes of all domains in the enzyme
- Diagram and annotation of the overall reaction
- Primary literature references

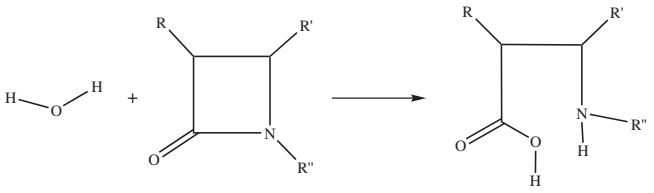

Fig. 1. The overall reaction for a β-lactamase.

- Diagram and annotation of all reaction steps, including:
  —The Ingold mechanism (Ingold, 1969)
  —Diagram and function of catalytic amino acid residues
  —Information on the reactive centres and bond changes
- Comments on the reaction (where applicable).

CONTENT

The criteria defined in the Design section initially produced a dataset of 100 entries. A single EC number may cover a plurality of MACiE entries when different mechanisms bring about the same overall chemical transformation, as with the two types of 3-dehydroquinate dehydratase, and thus 100 MACiE entries span only 96 EC numbers.

The 100 enzymes in Version 1 of MACiE incorporate domains from 140 CATH homologous superfamilies. MACiE currently covers 56 of the 174 EC sub-subclasses present in the PDB, thus, we feel that we have a representative coverage of EC reaction space (comparative EC wheels are available at URL http://www-mitchell.ch.cam.ac.uk/macie/ECCoverage/). We anticipate that all 158 sub-subclasses for which both structures and reliable mechanisms are available will be represented in the forthcoming MACiE Version 2.

SOFTWARE

The data are initially entered in MDL's ISIS/Base, a database package for chemical reactions, validated by at least two people, and then converted into CMLReact using the Jumbo Toolkit (Wakelin *et al*., 2005) to create an information and semantically rich database. At this stage we add extra fields of information to the CMLReact version of MACiE that are unavailable in the ISIS version, including the CATH code. Jumbo is a set of Java-based software which converts the MDL file format produced from ISIS/Base into CMLReact. The MacieConverter section of Jumbo performs the following functions:

- Integration of the files in the ISIS/Base version of MACiE
- Identification of reactant, product and spectator molecules
- Splitting of groups of molecules
- Automatic mapping of atoms within the reaction
- Checking for mass and charge conservation throughout the reaction (stoichiometry)
- Integration and checking of MACiE Dictionary entries.

Once the conversion process has been completed, a further tool in the Jumbo Toolkit, called CMLSnap (Holliday *et al*., 2004), can be used to create an animation of the reaction. This animation includes all of the atoms and bonds involved as well as the electron movements, which are calculated automatically. It is expected that CML will become our primary method of data entry and storage.

CURATION

The annotation process involves input and validation steps. Terms have been rigorously defined either from the IUPAC Gold Book (McNaught *et al*., 1997), such as chemical terms like hydrolysis, or from primary literature, such as mechanism, which is defined using Ingold's terminology (Ingold, 1969), originally put forward in the 1930s. All of the technical and scientific terms used in MACiE are contained in the MACiE dictionary, which is available at the URL http://www-mitchell.ch.cam.ac.uk/macie/glossary.html and is also available as a raw XML file.

The entries online are accessed via an HTML look-up table and include all of the information available in the database. The original ISIS/Base format file and the raw CML files can be supplied.

FUTURE WORK

Future work includes expanding the dataset to include a representative set of EC numbers (at the sub-subclass level), creating a search interface for MACiE and developing authoring tools for MACiE in CML. Ongoing research focuses on the evolution of enzyme catalysis and the classification of enzyme reaction mechanisms.

ACKNOWLEDGEMENTS

G.J.B. would like to thank Dr Jonathan Goodman for his invaluable help with organic chemistry queries. We would also like to thank the EPSRC (G.L.H. and J.B.O.M.), the BBSRC (G.J.B. and J.M.T.—CASE studentship in association with Roche Products Ltd; N.M.O.B. and J.B.O.M.—grant BB/C51320X/1), the Chilean Government's Ministerio de Planificación y Cooperación and Cambridge Overseas Trust (D.E.A.) for funding and Unilever for supporting the Centre for Molecular Science Informatics.

Conflict of Interest: none declared.

REFERENCES

Bartlett,G.J. *et al*. (2002) Analysis of catalytic residues in enzyme active sites. *J. Mol. Biol*., **324**, 105–121.

Berman,H.M. *et al*. (2000) The Protein Data Bank. *Nucleic Acids Res*., **28**, 235–242.

Holliday,G.L. *et al*. (2004) CMLSnap: animated reaction mechanisms. *Internet J. Chem*., **7**, Article 4.

Ingold,C.K. (1969) *Structure and Mechanism in Organic Chemistry*. 2nd edn, Cornell University Press, Ithaca, NY, Chapters 5–15.

Kanehisa,M. *et al*. (2004) The KEGG resource for deciphering the genome. *Nucleic Acids Res*., **32**, D277–D280.

