GI, the Gesellschaft für Informatik, publishes this series in order

- to make available to a broad public recent findings in informatics (i.e. computer science and information systems)
- to document conferences that are organized in cooperation with GI and
- to publish the annual GI Award dissertation.

Broken down into the fields of "Seminars", "Proceedings", "Monographs" and "Dissertation Award", current topics are dealt with from the fields of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure the high level of the contributions.

The volumes are published in German or English.

Information: http://www.gi-ev.de/LNI

ISSN 1617-5468
ISBN 978-3-88579-177-5

This volume contains papers presented at the German Conference on Bioinformatics, GCB 2006, held in Tübingen, September 19–22, 2006. GCB is an annual international conference providing a forum for the presentation of research results in Bioinformatics and Computational Biology. The meeting is organized on behalf of the "Fachgruppe Informatik in den Biowissenschaften" (BIOINF) of the German Society of Computer Science (GI), the "AG Computereinsatz in den Biowissenschaften" of the German Society of Chemical Technique and Biotechnology (DECHEMA) and the "Studiengruppe Bioinformatik" of the German Society for Biological Chemistry and Molecular Biology (GBM).

D. Huson, O. Kohlbacher, A. Lupas, K. Nieselt and A. Zell (eds.): GCB 2006

P-83

GI-Edition

Lecture Notes in Informatics

Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.)

German Conference on Bioinformatics

GCB 2006
September 19–22, 2006,
Tübingen, Germany

Proceedings

Daniel Huson, Oliver Kohlbacher, Andrei Lupas,
Kay Nieselt and Andreas Zell (eds.)

German Conference on Bioinformatics

GCB 2006

19.09.2006-22.09.2006
in Tübingen, Germany

Gesellschaft für Informatik 2006

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-83

ISBN 978-3-88579-177-5
ISSN 1617-5468

Volume Editors
Prof. Dr. Daniel H. Huson
Zentrum für Bioinformatik (ZBIT), Universität Tübingen, 72076 Tübingen, Germany
Email: huson@informatik.uni-tuebingen.de
Prof. Dr. Oliver Kohlbacher
Zentrum für Bioinformatik (ZBIT), Universität Tübingen, 72076 Tübingen, Germany
Email: kohlbacher@informatik.uni-tuebingen.de
Prof. Dr. Andrei Lupas
Max-Planck-Institut für Entwicklungsbiologie, 72076 Tübingen, Germany
Email: andrei.lupas@tuebingen.mpg.de
Dr. Kay Nieselt
Zentrum für Bioinformatik (ZBIT), Universität Tübingen, 72076 Tübingen, Germany
Email: nieselt@informatik.uni-tuebingen.de
Prof. Dr. Andreas Zell
Zentrum für Bioinformatik (ZBIT), Universität Tübingen, 72076 Tübingen, Germany
Email: zell@informatik.uni-tuebingen.de

printed by Köllen Druck+Verlag GmbH, Bonn

Contents

Preface

This volume contains papers presented at the German Conference on Bioinformatics, GCB 2006, held in Tübingen, September 19–22, 2006. This annual international conference provides a forum for the presentation of research results in Bioinformatics and Computational Biology. It is run on behalf of the Fachgruppe "Informatik in den Biowissenschaften (BIOINF)" of the German Society of Computer Science (GI), the AG "Computereinsatz in den Biowissenschaften" of the German Society of Chemical Technique and Biotechnology (DECHEMA), and the Studiengruppe "Bioinformatik" of the German Society for Biological Chemistry and Molecular Biology (GBM).

The conference opened on September 19th, 2006, with the following four tutorials: "Introduction to Phylogenetic Networks" given by Daniel Huson, "Kernel Methods for Predictive Sequence Analysis" given by Gunnar Rätsch and Cheng Soon Ong, "Mining the Biomedical Literature: State of the Art, Challenges and Evaluation Issues" given by Hagit Shatkay, and "Non-coding RNA - No Longer the Dark Matter in a Cellular Universe", given by Yu Wang.

Six leading scientists were invited to give keynote lectures. Gene Myers (HHMI, Janelia Farms) spoke on "Imaging-Based Systems Biology" and Detlef Weigel (MPI, Tübingen) on "Genome-wide Analysis of Sequence Variation in Arabidopsis". A thematic focus was placed on Protein-Protein Interactions and this was reflected by the topics of the other invited speakers: David Jones (University College, London) spoke on "Docking protein domains using a contact map representation", Cedric Notredame (CNRS IGS, Marseille) spoke on "Combining Sequence Information with T-Coffee", Rob Russell (EMBL, Heidelberg) spoke on "Pushing Details into Interaction Networks", and Mukund Thattai (NCBS, Bangalore) spoke on "Encoding evolvability: The hierarchical language of polyketide synthase protein interactions".

The technical program additionally contained seven short papers and 12 long papers, which were refereed and selected from 62 submissions by the program committee. Additionally, over 100 poster abstracts were accepted for presentation at the poster sessions. This volume contains all long papers and abstracts of the invited lectures. The poster abstracts appear in a special abstract book together with the short papers. The conference was concluded with a special session on "Bioinformatics in Germany – State of the Art", in which the five DFG-funded Bioinformatics Centers at Bielefeld, Leipzig, Munich, Saarbrücken and Tübingen reported on their work.

Thanks to the members of the program committee and their colleagues who gave their time to referee the submissions, and to all that helped locally to organize the meeting. We are also grateful to all presenters and participants, whose contributions and interactions made GCB 2006 a success.

Tübingen, August 2006

Daniel Huson
Oliver Kohlbacher (Local Chair, Program Co-Chair)
Andrei Lupas (Program Co-Chair)
Kay Nieselt
Andreas Zell

Organizing Committee

Daniel Huson
Oliver Kohlbacher
Andrei Lupas
Kay Nieselt
Andreas Zell

Program Committee

Janusz Bujnicki, IIMCB Warsaw, Poland
Arne Elofsson, Stockholm Bioinformatics Center, Sweden
Robert Giegerich, Bielefeld University
Daniel Huson, Tübingen University
Sorin Istrail, Brown University, Providence, RI, USA
Oliver Kohlbacher, Tübingen University (chair)
Kristin Koretke, GlaxoSmithKline, Collegeville, PA, USA
Hans-Peter Lenhof, Saarland University, Saarbücken
Andrei Lupas, MPI for Developmental Biology, Tübingen (chair)
Hans-Werner Mewes, GSF, TU Munich
Kay Nieselt, Tbingen University
Matthias Rarey, University of Hamburg
Knut Reinert, Free University of Berlin
Dietmar Schomburg, University of Cologne
Joachim Selbig, Potsdam University
Peter Stadler, University of Leipzig
Martin Vingron, MPI for Molecular Genetics, Berlin
Edgar Wingender, University of Göttingen
Andreas Zell, Tübingen University
Ralf Zimmer, LMU Munich

Additional Referees

Peter Arndt	Iwona A. Cymerman	Tim Conrad	Tobias Dezulian
Alexander Diemand	Janko Dietzsch	Andreas Döring	Markus Gruber
Clemens Gröpl	Andreas Hildebrandt	Andreas Keller	Jan Kosinski
Jan Küntzer	Hannes Luz	Philipp Messer	Dirk Neumann
Michal Pietal	Hannes Planatscher	Christian Rausch	Dirk Repsilber
Alexander Rurainski	Alexander Schliep	Marcel Schulz	Johannes Söding
Nora Speer	Christian Spieth	Stephan Steigele	Matthias Steinfath
Christine Steinhoff	Jochen Supper		

Sponsors of GCB 2006

Scientific societies


Non-profit sponsors

Commercial sponsors

Shape Distributions and Protein Similarity*

Stefan Canzar [1] and Jan Remy [2]

[1] Université Henri Poincaré
LORIA, B.P. 239
54506 Vandœuvre-lès-Nancy, France
canzar@loria.fr

[2] Institut für Theoretische Informatik
ETH Zürich
CH-8092 Zürich
jremy@inf.ethz.ch

Abstract: In this paper we describe a similarity model that provides the objective basis for clustering proteins of similar structure. More specifically, we consider the following variant of the protein-protein similarity problem: We want to find proteins in a large database $\mathscr{D}$ that are very similar to a given query protein in terms of geometric shape. We give experimental evidence, that the shape similarity model of Osada, Funkhouser, Chazelle and Dobkin [OFCD02] can be transferred to the context of protein structure comparison. This model is very simple and leads to algorithms that have attractive space requirements and running times. For example, it took 0.39 seconds to retrieve the eight members of the seryl family out of $26,600$ domains. Furthermore, a very high agreement with one of the most popular classification schemes proved the significance of our simplified representation of complex proteins structure by a distribution of C_α-C_α distances.

1 Introduction

Understanding the rapidly increasing number of protein three-dimensional structure data deposited in the Brookhaven Protein Data Bank (PDB) [BWF+00] poses a major challenge in the post-genome-sequence era. One reliable method to assign function to gene products that have no experimentally inferable molecular (biophysical or biochemical) function is on the basis of sequence similarity to proteins of known function. Since structure is evolutionary better conserved than sequence, the structural similarity to one or more proteins of known structures infers an even more powerful clue to the structure-function relationship. Clearly, the classifi cation of recurrent protein folds constitutes a major step towards the understanding of protein structure.

*This paper includes work done while the authors were at Technische Universit¨at M¨unchen, Institut f¨ur Informatik. Research was partially supported by the DFG project KN 309/1-1 “Information Mining”.

The placement in categories must be done according to a similarity criterion or distance (metric) that reflects the degree of shape affinity for pairs of proteins. The most popular classification systems either use a totally automated approach (FSSP) [HS97], classify manually (SCOP) [MBHC95] or are based on a combination of both (CATH) [OMJ+97]. The three-dimensional structures are usually compared by structural alignment algorithms such as CE [SB98], DALI [HS93], and VAST [MGB95], which is, mainly because of its intrinsic complexity, a time-consuming task.

Problem Statement. We consider a special variant of the molecular similarity problem. Let $\mathscr{D}$ be a database containing a collection of proteins. We want to find the proteins in $\mathscr{D}$ that are similar to a given query protein Q. There is no common definition of what "similarity of proteins" exactly means. As motivated above, we restrict ourselves to the similarity of three-dimensional structure. This kind of similarity is very "human oriented", since two objects - or in our case proteins - are usually said to be similar if a human observer thinks that they are. Thus, we have two criteria for performance: *i)* if $Q \in \mathscr{D}$ then Q should be recognized as the most similar and *ii)* molecules rated as very similar to Q should be also recognized by a human as being very similar. Note that the second criterion does not include the first. If the shape of Q is not very characteristic, it could be difficult for a human to recognize an identical structure. Since the database $\mathscr{D}$ contains usually thousands of proteins (the PDB contains currently $32,823$ structures) it is important that the comparison of a single pair of proteins is very fast. This usually requires some preprocessing of the database. It is desirable that the data structures produced during preprocessing have modest space consumptions.

Related Work. Geometric approaches to measure the similarity of proteins were extensively studied in various aspects. In order to give a representative selection, we like to mention geometric hashing [Wol90, NW91, FNW92, FNNW93, NLWN95], footprinting [BS97, BS99] and correlation techniques [KKSE+92, GJS97]. None of these algorithms has a running time that allows fast queries to a large database. Methods that do not depend on a structural alignment are based on graph theory [HPM+02], local feature profiles of C_α distance matrices [CKK04], C_α-C_α distances [CP02] or secondary structure matching [KH04]. Special algorithms for similarity search in protein database were considered by Kriegl and Seidl [KS98] and Ankerst, Kastenmüller, Kriegl and Seidl [AKKS99]. The first approach is based on parametric approximation of surface segments. In the second paper, proteins are described by density histograms that are robust under rotation.

Our Results. The concept of shape distributions was introduced by Osada, Funkhouser, Chazelle and Dobkin[OFCD02]. They evaluated their approach by comparing simple objects like cars, humans, phones or mugs. We have successfully transferred their similarity model to the protein similarity context. The main purpose of our work is to evaluate whether shape distributions are suitable means to compare the tree-dimensional structure of proteins or molecules. Our experiments give evidence that the performance criteria mentioned above are satisfied: The protein in the database with the most similar shape distribution was always the query protein itself. Furthermore, top ranked proteins could be observed to be structural similar to the query protein. The ability to distinguish CATH homologous superfamilies with a success rate of 98% confirmed this subjective evaluation.

We claim that this algorithm has some advantages compared to previous methods. First, the comparison step is fast enough for database search, since we are able to make around $100,000$ comparisons per second. Second, the algorithm is much more simple than most of the other approaches. Third, the space requirement of the data structure we generate in a preprocessing step is only linear in the number of proteins contained in the database. And fourth, our approach does not depend on any knowledge-based decisions, like the assignment of secondary-structure elements.

The remainder of the paper is organized as follows. In section 2 we review the concept of shape distributions. In section 3 we introduce the algorithm for similarity search. Finally, section 4 presents experimental results.

2 Shape Distributions

Osada, Funkhouser, Chazelle and Dobkin [OFCD02] introduced a simple model for shape similarity of objects. Let $\mathcal{S}$ be a set of points on the surface of an object. A *shape function* $\xi(\mathcal{S})$ measures a geometric property that depends on $\mathcal{S}$. A typical example for a shape function is the Euclidean distance $d(a, b)$ for $\mathcal{S} = \{a, b\}$. Other types of shape functions include angles, areas or volumes.

If $\mathcal{S}$ is chosen at random from all points on the surface of the object, then $\xi(\mathcal{S})$ is a random variable having some distribution $F(\xi(\mathcal{S}))$. Osada et. al. claim that this distribution, the *shape distribution*, is very characteristic for the shape of the object. Thus the shape matching problem can be reduced to the comparison of two probability distributions. The algorithmic side of shape distributions is very simple. For the sake of exposition, we assume that our shape function is the Euclidean distance of two points. As mentioned above, the distance of two random (surface) points is a random variable. The distribution of distances is reconstructed by choosing N pairs of surface points at random. Of course, for technical reasons, the distribution must be discretized into, say B many intervals. In essence, by counting the number of distances that fall into each interval, we obtain a histogram that consists of B bins that expresses the "probability" for a distance being within some interval. The similarity (or dissimilarity) of two objects can be computed by comparing their shape distribution, i.e., the histograms under an arbitrary metric. The most natural example is the Minkowski norm $\mathscr{L}_N$.

3 The Algorithm

In this section we give an overview of the algorithm. The input is a set $\mathscr{D}$ of 3D protein structures. The atomic coordinates are taken from the Brookhaven Protein Data Bank (PDB) [BWF$^+$00]. In our experiments we varied the definition of the point set $\mathcal{S}$ (cf. Section 2) to contain either all atoms, exclusively atoms located on the molecular surface or all C_α atoms. We have chosen the Euclidean distance as a shape function $\xi(\mathcal{S})$, since it seems to provide the best results.

Preprocessing The preprocessing is identical for each protein in $\mathscr{D}$ and only depends on the definition of $\mathcal{S}$. First we extract the coordinates of points in $\mathcal{S}$, which is a trivial step

in the case of $\mathcal{S}$ being equal to the set of all C_α atoms. To derive the shape distribution from the surface of the protein we determine the atoms that can be touched by a solvent molecule of fi xed size (e.g. 1.4Å). This can be done with an algorithm of Sanner, Olsen and Spehner [SOS96] in $\mathcal{O}(n \log n)$ time. Simply speaking, this algorithm computes the surface atoms as an intermediate result. Second we calculate the distances of each pair of atoms in $\mathcal{S}$. This yields a histogram with B bins each counting the number of occurrences of certain distances. By a normalization of the resulting shape distribution one could simply add an invariance under scaling, e.g. consider the shape of proteins independent of their size. Second we store the (not normalized) histogram as a sequence of B integers. The preprocessing of a protein with n atoms requires optionally time $\mathcal{O}(n \log n)$ for the computation of the surface plus time $\mathcal{O}\left(n^2\right)$ for the approximation of the shape distribution. The overall complexity can be reduced to $\mathcal{O}(n \log n)$ if we consider only $\mathcal{O}(n \log n)$ random pairs in $\mathcal{S}$ for the computation of the shape distribution.

Similarity Query Let Q denote the query protein. We compute the similarity measure between Q and each structure in $\mathscr{D}$ by comparing their shape distributions. We experimented with similarity measures based on the Minkowski $\mathscr{L}_N$ norms for $N = 1, 2, 10$.

It remains to discuss the complexity of the similarity query. The distance of two distributions f and g in the Minkowski norm is given by

$$D(f, g) = \left(\sum_{i=1}^{B} |f_i - g_i|^N \right)^{1/N} \tag{1}$$

In fact, this value is the distance of two points, f and g in the $\mathbb{R}^B$ under the $\mathscr{L}_N$ metric. Furthermore, the histograms of the proteins in $\mathscr{D}$ may be modeled as a set of points in a high dimensional space with coordinates determined by the approximations of the shape distributions. Also, the shape distribution of the query protein defi nes a point in the $\mathbb{R}^B$. Hence the similarity problem for proteins can be transformed into proximity problem among a set of points. This transformation is very helpful, as there are algorithms for proximity problems that have desirable asymptotics.

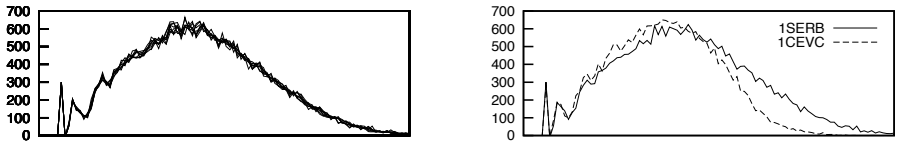
We want to query the database for the most similar proteins, i.e., proteins with scores that are lesser than a given threshold. So we have to solve a proximity problem which is known as *range searching*. There are data structures that provide fast queries for orthogonal search regions in spaces, provided the dimension is small. In our case, the search region is circular and $d = B$ is usually very large. Unfortunately there are no fast data structures for circular queries in high dimensional spaces. However, Arya and Mount [AMN$^+$94] proposed a data structure that allows queries with circular ranges if one is willing to accept some approximation. More precisely their data structure ensure that the following is true for all $\varepsilon > 0$. Let t denote the given threshold, i.e., diameter of the query range. Then points lying within distance $\varepsilon \cdot t$ around the boundary of the query range either may or may not be included in the output of the query. The running time of such a query is $\mathcal{O}\left((1/\varepsilon)^d + \log m\right)$ and it is also good in practice as Arya and Mount claim in their paper.

4 Experimental Results

We have implemented the algorithm described in section 3 in C++. The experiments were done on a system with a $1,60$ GHz Pentium M-Processor. The three-dimensional coordinate data was taken from the Brookhaven Protein Data Bank (PDB) [BWF$^+$00] and was dissected into domains according to CATH version v2.0. The resulting collection $\mathscr{D}$ of protein structures (about $26,600$ CATH domains) was preprocessed into shape distributions and finally stored on disk.

In contrast to [OFCD02], both the restriction to atoms on the molecular surface and the random sampling of $\mathcal{S}$ means a loss in characteristics of shape distribution for the complex structure of proteins. In contrast, the difference in classification accuracy depending on whether $\mathcal{S}$ contained all atoms or only the subset of C_α atoms was marginal. To shorten computation time we thus focused on the latter case which we will discuss now.

It turned out during the experiments, that using $B = 60$ bins for the representation of the shape distribution and the Minkowski $\mathscr{L}_2$-norm for measuring the dissimilarity between pairs of distributions is a good choice.



a) b)

Figure 1: a) The superposed shape distributions of the eight seryl family members. b) With respect to query protein 1SERB domain 1CEVC ranked on position 125. Their distributions can be distinguished visually.

4.1 Basic Similarity Search

In order to demonstrate the general applicability of shape distributions to the characteristic representation of the three-dimensional structure of proteins, we report on experiments on a group of molecules that are known to be related. We tried to retrieve the eight members of the *seryl-tRNA synthetase* family (1SERA, 1SERB, 1SESA, 1SESB, 1SRYA, 1SRYB, 1SETA, 1SETB) out of roughly $26,600$ domains contained in our database.

If the query molecule is 1SERB we obtained a ranking as depicted in Figure 2. The eight members of the seryl family rank on the top eight positions, followed by roughly $26,600$ molecules. This ranking is conform with the shape of the molecules. Furthermore, the shape distributions of the seryl family members are clearly distinguishable from those derived from higher ranked domains (Fig. 1). This kind of query could be the first step when searching for structural homologs of a given protein Q. Screening the whole PDB

by using shape distributions could result in a small number of structural homologs of Q (for example by range searching, as mentioned in section 3), which are further analyzed by rigid-body superposition (e.g. May and Johnson [MJ95]) to find the best possible alignment.

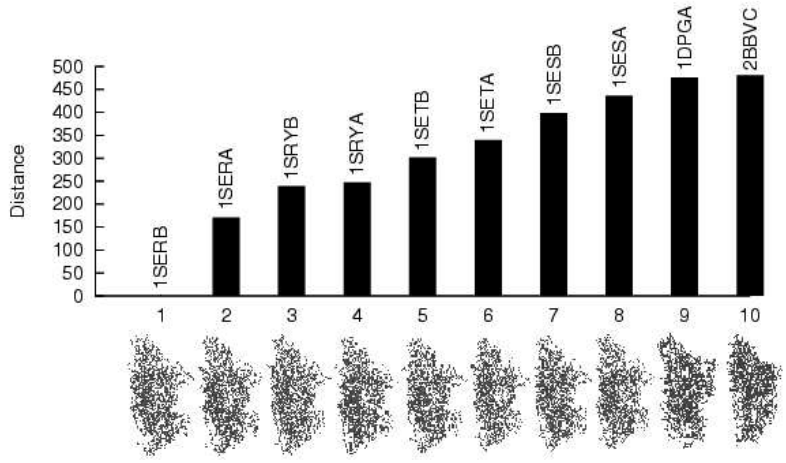


Figure 2: Similarity scores of the most similar molecules to 1SERB. The eight members of the seryl-tRNA-synthetase family rank on the top eight positions among $26,600$ domains. The first non-seryl protein 1DPGA is classified by CATH to fall into the same class.

4.2 Classification by Structural Similarity

The placement of protein structures in categories heavily depends on the nature of the underlying similarity model. In order to investigate whether the transformation of protein structures into points in B-dimensional Euclidean space $\mathbb{R}^B$ has a negative impact on the accuracy of classification, we performed an all-against-all comparison according to our distance measures on one of the most popular classification schemes, the CATH database [OMJ+97] ($353,766,700$ structural comparisons). CATH, as a hierarchical classification scheme, clusters protein structures in the PDB at four major levels, Class (C), Architecture (A), Topology (T) and Homologous superfamily (H). Based on our symmetric distance matrix (metric property of our distance measure) we determined the nearest neighbor N for every molecule in the database $\mathscr{D}$, ignoring the query structure Q itself, for which $d(Q,Q) = 0$ holds for all $Q \in \mathscr{D}$. When asking whether N and Q fall into the same CATH category on level l, $l = 1,2,\ldots,7$, we considered all those domains, that were labeled identically by CATH on levels $1,\ldots,l-1$.

From domains sharing the first six CATH labels, C, A, T, H, S, and N, 71% have been assigned the correct label on level seven (I) (cf. Table 1). Ascending the hierarchy, this value increases up to 98% at H-level, where the last three labels were allowed to vary. We attach great importance to the high categorization accuracy particularly at this level,

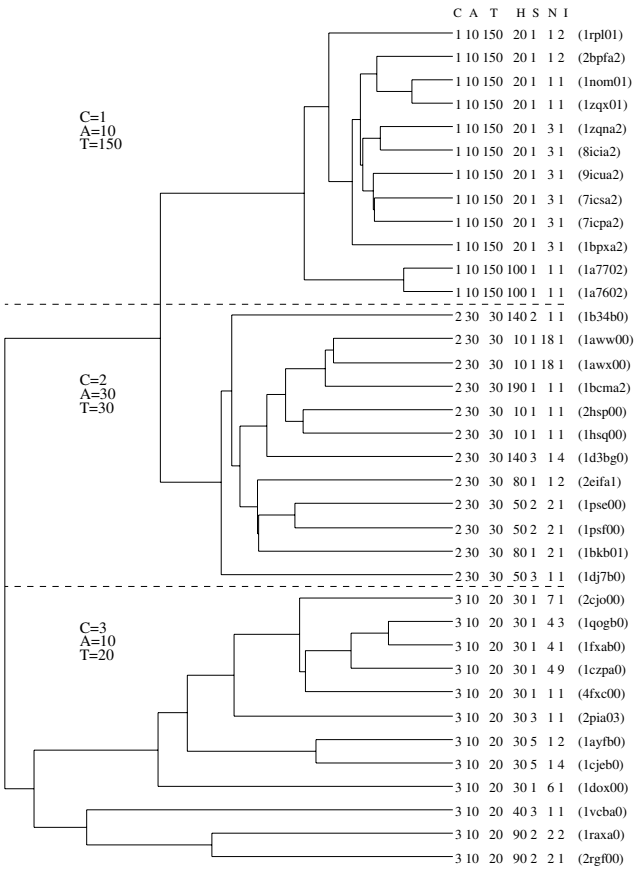


Figure 3: Cluster-analysis dendogram of randomly selected CATH domains. The shape distributions of the protein domain structures have been clustered by an agglomerative hierarchical algorithm using the single linkage similarity criterion.

as homologous superfamilies cluster proteins with highly similar structures and functions. Furthermore, we think that distinguishing different architectures with a success rate of 97% is a remarkable result, as label assignment at A-level is based on the human eye.

These features of our similarity measure are further illustrated by the cluster-analysis dendogram shown in Figure 3. We randomly selected 36 domains from three different nodes on the T-level of the CATH hierarchy (12 domains from each node), where the first node can be described by labels C=2, A=30 and T=30, the second node by C=3, A=10 and T=20 and the third node by C=1, A=10 and T=150. Not only that there was a clear discrimination between these three groups, but one can also associate lower CATH levels with subclusters in the clustering tree. For example, removing the longest edge from the minimum spanning tree of a graph, whose vertices correspond to protein domain structures from the third group (C=2, A=30, T=150) and whose edges are weighted with the distances based on our

CATH LABEL	CATH LEVEL	NEAREST NEIGHBOR AGREEMENT (%)
C	Class	97.1
A	Architecture	97.2
T	Topology	96.0
H	Homologous superfamily	98.0
S	Sequence families	96.8
N	Nearly-identical representatives	91.9
I	Identical representatives	71.5

Table 1: Nearest neighbor classification for CATH categories based on our similarity score. An agreement of $x\%$ on level l describes, that $x\%$ of domains sharing the first $l-1$ CATH labels have been assigned the correct label on level l.

similarity measure, results in two clusters, one containing the domains labeled H=20 and one with domains labeled H=100. Similarly this holds for domains sharing labels C=1, A=10, T=150, H=20, and S=1 and differing in N=1 or N=3.

4.3 Running Time

Since the shape distributions can be computed in a preprocessing step, we can perform the queries to the database very fast. In section 3 we have mentioned that the query time is roughly $\mathcal{O}(\log m)$ if we assume that ε and B are constant. In practice the constants are too large – at least for "small" databases. Nevertheless, the query time is still attractive. In our implementation, which was not optimized for speed, a query to a database of size $m \approx 26,600$ took only 0.39 seconds, ignoring the time spent on input/output operations. The computation of an all-against-all distance matrix ($353,766,700$ comparisons) was finished after less than an hour.

5 Concluding Remarks

We have given experimental evidence that the distribution of distances between C_α atoms provides a significant signature for the three-dimensional structure of proteins. By transferring the similarity model of Osada, Funkhouser, Chazelle and Dobkin [OFCD02] to the context of protein fold comparison, we were able to retrieve the eight members of the seryl family among $26,600$ domains in 0.39 seconds of CPU time. But despite the simplified representation of protein structure, our approach exhibits a classification accuracy of 98% for CATH homologous superfamilies.

Several alternative methods based on a simplified representation of protein structure have been proposed recently. The one of Carugo and Pongor [CP02] considers C_α-C_α distances between residues separated by a variable number of amino acid residues and is thus conceptually related to our approach. Nevertheless, they represent each molecule by a set of

28 histograms that have to be compared by a contingency table analysis. As a consequence, the comparison of a pair of proteins is more expensive both in terms of computation time and space consumption. The similarity score of Choi, Kwon and Kim [CKK04] is based on profiles of representative local features (LFF) of C_α distance matrices. Compared to shape distributions, LFF profiles necessitate an considerable preprocessing step and yield an agreement with CATH categories that ranges from 53.3% on Homology level to 70% on Class level.

In short, no other approach combines comparable high classification accuracy with approximate efficiency both in terms of time and space, while at the same time being independent of any sequence information or human input. These features allow for a quick categorization of recently determined structures by scanning large databases like the PDB and thus help to keep our ordering of the protein fold space always up to date, as opposed to knowledge-based schemes like SCOP and CATH.

References

[AKKS99] Mihael Ankerst, Gabi Kastenm¨uller, Hans-Peter Kriegel, and Thomas Seidl. Nearest Neighbor Classification in 3D Protein Databases. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 34–43, 1999.

[AMN+94] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1994.

[BS97] Gill Barequet and Micha Sharir. Partial Surface and Volume Matching in Three Dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):929–948, September 1997.

[BS99] Gill Barequet and Micha Sharir. Partial Surface Matching by Using Directed Footprints. *Computational Geometry: Theory and Applications*, 12(1–2):45–62, February 1999.

[BWF+00] H.M. Berman, J. Westbrook, Z. Feng, Gilliland G., T.N. Bhat, Weissig H., I.N Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[CKK04] In-Geol Choi, Jaimyoung Kwon, and Sung-Hou Kim. Local feature frequency profile: A method to measure structural similarity in proteins. In *Proceedings of the National Academy of Sciences*, volume 101, pages 3797–3802, March 2004.

[CP02] O. Carugo and S. Pongor. Protein Fold Similarity Estimated by a Probabilistic Approach Based on C^α-C^α Distance Comparison. *Journal of Molecular Biology*, 315(4):887–898, January 2002.

[FNNW93] Daniel Fischer, Raquel Norel, Ruth Nussinov, and Haim J. Wolfson. 3-D Docking of Protein Molecules. In *Proceedings of the 4th Annual Symposium on Combinatorial Pattern Matching*, Lecture Notes in Computer Science (684), pages 20–34. Springer, June 1993.

[FNW92] Daniel Fischer, Ruth Nussinov, and Haim J. Wolfson. 3-D Substructure Matching in Protein Molecules. In *Proceedings of the 3rd Annual Symposium on Combinatorial Pattern Matching*, Lecture Notes in Computer Science (644), pages 136–150. Springer, April/May 1992.

[GJS97] Henry A. Gabb, Richard M. Jackson, and Michael J.E. Sternberg. Modelling Protein Docking using Shape Complementary, Elektrostatics and Biochemical Information. *Journal of Molecular Biology*, 272(1):106–120, September 1997.

[HPM+02] A. Harrison, F. Pearl, R. Mott, J. Thornton, and C. Orengo. Quantifying the similarities within fold space. *Journal of Molecular Biology*, 323(5):909–26, November 2002.

[HS93] L. Holm and C. Sander. Protein-structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233:123–138, September 1993.

[HS97] L. Holm and C. Sander. DALI/FSSP classification of three-dimensional protein folds. *Nucleic Acids Research*, 25(1):231–234, January 1997.

[KH04] E. Krissinel and K. Henrick. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D*, 60(12):2256–2268, Dec 2004.

[KKSE+92] Ephraim Katchalski-Katzir, Isaac Shariv, Miriam Eisenstein, Asher A. Friesem, Claude Aflalo, and Ilya A. Vakser. Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. In *Proceedings of the National Academy of Sciences*, volume 89, pages 2195–2199, March 1992.

[KS98] Hans-Peter Kriegel and Thomas Seidel. Approximation-Based Similarity Search for 3-D Surface Segments. *GeoInformatica Journal*, 2(2):113–147, 1998.

[MBHC95] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

[MGB95] T. Madej, J. F. Gibrat, and S. H. Bryant. Threading a database of protein cores. *Proteins: Structure, Function and Genetics*, 23:356–369, 1995.

[MJ95] A.C. May and M.S. Johnson. Improved genetic algorithm-based protein structure comparison: pairwise and multiple superpositions. *Protein Engineering*, 8(9):873–82, Sep 1995.

[NLWN95] Raquel Norel, Shuo L. Lin, Haim J. Wolfson, and Ruth Nussinov. Molecular Surface Complementary at Protein-Protein Interfaces: The Critical Role Played by Surface Normals at Well Placed, Sparse, Points in Docking. *Journal of Molecular Biology*, 252(2):263–273, 1995.

[NW91] Ruth Nussinov and Haim J. Wolfson. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. In *Proceedings of the National Academy of Sciences*, volume 88, pages 10495–10499, 1991.

[OFCD02] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape Distributions. *ACM Transaction on Graphics*, 21(4):807–832, October 2002.

[OMJ+97] C.A. Orengo, A.D. Michie, S. Jones, D.T. Jones, M.B. Swindells, and Thornton J.M. CATH - a hierarchic classification of protein domains structures. *Structure*, 5(8):1093–108, August 1997.

[SB98] I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 12(9):739–747, 1998.

[SOS96] Michel F. Sanner, Arthur J. Olson, and Jean-Claude Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, March 1996.

[Wol90] Haim J. Wolfson. Model-Based Object Recognition by Geometric Hashing. In *Proceedings of the 1st European Conference on Computer Vision*, Lecture Notes in Computer Science (427), pages 526–536. Springer, April 1990.

Microarray Layout as Quadratic Assignment Problem

Sérgio A. de Carvalho Jr.[1,2,3] and Sven Rahmann[2,3]

[1] Graduiertenkolleg Bioinformatik,
[2] International NRW Graduate School in Bioinformatics and Genome Research,
[3] Algorithms and Statistics for Systems Biology group, Genome Informatics,
Technische Fakultät, Bielefeld University, D-33594 Bielefeld, Germany.

Abstract: The production of commercial DNA microarrays is based on a light-directed chemical synthesis driven by a set of masks or micromirror arrays. Because of the natural properties of light and the ever shrinking feature sizes, the arrangement of the probes on the chip and the order in which their nucleotides are synthesized play an important role on the quality of the final product. We propose a new model called *conflict index* for evaluating microarray layouts, and we show that the probe placement problem is an instance of the *quadratic assignment problem* (QAP), which opens up the way for using QAP heuristics. We use an existing heuristic called GRASP to design the layout of small artificial chips with promising results. We compare this approach with the best known algorithm and describe how it can be combined with other existing algorithms to design the latest million-probe microarrays.

1 Introduction

An oligonucleotide microarray is a piece of glass or plastic on which single-stranded fragments of DNA, called *probes*, are affixed or synthesized. The chips produced by Affymetrix, for instance, can contain more than one million spots (or *features*) as small as 11 μm, with each spot accommodating several million copies of a probe. Probes are typically 25 nucleotides long and are synthesized in parallel, on the chip, in a series of repetitive steps. Each step appends the same nucleotide to probes of selected regions of the chip. Selection occurs by exposure to light with the help of a photolithographic mask that allows or obstructs the passage of light accordingly [3].

Formally, we have a set of probes $\mathcal{P} = \{p_1, p_2, ...p_n\}$ that are produced by a series of masks $\mathcal{M} = (m_1, m_2, ...m_T)$, where each mask m_t induces the addition of a particular nucleotide $\mathcal{S}_t \in \{A, C, G, T\}$ to a subset of $\mathcal{P}$. The *nucleotide deposition sequence* $\mathcal{S} = \mathcal{S}_1\mathcal{S}_2 \ldots \mathcal{S}_T$ corresponding to the sequence of nucleotides added at each masking step is therefore a supersequence of all $p \in \mathcal{P}$ [10].

In general, a probe can be *embedded* within $\mathcal{S}$ in several ways. An embedding of p_k is a T-tuple $\varepsilon_k = (e_{k,1}, e_{k,2}, ...e_{k,T})$ in which $e_{k,t} = 1$ if probe p_k receives nucleotide $\mathcal{S}_t$ (at step t), or 0 otherwise (Figure 1). The deposition sequence is often taken as a repeated permutation of the alphabet, mainly because of its regular structure and because such sequences maximize the number of distinct subsequences.

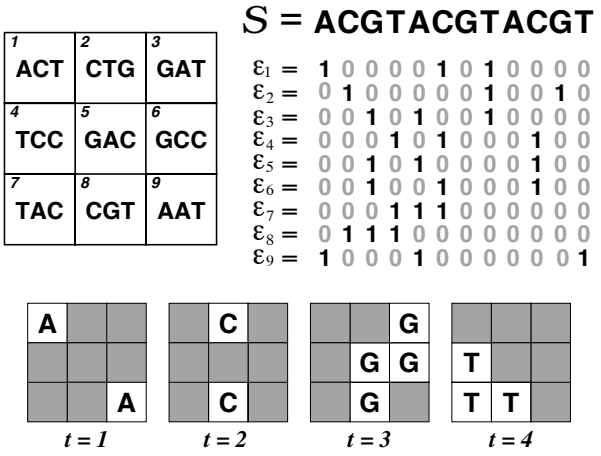


Figure 1: Synthesis of a hypothetical 3×3 chip. Top left: chip layout and the 3 nt probe sequences. Top right: deposition sequence and probe embeddings. Bottom: first four resulting masks.

We distinguish between *synchronous* and *asynchronous* embeddings. In the first case, each probe has exactly one nucleotide synthesized in every cycle of the deposition sequence; hence, 25 cycles or 100 steps are needed to synthesize probes of length 25. In the case of asynchronous embeddings, probes can have any number of nucleotides synthesized in any given cycle, allowing shorter deposition sequences. All Affymetrix chips that we know of can be asynchronously synthesized in 74 steps (18.5 cycles), which is probably due to careful probe selection.

Because of diffraction of light or internal reflection, untargeted spots can be accidentally activated in a certain masking step, producing unpredicted probes that can compromise the results of an experiment. This problem is more likely to occur near the borders between masked and unmasked spots [3]; this observation has given rise to the term *border conflict*.

We are interested in finding an *arrangement* of the probes on the chip together with *embeddings* in such a way that the chances of unintended illumination during mask exposure steps are minimized. The problem appears to be hard because of the exponential number of possible arrangements, although we are not aware of an NP-hardness proof (and our QAP formulation has several special properties). Optimal solutions are thus unlikely to be found even for small chips and even if we assume that all probes have a single predefined embedding.

If we consider all possible embeddings (up to several million for a typical Affymetrix probe), the problem is even harder. For this reason, the problem has been traditionally tackled in two phases. First, an initial embedding of the probes is fixed and an arrangement of these embeddings on the chip with minimum border conflicts is sought. This is usually referred to as the *placement*. Second, a *post-placement* optimization phase re-embeds the probes considering their location on the chip, in such a way that the conflicts with the neighboring spots are further reduced.

In the next section, we review the Border Length Minimization Problem [4], and define an extended model for evaluating microarray layouts. In Section 3, we briefly review existing placement strategies. In Section 4, we present a new formulation of the microarray placement problem based on the quadratic assignment problem (QAP). The results of using a QAP heuristic algorithm, called GRASP, to design small artificial chips are presented in Section 5, where we compare its performance with the best known placement algorithm and discuss how this approach can be used to design and improve larger microarrays.

2 Modeling

Border length. Hannenhalli and co-workers [4] were the first to give a formal definition of the problem of unintended illumination in the production of microarrays. They formulated the *Border Length Minimization Problem*, which aims at finding an arrangement of the probes together with their embeddings in such a way that the number of border conflicts during mask exposure steps is minimal.

The *border length* $\mathcal{B}_t$ of mask m_t is defined as the number of borders shared by masked and unmasked spots at masking step t. The total border length of a given arrangement is the sum of border lengths over all masks. For example, the initial four masks shown in Figure 1 have $\mathcal{B}_1 = 4$, $\mathcal{B}_2 = 6$, $\mathcal{B}_3 = 6$ and $\mathcal{B}_4 = 4$. The total border length of that arrangement is 50 (masks 5 to 12 not shown).

Conflict Index. The border length of an individual mask measures the quality of that mask. We are more interested in estimating the risk of synthesizing a faulty probe at a given spot, that is, we need a per-probe measure instead of a per-mask measure. Additionally, the definition of border length does not take into account two important practical considerations [6]: a) stray light might activate not only adjacent neighbors but also probes that lie as far as three cells away from the targeted spot; b) imperfections produced in the middle of a probe are more harmful than in its extremities.

This motivates the following definition of the *conflict index* $\mathcal{C}(p)$ of a probe p of length ℓ_p that is synthesized in T masking steps. First we define a distance-dependent weighting function, $\delta(p, p', t)$, that accounts for observation a) above:

$$\delta(p, p', t) := \begin{cases} (d(p, p'))^{-2} & \text{if } p' \text{ is unmasked at step } t, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $d(p, p')$ is the Euclidean distance between the spots of p and p'. This form of weighting function is the same as suggested in [6]. Note that δ is a "closeness" measure between p and p' only if p' is not masked (and thus creates the potential of illumination at p). To limit the number of neighbors that need to be considered, we restrict the support of $\delta(p, p', \cdot)$ to those $p' \neq p$ that are in a 7×7 grid centered around p (see Figure 2 left).

We also define position-dependent weights to account for observation b):

$$\omega(p, t) := \begin{cases} c \cdot \exp\left(\theta \cdot \lambda(p, t)\right) & \text{if } p \text{ is masked at step } t, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

0.06	0.08	0.10	0.11	0.10	0.08	0.06
0.08	0.13	0.20	0.25	0.20	0.13	0.08
0.10	0.20	0.50	1.00	0.50	0.20	0.10
0.11	0.25	1.00	p	1.00	0.25	0.11
0.10	0.20	0.50	1.00	0.50	0.20	0.10
0.08	0.13	0.20	0.25	0.20	0.13	0.08
0.06	0.08	0.10	0.11	0.10	0.08	0.06

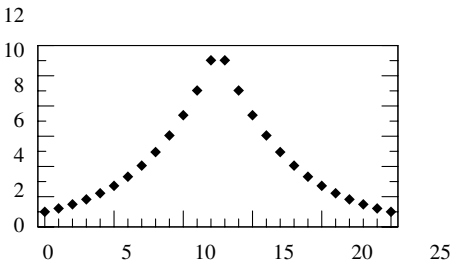


Figure 2: Ranges of values for both δ and ω on a typical Affymetrix chip where probes of length 25 are synthesized in 74 masking steps. Left: approximate values of the distance-dependent weighting function $\delta(p, p', t)$ for a spot containing probe p (shown in the center) and close neighbors p', assuming that p' is unmasked. Right: position-dependent weights $\omega(p, t)$ on the y-axis for each value of $b_{p,t}$ on the x-axis, assuming that p is masked at step t.

where $c > 0$ and $\theta > 0$ are constants, and

$$\lambda(p, t) := 1 + \min(b_{p,t}, \ell_p - b_{p,t}) \tag{3}$$

is the distance, from the start or end of the final probe sequence, of the last base synthesized before step t, i.e., $b_{p,t}$ denotes the number of nucleotides synthesized within p up to and including step t, and ℓ_p is the probe length (see Figure 2 right).

The motivation behind an exponentially increasing weighting function is that the probability of a successful stable hybridization of a probe with its target should increase exponentially with the absolute value of its Gibbs free energy, which increases linearly with the length of the longest perfect match between probe and target. The parameter θ controls how steeply the exponential weighting function rises towards the middle of the probe. In our experiments, we set $\theta := 5/\ell_p$ and $c := 1/\exp(\theta)$.

We now define the conflict index of a probe p as

$$\mathcal{C}(p) := \sum_{t=1}^{T} \Big(\omega(p, t) \sum_{p'} \delta(p, p', t) \Big), \tag{4}$$

where p' ranges over all probes that are at most three cells away from p. $\mathcal{C}(p)$ can be interpreted as the fraction of faulty p-probes.

Note the following relation between conflict index and border length. Define $\delta(p, p', t) := 1$ if p' is a direct neighbor of p and is unmasked at step t, and $:= 0$ otherwise. Define $\omega(p, t) := 1$ if p is masked at step t, and $:= 0$ otherwise. Then $\sum_p \mathcal{C}(p) = 2 \sum_{t=1}^{T} \mathcal{B}_t$, as each border conflict is counted twice, once for p' and once for p. Therefore, border length and total conflict are equivalent for this particular choice of δ and ω. For our choice (1) and (2), they are not equivalent but still correlated: a good layout has both low border lengths and low conflict indices.

3 Review of Placement and Partitioning Algorithms

Placement Algorithms. All methods mentioned here assume fixed (synchronous, left-most, or otherwise pre-computed) embeddings. Post-placement optimizations such as the Chessboard [5] exist but separate the embedding problem from the placement problem.

The Border Length problem was first formally addressed in [4]. The article reports that the first Affymetrix chips were designed using a heuristic for the traveling salesman problem (TSP). The idea consists of building a weighted graph with nodes representing probes, and edges containing the Hamming distance between the probe sequences. A TSP tour is approximated, resulting in consecutive probes in the tour being likely similar. The TSP tour is then *threaded* on the array in a row-by-row fashion. A different threading of the TSP tour, called *1-threading*, is suggested to achieve up to 20% reduction in border length [4].

A different strategy called *Epitaxial* placement [5] places a random probe in the center of the array and continues to insert probes in spots adjacent to already filled spots. Priority is given to spots with the largest numbers of filled neighbors. At each iteraction, it examines all non-filled spots and finds a non-assigned probe with minimum sum of Hamming distances to the neighboring probes, employing a greedy heuristic to select the next spot to be filled. A further 10% reduction in border conflict over TSP + 1-threading is claimed.

Both the Epitaxial algorithm and the TSP approach do not scale well to large chips. For this reason, [6] proposes a simpler variant of the Epitaxial algorithm, called *Row-epitaxial*, with two main differences: spots are filled in a pre-defined order, namely from top to bottom, left to right, and only probes of a limited list of candidates are considered when filling each spot. Experiments show that Row-epitaxial is the best large-scale placement algorithm, achieving up to 9% reduction in border length over the TSP + 1-threading.

Partitioning Algorithms. The placement problem can be partitioned by dividing the set of probes into smaller sub-sets, and assigning these sub-sets to sub-regions of the chip. Each sub-region can then be treated as an independent chip or recursively partitioned. In this way, algorithms with non-linear time or space complexities can be used to compute the layout of larger chips that otherwise would not be feasible.