McNaught,A.D. and Wilkinson,A. (1997) International Union of Pure and Applied Chemistry Compendium of Chemical Terminology (''The Gold Book''). 2nd edn, ISBN 0-8-654-26848.

Nagano,N. (2005) EzCatDB: the Enzyme Catalytic-mechanism Database. *Nucleic Acids Res*., **33**, D407–D412.

Orengo,C.A. *et al*. (1997) CATH—a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1108.

Pegg,S.C-H. *et al*. (2005) Representing structure-function relationships in mechanistically diverse enzyme superfamilies. *Pac. Symp. Biocomput*., 358–369.

Porter,C.T. *et al*. (2004) The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res*., **32**, D129–D133.

Schomburg,I. *et al*. (2004) BRENDA, the enzyme database: updates and major new developments. *Nucleic Acids Res*., **32**, D431–D433.

Wakelin,J. *et al*. (2005) CML tools and information flow in atomic scale simulations. *Mol. Simul*., **31**, 315–322.

BIOINFORMATICS

Vol. 21 no. 23 2005, page 4317
doi:10.1093/bioinformatics/bti734

Erratum

Shifting and scaling patterns from gene expression data

Jesús S. Aguilar-Ruiz

Bioinformatics **21**(20), 3840–3845

An error occurred in the text of the above article, in Section 4.

The mean of a row (condition) i is $\mu_{r_i} = {}_{\alpha i}\mu_\pi + \beta_i$, and the mean of a column (gene) j is given by $\mu_{c_j} = {}_{\pi j}\mu_\alpha + \mu_\beta$,

should have been represented as follows:

The mean of a row (condition) i is $\mu_{r_i} = \alpha_i\mu_\pi + \beta_i$, and the mean of a column (gene) j is given by $\mu_{c_j} = \pi_j\mu_\alpha + \mu_\beta$.

Also, the following text, in Theorem 1.

In the bicluster illustrated in Equation (11), $w_{ij} = \pi_j\alpha_i + \beta_i$, $w_{iJ} = \mu_{r_i = \alpha\mu\pi} + \beta_i$, $w_{Ij} = \mu_{c_j = \pi j}\mu_\alpha + \mu_\beta$ and $w_{IJ} = \mu_\pi\mu_\alpha + \mu_\beta$,

should have read as follows:

In the bicluster illustrated in Equation (11), $w_{ij} = \pi_j\alpha_i + \beta_i$, $w_{iJ} = \mu_{r_i} = \alpha_i\mu_\pi + \beta_i$, $w_{Ij} = \mu_{c_j} = \pi_j\mu_\alpha + \mu_\beta$ and $w_{IJ} = \mu_\pi\mu_\alpha + \mu_\beta$.

The publisher wishes to apologise for this error.

BIOINFORMATICS

Vol. 21 no. 23 2005, page 4318
doi:10.1093/bioinformatics/bti735

Erratum

Statistical evaluation of pairwise protein sequence comparison with the Bayesian bootstrap

Gavin A. Price, Gavin E. Crooks, Richard E. Green and Steven E. Brenner

Bioinformatics **21**(20), 3824–3831

An error occurred in the typesetting of the above article. These following equations were incorrectly represented:

	Standardbootstrap	Bayesianbootstrap
No normalization	$\frac{1}{n^2-n}$	$\frac{w_i w_j}{\left(\sum_{k=1}^{n} w_k\right)^2 - \sum_{k=1}^{n} (w_k)^2}$
Linear normalization	$\frac{1}{n(s-1)}$	$\frac{w_i w_j}{\left(\sum_{k=1}^{s}\right)\left(\left(\sum_{k=1}^{s} w_k\right) - w_i\right)}$
Quadratic normalization	$\frac{1}{(s^2-s)S}$	$\frac{w_i w_j}{\left(\left(\sum_{k=1}^{s}\sum_{k=1}^{s} w_k \cdot w_l\right) - \sum_{k=1}^{n} (w_k)^2\right)S}$,

They should have been displayed as follows:

	Standard bootstrap	Bayesian bootstrap
No normalization	$\frac{1}{n^2-n}$	$\frac{w_i w_j}{\left(\sum_{k=1}^{n} w_k\right)^2 - \sum_{k=1}^{n} (w_k)^2}$
Linear normalization	$\frac{1}{n(s-1)}$	$\frac{w_i w_j}{\left(\sum_{k=1}^{n} w_k\right)\left(\left(\sum_{k=1}^{s} w_k\right) - w_i\right)}$
Quadratic normalization	$\frac{1}{(s^2-s)S}$	$\frac{w_i w_j}{\left(\left(\sum_{k=1}^{s}\sum_{l=1}^{s} w_k \cdot w_l\right) - \sum_{k=1}^{s} (w_k)^2\right)S}$

The publisher wishes to apologise for this error.

Please note that high definition figures are available at *Bioinformatics* online.