The first known partitioning algorithm is called Centroid-based Quadrisection [7]. It starts by randomly selecting a probe $c_1 \in \mathcal{P}$. Then, it selects another probe c_2 maximizing $h(c_1, c_2)$, the Hamming distance between their embeddings. Similarly, it selects c_3 and c_4 maximizing the sum of Hamming distance between these four probes, which are called centroids. All other probes $p \in \mathcal{P}$ are then compared to the centroids and assigned to the sub-set $\mathcal{P}_k$ associated with c_k that has minimum $h(p, c_k)$. The chip is divided into four quadrants, each being assigned to a sub-set $\mathcal{P}_k$. The procedure is repeated recursively on each quadrant until a given recursion depth is reached. In the end, the Row-epitaxial algorithm is used to produce the placement of the probes in each final sub-region.

We recently developed an approach that, for the first time, combines the partitioning of the chip with the embedding of the probes [1]. Our algorithm, called Pivot Partitioning, achieves up to 6% reduction in conflicts when compared to the best known algorithms.

4 Quadratic Assignment Problem

We now explore a different approach to the design of microarrays based on the quadratic assignment problem (QAP), a classical combinatorial optimization that can be stated as follows. Given $n \times n$ real-valued matrices $F = (f_{ij}) \geq 0$ and $D = (d_{kl}) \geq 0$, find a permutation π of $\{1, 2, \ldots n\}$ such that

$$\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot d_{\pi(i)\pi(j)} \rightarrow \min . \tag{5}$$

The attribute *quadratic* stems from the fact that the target function can be written with n^2 binary indicator variables $x_{ik} \in \{0, 1\}$, where $x_{ik} := 1$ if and only if $k = \pi(i)$. The objective (5) then becomes $\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot \sum_{k=1}^{n} \sum_{l=1}^{n} d_{kl} \cdot x_{ik} \cdot x_{jl} \rightarrow \min$, such that $\sum_k x_{ik} = 1$ for all i, $\sum_i x_{ik} = 1$ for all k and $x_{ik} \in \{0, 1\}$ for all (i, k). The objective function is a quadratic form in x.

The QAP has been used to model a variety of real-life problems. One of its major applications is to model the facility location problem where n facilities must be assigned to n locations. In this scenario, F is called the flow matrix as f_{ij} represents the flow of materials from facility i to facility j. One unit of flow is assumed to have an associated cost proportional to the distance between the facilities. Matrix D is called the distance matrix, as d_{kl} gives the distance between locations k and l. The optimal permutation π defines a one-to-one assignment of facilities to locations with minimum cost.

QAP Formulation of Probe Placement. The probe placement problem can be seen as an instance of the QAP, where we want to find a one-to-one correspondence between spots and probes. In a realistic setting, we may have more spots available than probes to place. Below we show that this does not cause problems as we can add enough "empty" probes and define their weight functions appropriately.

Perhaps more severely, we assume that all probes have a single pre-defined embedding in order to force a one-to-one relationship. A more elaborate formulation would consider all possible embeddings of a probe, but then it becomes necessary to ensure that only one embedding of a probe is assigned to a spot. This still leads to a quadratic integer programming problem, albeit with slightly different side conditions.

Our goal is to design a microarray minimizing the sum of conflict indices over all probes k, i.e., $\sum_k \mathcal{C}(k) \rightarrow \min$.

The "flow" f_{ij} between spots i and j depends on their Euclidean distance $d(i, j)$ on the array; in accordance with the conflict index model, we set

$$f_{ij} := \begin{cases} (d(i,j))^{-2} & \text{if spot } j \text{ is "near" spot } i, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

where "near" means that spot j is at most three cells away from i. Note that most of the flow values on large arrays are zero. For Border Length Minimization, the case is even simpler: We set $f_{ij} := 1$ if spots i and j are direct neighbors, and $f_{ij} := 0$ otherwise.

The "distance" d_{kl} between probes k and l depends on the (weighted) Hamming distance of their embeddings. To account for possible "empty" probes to fill up surplus spots, we set $d_{kl} := 0$ if k or l or both refer to an empty probe (i.e., empty probes never contribute to the target function, we do not mind if nucleotides are erroneously synthesized on spots assigned to empty probes). For real probes, we set

$$d_{kl} := \sum_{t=1}^{T} d_{klt},$$

where d_{klt} is the potential contribution of probe l's embedding to the failure risk of probe p_k in the t-th synthesis step. According to the conflict index model,

$$d_{klt} = \begin{cases} c \cdot \exp(\theta \cdot \lambda(p_k, t)) & \text{if } p_k \text{ is masked and } p_l \text{ unmasked in step } t, \\ 0 & \text{otherwise.} \end{cases}$$

In the special case of the Border Length Minimization Problem, where $\theta = 0$ and $c = 1/2$, we obtain that $d_{kl} + d_{lk} = H_{kl} = H_{lk}$, where H_{kl} denotes the Hamming distance between the embeddings of probes p_k and p_l.

It now follows that for a given assignment π, we have, in the notation of Section 2, $f_{ij} d_{\pi(i)\pi(j)} = \sum_{t=1}^{T} \delta(p_{\pi(i)}, p_{\pi(j)}, t) \cdot \omega(p_{\pi(i)}, t)$. The objective function (5) then becomes

$$\begin{aligned} & \sum_i \sum_j f_{ij} \cdot d_{\pi(i)\pi(j)} = \sum_i \sum_j \Big(\sum_{t=1}^{T} \delta(p_{\pi(i)}, p_{\pi(j)}, t) \cdot \omega(p_{\pi(i)}, t) \Big) \\ = \quad & \sum_i \sum_{t=1}^{T} \Big(\omega(p_{\pi(i)}, t) \cdot \sum_j \delta(p_{\pi(i)}, p_{\pi(j)}, t) \Big) = \sum_i \mathcal{C}(\pi(i)) = \sum_k \mathcal{C}(k), \end{aligned}$$

and indeed equals the total conflict index with our definitions of $F = (f_{ij})$ and $D = (d_{kl})$. Note that it is technically possible to switch the definitions of F and D, i.e., to assign probes to spots instead of spots to probes as we do now, without modifying the mathematical problem formulation. However, this would lead to high distance value for neighboring spots and many zero distance values for independent spots, a somewhat counterintuitive model. Also, QAP heuristics tend to find pairs of objects with large flow values and place them close to each other, initially. Therefore, the way of modeling F and D may be significant.

QAP Heuristics. We have shown how the microarray placement problem can be modeled as a quadratic assignment problem. The QAP is known to be NP-hard and particularly hard to solve in practice. Instances of size larger than $n = 20$ are generally considered to be impossible to solve to optimality. Nevertheless, our formulation is of interest because we can now use existing QAP heuristics (see [2] for a survey) to design the layout of microarrays minimizing either the sum of border lengths or conflict indices.

Table 1: Border length of random chips compared with the layouts produced by Row-epitaxial and GRASP with path-relinking. Reductions in border length are reported in percentages compared to the random layout.

		Random	Row-epitaxial			GRASP with path-relinking		
Chip dimension	Number of probes	Border length	Border length	Reduction (%)	Time (sec.)	Border length	Reduction (%)	Time (sec.)
6×6	36	1 989.20	1 714.60	13.80	0.01	1 672.20	15.94	2.73
7×7	49	2 783.20	2 354.60	15.40	0.02	2 332.60	16.19	6.43
8×8	64	3 721.20	3 123.80	16.05	0.03	3 099.13	16.72	12.49
9×9	81	4 762.00	3 974.80	16.53	0.05	3 967.20	16.69	25.96
10×10	100	5 985.20	4 895.60	18.20	0.06	4 911.40	17.94	47.57
11×11	121	7 288.40	5 954.40	18.30	0.10	5 990.73	17.80	87.48
12×12	144	8 714.00	7 086.20	18.68	0.11	7 159.80	17.84	152.42

As an example, we used a general QAP heuristic known as GRASP [8] (Greedy Randomized Adaptive Search Procedure), and an improved version called GRASP with path-relinking [9]. GRASP is comprised of two phases: a construction phase where a random feasible solution is built, and a local search phase where a local optimum in the neighborhood of that solution is sought.

Initially, the elements of the distance and flow matrices are sorted in increasing and decreasing order, respectively. The first β elements of each are kept (where $0 < \beta < 1$) and their products are computed. A simultaneous assignment of a pair of facilities to a pair of locations is selected at random among those with the α smallest costs, where $0 < \alpha < 1$. A feasible solution is then built by making a series of greedy assignments.

The construction and local search phases are repeated for a given number of times. Each iteration is independent in the sense that a new solution is always built from scratch. GRASP with path-relinking is an extension of the basic algorithm that uses an "elite set" to store the best solutions found. It incorporates a third phase that chooses, at random, one elite solution that is used to improve the solution produced at the end of the local search phase.

5 Results and Discussion

We present experimental results of using GRASP with path-relinking (GRASP-PR) for designing the layout of small artificial chips, and compare them with the layouts produced by Row-epitaxial. We used a C implementation of GRASP-PR provided by [9] with default parameters (32 iterations, $\alpha = 0.1$, $\beta = 0.5$, and elite set of size 10) and our own implementation of Row-epitaxial. The chips have 25-nt probes uniformly generated and asynchronously embedded in a deposition sequence of length 74. The running times and the border lengths of the resulting layouts are shown in Table 1 (all results are averages over a set of ten chips).

Our results show that GRASP-PR produces layouts with lower border lengths than Row-epitaxial on the smaller chips. On 6 x 6 chips, GRASP-PR outperforms Row-epitaxial by 2.14 percentage points on average, when compared to the initial random layout. On 9 x 9 chips, however, this difference drops to 0.16 percentage point, while Row-epitaxial

Table 2: Average conflict indices of random chips compared with the layouts produced by Row-epitaxial and GRASP with path-relinking.

		Random	Row-epitaxial			GRASP with path-relinking		
Chip dimension	Number of probes	Avg. C. Index	Avg. C. Index	Reduction (%)	Time (sec.)	Avg. C. Index	Reduction (%)	Time (sec.)
6×6	36	524.28	495.15	5.56	0.05	467.08	10.91	3.68
7×7	49	558.25	521.90	6.51	0.07	489.32	12.35	8.84
8×8	64	590.51	551.84	6.55	0.09	515.69	12.67	19.48
9×9	81	613.25	568.62	7.28	0.11	533.79	12.96	38.83
10×10	100	628.50	576.49	8.28	0.11	539.69	14.13	73.09
11×11	121	642.72	588.91	8.37	0.12	551.41	14.21	145.67
12×12	144	656.86	598.21	8.93	0.12	561.21	14.56	249.19

generates better layouts on 11 x 11 or larger chips. In terms of running time, Row-epitaxial is faster and shows little variation as the number of probes grows. In contrast, the time required to compute a layout with GRASP-PR increases at a fast rate.

Table 2 shows improved results in terms of conflict indices. For these experiments, we used the same implementation of GRASP-PR and a version of Row-epitaxial implemented for conflict index minimization, which fills every spot with a probe minimizing the resulting conflict index on that spot. GRASP-PR consistently produces better layouts on all chip dimensions, achieving up to 6.38% less conflicts on 10 x 10 chips, for example, when compared to Row-epitaxial. In terms of running times, however, GRASP-PR is even slower for the case of conflict index minimization. Reasons are two-fold. First, the definitions of matrices F and D are more elaborate in the conflict index model. Second, the distance matrix contains less zero entries, which seems to increase the running time of GRASP.

The gains in terms of conflict index of both approaches are clearly less than the gains in terms of border length. This may be because the embeddings are fixed and the reduction of conflicts is restricted to the relocation of the probes, which only accounts for one part of the conflict index model. The fact that the distance matrix contains less zero entries, however, might explain why GRASP-PR performs better in terms of conflict index minimization when compared to Row-epitaxial.

Because of the large number of probes on industrial microarrays, it is not feasible to use GRASP-PR (or any other QAP method) to design an entire microarray chip. However, we showed that it is certainly possible to use it on small sub-regions of a chip, which opens up the way for two interesting alternatives. First, our QAP approach could be used combined with a partitioning strategy such as the Centroid-based Quadrisection or our new Pivot Partitioning, to the design the smaller regions that result from the partitioning.

Second, an existing layout could be improved, iteratively, by relocating probes inside a defined region of the chip, in a sliding-window fashion. Each iteration produces an instance of a QAP whose size equals the size of the window. The QAP heuristics can be used to check whether a different arrangement of the probes inside the window can reduce the conflicts. For this approach to work, we also need to take into account the conflicts due to the spots around the window. Otherwise, a new layout with less internal conflicts could be achieved at the expense of an increase of conflicts on the borders of the window.

A simple way of preventing this problem is to solve a larger QAP instance consisting of the

spots inside the window as well as those around it. The spots outside the window obviously must remain unchanged, and that can be done by fixing the corresponding elements of the permutation π. Note that there is no need to compute f_{ij} if spots i and j are both outside the window, nor d_{kl} if probes k and l are assigned to spots outside the window.

Summary. We have identified the probe placement or microarray layout problem with general distance-dependent and position-dependent weights as a (specially structured) quadratic assignment problem. QAPs are notoriously hard to solve, and currently known exact methods start to take prohibitively long already for slightly more than 20 objects, i.e., we could barely solve the problem for 5×5 arrays. However, the literature on QAP heuristics is rich, as many problems in operations research can be modeled as QAPs. Here we used one such heuristic to identify the potential of the probe-placement-QAP-relation.

References

[1] de Carvalho Jr.,S., Rahmann,S. (2006) Improving the Layout of Oligonucleotide Microarrays: Pivot Partitioning. In *Workshop on Algorithms in Bioinformatics (WABI)*, LNBI, **4175**. Springer.

[2] Çela,E. (1998) *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer, Massachessets, USA.

[3] Fodor,S., Read,J., Pirrung,M., Stryer,L., Lu,A. and Solas,D. (1991) Light-directed, spatially addressable parallel chemical synthesis. *Science*, **251**, 767–73.

[4] Hannenhalli,S., Hubell,E., Lipshutz,R. and Pevzner,P. (2002) Combinatorial algorithms for design of DNA arrays. *Advances in Biochemical Engineering / Biotechnology*, **77**, 1–19.

[5] Kahng,A.B., Mandoiu,I.I., Pevzner,P.A., Reda,S. and Zelikovsky,A.Z. (2002) Border length minimization in DNA array design. In *Proceedings of the Second Workshop on Algorithms in Bioinformatics*.

[6] Kahng,A.B., Mandoiu,I., Pevzner,P., Reda,S. and Zelikovsky,A. (2003a) Engineering a scalable placement heuristic for DNA probe arrays. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology*, 148–156.

[7] Kahng, A.B., Mandoiu,I., Reda,S., Xu,X. and Zelikovsky,A. (2003b), Evaluation of placement techniques for DNA probe array layout. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 262–269.

[8] Li,Y., Pardalos,P.M. and Resende,M.G.C. (1994) A greedy randomized adaptive search procedure for the quadratic assignment problem. In Pardalos,P. and Wolkowicz,H. (eds.), *Quadratic Assignment and Related Problems*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **16**, 237–261.

[9] Oliveira,C.A.S., Pardalos,P.M. and Resende,M.G.C. (2004) GRASP with path-relinking for the quadratic assignment problem. In Ribeiro,C.C. and Martins,S.L. (eds.), *Efficient and Experimental Algorithms*, LNCS, **3059**, 356–368, Springer-Verlag.

[10] Rahmann,S. (2003) The shortest common supersequence problem in a microarray production setting. In *Proceedings of the 2nd European Conference in Computational Biology* (ECCB 2003), volume 19 Suppl. 2 of *Bioinformatics*, pages ii156–ii161.

Comparison of Human Protein-Protein Interaction Maps

Matthias E. Futschik[1], Gautam Chaurasia[1,2], Erich Wanker[2] and Hanspeter Herzel[1]

[1]Institute for Theoretical Biology, Charité, Humboldt-Universität and [2]Max-Delbrück-Centrum,
Invalidenstrasse 43
10115 Berlin, Germany
m.futschik@biologie.hu-berlin.de

Abstract: Large-scale mappings of protein-protein interactions have started to give us new views of the complex molecular mechanisms inside a cell. After initial projects to systematically map protein interactions in model organisms such as yeast, worm and fly, researchers have begun to focus on the mapping of the human interactome. To tackle this enormous challenge, different approaches have been proposed and pursued. While several large-scale human protein interaction maps have recently been published, their quality remains to be critically assessed. We present here a first comparative analysis of eight currently available large-scale maps with a total of over 10000 unique proteins and 57000 interactions included. They are based either on literature search, orthology or by yeast-two-hybrid assays. Comparison reveals only a small, but statistically significant overlap. More importantly, our analysis gives clear indications that all interaction maps suffer under selection and detection biases. These results have to be taken into account for future assembly of the human interactome.

1 Introduction

Interactions between proteins underlie the vast majority of cellular processes. They are essential for a wide range of tasks and form a network of astonishing complexity. Until recently, our knowledge of this complex network was rather limited. The emergence of large scale protein-protein interaction maps has given us new possibilities to systematically survey and study the underlying biological system. The first attempts to collect protein-protein interactions on large scale were initiated for model organisms such as *S. cerevisiae*, *D. melanogaster* and *C. elegans* [Gavin *et al.* '02,Giot *et al.* '03,Ito *et al.* '01,Li *et al.* '04,Uetz *et al.* '00]. Evidently, the generated interaction maps offered a rich resource for systematic studies.

After these initial efforts, the focus has moved towards deciphering the human interactome. Recently, the first large-scale human protein interaction network has been constructed following alternative mapping strategies. Most currently available human interaction maps can be divided into three classes: i) maps obtained from literature search [Bader *et al.* '01,Peri *et al.* '03,Ramani *et al.* '05], ii) maps derived from interactions between orthologous proteins in other organisms [Brown and Jurisica '05,

Lehner and Fraser '04, Persico *et al.* '05] and iii) maps based from large scans using yeast-two-hybrid (Y2H) assays [Rual *et al.* '05, Stelzl *et al.* '05]. All of these different mapping strategies have their obvious advantages as well as disadvantages. For example, Y2H-based mapping approaches offer rapid screens between thousands of proteins, but might produce a high false positive rate. The extent, however, how much the resulting interaction maps are influenced by the choice of mapping strategy, is less clear. Thus, it is important to critically assess the quality and reliability of produced maps.

For yeast interaction maps, several of such critical comparisons have been performed [Bader and Hogue '02, von Mering *et al.* '02]. They revealed a surprising divergence between different interaction maps. They also indicated that functional coherency of maps is severely influenced by the choice of mapping scheme. Such comparison is still lacking for human protein interaction maps despite their expected importance for biomedical research [Goehler *et al.* '04]. Therefore, we compared several currently available large-scale interactions maps regarding their concurrence and divergence. We assess especially potential selection and detection biases as they might interfere with future applications of these maps.

2 Materials and Methods

2.1 Assembly of Protein-Protein Interaction Maps

To evaluate the different mapping approaches listed above, we selected eight publicly available large-scale interaction maps: three literature-based, three orthology-based and two Y2H-based maps. We restricted further our analysis to binary interactions in order to compare Y2H-based maps directly with the remaining interaction maps.

Two literature-based interaction maps were derived from the Human Protein Reference Database (HPRD) and Biomolecular Interaction Network Database (BIND) [Bader *et al.* '01, Peri *et al.* '03]. These manually curated databases are mainly based on literature reviews performed by human experts. At the time of analysis, interactions included in these databases were predominantly from small scale experiments. As third literature-based interaction map, we used the set of interactions found by Ramani and co-workers using a text-mining approach [Ramani *et al.* '05]. As HPRD and BIND, it is based on literature, but computationally generated. In our study, we will refer to it as the COCIT map.

The first orthology-based protein interaction map was proposed by Lehner and Fraser [Lehner *et al.* '04]. Interactions included were predicted based on interactions observed between orthologous proteins in yeast, worm and fly. We used only interactions that were assigned to core map by Lehner and Fraser, as these were identified with high confidence. Besides this map (here referred to as the ORTHO map), we included two alterative orthology-based large-scale maps from in the Online Predicted Human Interaction Database (OPHID) and HOMOMINT database [Brown *et al.* '05, Persico *et al.* '05]. Both mappings were derived following the approach by Lehner and Fraser with some deviations. We extracted from the two databases only the interactions that were based on orthology assignment to ensure conformity of the resulting maps.

The Y2H-based interaction maps included in our comparison were generated in the recent large-scale scans by Stelzl *et al*. and Rual *et al*. [Rual *et al.* '05, Stelzl *et al.* '05] We will refer to these maps as MDC-Y2H and CCSB-H1 in our study. Although both scans are based on Y2H-assay, it should be noted that considerable differences exist in regard to experimental procedures.

To enable comparison, all proteins were mapped to their corresponding EntrezGene ID. For efficient computational analysis, we converted all interaction maps into graphs using the Bioconductor *graph* package [Balasubramanian *et al.* '04, Carey *et al.* '05, Gentleman *et al.* '04].

2.2 Overlap of Interaction Maps

Protein interaction maps are formed by both their proteins and interactions included. Thus, any comparison of maps should assess the concurrence of proteins as well as of interactions in different maps.

Comparison of the proteins in different maps is based on following procedure: Given the sets of proteins (P_A, P_B) in map A and B, their intersection is $P_{AB} = P_A \cap P_B$. To facilitate assessment, the intersection was normalized in regard to the total number of proteins in A or B ($P^A_{AB} = P_{AB} / P_A$; $P^B_{AB} = P_{AB} / P_B$). Thus, the normalized intersection is simply the percentage of proteins that can be found in the other map. In our study, we will refer to the average of P^A_{AB} and P^B_{AB} as the *(relative) protein overlap* between A and B.

For the comparison of interactions, we could proceed similarly by counting common interactions in two maps. However, it is important to note that network structure is not only determined by existing interactions, but also by missing interactions. As we want to assess the concurrence of maps for both observed as well as missing interactions, we used a log-likelihood ratio (*LLR*) score [Lee *et al.* '04]. The *LLR* provides a similarity measure for two sets of interactions (I_1,I_2). It is defined as

$$LLR(I_1, I_2) = \ln\left(\frac{P(I_1 \mid I_2)}{P(I_1 \mid \sim I_2)}\right)$$

where $P(I_1|I_2)$ is the probability of observing an interaction in I_1 conditioned on observing the same interaction in I_2. Respectively, $P(I_1|\sim I_2)$ is the probability of observing an interaction in I_1 conditioned on not observing the same interaction in I_2. For highly similar interaction networks, *LLR* produces large scores. For absence of similarity, the *LLR* score is zero. The latter is the case if random interactions networks are compared.

Additionally to the *LLR* score, we used two permutation tests to stringently assess the statistical significance of observed concurrence of interactions [Balasubramanian *et al.* '04]. For both tests, a large set of random networks are generated based on the original networks, either by re-labelling of nodes (*node label permutation*) or by randomly permuting the edges (*edge permutation*). In contrast to node label permutation, the implemented scheme for edge permutation does not conserve the degree distribution i.e. the number of interactions of proteins. Subsequently, the number of common interactions between the original networks is compared to the corresponding number for randomized networks. The probability of observing at least the same number of common interactions for random networks determines the significance. Although their permutation schemes are different, the two tests usually produce similar results.

2.3 Gene Ontology Analyses

Protein interaction maps can be compromised by several types of biases. For example, selection bias arises if certain protein categories are over- or underrepresented in a chosen map. To assess stringently the significance of such potential biases, we utilized Fisher's exact test. It is based on the hypergeometric distribution and delivers the probability P to observe k or more proteins of chosen category in case of random drawings:

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{l-i}}{\binom{N}{l}}$$

where M is the total number of proteins attributed to the category, N is the total number of proteins annotated and l is the number of proteins in the corresponding map. The significance of under-represented GO categories in maps can be calculated accordingly. Since we tested simultaneously for multiple GO categories, the p-values were converted to false discovery rates applying the Benjamini-Hochberg procedure [Benjamini and Hochberg '95]. As reference, the set of all proteins tested for interactions could be used. However, such sets are explicitly known only for Y2H-based maps comprising the proteins in a matrix screen. For literature- and orthology-based maps, these sets are not available. Hence, we used the set of all genes annotated in GO as reference to facilitate direct comparison.

We also assessed whether interactions between protein classes were overrepresented. We determined the number of interactions k_{mn} between proteins of GO category m and proteins of GO category n. Log_2-odds were calculated to assess deviation of the observed number of interactions k_{mn} with the number k^0_{mn} of interactions expected for randomized networks:

$$LOD_{mn} = \log_2 \frac{k_{mn}}{k^0_{mn}}$$

MAP	REFERENCE	*P*	*I*	D_{AV}	METHOD
MDC-Y2H	Stelzl *et al.* 2005 *Cell*	1703	3186	1.9	Y2H-ASSAY
CCSB-H1	Rual *et al.* 2005 *Nature*	1549	2754	1.8	Y2H-ASSAY
HPRD	Peri *et al.* 2003 *Genome Res*	5908	15658	2.7	LITERATURE
BIND	Bader *et al.* 2001 NAR	2677	4233	1.7	LITERATURE
COCIT	Ramani *et al.* 2004 *Genome Biology*	3737	6580	1.8	LITERATURE
OPHID	Brown and Jurisica 2005 *Bioinformatics*	2284	8962	3.9	ORTHOLOGY
ORTHO	Lehner and Fraser 2003 *Genome Biology*	3503	9641	2.8	ORTHOLOGY
HOMOMINT	Persico *et al.* 2005 *BMC Bioinformatics*	2556	5582	2.3	ORTHOLOGY

Table 1: List of human protein-protein interactions maps compared in this study. The number of proteins *P* and interactions *I* result after mappings of proteins to their corresponding Entrez ID. D_{av} denotes the average number of interactions per protein.

The randomized networks had the same number of proteins and interactions as the corresponding maps. The proteins' connectivity (number of interactions per protein) was also conserved.

Alternatively, we can evaluate the tendency that proteins of similar function interact. Although difficult to define rigorously, similarity of function may be approximated by following procedure [Jansen *et al.* '03]: After mapping proteins to their GO terms (categories), their functional similarity is determined by the positions of corresponding GO terms within the GO graph. Similar GO terms are expected to be located in proximity to each other. Measuring the shared paths to the GO terms (from the root term), we would expect that similar GO terms have larger shared paths than unrelated GO terms. Thus, if proteins of similar function tend to interact in a network, the average shared paths lengths will be larger than random networks. To test the significance, we compared therefore the distribution of shared path lengths to those measured for randomized networks. Note that we counted the largest shared path length in case of multiple GO assignments for proteins.

3. Results

In total, we were able to map 57095 interactions between 10769 proteins uniquely identified by the corresponding Entrez IDs (table 1). The size of the interactions maps varied between 2754 (CCSB-H1) and 15658 (HPRD) interactions. Proteins had an average number of 1.8 to 3.8 interactions. Considering previous estimates of an average of 3-10 interactions per proteins, the result indicates that interactions maps are currently still highly unsaturated [Bork *et al.* '04].

3.1 Common Proteins and Interactions

We examined first how many proteins and interactions were common to the different

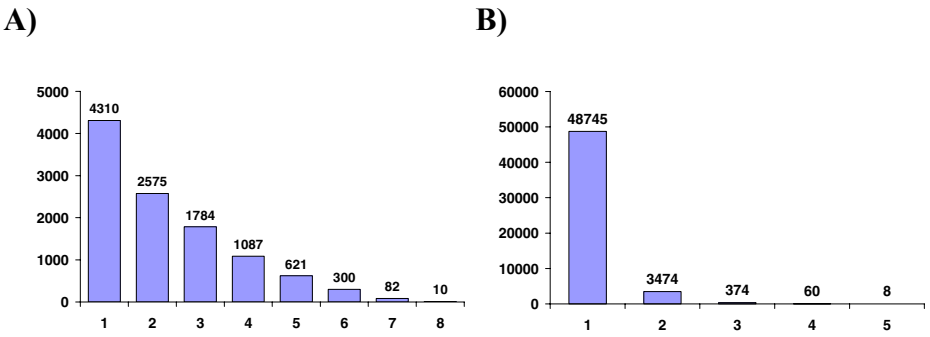


Figure 1: Number of proteins (A) and interactions (B) common to multiple maps. The x-axis shows the number of maps in which proteins or interactions are included.

maps in our comparison (figure 1). We found that a large part (60%) of all proteins can be found in at least two maps. The number of proteins included in all eight maps, however, is diminishingly small: Only 10 proteins (i.e. 0.001% of all proteins) fulfill this criterion. Even more striking were the small numbers of common interactions. The vast majority of interactions (85%) are cataloged in only a single map. No interaction can be found in six or more maps; and just 8 interactions are common to five maps.

3.2 Protein Overlap

To investigate whether some maps tend to share more proteins than others, we calculated the relative protein overlap for each pair of maps. We detected considerable variation of protein overlap ranging from 16% to 58%. Comparison of overlaps gave us first indications that maps could be ordered into distinct groups. To examine this possibility, a clustering approach was applied. First, we converted protein overlaps O_{ij} (between maps i and j) into distances Δ_{ij} defined as $\Delta_{ij} = 1\text{-}O_{ij}$. Thus, maps having large protein overlap are assigned a small distance between each other. After conversion, the interaction maps were hierarchically clustered. The resulting cluster structure showed a surprisingly clear pattern: All maps are grouped in accordance to the mapping approach used for their generation. We obtained two clusters that either included only literature-based or orthology-based maps. The Y2H-based maps formed own clusters. The CCSB-H1 has the most distinguished set of proteins, whereas MDC-Y2H is placed closer to the remaining maps. These observations indicate that all mapping approaches show their own characteristic preference for proteins included or, in others words, a prominent selection bias.

We verified this conjecture by testing systematically for over- and under-representation of protein categories in interaction maps. The categories used were based on Gene Ontology (GO) that currently represents the most comprehensive system of annotation for the human genome [Ashburner *et al.* '00]. Gene Ontology assigns defined categories

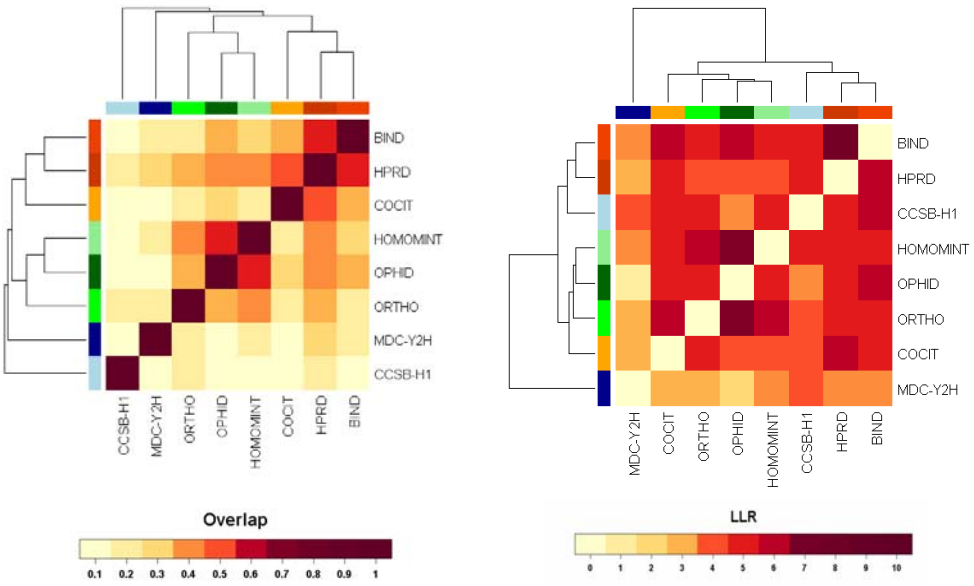


Figure 2: Hierarchical clustering of maps based on protein overlap (A) and log likelihood ratio *LLR* (B) as defined in *Materials and Methods*. The matrices display the (relative) protein overlap, respectively the *LLR* between all possible pairs of maps. Their numerical values are represented according to color-bars at the bottom. On top and right side of each matrices, dendrograms resulted from the clustering are shown. Clustering of protein overlap was based on the distance Δ between map *i* and *j* defined as $\Delta_{ij} = 1\text{-}O_{ij}$ where O_{ij} is protein overlap between maps *i* and *j*. For clustering of *LLR*, the distance Δ was defined as $\Delta_{ij} = 1/LLR(I_i, I_j)$ where $I_{i,j}$ are the sets of interactions included in map *i* or *j*. For both cluster analysis, average linkage was used

to genes according to their molecular function (MF), biological process (BP) or cellular component (CC). First, we tested whether proteins of MF categories are overrepresented in maps using Fisher's test (FDR = 0.01). As reference, the set of all annotated human genes in GO was used. Most maps showed significant enrichment for proteins involved in nucleotide binding (all maps except CCSB-H1) and protein binding (all except ORTHO). Likewise, all maps were found to be enriched by proteins related to metabolism and cell cycle (BP categories) or located in the nucleus (CC category). Interestingly, signal transducers are significantly underrepresented in Y2H- and orthology-based maps, whereas they are significantly overrepresented in literature-based maps. Whereas the reasons for the observed underrepresention are less clear, a possible explanation for the overrepresentation in literature-based maps is the existence of an inspection bias towards 'popular' signaling proteins in the literature. Surprisingly, we detected a highly significant depletion of membrane proteins in all maps including pharmaceutically important classes as the G-protein coupled receptors.

3.3 Concurrence of Interactions

After the comparison of maps based on proteins included, we focused on the concurrence of interactions. To assess the similarity between maps, the *LLR* was calculated for each pair. It ranged from 1.5 (MDC-Y2H- OPHID) to 7.1 (BIND-HPRD) having an average value of 4.6. For all comparisons, it was notably larger than zero, which is the expected value for comparison of random maps. This signifies that the observed concurrence of interaction maps did not occur merely by chance despite of being rather small. To confirm this finding, we applied two permutation tests (described the *Materials and Methods*) for pair-wise comparison of graphs. These results showed that the observed overlap of interactions is highly significant for all comparisons ($p < 0.01$).

Inspection of the *LLR*s also suggested that the interaction maps can be divided into distinct groups. Similarly as before, we subsequently clustered interaction maps to detect common tendencies. The distance was defined as the reciprocal *LLR*. Similar maps score a large *LLR* resulting in a small distance. Hierarchical clustering resulted again in the formation of distinct cluster. However, the detected clusters were differently composed compared to the clusters based on protein overlap. This time, COCIT was found in the group of orthology-based clusters, whereas CCSB-H1 was assigned to the cluster of literature–based HPRD and BIND. MDC-Y2H formed its own separate cluster displaying the weakest similarity to remaining maps. Interestingly, the two large clusters follow exactly the division into computationally generated maps (COCIT, ORTHO, OPHID, HOMOMINT) and maps based on experiments (HPRD, BIND, CCSB-H1). An explanation for this observation is still lacking.

3.4 Coherency of Interaction Maps

Next, we examine the functional coherency of maps. The observation that interacting proteins tend to have common functions has previously been utilized for assessing the quality of interaction maps as well as for *de novo* prediction [Schwikowski *et al.* '00,von Mering *et al.* '02]. To test whether current human interactions maps also display such functional coherency, we employed the gene annotations available in GO. We followed two alternative approaches: First, we assessed the similarity of GO annotations of interacting proteins. In case that interacting proteins have similar functions, their MF annotations should be more similar than expected for random pairs of proteins. This can be measured by the shared path length of GO categories for interacting proteins (see *Materials and Methods*): Assuming a strong correlation between function and interaction (i.e. large functional coherency), we would observe that short shared path length are less likely and long shared path length are more likely than expected. The results of this analysis are shown in figure 3. Indeed, all maps follow this pattern. However, considerable differences can be observed. COCIT showed the largest functional coherency of all maps whereas MDC-Y2H and OPHID showed only modest coherency. A similar analysis was performed for maps with regard to shared process (BP) and location (CC) of interacting proteins. Here, all maps displayed large coherency with only minor differences between maps (figure 3).

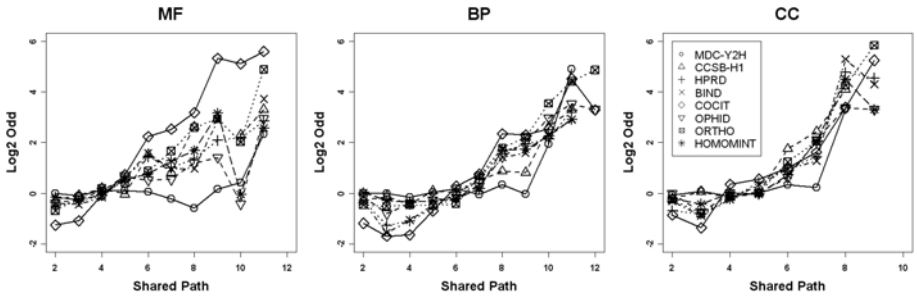


Figure 3: Assessment of coherency based on GO annotations for molecular function (MF), biological process (BP) and cellular component (CC). For interacting proteins, the shared path lengths of GO categories were calculated as described in *Materials and Methods*. The figures show the $\log_2$ odds for the observed path lengths with respect to path lengths derived for random networks. Log_2 odds are plotted as function of shared path lengths.

An alternative approach to study the coherency of interaction maps is the examination whether interacting proteins share a common location. It is based on inspection of the interaction matrix as described in *Materials and Methods*. A similar strategy was introduced by von Mering and co-workers counting the interactions within and between functional categories for yeast interaction maps [von Mering *et al.* '02]. If only interactions of proteins of the same category occur, a diagonal pattern emerges in the corresponding interactions matrices. However, this assumes that proteins are assigned to a single category and not to multiple categories as it is frequently the case for GO annotations. Thus, we modified the approach and compared the observed interaction matrices to matrices of the corresponding randomized networks. Figure 4 displays the log odds for interactions between CC categories of the third level. Interestingly, some compartments (e.g. cytoskeleton) are enriched by internal interactions independently of the map chosen. Generally, however, literature-based networks displayed most prominently enrichment of interactions within proteins of the same component. Less clear patterns for enrichment were found for MC-Y2H and OPHID. This result seems to contradict the previous observation that the coherency for location is similar in all interaction maps (figure 3). However, it is important to note that the interaction matrix approach only assesses the coherency at one particular level of the GO hierarchy. This is contrasted by the previous approach that integrates over all levels. Moreover, overrepresentation of interactions between different categories might not always derive from poor quality of interaction maps, but may point to true biological coupling of cellular compartments. For example, the repeatedly observed enrichment in protein interactions between endomembrane and plasma membrane most likely reflects the close biological connection of both membrane systems.

4. Discussion and Conclusions

Large-scale maps of protein-protein interactions promise to have a considerable impact on the revelation of molecular networks. Similar to fully sequenced genomes serving nowadays the base for genomics, large-scale maps of the interactome might become the foundation for any systematic approach to model cellular networks. Thus, they are likely

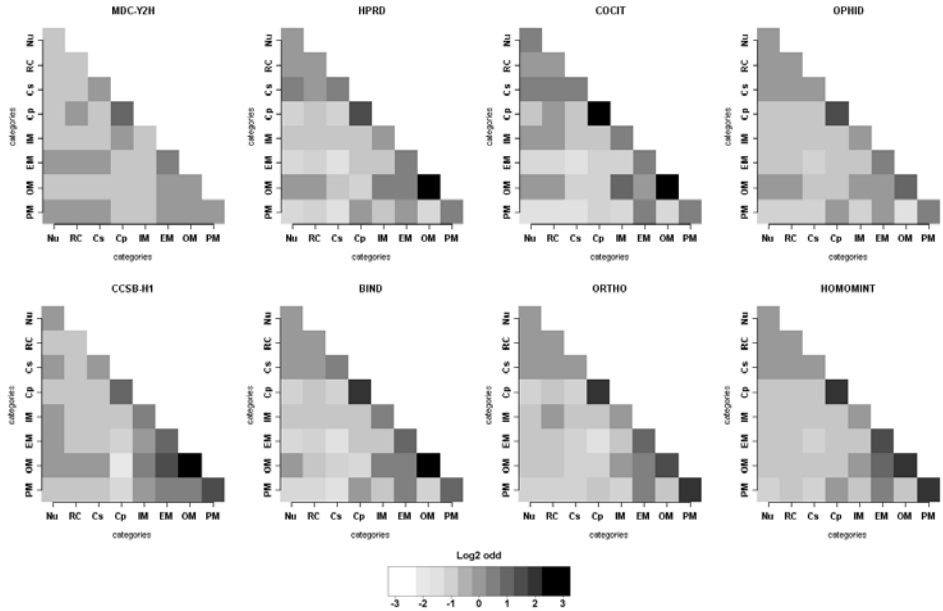


Figure 4: Cellular components of interacting proteins. Pairs of interacting proteins were mapped to the pairs of cellular components to which the proteins are assigned in Gene Ontology. The plots display the $\log_2$ odds ratios of the observed distribution compared to distribution obtained for randomized networks with conserved degree distribution. Categories of the third level of GO were chosen. The following abbreviations were used: Nu – *Nucleus*, RC – *Ribonucleoprotein complex*, Ck – *Cytoskeleton*, Cp – *Cytoplasm*, IM – *Intrinsic to membrane*, EM – *Endomembrane system*, OM – *Organelle membrane* and PM – *Plasma membrane*. For simplicity, only GO categories are shown including more than 2% percent of total number of proteins.

to be of substantial benefit for biomedical researchers. However, a requisite for future application of large-scale human interaction maps is a critical assessment of their quality and reliability. Therefore, we presented here a first comparison of eight currently available large-scale interaction maps. Our comparison is distinguished from previous studies, as it includes all three main approaches currently used for assembly of the human interactome.

A general analysis showed a distinct picture for the concurrence of proteins and interactions in different maps. While a large part of proteins are shared between maps, the interactions included are largely complementary. Only a small percentage of all interactions can be found in multiple maps. This finding has two direct consequences for the integration of maps: The previously proposed approach of assigning higher confidence to interactions found in multiple maps is strongly restricted by the low number of shared interactions [von Mering *et al.* '02]. At the same time, however, the complementary of interactions based on a large overlapping set of proteins indicates that unifying interaction maps will be highly beneficial.

We detected strong sampling and detection biases linked to the approaches used for the generation of the maps. This is reflected by the appearance of distinct groups when

cluster analysis was applied to interaction maps. Such biases have to be observed when interaction maps are utilized. Nonetheless, our analysis showed that most interaction maps display a high internal coherency regarding function, process and location of proteins. This result gives justification for future de novo annotation of proteins based on interaction maps. We like to note that the use of GO for assessment might lead to overestimation of the coherency of literature-based maps, as GO annotations are frequently also based on literature reviews and, thus, do not represent a truly independent benchmark set. In this case, the apparent lack of coherency in other maps could be interpreted that these maps may provide more novel information about the observed interactions.

Although the overlap of protein interactions is statistically significant, it remains small even for maps derived by similar approach. Only 12% - 36% of interactions are shared between orthology-based maps. Possible causes are the use of different data sets and methods for prediction of interactions. Likewise, literature-based maps have only 10% - 28% of their interactions in common. This might result from inspection bias, such as the focus of HPRD towards disease-related genes. Notably, two earlier studies reported contradicting findings for the overlaps between HPRD and BIND. Whereas our study agrees well with results by Ramani and co-workers detecting an overlap of 25%, we cannot confirm the results by Gandhi and colleagues claiming that 85% of interactions in BIND were included in HPRD [Gandhi *et al.* '06, Ramani *et al.* '05]. Finally, a more worrying finding is the minute overlap of mere 1% between interactions in Y2H-based maps underlining the importance of stringent validation of high-throughput data.

In conclusion, this study is aimed to provide a first groundwork for future integration of large-scale human interaction maps [Chaurasia *et al.*]. As we saw, the combination of different maps can be expected to offer great assets. Nevertheless, researchers should be aware of the shortcomings of the underlying mapping approaches.

Acknowledgements

The work presented was supported by the *Deutsche Forschungsgemeinschaft (DFG)* by the SFB 618 grant.

Literature

Ashburner, M. et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* **25:** 25-29.
Bader, G.D. et al. 2001. BIND--The Biomolecular Interaction Network Database. *Nucleic Acids Res* **29:** 242-245.
Bader, G.D. and C.W. Hogue. 2002. Analyzing yeast protein-protein interaction data obtained from different sources. *Nat Biotechnol* **20:** 991-997.
Balasubramanian, R. et al. 2004. A graph-theoretic approach to testing associations between disparate sources of functional genomics data. *Bioinformatics* **20:** 3353-3362.
Benjamini, Y. and Y. Hochberg. 1995. Controlling the false discovery rate - a practical and powerful approach to multiple testing. *J Roy Stat Soc B* **57:** 289-300.
Bork, P. et al. 2004. Protein interaction networks from yeast to human. *Curr Opin Struct Biol* **14:** 292-299.

Brown, K.R. and I. Jurisica. 2005. Online predicted human interaction database. *Bioinformatics* **21:** 2076-2082.
Carey, V.J. et al. 2005. Network structures and algorithms in Bioconductor. *Bioinformatics* **21:** 135-136.
Chaurasia, G. et al. UniHI: An entry gate to the Human Protein Interactome. *submitted*.
Gavin, A.C. et al. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415:** 141-147.
Gentleman, R.C. et al. 2004. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol* **5:** R80.
Giot, L. et al. 2003. A protein interaction map of Drosophila melanogaster. *Science* **302:** 1727-1736.
Goehler, H. et al. 2004. A protein interaction network links GIT1, an enhancer of huntingtin aggregation, to Huntington's disease. *Mol Cell* **15:** 853-865.
Ito, T. et al. 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U S A* **98:** 4569-4574.
Jansen, R. et al. 2003. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* **302:** 449-453.
Lee, I. et al. 2004. A probabilistic functional network of yeast genes. *Science* **306:** 1555-1558.
Lehner, B. and A.G. Fraser. 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.
Peri, S. et al. 2003. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Res* **13:** 2363-2371.
Persico, M. et al. 2005. HomoMINT: an inferred human network based on orthology mapping of protein interactions discovered in model organisms. *BMC Bioinformatics* **6 Suppl 4:** S21.
Ramani, A.K. et al. 2005. Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biol* **6:** R40.
Rual, J.F. et al. 2005. Towards a proteome-scale map of the human protein-protein interaction network. *Nature* **437:** 1173-1178.
Schwikowski, B. et al. 2000. A network of protein-protein interactions in yeast. *Nat Biotechnol* **18:** 1257-1261.
Stelzl, U. et al. 2005. A human protein-protein interaction network: a resource for annotating the proteome. *Cell* **122:** 957-968.
Uetz, P. et al. 2000. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature* **403:** 623-627.
von Mering, C. et al. 2002. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* **417:** 399-403.

MPI-ClustDB: A fast String Matching Strategy utilizing Parallel Computing

Thomas Hamborg, Jürgen Kleffe

Institut für Molekularbiologie und Bioinformatik
Charité-Universitätsmedizin Berlin
Arnimallee 22
{thomas.hamborg,juergen.kleffe}@charite.de

Abstract: ClustDB is a tool for the identification of perfect matches in large sets of sequences. It is faster and can handle at least 8 times more data than VMATCH, the most memory efficient exact program currently available. Still ClustDB needs about four hours to compare all Human ESTs. We therefore present a distributed and parallel implementation of ClustDB to reduce the execution time. It uses a message-passing library called MPI and runs on distributed workstation clusters with significant runtime savings. MPI-ClustDB is written in ANSI C and freely available on request from the authors.

1 Introduction

Since many bioinformatics problems deal with the analysis of large amounts of data, parallel computing has proven to be an important tool to ensure computation capability and computation in reasonable time. In addition to traditional massively parallel computers, distributed workstation clusters play an increasing role in scientific computing. But so far, there had been little success in using distributed computing for large scale sequence matching. MUMMER [Ku04] and VMATCH [AKO04] are the most sophisticated programs implementing suffix tree and suffix array algorithms for simultaneous sequence matching. But due to their high memory usage these algorithms are not able to deal with datasets as large as necessary. Moreover both algorithms are not parallelizable for distributed memory architectures. Futamura et al. [FAK98] suggested an algorithm for parallel sorting of suffixes which performs a bucket sort of suffixes followed by parallel sorting of buckets. Still the algorithm requires large RAM by storing all sequences and suffix positions simultaneously. Another drawback is that the sorting of buckets cannot be performed using optimal algorithms. Hence, depending on the data, the method does not always improve overall computing time. Two other attempts of using massively parallel computation are the approaches by Iliopoulos et al. [IK02] and Kalyanaraman et al. [Ka03] which use far more memory than is available for practical input sizes. The latter publication estimates the demand of 512 parallel processors each with 512 MB RAM in order to compare five million human ESTs. In contrast ClustDB [KMW06] uses a new sorting algorithm named

partitioned suffix array method. It permits working with at least 8 times more data than VMATCH and MUMMER and offers several ways of efficient parallel computing. We therefore developed the program MPI-ClustDB that significantly reduces time consumption for a group of loosely coupled computers. This parallel approach allows to compare about six million human ESTs using only 7 personal computers each equipped with a 2.6 GHz Intel Pentium 4 CPU and 2 GB RAM. MPI-ClustDB is designed as a data-parallel approach where in certain parts of the program one has to cope with a variable number of identical tasks and each process executes more or less the same set of commands on its data (task). MPI, the de facto standard for distributed memory systems, is used for interprocess communication. It supports dynamic assignment of tasks to processes and has the advantage of running on several platforms without code alteration.

2 Algorithm

MPI-ClustDB is designed in a Master/Slave-manner where one process coordinates the scheduling and allocates tasks to a number of slave processes. It is assumed that all processes have access to the entire data. Therefore the master converts the input data into a fast accessable and space saving format called DNA_Stat database and distributes suchlike data across the slaves. The aim is the identification of all matching substrings of a certain minimal length in a large set of sequences which are derived by performing two computational steps called *Start Word Sort* and *Substring Detection*. We describe these steps together with the associated parallelization strategies in section 2.1 and 2.2. Subsequently the results are converted into a user-friedly output format. The parallelization of the conversion is described in section 2.3.

2.1 Start Word Sort

Let L be the number of nucleotides of the concatenated sequence formed from all considered sequences separated by a dot character. Conventional suffix array methods store and sort a vector of pointers of length L into the concatenated sequence. These pointers are called suffix positions and require at least four times more memory than needed to store the sequence. We therefore cut the vector of suffix positions into N pieces of length L/N which are processed one by one. A word length W between 3 and 10 is fixed and the pointers in each subvector are sorted in lexicographic order of the first W characters the suffixes begin with (Fig.1 - Step 1). Then each of the N sorted subvectors splits into blocks of suffix positions which start with the same word and are equally colored in Fig. 1. The parameter N is determined by the RAM size as each subvector has to fit into RAM for efficient sorting. In the parallel approach we choose N at least as large as the number of slave processes. The master process sends the appropriate sequence regions (two numbers) to the slaves and receives the sorted suffix positions. Each slave processor works in linear time $O(L/N)$ and requires $L/3 + 4(L/N + Z)$ bytes of RAM in order to store the

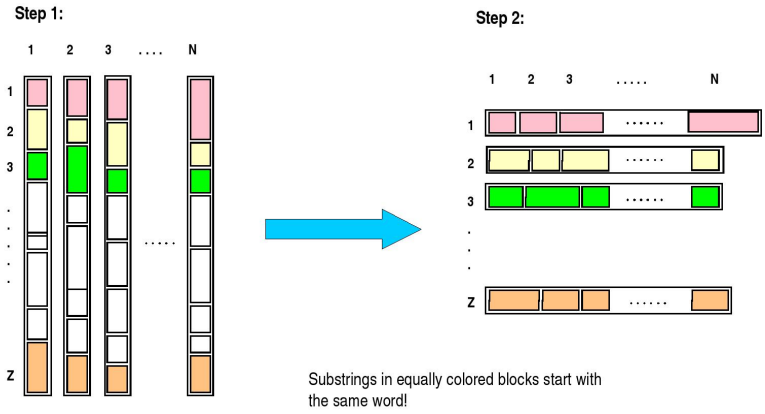


Figure 1: partitioned suffix array algorithm

complete sequence in compressed form (1 byte for 3 nucleotides), L/N suffix positions and $Z = 4^W$ different word counts. Let p be the number of available processors, then the total runtime for this step is roughly $O(L/p)$.

2.2 Substring Detection

In step 2 all blocks of suffix positions starting with the same word are collected from the N subvectors formed in step 1 and merged into Z new vectors displayed horizontally in Fig. 1 - Step 2. Z is the number of different words of length W. Each of these vectors is individually scanned for repeats of length M. Details about the calculations are given in [KMW06]. In case of parallel computing the master sends the Z vectors to the slaves and the slaves calculate the positions of multiple substrings which form clusters. Contrary to parallelization in step 1, dynamic allocation of tasks is indispensable here. Some words occur more frequently than others and hence the Z vectors differ greatly in length. But even if two start words occur with equal frequencies, the clusters originating from them will usually differ in size and so will the corresponding times of computation.
This step is performed in $O(M * L/(p * W))$ time requiring $L/3 + 8 * F$ bytes of RAM for each slave process where F is the maximum frequency of all considered start words. Assuming we have L bytes of RAM, we can use approximately $2 * L/3$ bytes for a table of size $8 * F$ that is necessary for the iterated suffix sort algorithm described in [KMW06], i.e. F can be as large as $L/12$. About every twelfth of all overlapping start words must be the same in order to cause failure of the algorithm with L bytes of RAM. In general F rapidly decreases by increasing word length.

2.3 Output Conversion

Subsequent to the partitioned suffix array algorithm each set of multiple substrings is represented by a cluster number and a set of global sequence positions in the concatenated sequence. These positions have to be turned into sequence numbers according to the succession in the input file and local sequence positions in the respective sequences. This task is carried out by means of a binary search algorithm. We use a scheduling strategy called *fixed-size chunking* [Ha97] here. A fixed amount of positions from substring cluster elements is sent to a slave, the sequence numbers and local positions are calculated and sent back to the master. The computation time for this part is $O(L * logS/p)$ where S denotes the number of sequences. Each processor requires $4 * S$ bytes of RAM in order to store the start positions of all individual sequences.

3 Implementation

MPI-ClustDB processes DNA-sequence data in the established formats Genbank, EMBL and FASTA or in DNA_Stat database format. The latter is an inhouse binary format that allows for fast direct access of individual sequences and playes a keyrole in fast data communication. It significantly reduces runtime especially if MPI-ClustDB repeatedly runs on the same data. Results of the substring calculation are presented in a tabular form with the three columns cluster number, sequence number and match position. It is possible to obtain the results in text file and/or DNA_Stat database format. Summary results are written to a seperate log file and several options of the program are described in [KMW06].
In order to execute MPI-ClustDB, an implementation of the Message Passing Interface communication protocol has to be installed. A large number of implementations is freely available. We have choosen the widely spread MPICH2 implementation that can be obtained from http://www-unix.mcs.anl.gov/mpi/mpich2/. As we make use of standard MPI commands only, it should be possible to link against any other MPI library, too. However, it is important to use buffered and blocking MPI send/receive functions in order to avoid deadlocks.

4 Results

We investigate the speedup of MPI-ClustDB compared to the serial ClustDB implementation for a 100 MBit/s and 1000 MBit/s ethernet network connection. If T_0 denotes the runtime of the serial solution and T_p denotes the runtime of the parallel solution with p processes, speedup is defined as $S_p = T_0/T_p$. All computations were performed on a test cluster consisting of seven standard personal computers. Each of them has a 2.6 GHz Intel Pentium 4 CPU and 2 GB of RAM running the operating system Mandriva Linux 2006.
We report application to the set of all 6,054,053 human ESTs stored in Genbank of date

	100 MBit/s		1000 MBit/s	
# slaves	complete runtime	speedup	complete runtime	speedup
0 (serial)	13360 sec	1	13360	1
2	7738 sec	1.73	6352	2.10
3	6264 sec	2.13	5011	2.57
4	5263 sec	2.54	4215	3.17
5	4785 sec	2.79	3741	3.57
6	4392 sec	3.04	3410	3.92

Table 1: Runtime and speedup for MPI-ClustDB results of detecting all common substrings of length $M = 50$ in all human ESTs considering two network velocities.

2005-04-06. The task is the identification of all common substrings of length 50 in the test set. The serial ClustDB programm needs a total of 3 hours and 42 minutes to solve the problem of detecting all 7,059,622 substring clusters of match length 50 for all human ESTs. Table 1 shows how the runtime of MPI-ClustDB alters for employing different numbers of CPUs. The complete runtime decreases for any addition of a CPU in the cluster leading to an overall runtime of 1 hours and 15 minutes for 7 personal computers and a 100 MBit/s network. Using the gigabit connection the runtime decreases to 56 minutes. Thus we are approximately four times faster with MPI-ClustDB than with the serial ClustDB software. Figure 2 analyses the reasons of the performance gains. The bisecting line presents the optimal speedup. Ideally parallel computing using p processors should be p times faster than the serial program. The left plot displays the achieved speedup for a 100 MBit/s network. Employing only one slave increases time of computation. But for at least two slaves we see a sound speedup for the parallelization of the Substring Detection step (square symbol). The other two parallel steps Start Word Sort (diamond) and Output Conversion (triangle) do not scale well. This results from an excessive amount of overhead that is due to communication among the processes. The amount of data that has to be distributed in these parts is comparatively large and the period for sending the data to another process is out of scale compared to the calculations performed on the data. The right plot in Fig. 2 shows the results for a 1000 MBit/s network connection. A significant speedup for each step is achieved resulting in a greater overall speedup. Compared to the slower network, the speedup for step 1 increases best while the speedup for step 2, that already scaled well for 100 Mbit/s, improves just slightly. The reason is that the time consumption for step 2 is mainly due to computation and not communication.
To account for diverse network velocities, our program optionally utilizes parallel computing for Substring Detection only (overall speedup 1) or with all three parts being parallelized (overall speedup 2). The two resulting speedups are displayed in Fig. 2. For the slower network connection overall speedup 1 is superior to overall speedup 2. Although step 1 and 3 scale poorly for the slower network, overall speedup 1 is just slightly better. This results from the fact that the Substring Detection step takes about 83% of the overall CPU time. By contrast omitting the parallelization of part 1 and 3 leads to a notedly larger runtime for the 1000 MBit/s ethernet.

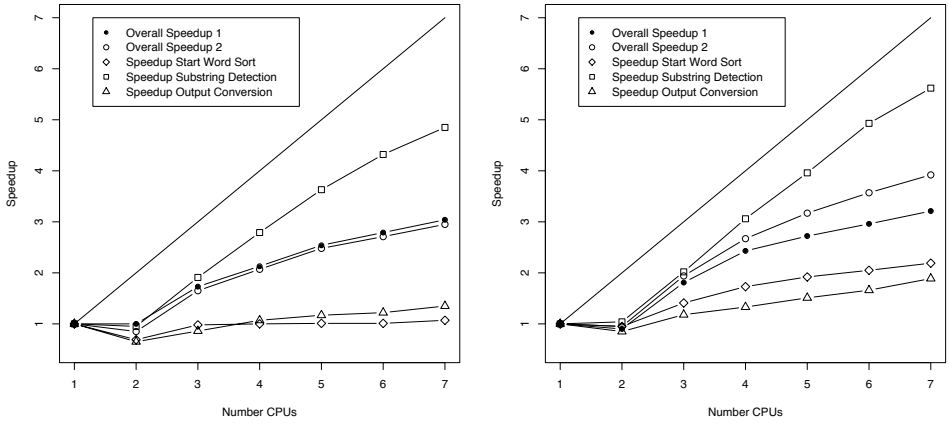


Figure 2: Speedup, as a function of the number of processors, for a 100 (left) and a 1000 (right) MBit/s network connection. The task is detecting all common substrings of length $M = 50$ in all human ESTs. Speedup 1 displays results for only Substring Detection being parallelized and speedup 2 displays results with parallel computation of all three algorithmic steps.

5 Discussion and Summary

Nowadays Bioinformatics in general and sequence comparison in particular is faced with very large datasets. ClustDB, our tool for finding common substrings in DNA-sequences, is able to work on a greater amount of data than the competing programs. Additionally we achieved to significantly reduce the runtime via our parallel implementation MPI-ClustDB using a relatively inexpensive PC cluster. We parallelized the three most time consuming parts of the programm, but for a 100 MBit/s network connection only one part shows a speedup and is therefore used in it‘s parallel implementation. Increasing the network speed to 1000 MBit/s yields significant speedups for all parallelized parts and cleary an ascending overall speedup.

The parallel computation of each of the three parts is a problem of allocating independant tasks to processors. The goal is to execute the tasks as quickly as possible. In the Output Conversion step the fixed size chunking strategie is used that is theoretically preferable compared to the others. But due to sundry constraints fixed size chunking is not applicable in steps one and two. For example in the first parallel part N could be enlarged to reduce processor idle time. But that would lead to a larger number of files to be read from in the next step and overall runtime was observed to increase. Nevertheless more sophisticated scheduling strategies may be possible and will be analysed in further developements.

We will investigate and optimize the scaleability of MPI-ClustDB next by running it with a greater number of processors. Furthermore we will try to parallelize additional parts of ClustDB. First aims are the calculation of sequence clusters (a subset of sequences having no substring of length M in common with any sequence outside the subset) derived from the substring clusters and extending pairs of matching substrings with errors. Based on

MPI-ClustDB a parallel solution for 64 bit shared memory systems is intended afterwards.

Acknowledgment

This project was supported by the BMBF Germany under contract number 0312705A. The authors would like to thank Friedrich Möller for technical assistance.

References

[AKO04] M.I. Abouelhoda, S. Kurtz, E. Ohlebusch. "Replacing Suffix Trees with Enhanced Suffix Arrays" Journal of Diskrete Algorithms, No. 2, 53-86, 2004.

[De02] A.L. Delcher, A. Philippy, J. Carlton, S.L. Salzberg. "Fast algorithms for large scale genome alignment and comparison", Nucleic. Acids Research, Vol. 30, No. 11, 2002.

[FAK98] N. Futamura, S. Alura, S. Kurtz. "Parallel Suffix Sorting", Proc. 9th International Conference on Advanced Computing and Communications, 76-81, 2001.

[GSN98] W. Gropp, M. Snir, B. Nitzberg, "MPI: The Complete Reference", 2nd edn, MIT Press, Cambridge, MA, 1998.

[GTL01] W. Gropp, R. Thakur, E. Lusk. "Using MPI-2", MIT Press, Cambridge, MA, 2001.

[Ha97] T. Hagerup. "Allocating independent tasks to parallel processors: an experimental study", J. Parallel Comput., Vol. 47, 185-197, 1997.

[IK02] C. S. Iliopoulos, M. Korda. "Massively Parallel Suffix Array Construction", Proc. 25th Conference on Current Trends in Theory and Practice of Informatics, 371 - 380, 1998.

[Ka03] A. Kalyanaraman, S. Alura, S. Kothari, V. Brendel. "Efficient clustering of large EST data sets on parallel computers", Nucleic Acid Research, Vol. 31, No. 11, 2963-2974, 2003.

[KMW06] J. Kleffe, F. Möller, B. Wittig. "ClustDB: A high performance tool for large scale sequence matching", Proceedings DEXA 2006.

[Ku04] S. Kurtz, A. Philippy, A.L. Delcher, M. Smoot, M. Shumway, C. Antonescu, S.L. Salzberg. "Versantile and Open Software for Comparing Large Genomes", Genome Biology, 5 (R12), 2004.

Ab Initio Prediction of Molecular Fragments from Tandem Mass Spectrometry Data

Markus Heinonen[a,*] Ari Rantanen[a] Taneli Mielikäinen[a] Esa Pitkänen[a]
Juha Kokkonen[b] Juho Rousu[a]

[a] Department of Computer Science, University of Helsinki, Finland
[b] VTT, Technical Research Centre of Finland

Abstract: Mass spectrometry is one of the key enabling measurement technologies for systems biology, due to its ability to quantify molecules in small concentrations. Tandem mass spectrometers tackle the main shortcoming of mass spectrometry, the fact that molecules with an equal mass-to-charge ratio are not separated. In tandem mass spectrometer molecules can be fragmented and the intensities of these fragments measured as well. However, this creates a need for methods for identifying the generated fragments.

In this paper, we introduce a novel combinatorial approach for predicting the structure of molecular fragments that first enumerates all possible fragment candidates and then ranks them according the cost of cleaving a fragment from a molecule. Unlike many existing methods, our method does not rely on hand-coded fragmentation rule databases. Our method is able to predict the correct fragmentation of small-to-medium sized molecules with high accuracy.

1 Introduction

One of the enabling measurement technologies for the new era of systems biology is mass spectrometry (MS). Mass spectrometer measures the abundances of molecules with different masses in the sample with very high precision [MZSL98]. Mass spectrometry has an integral role in many biological analysis tasks, such as in protein identification [GV00, HZM00, Swe03]. In the study of metabolism mass spectrometry can be used to identify intracellular small molecules by comparing the intensity spectrum of unknown metabolite to a spectra residing in reference library [Fie02, MZSL98, SS94]

More information about an unknown metabolite can be obtained by applying *tandem mass spectrometer* (also known as *MS/MS*) techniques where metabolite molecules are collided with e.g. neutral gas to fragment the molecules and also the abundances of fragments are measured [dH96]. For example, the product ion spectrum produced by tandem MS can be used to improve the accuracy of library-based identification of unknown metabolites [Fie02, JS04] and to deduce structural information about them [KPH$^+$03, SP99,

*Author to whom correspondence should be directed. E-mail: markus.heinonen@cs.Helsinki.FI

vRLDZ$^+$04]. In addition, the elemental composition of a metabolite can be accurately inferred from product ion spectrum [ZGC$^+$05].

Tandem MS has also great potential in the area of ^{13}C metabolic flux analysis [RMR$^+$06, SCNV97, WMPdG01] where the velocities of metabolic reactions are estimated from the isotopomer distributions[1] of the metabolites. The isotopomer distribution of a metabolite can be accurately derived from tandem MS data [CN99, RRKK05]. Before the isotopomer distribution of a metabolite can be computed, the exact structures of molecular fragments produced by tandem MS have to be identified. The identification of fragments produced by tandem MS is also a problem of interest in e.g. structural elucidation [Swe03].

The manual identification of molecular fragments is a very time-consuming process even for an expert [McL80]. In this article we propose a novel method for the identification of molecular fragments produced by tandem MS from a known parent molecule. In the existing commercial tools Mass Frontier [Hig05] and MS Fragmenter [ACD05, Wil02] fragment identification is based on the fragmentation rules stored into a database. However, small changes in the structure of a molecule can result in significant differences in the fragmentation process [McL80]. Rule based systems will err if the fragmentation of a new molecule does not follow the rules found by studying other kinds of molecules. Deduction of fragmentation rules for each molecule and for each different MS technique is also a laborious task.

Our approach for tandem MS fragment identification is not based on a prior knowledge about common fragmentation rules but on the utilization of the combinatorial structure of the problem. Shortly, we first generate candidate fragments whose masses correspond to the observed peaks in a product ion spectrum and rank the candidate fragments according to the cost of cleaving a fragment from a molecule. Our experiments indicate that when molecules are reasonably small and the masses of molecular fragments can be measured with accuracy characteristic to modern high resolution MS devices, tandem MS fragments can be identified with good precision without a priori knowledge about common fragmentation mechanisms.

2 Fragment identification problem

Molecules can be modeled as undirected, connected, weighted and labeled graphs with the vertices being the atoms of the molecules and edges the bonds:

Definition 2.1 (Molecule)**.** A molecule M is an undirected, connected, weighted and labeled graph $\langle V, E, t_V, t_E, w_V, w_E \rangle$, where V is the set of vertices corresponding to the atoms and E is the set of undirected edges corresponding to the bonds between the atoms. The function $t_V : V \to A$ assigns each atom a type (e.g., carbon, hydrogen, etc.) and $t_E : E \to B$ assigns each bond a type (e.g., single, double, triple, aromatic, etc.). Vertices have atomic weights $w_V : V \to \mathbb{R}_+$ and edges have values $w_E : E \to \mathbb{R}_+$ assigning each

[1] By different isotopomers of a metabolite we mean molecules having specific combination of ^{12}C and ^{13}C atoms in different positions of the carbon chain of the metabolite. Isotopomer distribution of the metabolite then gives the relative concentrations of different isotopomers.

edge the strength of the corresponding bond.

The mass of the molecule is the sum of the weights of its atoms, i.e.,

$$w(M) = \sum_{v \in V} w_V(v). \tag{1}$$

We define a *fragment* F of M as a connected subgraph of M.[2]

The output of tandem MS is a spectrum where the locations of peaks correspond to observed weights $W \subset \mathbb{R}_+$ of molecular and fragment ions.[3] On a high level, the fragment identification problem of a molecule M can be formulated as follows:

Problem 2.2. Given a molecule M and a set $W \subset \mathbb{R}_+$ of observed weights of fragments of the molecule, find fragments $F_1, \ldots, F_{|W|}$ of M that most likely correspond to the weights in W.

Formally, a molecule M induces a fragmentation graph G_M containing all fragments of M (see Figure 1 for an example):

Definition 2.3 (Fragmentation graph)**.** A fragmentation graph G_M for a molecule M is a directed acyclic graph $\langle \mathcal{F}, \prec, c \rangle$ where

- $\mathcal{F}$ is the set of nodes corresponding to the fragments of the molecule M, i.e., the subsets of edges in M. That is, $\mathcal{F}$ is the collection of sets $E' \subseteq E$ such that E' forms a connected component in the molecule M;
- $\prec$ is the set of directed edges from each fragment $F \in \mathcal{F}$ to its subfragments $F' \in \mathcal{F}$. Hence, $\prec$ is binary relation over $\mathcal{F}$ such that $F \prec F' \iff F' \subset F$ for all $F, F' \in \mathcal{F}$;
- $c : \prec \rightarrow \mathbb{R}_+$ associates a cost to each edge in the graph giving the cost of producing the fragment F' from the fragment F for each $\langle F, F' \rangle \in \prec$ (i.e., for each $F' \subset F \subseteq E$ where F and F' form connected components).

We use several heuristic cost functions for producing the fragment F' from the fragment F. All functions are based on the assumption that, during the fragmentation process, weak bonds between the atoms of a molecule are more likely to be cleaved than the stronger ones. We approximate the strength of a bond with the standard covalent bond energy.

The simplest cost function for producing F' from F is the sum of energies of all cleaved bonds:

$$c(F, F') = \sum_{C_{F,F'}} w_E(e) \tag{2}$$

[2] Although not all fragments produced by tandem MS are necessary connected subgraphs, the assumption holds quite often. See Section 5 for further discussion.

[3] More precisely, MS separates molecular and fragment ions according to their mass-to-charge (m/z) ratio. However, when analyzing small molecules like metabolites, ions almost always get a single charge. In the following we assume that ions have a single charge.

where $C_{F,F'}$ consists of the bonds that must be cleaved to cut F' from F, i.e., $C_{F,F'} = \left\{e \in F : \left|e \cap \bigcup_{e' \in F'} e'\right| = 1\right\}$.

The total cost of a fragmentation graph G_M is the sum of the costs of its edges:

$$c(G_M) = \sum_{F \prec F'} c(F, F'). \tag{3}$$

With the notion of the fragmentation graph, the task of finding the best fragmentation for a molecule M and the weight set W can be formulated as follows:

Problem 2.4 (Fragment identification). Given a molecule M and a set $W \subset \mathbb{R}_+$ of weights, find a connected subgraph G^*_M of the fragmentation graph G_M such that G^*_M contains at least one fragment for each weight in W and the total cost $c(G^*_M)$ is minimized.

The actual form of the problem relies strongly on the cost function for the fragmentation graphs. We discuss different ways of defining the cost functions and fragmentation models in more detail in Section 3.

3 Models for the fragmentation process

The fragmentation of a molecule in tandem MS is a complex, stochastic and multistep process where ions are decomposed to smaller fragments. In general there exists many competing fragmentation pathways which a single molecule can take. The likelihood of the competing fragmentation pathways depends on many factors, including the amount of internal energy an ion obtains during the fragmentation, the stability of a product ion, steric requirements of fragmentation pathways and charge or radical sites of parent ion [McL80]. The accurate modeling of all these factors is very tedious [RHO00, SHS01] and is not done in practice when fragments are identified in every day laboratory work.

Next we give two alternative models for fragmentation and define the cost $c(G'_M)$ for a connected subgraph $G'_M \subseteq G_M$ of molecule M according to these models.

3.1 Single step fragmentation

Our primary model for fragmentation is based on the consensus that in tandem MS usually weak bonds are cleaved [MFH$^+$99] and that with low collision energies fragments are usually cleaved directly from the parent molecule [dH96]. Thus we can best explain the detected fragment peaks by fragments that can be cleaved from a parent molecule using the smallest amount of energy possible. With the notion of fragmentation graph, *single step fragmentation model* leads to a star-shaped graph, where each fragment originates directly from the original molecular ion in a single reaction. (See Figure 1.)

Unfortunately, even finding one weight-w minimum cost fragment F of a molecule M

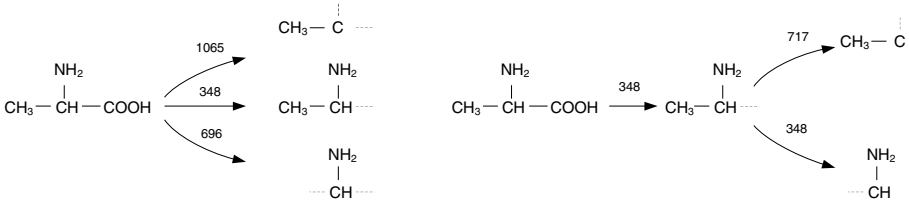


Figure 1: Example of fragmentation graphs of Alanine using single step (left) and multistep fragmentation model (right). Both graphs have four fragment nodes. Fragmentation is indicated by arrows with accompanying weights corresponding to the fragmentation graph edge weights, i.e. sum of cleaved bonds energies. For example, on the left the alanine is fragmented into CH3N (bottom arrow), by two cleavages: the COOH-group with C-C cleavage and the CH3-group with C-C cleavage. Both have energetic value of 348 kJ/mol thus making the total cost of producing CH3N 696 kJ/mol. Dashed arrows indicate cleaved bonds.

for certain weight $w \in \mathbb{R}_+$ is NP-hard. We show that by a polynomial-time computable reduction from the 3-satisfiability problem that is known to be NP-complete [GJ79]:

Problem 3.1 (3-satisfiability). Given a set $U = \{x_1, \ldots, x_n\}$ of boolean variables x_i and a collection $C = \{c_1, \ldots, c_m\}$ of clauses $c_i, |c_i| = 3$, over U, decide whether of not there is a a satisfying truth value assignment for C.

Theorem 3.2. *Given the molecule M, a weight $w \in \mathbb{R}_+$ and a cost c, it is NP-complete to decide whether or not there is a fragment F of M of the weight w with the cost $c(M, F)$ being at most c, where the cost is defined by Equation 2.*

Proof. The problem is clearly in NP, since any subgraph of M is at most as large as M itself and the weight (as defined by Equation 2) of any fragment F of M can be computed in time linear in F.

We reduce the instances $\langle U, C\rangle$ of 3-satisfiability to the instances $\langle M, w, c\rangle$ of finding a fragment of M of weight w and cost at most c.

The set V consists of vertices $c_{i,1}$, $c_{i,2}$, and $c_{i,3}$ for each clause $c_i \in C$, and dummy vertices $d_1, \ldots, d_\delta$. (δ is a constant that shall be determined later.) The vertex $c_{i,k}$ corresponds to setting the truth value of the boolean variable of the kth literal of the clause c_i in such a way that the literal is satisfied.

The weights of the atoms in the molecule are determined as follows. Let $p_1, \ldots, p_{n+1}$ be distinct primes. $w_V(c_{i,k}) = \log p_i$ for each $c_{i,k} \in V$, $w_V(d_j) = \log p_{n+1}$ for each $j = 1, \ldots, n$, and $w = \sum_{i=1}^{n} w_V(c_{i,1})$. Hence, any fragment consisting one vertex for each clause c_i.

Now we need to define the set E of edges and their weights appropriately. There is an edge $\{c_{i,k}, c_{j,l}\}$ in E if and only if kth literal in the clause c_i is not the negation of the lth literal of the clause c_j. Let δ be maximum degree of a vertex in the subgraph induced by the vertices $v_{i,k}$. Each vertex $c_{i,k}$ is connected to so many dummy vertices that the degree of $c_{i,k}$ is δ. The weights of the edges are all one.

The clauses in C are satisfiable if and only if there is a fragment F of weight w with cost $c(M, F) = n(\delta - n + 1)$. To see that, notice that fragment F is a clique if and only if the vertices in F determine a (partial) satisfying truth value assignment for C. □

Fortunately, in practice all fragments of the molecule M can be often generated and computing the cost $c(M, F)$ for a given F is easy. Thus, by generating all fragments (with weights in W) we can solve the problem. This observation leads to a conceptually simple algorithm where for each observed weight $w \in W$ a fragment F of weight w that minimizes $c(M, F)$ is found. The algorithm has three steps for each weight $w_i \in W$:

1. Find a set $\mathcal{F}_i$ of all connected subgraphs of M that have a weight w_i.
2. For each fragment $F \in \mathcal{F}_i$, compute a cost $c(M, F)$ of cleaving F from M.
3. For each $\mathcal{F}_i$, return $F_i \in \mathcal{F}_i$ with the smallest cost among the fragments in $\mathcal{F}_i$.

We find sets $\mathcal{F}_i$ of all fragments of weight w_i by enumerating all fragments, that is, all connected subgraphs induced by M with a depth-first traversal algorithm briefly mentioned in [BV97] and elaborated in [RR00]. The algorithm can easily be modified to give k_i least expensive fragments for each observed weight or all fragments with minimum cost w_i.

In our experiments, the cost $c(M, F)$ was based on five key figures derived from the bonds of M that have to be cleaved to form F from M, that is, bonds that connects elements in F to elements in $M \setminus F$. The key figures are: (1) the number of cleaved bonds, (2) the sum of strengths of cleaved bonds, (3) the strength of strongest cleaved bond, (4) the average strength of cleaved bonds and (5) the difference of strength between strongest intact bond versus the weakest cleaved bond in our candidate fragment. We defined $c(M, F)$ to be an average rank of F according to these key figures among the fragments of same weight.

3.2 Multistep fragmentation

As an alternative to single step fragmentation model, we experimented with a model where we assume that many fragmentation pathways consist of two or more consecutive reactions. Consecutive fragmentation reactions are thought to be common when higher collision energies are applied [dH96]. In this *multistep fragmentation model* we also assume that in intermediate reaction steps of a fragmentation pathway usually not all molecular fragments are further cleaved but some proportion of them is observed as a peak in tandem MS spectrum. These assumptions allow us to construct a model where pathways of consecutive reaction steps that (1) explain observed fragment peaks by intermediates of the pathway and (2) that cleave only weak bonds, are favored. This approach can be thought to mimic the decision process an expert goes through while identifying fragments manually: a proposed fragmentation pathway is more likely correct if peaks matching to intermediate steps of the pathway are present in the spectrum [SHS01].

Multistep fragmentation process can be computationally modeled by allowing fragmentation graphs where fragments are cleaved from other fragments and defining the cost of a

fragmentation subgraph $G'_M = \langle \mathcal{F}', \prec', c \rangle$ to be the sum of the costs of edges in G'_M, i.e.,

$$c(G'_M) = \sum_{e \in \prec'} c(e). \tag{4}$$

We use the sum of all cleaved bonds energies (see Equation 2) as the cost of an edge.

In the multistep fragmentation model the cost of fragment F depends on the other fragments in the fragmentation subgraph while in the single step fragmentation model, where fragments are always cleaved directly from the parent molecule, the cost of F depended only of its own structure. Thus instead of ranking the fragment of observed weight by comparing it to the other fragments of equal weight, we search for the optimal fragmentation subgraph G^*_M that minimizes the cost given in Equation 4.

Proposition 3.1. *The minimum cost connected subgraph G^*_M of the fragmentation graph G_M of a molecule M is a tree with at most $|W|$ leaves, where the cost of G^*_M is defined by Equation 4.*

Proof. Let G^*_M be the minimum cost connected subgraph of G_M. To see that G^*_M is necessarily tree, assume that G^*_M is not a tree.

If G^*_M is not a tree, then there must be a cycle C in G^*_M. However, then also the graph $G^*_M \setminus \{e\}, e \in C$, is connected. As the costs of the edges in G^*_M are strictly positive, the cost of $G^*_M \setminus \{e\}$ strictly smaller than the cost of G^*_M. Thus, if G^*_M is not a tree, then it is not the minimum cost connected subgraph of G_M.

The number of leaves can be at most W, since each leaf corresponds to some weight in W. □

An optimal fragmentation subgraph G^*_M can be found from the fragmentation graph G_M with mixed integer linear programming (MILP) by formulating the problem as a mixed integer linear program. (There exist well-developed techniques for solving MILP reasonably fast in practice [Mar01].)

The MILP formulation of the problem is as follows. We partition the fragments whose weight correspond to observed weights into sets $L_1, \ldots, L_{|W|}$ according to their weights. We denote by $\mathcal{L}$ a collection of sets L_k. Let f_i be a binary variable indicating whether a fragment $F_i \in L_k$ is chosen to be a fragment corresponding an observed weight w_k. We set $f_M = 1$ for the whole molecule. Let binary variable $p_{i,j}$ indicate whether an edge from F_i to F_j in G_M is chosen to G^*_M and $c_{i,j} \in \mathbb{R}$ the cost of $F_i \prec F_j$. The function to be minimized corresponds to the total cost of edges of G_M that are selected to G^*_M (see Equation 4).

We then obtain the following integer linear program:

$$\begin{aligned}
\min & \sum_{F_i \prec F_j} c_{i,j} p_{i,j} & \\
\text{s.t.} & \sum_{f_i \in L_k} f_i = 1 & \forall L_k \in \mathcal{L} \\
& f_j - \sum_{F_i \prec F_j} p_{i,j} = 0 & \forall F_j \in \mathcal{F} \\
& p_{i,j} - f_i \leq 0 & \forall F_i \prec F_j \in G_M
\end{aligned}$$

The first constraint of the above program states that exactly one fragment from each observed weight needs to be selected. The second constraint states that for each selected fragment F_j exactly one parent fragment F_i, from which F_j is cleaved, have to be selected. The third constraint states that if $F_i \prec F_j$ is selected to G_M^*, also F_j have to be in G_M^*. The solution to the above program is a minimal cost set G_M^* of pathways which form a connected tree in the fragmentation graph and cover each weight class of fragments with exactly one fragment. Note that either all f_i's or $p_{i,j}$'s can be relaxed to be real-valued (in the interval $[0, 1]$) in order to speed up the optimization. We relax $p_{i,j}$ as the number of $p_{i,j}$'s is quadratic to the number of f_i's in the worst case.

In practice the mixed integer linear programs tend to be very large. A major optimization for the model is to notice the specialty of hydrogen atoms in the fragments. As hydrogens connect to at most one other element, their removal from the model do not split a molecule or fragment to two fragments. Thus hydrogens do not need to be included when all fragments are enumerated. By using hydrogen-suppressed fragments, the amount of fragments drops drastically.

To cover the loss of hydrogen specificity in fragments, we add variables and constraints to integer linear program requiring that the correct number of hydrogens is cleaved from each selected fragment and that the cleaved hydrogen of parent fragment in G_M^* stays cleaved in its daughter fragments. Also, the objective function is modified such that the costs of hydrogen cleavages are correctly accounted for.

Let $h_{n,j}$ be a binary variable indicating whether a hydrogen n directly connected to fragment F_j is cleaved. Let $\mathcal{H}$ be the set of all hydrogens in M and $|H_i|$ the (precomputed) number of hydrogens connected to F_i that should be cleaved in order to obtain F_i.

We add to MILP a constraint to ensure that the correct amount of hydrogens will be chosen for the fragment:

$$\sum_{n \in \mathcal{H}} h_{n,i} - |H_i| \, f_i = 0 \qquad \forall F_i \in \mathcal{F}.$$

We also add a constraint ensuring that a hydrogen cleaved in F_i is cleaved in all F_j's that have selected to be its children in the solution:

$$p_{i,j} + h_{n,i} - h_{n,j} \leq 1 \qquad \forall F_i \prec F_j, \forall n \in \mathcal{H}.$$

Finally, the cost of the solution is modified to take the costs of cleaved hydrogens into account:

$$\min \sum_{F_i \prec F_j} c_{i,j} p_{i,j} + \sum_{h \in \mathcal{H}} \sum_{F_i \prec F_j} c_h (h_{n,j} - h_{n,i}).$$

Again, the variables $p_{i,j}$ can be relaxed to be in $[0, 1]$.

4 Experiments

We tested our method of identifying tandem MS fragments with 20 amino acids and 7 sugar phosphates. Molecular masses ranged from 75 Da to 340 Da, 160 Da being the average. In particular, the most massive molecule Fructose-1,6-bisphosphate had 34 atoms and 34 bonds. Out of the 27 molecules, 8 were cyclic. The number of connected subgraphs of the molecules varied from hundreds to millions, depending on the cyclicity and size of the molecules. The run times of the above algorithms for candidate fragment enumeration and ranking varied accordingly from seconds to days.

Compounds were fragmented with the collision-induced dissociation (CID) method by using a Micromass Quattro II triple quadrupole MS equipped with an electrospray ionization interface. The spectra of compound were measured in a positive ionization mode. The collision gas for CID fragmentation was argon and collision energies varied between $10 - 50$ eV. The number of peaks in the product ion spectra of the molecules varied from one to 15, average being 7.1 peaks/molecule. Domain experts first manually identified the fragmentation pathways for each of the 27 molecules and the weights of the manually identified fragments were calculated with high precision for comparison of the effect of measurement accuracy to fragment identification. We then predicted the fragments with both of our models and compared the results against the manually identified fragments. A predicted fragment was deemed correct if its chemical formula and carbon backbone matched the manually identified one as this level of accuracy is sufficient for applications such as ^{13}C metabolic flux analysis. We used the off-the-shelf MILP solver `lp_solve` [BEN05] to solve the MILPs introduced in Section 3.2.

Our methods for identifying fragments agreed well with the domain experts when atom weights of peaks were assumed to be measurable at 0.01 Da (mass) accuracy. This is a realistic assumption in the current high resolution mass spectrometers and in our dataset. In high accuracy there were 6.5 fragments for each peak in fragment spectra, on average ($\sigma = 9.8$). If the fragments corresponding to observed peaks were selected randomly from the sets of fragments with the lowest cost suggested by the single step fragmentation method (Section 3.1), the fragmentations of the metabolites would be 88.7% correct, on average. If the best fragment among the fragments with the lowest cost was selected for each peak, metabolites would get 90.8% of correct fragments, on the average. On average, there were 1.4 fragments with the equal lowest cost per peak ($\sigma = 0.9$).

With the multistep fragmentation method (Section 3.2) fragmentation subgraphs with the lowest cost consisted of 82.8% correct fragments, on the average. The fragmentation subgraph in best agreement with manual identification among the subgraphs whose cost

was among the top-3 costs consisted 93.8% of correct fragments, on average. (There were 17.0 subgraphs in top-3 cost classes.)

In comparison, randomly constructed fragmentation subgraph of fragments whose weight match with observed peaks would have 36.8% ($\sigma = 36.3$) of correct fragments, on average.

If we assume that the mass spectrometer can separate compounds only at integer accuracy, the number of fragments with the same weight is considerably larger, namely 19.3 versus 6.5 fragments/peak on the average. This makes combinatorial identification of fragments much harder. With integer accuracy and single step model the fragmentations of the metabolites would be 66.4% correct on average, if the fragments corresponding peaks were selected randomly from the sets of fragments with lowest cost. Again, there were for each observed peak 1.4 fragments that had the lowest cost, on average. With multistep model the fragmentation subgraphs with the lowest cost yield an average accuracy of 55.9% and with the best subgraph among the subgraphs with top three lowest cost an average accuracy of 70.7%. (There were 25.7 subgraphs in the three lower cost classes on average.) Randomly constructed fragmentation subgraph of fragments that have an observed weight, has an average accuracy of 12.3% ($\sigma = 9.9$).

Figure 2 and Table 1 summarize the results of the experiments. Table 1 shows the prediction accuracies of fragmentation subgraphs with the lowest costs. In Figure 2, prediction accuracies of fragmentation subgraphs that had the cost among k lowest costs are shown. For example, with high mass accuracy and the single step model and examining the best fragmentation subgraphs with the cost in $k = 3$ lowest cost classes for each peak, 94.6% of predicted fragments match the manually identified ones. The reported accuracies are averages over 27 metabolites.

As a conclusion, most of the molecules can be resolved without difficulties and near 90% prediction rates are achieved, when high resolution MS is available. With our dataset the single step fragmentation model gives more accurate prediction than the multistep model.

Table 1: Single step and multistep model accuracies with integer and high mass (0.01 Da) accuracy. The best, the worst and the average accuracies of the fragmentation subgraphs that had the lowest cost according to single step or multistep models are shown. Reported accuracies are averages over 27 metabolites.

Scheme	Best	Average	Worst	σ_B	σ_A	σ_W
Single step, integer	68.2%	66.4%	64.5%	19.2%	21.2%	23.9%
Single step, high	90.8%	88.7%	86.3%	11.3%	12.0%	14.3%
Multistep, integer	62.0%	55.9%	51.1%	22.4%	23.5%	26.1%
Multistep, high	87.0%	82.8%	78.0%	20.4%	21.6%	24.8%

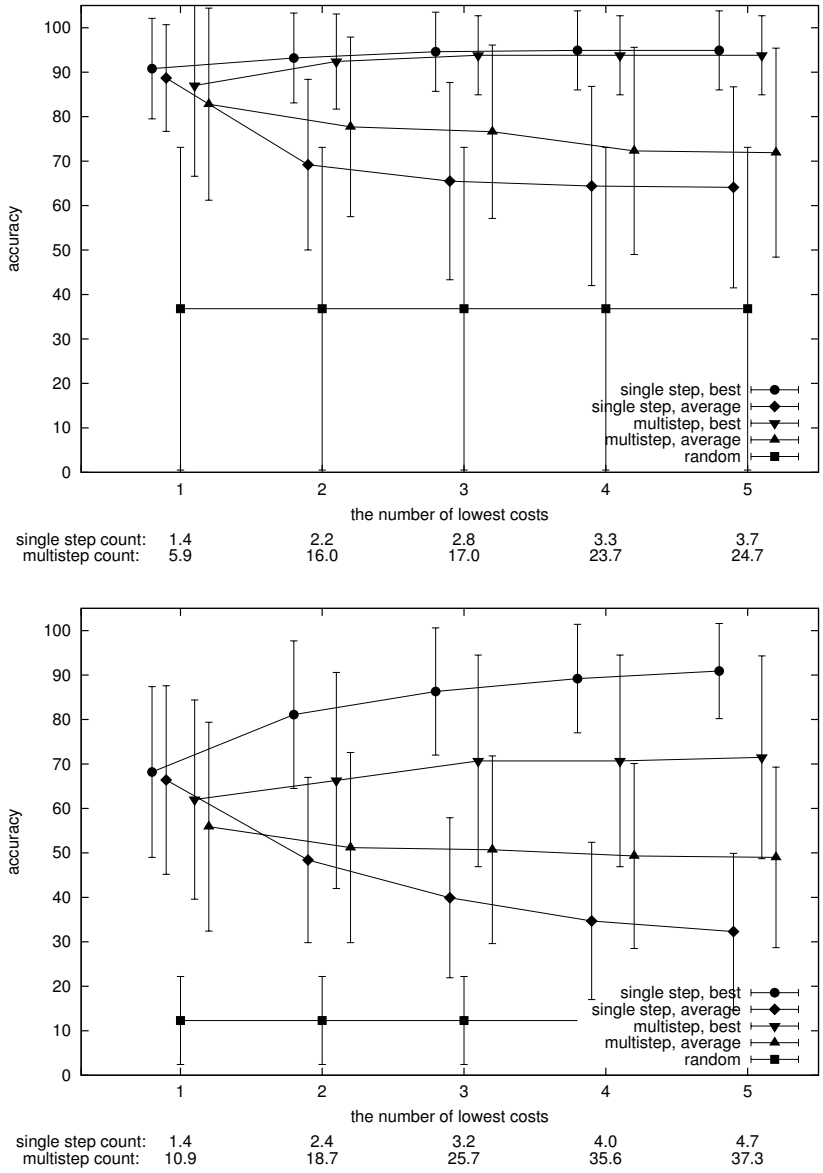


Figure 2: Figures depict the accuracy of single step and multistep fragmentation models when fragmentation subgraphs whose cost was among k lowest costs (x-axis) were taken into account. On top, fragment weights are assumed to be measurable with 0.01 Da accuracy, on bottom with integer accuracy. *Single step count* and *multistep count* below x-axes show the cumulative number of fragments per peak (single step) and the cumulative number of fragmentation subgraphs (multistep) with the cost among k lowest costs. The reported accuracies are averages over 27 metabolites. The lines connecting the points are only to improve the readability of the figures.

5 Discussion

The fragmentation of a molecule in mass spectrometer is a complex process which is not fully understood. We have shown that a combinatorial approach gives good results when the molecules analyzed are sufficiently small and the resolution of the mass spectrometer is characteristic to modern mass spectrometers. The combinatorial method given above automatically generates good hypotheses of the fragmentation patterns, thus aiding an experimentalist to evaluate all relevant possibilities of the fragmentation. Furthermore, our approach does not make assumptions on the MS technique used and is thus potentially applicable to a wide variety of problems.

The number of connected subgraphs of a molecule graph easily explodes when the size of the graph grows, even if hydrogen atoms are disregarded. Thus, the applicability of the combinatorial method is limited to small or medium–sized molecules. The number of connected subgraphs depends heavily on the cyclicity of the graph. As a rule of thumb, the method requires that the size of the molecule does not exceed 50 atoms, excluding hydrogens. Thus the method is suitable for many metabolites, but unsuitable for proteins. Additionally, as a result of element rearrangements, that is, by formation of new bonds during the fragmentation [MZSL98], not all fragments are necessarily connected subgraphs of the parent molecule. Fortunately, the most common example of such bond formation is hydrogen rearrangement. Again, hydrogen rearrangements can be handled as special cases as hydrogen atoms can only be transferred from one position to another, not creating cycles. For more complex rearrangements involving cyclizations, our software implementation of the above methods provides the user a tool to manually add bonds that are formed during the fragmentation to the molecule. Comparing our method against the commercial rule based systems proved problematic. To the authors knowledge, no public data on the performance or accuracy of existing tools is available.

Taking advantage of fragment intensities provides an interesting direction for further development of our combinatorial fragment identification method. In addition, we are investigating the possibility of combining the combinatorial approach with stochastic modeling to improve the accuracy of identification. Also combining the local ranking heuristics in a more advanced way than computing the average rankings is a promising direction [FISS03, FKM+04]. The software implementing the methods described in this paper is available from the authors and from a web site `http://www.cs.helsinki.fi/group/sysfys/software/fragid/`.

Acknowledgments. This work was supported by grant 203668 from the Academy of Finland (SYSBIO program) and by European Commission IST programme FET arm, contract no. FP6-508861 (APrIL II).

References

[ACD05] ACD/Labs. ACD/MS Fragmenter. `http://www.acdlabs.com`, 2005.

[BEN05] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. lp_solve: Open source (Mixed-Integer) Linear Programming system., 2005. Multi-platform, pure ANSI C / POSIX source code, Lex/Yacc based parsing. Version 5.5.0.0 dated 17 may 2005. GNU LGPL (Lesser General Public Licence). http://groups.yahoo.com/group/lp_solve/.

[BV97] Richard G. A. Bone and Hugo O. Villar. Exhaustive Enumeration of Molecular Substructures. *Journal of Computational Chemistry*, 18(1):86–107, 1997.

[CN99] Bjarke Christensen and Jens Nielsen. Isotopomer analysis using GC-MS. *Metabolic Engineering*, 1:E6–E16, 1999.

[dH96] Edmond de Hoffmann. Tandem Mass Spectrometry: a Primer. *Journal of Mass Spectrometry*, 31:129–137, 1996.

[Fie02] Oliver Fiehn. Metabolomics – the link between genotypes and phenotypes. *Plant Molecular Biology*, 48:155–171, 2002.

[FISS03] Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An Efficient Boosting Algorithm for Combining Preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[FKM+04] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing and Aggregating Rankings with Ties. In Alin Deutsch, editor, *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 14-16, 2004, Paris, France*, pages 47–58, 2004.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[GV00] Kris Gevaert and Joël Vandekerckhove. Protein identification methods in proteomics. *Electrophoresis*, 21:1145–1154, 2000.

[Hig05] HighChem. HighChem Mass Frontier 4.0. `http://www.highchem.com`, 2005.

[HZM00] David Horn, Roman Zubarev, and Fred McLafferty. Automated de novo sequencing of proteins by tandem high-resolution mass spectrometry. *Proceeding of the National Academy of Sciences*, 97(19):10313–10317, 2000.

[JS04] Jonathan Josephs and Mark Sanders. Creation and comparison of MS/MS spectral libraries using quadrupole ion trap and triple-quadrupole mass spectrometers. *Rapid Communications in Mass Spectrometry*, 18:743–759, 2004.

[KPH+03] Katerina Klagkou, Frank Pullen, Mark Harrison, Andy Organ, Alistair Firth, and John Langley. Approaches towards the automated interpretation and prediction of electrospray tandem mass spectra of non-peptidic combinatorial compounds. *Rapid Communications in Mass Spectrometry*, 17:1163–1168, 2003.

[Mar01] Alexander Martin. General Mixed Integer Programming: Computational Issues for Branch-and-Cut Algorithms. In Michael Jünger and Denis Naddef, editors, *Computational Combinatorial Optimization: Optimal and Provably Near-Optimal Solutions*, volume 2241 of *Lecture Notes in Computer Science*, pages 1–25. Springer, 2001.

[McL80] Fred McLafferty. *Interpretation of Mass Spectra*. University Science Books, 3rd edition, 1980.

[MFH+99] Fred McLafferty, Einar Fridriksson, David Horn, Mark Lewis, and Roman Zubarev. Biomolecule Mass Spectrometry. *Science*, 284(5418):1289–1290, 1999.

[MZSL98] Fred McLafferty, Mei-Yi Zhang, Douglas Stauffer, and Stanton Loh. Comparison of Algorithms and Databases for Matching Unknown Mass Spectra. *American Society for Mass Spectrometry*, 9:92–95, 1998.

[RHO00] Françoise Rogalewicz, Yannik Hoppilard, and Gilles Ohanessian. Fragmentation mechanisms of α-amino acids protonated under electrospray ionization: a collision activation and ab initio theoretical study. *International Journal of Mass Spectrometry*, 195/196:565–590, 2000.

[RMR+06] Ari Rantanen, Taneli Mielikäinen, Juho Rousu, Hannu Maaheimo, and Esko Ukkonen. Planning optimal measurements of isotopomer distributions for estimation of metabolic fluxes. *Bioinformatics*, 22(10):1198–1206, 2006.

[RR00] Gerta Rücker and Christoph Rücker. Automatic Enumeration of All Connected Subgraphs. *MATCH Communications in Mathematical and Computer Chemistry*, 41:145–149, 2000.

[RRKK05] Juho Rousu, Ari Rantanen, Raimo Ketola, and Juha Kokkonen. Isotopomer distribution computation from tandem mass spectrometric data with overlapping fragment spectra. *Spectroscopy*, 19:53–67, 2005.

[SCNV97] Karsten Schmidt, Morten Carlsen, Jens Nielsen, and John Villadsen. Modeling isotopomer distributions in biochemical networks using isotopomer mapping matrices. *Biotechnology and Bioengineering*, 55:831–840, 1997.

[SHS01] Tamer Shoeib, Alan Hopkinson, and Michael Siu. Collision-Induced Dissociation of the AG^+ –Proline Complex: Fragmentation Pathways and Reaction Mechanisms – A Synergy between Experiment and Theory. *The Journal of Physical Chemistry B*, 105:12399–12409, 2001.

[SP99] Bernhard Seebass and Ernö Pretsch. Automated Compatibility Tests of the Molecular Formulas or Structures of Organic Compounds with Their Mass Spectra. *Journal of Chemical Information and Computer Sciences*, 39:713–717, 1999.

[SS94] S.E. Stein and D. Scott. Optimization and Testing of Mass Spectral Library Search Algorithms for Compound Identification. *Journal of American Society of Mass Spectrometry*, 5:859–866, 1994.

[Swe03] Daniel Sweeney. Small molecules as Mathematical Partitions. *Analytical Chemistry*, 75:5362–5373, 2003.

[vRLDZ+04] Edda von Roepenack-Lahaye, Thomas Degenkolb, Michael Zerjeski, Mathias Franz, Udo Roth, Ludger Wessjohann, Jürgen Schmidt, Dierk Scheel, and Stephan Clemens. Profiling of Arabidopsis Secondary Metabolites by Capillary Liquid Chromatography Coupled to Electrospray Ionization Quadrupole Time-of-Flight Mass Spectrometry. *Plant Physiology*, 134:548–557, 2004.

[Wil02] Antony Williams. Applications of Computer Software for the Interpretation and Management of Mass Spectrometry Data in Pharmaceutical Science. *Current Topics in Medicinal Chemistry*, 2:99–107, 2002.

[WMPdG01] Wolfgang Wiechert, Michael Möllney, Sören Petersen, and Albert de Graaf. A Universal Framework for ^{13}C Metabolic Flux Analysis. *Metabolic Engineering*, 3:265–283, 2001.

[ZGC+05] Jingfen Zhang, Wen Gao, Jinjin Cai, Simin He, Rong Zeng, and Runsheng Chen. Predicting Molecular Formulas of Fragment Ions with Isotope Patterns in Tandem Mass Spectra. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:217–230, 2005.

Classifying permanent and transient protein interactions

Samatha Kottha and Michael Schroeder
Biotec and ZIH, TU Dresden, {samatha.kottha,michael.schroeder}@tu-dresden.de

Abstract: Currently much research is devoted to the characterization and classification of transient and permanent protein-protein interactions. From the literature, we take data sets consisting of 161 permanent (65 homodimers, 96 heterodimers) and 242 transient interactions. We collect over 300 interface attributes relating to size, physiochemical properties, interaction propensities, and secondary structure elements.

Our major discovery is a surprisingly simple relationship not yet reported in the literature: interactions with the same molecular weight or very big interfaces are permanent and otherwise transient. We train a support vector machine and achieve the following results: Molecular weight difference alone achieves 80% success rate. Together with the size of the buried surface the success rate improves to 89%. Adding water at the interface and the number of hydrophobic contacts we achieve a success rate of 97%.

1 Introduction

Protein-protein interactions are fundamental to most cellular processes such as recognition of foreign molecules, host response to infection, transport machinery across various biological membranes, packaging of chromatin, the network of sub-membrane filaments, muscle contraction, signal transduction, and regulation of gene expression. Aberrant or lack of certain protein-protein interactions leads to the neurological disorders such as Alzheimer's disease. The forces that are responsible for these interactions include electrostatic forces, hydrogen bonds, van der Waals forces, and hydrophobic effects. The understanding of these interactions will provide the clues to their biological function. Several groups have been analyzing protein-protein interactions by categorizing them as homo-complexes, homo-oligomers, hetero-complexes, hetero-oligomers, obligate and non-obligate complexes, transient and permanent complexes, folding type and recognition type complexes (10; 14; 17; 13; 5; 2; 3; 1; 4).

A fundamental distinction in the nature of protein-protein interfaces is the separation into permanent and transient interfaces which are also called two-state and three-state complexes, respectively (21). Folding and binding are inseparable for two-state complexes. However, in case of three-state complexes, proteins fold independently and then bind. It is widely believed that permanent interactions can occur in homomers and heteromers, and transient interactions mostly in heteromers. However, Nooreen et al. and and Schreiber et al. collected 13 experimentally validated homodimers with transient interactions (15; 20).

Several studies analyze protein-protein interactions using interface properties like size, shape, residue and atomic contact propensities, hydrophobicity, hydrogen bonds, and sec-

ondary structure (10; 17; 13; 15; 5; 2; 3). Not a single feature analyzed in these studies differentiates permanent interactions from transient interactions or vice versa. As Nooreen and Thornton point out (15), it is difficult to discriminate, especially the strong transient from permanent interactions or the weak permanent from transient interactions. Mintseris and Weng propose atomic contact vectors to tackle this difficult problem and achieve a 91% success rate (13). However, they use 171 features to classify 340 interactions.

In this paper, we derive a data set of transient and permanent interactions from literature and initially capture over 300 attributes for the interfaces. We analyse the most predictive attributes in detail and show that the four attributes of molecular weight difference of the chains, size of the buried surface, number of water molecules at the interface, and number of hydrophobic contacts achieve a classification success rate of 97% - to our knowledge the best success rate reported. Moreover, the difference in molecular weight of the two interacting chains is the single most predictive attribute, which achieves a success rate of 80% on its own. This is particularly remarkable, as it can be derived from sequence information only.

2 Materials and Methods

We use five datasets introduced in (13; 20; 15; 1; 4). Even though all these datasets are generated by applying stringent criteria, some of them are contradicting each other. For example, the transferase 1d09 A:B is classified as permanent in (13) and transient in (20) and the toxin 1bun A:B is classified as permanent in (4) and as transient in (20). We carefully examine all the interactions with contradicting classification and label them according to the literature. Overall, only 9 out of over 400 interactions are affected.

To obtain a non-redundant dataset, all the interacting chains' sequences are clustered using BLASTCLUST (`ftp.ncbi.nih.gov/blast/`). The interactions which have both interacting chains with $\geq$25% sequence identity are clustered together and one interaction from each cluster is selected. As a result, we have 161 permanent and 242 transient interactions in our dataset. For these two classes, it is important to cover both homo- and heterodimers. This is indeed the case for our dataset, as the breakdown below shows:

	transient	permanent	sum
homo	13	65	81
hetero	229	96	322
sum	242	161	403

Feature Collection. We collect over 300 attributes about the interacting chains, residues, interfaces, and secondary structure elements and categorize them into the following four sets:

Size. Number of residues per chain, molecular weight and Accessible Surface Area (ASA) of each interacting chain, molecular weight difference, interface area ΔASA, number of residues at interface compared to individual chains, number of residues at interface compared to total residues, contact surface area, contact volume, total number of residue con-

tacts, number of residues at interface.

Physiochemical properties. Isoelectric Point of each interacting chain, hydrophobicity of the interface, normalized hydrophobicity by the interface size, hydrogen bonds, salt bridges, disulfide bonds and hydrogen bonds per 100 ASA in interface, water at interface, interaction strength, number of aromatic, charged, polar, hydrophobic, hydrophilic, hydro-neutral residues in interface, and the contacting residues pairs properties like aromatic_aromatic etc.

Amino acid propensities. Counts of residues A,C, . . . , Y and contacts A-A, A-C, . . . , Y-Y at interface.

Secondary structure elements. The absolute and normalized counts of interacting residues' secondary structure elements (helix, strand, coil, and turn).

The above attributes range from very general attributes like the number of hydrophobic-hydrophobic contacts to very special ones like the individual residue pair propensities including all pairs of hydrophobic residues, which appears redundant. However, the objective behind collection of both specific and general attributes is that all of them may play a role. If permanent interactions have large interfaces, there should be hydrophobic cores and hence hydrophobic-hydrophobic contacts could be important. Residue propensities vary strongly for different pairs and hence individual counts of residue-residue interactions may also be important. In the end, all of these attributes are collected, so that the algorithm can select the most predictive ones.

The molecular weight and the isoelectric points are calculated using the bioperl module with the EMBOSS value set. Accessible surface areas and ΔASA are determined using NACCESS (`wolf.bms.umist.ac.uk/naccess/`). The contact surface area and volume are derived by computing convex hulls of interaction interfaces (7). A novel, experimentally determined Stephen-White hydrophobicity scale (9) is used to calculate hydrophobicity. It does not lead to different results compared to the Kyte–Doolittle scale (12). The number of hydrophobic contacts is computed at residue level (F, A, I, M, L, V, C are hydrophobic) and if a residue participates in several hydrophobic-hydrophobic contacts, all of them are counted. While hydrophobic-hydrophobic contacts are a count, hydrophobicity is the sum of all interface residues' hydrophobicity according to (9).

Different types of bonds between two chains are determined using WHATIF (22). The interaction strength is calculated based on the bonds formed between two chains. The bond strength is measured by the amount of energy required to break the bond. Although the strength of a bond depends on the environment, a covalent bond is nearly 90 times stronger than a single hydrogen bond in water. Therefore, we consider disulfide bridges with a strength of 90, salt bridges with 3 and hydrogen bonds with 1.

Water at the interface is the number of water molecules which are $\leq 5\mathring{A}$ distance to both interacting chains.

The absolute and normalized counts of all amino acids in the interface are considered along with the contacting residue pairs. The two residues are said to be in contact if their atoms are within or equal to 5 $\mathring{A}$ distance.

Using STRIDE (8), the secondary structure elements of the interacting residue pairs are

determined. We consider both the absolute and the normalized counts.

Algorithms. We have 161 instances of permanent and 242 instances of transient interactions each with a vector of over 300 attributes in the training set. To identify the most relevant attributes for the classification task, we use relief estimation (11), which ranks the most predictive features independent of any learning algorithm. For the classification of permanent and transient interactions we use decision trees (C4.5) (18) to derive specific rules and support vector machines (SVM) to carry out an overall classification. For the SVM we use the LIBSVM library (6). We use a Radial Basis Function (RBF) kernel to map data into a higher dimensional space. We perform a grid search on internal parameters C and γ using cross validation and the value set with the best cross validation accuracy is picked. To avoid the problem of overfitting we use stratified 10-fold cross validation for both, the SVM and C4.5 algorithms.

Evaluation. In the results section we apply support vector machines to compute the overall success rate for a set of attributes, as well as sensitivity and specificity of built model and decision trees to derive intuitive classification rules. For these rules we report accuracy and support. Accuracy assesses how good the rule's classification is and support assesses to how many examples in the data set the rule applies.

The success rate is defined as the number of correctly predicted interfaces divided by all interactions: Success rate = Correct predictions / All interactions i.e. the success rate assesses the overall percentage of correct predictions. The sensitivity = TP / TP+FN and the specificity = TN / TN+FP .

To define the accuracy and support of a rule, let us denote the correct predictions of the rule as TP (True Positives) and the incorrect predictions as FP (False Positives). Then, the accuracy of a rule's prediction is defined as the percentage of correctly predicted examples for the rule: Accuracy = TP / TP+FP. The support indicates how general a rule is, i.e. to how much of the data it applies to: Support = TP+FP / All interactions. Generally, we wish to define rules with high accuracy and support.

3 Results

Molecular weight difference achieves 80% classification success rate. The ten most highly predictive attributes (in descending order) are molecular weight difference, ΔASA, hydrophobic-polar contacts, hydrophobic-hydrophobic contacts, water at interface, no. alanine-lysine contacts, no. isoleucine-tyrosine contacts, no. helix-helix contacts, no. methionine at interface, and no. leucine-serine contacts. The difference in molecular weights is the most outstanding feature separating permanent from transient interactions - both for homo- and heterodimers. Consider the scatterplot in Fig. 1a. Most permanent interactions are located on or close to the diagonal, i.e. both chains are of (nearly) equal molecular weight. This is not surprising for homodimers, but the majority (96 out of 161) of permanent interactions in the data set are actually heterodimers. Using a support vector machine (see materials and methods), the molecular weight difference alone can classify 80% of interactions correctly with a sensitivity of 71% and specificity of 86%. A closer exami-

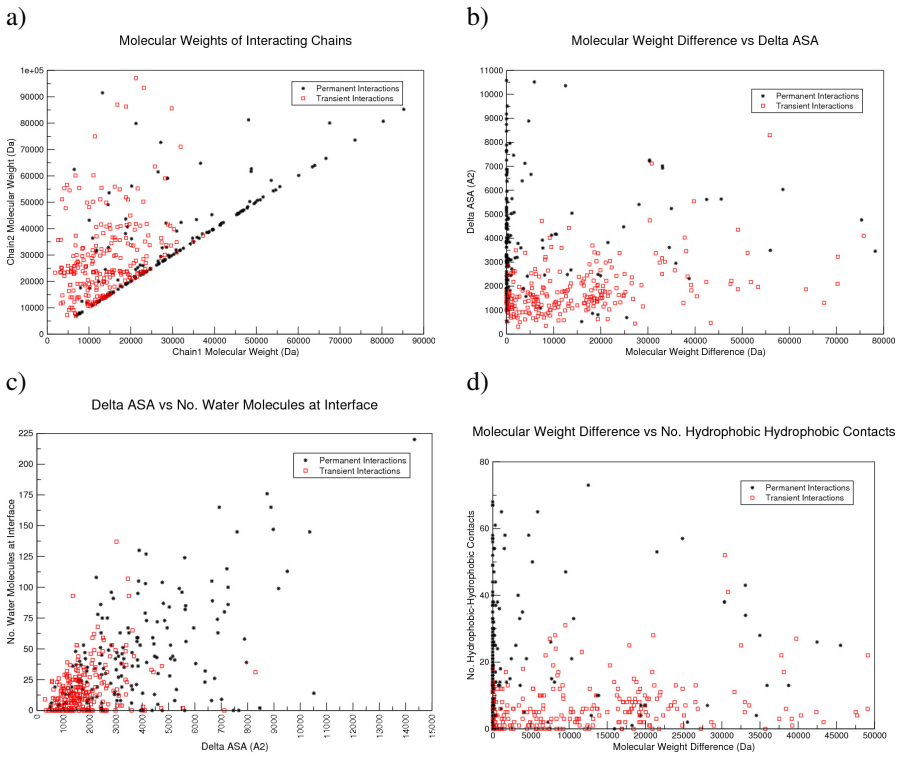


Figure 1: **a)** Scatterplot for molecular weight difference of interacting chains. Permanent interactions are close to the diagonal as they have similar weights. This is particularly remarkable as 96 out of 161 permanent interactions are heterodimers. Transient interactions mostly involve a lighter and a heavier chain. **b)** Scatterplot for molecular weight difference of interacting chains against ΔASA. Permanent complexes loose more surface accessible surface area upon complexation than the transient ones. Permanent interactions with more than 5 kDa molecular weight difference have mostly large interface of greater than 2000 $\mathring{A}^2$. **c)** Scatterplot for absolute counts of water at the interface plotted against ΔASA. There is some correlation (0.486) between the two attributes. **d)** Scatterplot for the number of hydrophobic contacts plotted against molecular weight difference. The plot shows that permanent interfaces have more hydrophobic contacts and are therefore a useful additional feature in the classification task.

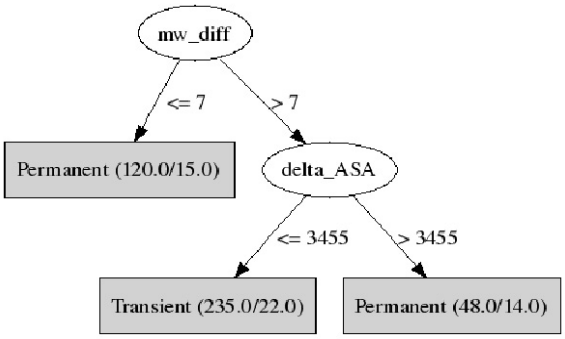


Figure 2: Decision tree with molecular weight difference and ΔASA. The boxes contain the predicted class. The total number of interactions and the number of incorrectly classified examples are in brackets. The ovals are the decision points defined by the algorithm. It identifies more or less than 7 Da molecular weight difference as main separating feature for transient and permanent interactions. It also automatically separates very big interfaces from other interfaces.

nation of the distribution also reveals that the interaction between chains with less than 7 Da weight difference are mostly permanent (88% accuracy and 30% support), while interactions between chains with more than 10 kDa molecular weight difference are usually transient (83% accuracy and 40% support).

The interesting aspect of these two rules is that they do not require any structural information and as only 65 out of 161 permanent interactions are homodimers.

Molecular weight difference and buried surface achieve 89% classification success rate.. As stated above, ΔASA is the second most predictive feature. The scatterplot in Fig. 1b shows that permanent complexes loose more solvent accessible surface area than transient complexes. In particular, nearly all permanent interactions with more than 5 kDa molecular weight difference have interfaces bigger than 2000 $\mathring{A}^2$, while most transient interactions have smaller interfaces.

To quantify this observation, we trained a support vector machine (see materials and methods) for molecular weight difference and ΔASA and achieved a classification success rate of 89% (sensitivity 84% and specificity 93%). In order to capture intuitive rules for this classification task, we also generated a decision tree (see materials and methods) shown in Fig. 2. The decision tree procedure automatically derives cut-off values. For ΔASA, it distinguishes very big ($3455\mathring{A}^2$) or not and for molecular weight differences small (≤ 7 Da) or not. Overall, the decision tree consists of three rules as shown in Fig. 3, which can be summarized as follows: Interactions with very small molecular weight difference (≤ 7 Da) or very big interfaces ($\geq 3455\mathring{A}^2\Delta$ASA) are permanent, otherwise they are transient. This single rule on its own achieves accuracy of 87% and a support of 100%.

No	Weight Difference		ΔASA		Class.	Acc.	Supp.
1	Very small	$\leq$7 Da	Does not matter		Permanent	88	30
2	Not small	>7 Da	Very big	$> 3455\mathring{A}^2$	Permanent	71	12
3	Not small	>7 Da	Not very big	$\leq 3455\mathring{A}^2$	Transient	91	58

Figure 3: Classification rules derived from a decision tree with their accuracy (Acc.) and support (Supp.). Rule 1 and 3 have the biggest support, i.e. they capture a large portion of the data set. Rule 1 states that if the molecular weight difference is very small the interaction is permanent. Rule 3 states that a difference in molecular weights and an interface, which are not very big, imply a transient interaction.

Adding hydrophobic contacts and water achieves 97% classification success rate.. To further improve the classification results we added two more features: water at the interface, which is a feature for transient interfaces (16), and the number of hydrophobic contacts, which is important for permanent interactions. As stated above, the number of hydrophobic-polar contacts is the third most predictive feature. However, molecular weight difference, ΔASA, water at the interface and hydrophobic-hydrophilic contacts are performing slightly worse (96.03%) than hydrophobic-hydrophobic contacts (97.27%). Both features achieve roughly similar results as they are highly correlated (0.8), but hydrophobic-hydrophobic contacts are slightly less correlated to water at the interface (0.35) than hydrophobic-hydrophilic contacts are (0.43). It is also established that large interfaces have hydrophobic cores (see e.g. (10)), so that the better performance of hydrophobic-hydrophobic contacts and its role in large interfaces led us to choose it over hydrophobic-hydrophilic contacts. So, the attributes molecular weight difference, ΔASA, water at the interface, and hydrophobic-hydrophobic contacts could classify 97% of interactions (sensitivity 95% and specificity 99%) correctly.

Although the absolute number of water molecules at the interface correlates to some degree (0.486) with the interface size ΔASA (see Fig. 1c), it improves the classification success rate as shown below. As an additional feature relating to the role of water, we also checked water mediated contacts. These are contacts between two residues from different interacting chains, which are in contact through a single water molecule but not in direct contact ($> 5\mathring{A}$ distance).

However, water-mediated contacts do not play a role in this classification task, which is consistent with Rodier et al. (19), who found that water density at homodimeric interfaces and protein-protein complexes is the same. Note, that the number of water molecules at the interface and the number of water-mediated contacts are not highly correlated (only 0.424).

Besides water, we investigated hydrophobic contacts as it is widely believed that permanent interfaces are more hydrophobic than transient ones. For the analyses of hydrophobicity we used the Stephen-White hydrophobicity scale (9). Fig. 1d shows that the feature of hydrophobic contacts separates transient and permanent interfaces well.

As a final step, we trained a support vector machine (see materials and methods) with the four attributes molecular weight difference, ΔASA, number of water molecules at the

Molecular weight difference

	transient	permanent	sum
homo	0/13	65/65	65/78
hetero	207/229	50/96	257/325
sum	207/242	115/161	322/403

Molecular weight difference, ΔASA

	transient	permanent	sum
homo	8/13	58/65	66/78
hetero	217/229	77/96	257/325
sum	225/242	135/161	360/403

Weight diff., ΔASA, hydrophobic-hydrophobic contacts, water at interface

	transient	permanent	sum
homo	11/13	61/65	72/78
hetero	229/229	91/96	320/325
sum	240/242	152/161	392/403

Figure 4: Breakdown of correctly classified protein-protein interactions for transient homodimers, transient heterodimers, permanent homodimers, and permanent heterodimers. The overall success rates achieved are consistent with all these subclasses. Molecular weight difference alone classifies permanent homodimers and transient heterodimers very well and permanent heterodimers reasonably well. Adding the other three attributes, success rates for all these subclasses are in the 90s.

interface (within $5\mathring{A}$), and number of hydrophobic contacts. We achieve a classification success rate of 97% for over 400 interactions in the data sets taken from (13; 20; 15; 1; 4).

Heterodimers vs. Homodimers and Transient vs. Permanent.. To test whether the above results also hold for heterodimers only, we considered 96 transient and 96 permanent heterodimer interactions. Thus, a random predictor achieves an expected success rate of 50%. The four attributes considered above perform as follows: Molecular weight difference alone achieves 73%. Molecular weight difference and delta ASA achieve 84%. Molecular weight difference, delta ASA, water at the interface and hydrophobic contacts achieve 88%. These results are in line with the ones for hetero- and homodimers reported above, in particular as homodimer interactions are not always permanent and as our dataset contains 13 such transient homodimer interactions, which are difficult to classify.

Indeed, it is an interesting questions how the success rates for the classification of the full 403 interactions break down between the classes of homo-transient, hetero-transient, homo-permanent, and hetero-permanent. Figure 4 shows three tables with these success rates for the three combinations of the four attributes. The first table shows that molecular weight difference alone classifies permanent homodimers and transient heterodimers very well and permanent heterodimers reasonably well. It does not handle the transient homodimers well. Adding ΔASA, the success rates for transient homodimers and permanent heterodimers greatly increase. Finally, the third table in Fig. 4 shows that the overall success rate of 97% is consistently achieved in all subclasses of transient homodimers (85%), transient heterodimers (100%), permanent homodimers (94%), and permanent heterodimers

(95%). Also, homo- and heterodimers achieve consistent success rates (92% and 99%, respectively) and transient and permanent interactions, too (99% and 94%, respectively).

4 Conclusion

There is great interest in characterizing and classifying protein interactions as transient or permanent (10; 14; 17; 13; 5; 2; 3; 1; 4). In particular, Mintseris and Weng achieve 91% prediction success rate using their atomic contact model with 171 features to classify 340 interfaces (13).

In this paper, we have assembled a data set consisting of 161 permanent and 242 transient interactions taken from the literature (13; 20; 15; 1; 4). For the interfaces we collected over 300 attributes relating to the size, physiochemical properties, residue propensities, and secondary structure elements.

Based on these data, we made a surprisingly simple discovery not yet reported in the literature: The difference in molecular weight between the interacting chains is the single most informative feature to distinguish transient from permanent interactions. Using this feature, 80% of interactions can be correctly classified. This is particularly important, as the molecular weight can be derived from sequence alone, so that no structural data is needed. Together with attributes known to play a role such as the size of the solvent accessible surface area lost upon complex formation, we can formulate the simple rule that interactions with small molecular weight difference or very big interfaces are permanent and otherwise they are transient. This simple rule achieves 87% success rate.

Finally, we added two more attributes known to be important, namely water at the interface and number of hydrophobicity contacts. Overall, we achieve a classification success rate of 97%, thus improving on other results previously published.

As next step, we wish to underpin our key insight that permanent interactions - like lasting marriages - require equal partners by developing physical models of the protein masses and moments, which can shed further light on this observation.

Acknowledgment: We gratefully acknowledge support of the EFRE Project CODI. We would like to thank Wan Kyu Kim, Joan Teyra, Gihan Dawelbait and Christoph Winter for helpful discussions and comments.

References

[1] S Ansari and V Helms. Statistical analysis of predominantly transient protein-protein interfaces. *Proteins: Structure, Function, and Bioinformatics*, 61(2):344–355, November 2005.

[2] R P Bahadur, P Chakrabarti, F Rodier, and J Janin. Dissecting subunit interfaces in homodimeric proteins. *PROTEINS: Structure, Function, and Genetics*, 53(3):708–719, November 2003.

[3] R P Bahadur, P Chakrabarti, R Rodier, and J Janin. A dissection of specific and non-specific protein-protein interfaces. *Journal of Molecular Biology*, 336(4):943–955, February 2004.

[4] J R Bradford and D R Westhead. Improved prediction of protein-protein binding sites using a support vector machines approach. *Bioinformatics*, 21(8):1487–1494, April 2005.

[5] P Chakrabarti and J Janin. Dissecting protein-protein recognition sites. *PROTEINS: Structure, Function, and Genetics*, 47(3):334–343, May 2002.

[6] C C Chang and C J Lin. *LIBSVM : A Library for Support Vector Machines (Version 2.6)*, 2004. www.csie.ntu.edu.tw/ cjlin/libsvm.

[7] P Dafas, D Bolser, J Gomoluch, J Park, and M Schroeder. Using convex hulls to extract interaction interfaces from known structures. *Bioinformatics*, 20(10):1486–1490, July 2004.

[8] M Heinig and D Frishman. Stride: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic Acids Research*, 32:W500–502, July 2004.

[9] T Hessa, H Kim, K Bihlmaier, C Lundin, J Boekel, H Andersson, I Nilsson, S H White, and G von Heijne. Recognition of transmembrane helices by the endoplasmic reticulum translocon. *Nature*, 433(7024):377–381, January 2005.

[10] S Jones and J M Thornton. Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences USA*, 93(1):13–20, January 1996.

[11] I Kononenko. Estimating attributes: analysis and extensions of relief. In F Bergadano and L De Raedt, editors, *Proceedings of Machine Learning: ECML-94*, pages 171–182. Springer Verlag, 1994.

[12] J Kyte and R F Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, May 1982.

[13] J Mintseris and Z Weng. Atomic contact vectors in protein-protein recognition. *Proteins*, 53(3):629–639, November 2003.

[14] I M A Nooren and J M Thornton. Diversity of protein-protein interactions. *The EMBO Journal*, 22(14):3486–3492, July 2003.

[15] I M A Nooren and J M Thornton. Structural characterisation and functional significance of transient protein-protein interactions. *Jorunal of Molecular Biology*, 325(5):991–1018, January 2003.

[16] R Nussinov, C J Tsai, and D Xu. Hydrogen bonds and salt bridges across protein-protein interfaces. *Protein Engineering*, 10(9):999–1012, September 1997.

[17] Y Ofran and B Rost. Analyzing six types of protein-protein interfaces. *Journal of Molecular Biology*, 325(2):377–387, January 2003.

[18] J R Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, San Francisco, 1993.

[19] F Rodier, R P Bahadur, P Chakrabarti, and J Janin. Hydration of proteinprotein interfaces. *Proteins: Structure, Function, and Bioinformatics*, 60(1):36–45, July 2005.

[20] G Schreiber, R Raz, and H Neuvirth. Promate: A structure based prediction program to identify the location of protein-protein bindings. *Journal of Molecular Biology*, 338(1):181–199, April 2004.

[21] C J Tsai, D Xu, and R Nussinov. Protein folding via binding and vice versa. *Folding & Design*, 3(4):71–80, 1998.

[22] G Vriend. What if: A molecular modeling and drug design program. *Journal of Molecular Graphics*, 8(1):52–56, March 1990.

Characterization of Protein Interactions

Robert Küffner, Timo Duchrow, Katrin Fundel, Ralf Zimmer

Institut für Informatik, Ludwig-Maximilians-Universität München,
Amalienstrasse 17, 80333 München, Germany

Abstract. Available information on molecular interactions between proteins is currently incomplete with regard to detail and comprehensiveness. Although a number of repositories are already devoted to capture interaction data, only a small subset of the currently known interactions can be obtained that way. Besides further experiments, knowledge on interactions can only be complemented by applying text extraction methods to the literature. Currently, information to further characterize individual interactions can not be provided by interaction extraction approaches and is virtually nonexistent in repositories.
We present an approach to not only confirm extracted interactions but also to characterize interactions with regard to four attributes such as activation vs. inhibition and protein-protein vs. protein-gene interactions. Here, training corpora with positional annotation of interacting proteins are required. As suitable corpora are rare, we propose an extensible curation protocol to conveniently characterize interactions by manual annotation of sentences so that machine learning approaches can be applied subsequently. We derived a training set by manually reading and annotating 269 sentences for 1090 candidate interactions; 439 of these are valid interactions, predicted via support vector machines at a precision of 83% and a recall of 87%. The prediction of interaction attributes from individual sentences on average yielded a precision of about 85% and a recall of 73%.

1 Introduction

The discovery or extension of molecular pathways and disease models requires the detailed knowledge on molecular interactions and their properties. Databases already capture many thousands of interactions between molecules [1,2,3], sometimes organized as pathways [3,4,5]. Most interactions were derived from large scale experiments, lacking additional details, e.g. to distinguish activation from inhibition. On the other hand, the bulk of the knowledge on interactions resides in the literature and can be accessed systematically only by automated extraction techniques. A number of such approaches have been published (a brief review can be found in [6]) but they usually do not predict any additional details on interactions. As common in the field, interactions are extracted from sentences that in turn are derived from publication abstracts as provided by Medline. We subdivide the extraction of interactions from sentences into the following steps for which we provide novel solutions:

1. We present a novel curation protocol (section 2.2) for a positional annotation of the interacting proteins. Manual annotation and systematic curation protocols are necessary as suitable training corpora are rare, e.g. as provided by the LLL challenge [7] dataset on procaryotic gene interactions.
2. Following this protocol, 269 sentences including 1090 possible interactions have been carefully read and annotated to derive a training data set (section 3.1). The large number of possible interactions is due to the fact that sentences tend to be long and frequently contain more than two proteins, and therefore $\binom{n}{2}$ co-occurrences for n proteins, and it might be difficult to decide which of the respective pairs of proteins actually interact and in which way.
3. To distinguish interactions from co-occurrences, we first identify the relevant part of sentences via RelEx [6]. Subsequently, each co-occurrence is evaluated in turn to predict interactions using support vector machines (Section 2.4).
4. Sentences frequently provide additional information to characterize individual interactions. Here we aimed to derive four attributes from the texts (Table 1): (a) directed vs. nondirected, (b) activation vs. inhibition, (c) immediate vs. long range and (d) protein-protein vs. protein-gene.

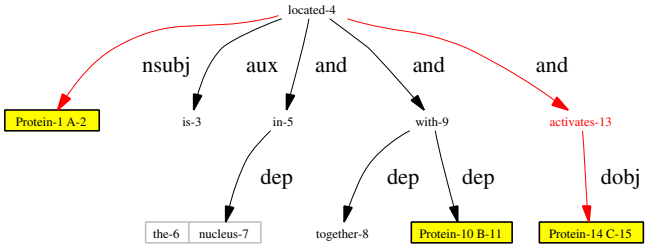
2 Methods

2.1 Preparation of data

In order to create a training set we compiled a list of PubMed abstracts likely to contain protein interactions. To this end, some preprocessing and preselection is required. In the context of this paper we were interested in human protein interactions, so we first screened Medline abstracts for human proteins via ProMiner [8]. The protocol to derive mappings between Medline abstracts and human proteins has been described in detail in [9]. Then, so called *interaction paths* have been derived via RelEx [6] based on dependency parse trees contructed by the Stanford Lexicalized Parser [10]. RelEx extracts chains of dependencies (*paths*) that connect two proteins to create candidate interactions (Figure 1). Thus, paths should contain the semantic dependencies and the corresponding subset of words from the original sentence necessary and sufficient to describe the relationship for each pair of protein entitites. This allows to subdivide the extraction of interactions from texts into (a) extracting the corresponding path and (b) to distinguish protein interactions from mere co-occurrences based on features from the path. The second subproblem will be described below (section 2.4). The following protocol has been applied to select sentences with an increased probability to contain descriptions of interactions:

1. We screened molecular biology databases [1,2,3] for PubMed references.
2. Sentences are selected according to one of the following two criteria. Variant 1 selects sentences matching HPRD interactions denoted as (PubMedId, protein 1, protein 2). Here, sentences from abstracts defined by PubMedId were selected that include both proteins. As we encountered some difficulties with this criterion (compare section 3.1) most sentences have been selected by variant 2 that simply requires sentences to contain at least two proteins.

3. Valid sentences were further required to contain at least one RelEx path and an interaction keyword (such as 'activate', 'formylation' etc). For this purpose we compiled a list of some 300 keywords, which we consider almost exhaustive. We randomly selected 4500 Medline abstracts that satisfy the above criteria as the source for an inital training data set.
4. From each of the selected abstracts only one sentence was selected at random. This constraint intends to avoid bias from abstracts referring to a particular interaction several times or containing many proteins.



Protein A is located in the nucleus together with protein B and activates protein C.

Fig. 1. Dependency parse tree of an example sentence as contructed by the Stanford Lexicalized Parser [10]. Arrows represent dependencies between terms. Proteins (yellow boxes) and noun phrase chunks containing several words are combined into larger nodes. The sentence contains one interaction keyword (*activates*) and one corresponding dependency path extracted by RelEx [6] that is marked in red. The path correctly maps *activates* to {A, C}, but not B.

2.2 Manual annotation

A simple textual annotation form is generated for each sentence selected in section 2.1. Proteins have already been detected via ProMiner [8] during sentence selection. Pairs of detected proteins yield candidate interactions that are manually annotated by five different attributes (Table 1). We use five labels that denote different levels of confidence to describe each attribute thereby providing some flexibility for the annotation of difficult cases. Figure 2 shows the annotation of a sample sentence. In addition to the five confidence labels the curator can indicate additional hints .

In the following we will introduce the concept of ***hints*** that are used to safeguard the selection of meaningful training contexts. During the development of our annotation protocol we had to ensure that results from curation are suitable for a subsequent classification/prediction setting. We need to keep in mind that

```
7838715.2.5 Chemical-1 sequencing-2 and mass-4 spectral-5 analysis-6 of
tryptic-8 peptides-9 derived-10 from the-12 purified-13 polypeptides-14
identifies-15 the-16 ARF6-17 complex-18 as a-20 heterodimer-21 of the-23
retinoid-24-X-25 receptor-26-alpha-27      (RXR-29-alpha-30)-31 and the-33
murine-34 peroxisome-35-proliferator-36-activated-37-receptor-38-gamma-39
(PPAR-41-gamma-42)-43.
27 39 interacting=5 18 21
27 39 directed   =1
27 39 activating =3
27 39 immediate  =5
27 39 expression =1
```

Fig. 2. Annotation of entry 7838715.2.5 (PubMedId, <1=title, 2=abstract>, SentenceNo), an undirected, immediate protein-protein interaction. Two proteins have been detected by ProMiner [8], thus a single annotation slot below the sentence has been generated. Here, names ending at token positions 27 and 30 as well as 39 and 42, respectively, are consecutive synonyms referring to the same entity and thus do not yield additional candidate interactions. Each interaction slot is defined by the token positions of the two proteins as denoted by the first two columns of integers. The third column specifies the attribute that is to be labelled. The attribute value is manually entered into the fourth column, here already filled in. Further columns are reserved for hints (token *complex-18* or token *heterodimer-21*), required to be present on paths for training classifiers.

potentially not all the words from a sentence might be avaliable to a classifier, e.g. features might be generated from RelEx paths only. At the same time, we had to ensure that the curation process is independent from feature generation/classification as the exact specifications of RelEx or other underlying tools might be subject to change. Frequently, the decision if a particular label should be attributed (e.g. *expression*) depends on the presence of an essential term (e.g. *gene*) as in the sentence *The gene coding for A is regulated by B.* By denoting the keyword *gene* as a hint for the decision protein-protein vs. protein-gene interaction this sample would be valid only if the keyword *gene* is part of the respective set of features, or path, as the assertion of the attribute *expression* would not be possible based on the second part of the sentence (*A is regulated by B*) alone. In the classification setting, instances are removed from the training and classification pools if they lack features annotated as hints.

2.3 Generation of features

Features are generated for all sentences chosen by our selection protocol (section 2.1). Our approach is to define generic *feature sources* that are applied to each candidate interaction (i.e. (PubMedId, position protein 1, position protein 2)). Each feature source generates features that are added to a global feature list for this candidate. This makes it possible to combine several feature sources

with each other to define a feature space. Protein names are excluded to avoid overfitting. Features are derived from words stemmed by the Porter [11] stemmer.

Bag-of-words (BOW) creates features from all words in a sentence.

Bag-of-words-path (BOW-path) only creates features for a subset of the words in a sentence, i.e. for a path determined by RelEx. Given a sentence and a pair of proteins (candidate interaction), a subset of paths from the set of all paths for the given sentence are selected that contain the proteins. This feature source also uses hints entered into the curation forms. If no hints are given all applicable paths are selected. If hints are defined specifically for an attribute only those paths are admitted that contain at least one of the hints.

2.4 Classification procedure

Besides predicting protein interactions from co-occurrences we also predict the type of interaction with respect to four attributes: (a) directed vs. nondirected, (b) activating vs. inhibiting, (c) immediate vs. long range and (d) protein-protein vs. protein-gene. Training and predictions for the latter 4 attributes are performed even if a candidate interaction is annotated or predicted as invalid.

For learning, a reduced set of labels is constructed by combining 1+2 as well as 4+5. The prediction of interactions is a two class problem and has been realized by training a single SVM classifier. The other four attributes each constitute three class problems, e.g. *activating* (1+2), vs. *inhibiting* (4+5) vs. *not specified* (3). A three class problem can be reduced to a set of two class problems using the one-versus-rest (OVR) strategy. Two binary SVM classifiers are constructed for each class vs. the other classes, i.e. 1+2 vs. 3+4+5 and 4+5 vs. 1+2+3. No classifiers were constructed for *not specified* vs. *rest*, though, so that two classifiers are required for each of the three class problems. Thus, a total of nine classifiers are required for the five attributes. To combine the outputs of the two classifiers for a specific attribute we use the following rule: *not specified* is predicted if a new sample is located on the side of the negative training samples with regard to the decision hyperplane for both classifiers. Otherwise, the class is selected that corresponds to the maximum value of the SVM decision functions of the two respective classifiers.

All training and classification using support vector machines has been performed using svmlight [12]. We used the default parameters (linear kernels), except that the cost-ratio for training errors on positive samples has been set to the ratio of the corresponding class sizes, i.e. #negative examples / #positive examples.

3 Results

3.1 Construction of a test set

In order to increase the probability that selected sentences indeed describe interactions, we first used variant 1 of our sentence selection protocol (Section

Attribute	label 1	label 2	label 3	label 4	label 5
interacting	no=**661**	**0**	**0**	**37**	yes=**392**
directed	undirected=**186**	**4**	**3**	**6**	directed=**240**
activating	inhibiting=**36**	**0**	**280**	**10**	activating=**113**
immediate	indirect=**101**	**13**	**33**	**64**	direct=**228**
expression	protein-protein=**258**	**32**	**44**	**9**	protein-gene=**96**

Table 1. Attribute labels and their distribution in the training data. Label 3 indicates that an attribute is not specified in the given sentence. Intermediate labels 2 and 4 indicate that the annotation has been attributed with only moderate confidence by the curator. The 661 samples labeled as not interacting are assigned label 3 for the other attributes (not counted here).

2.2) to select 50 abstracts and one sentence from each abstract. Thereby, sentences are selected that were likely sources for interactions derived by HPRD [3]. We manually labelled these sentences and analyzed the results with regard to the five interaction attributes. This analysis showed that about 90% of the selected sentences described interactions. Unfortunately, the analysis also showed that the distribution of attribute labels was significantly imbalanced towards protein-protein interactions based almost exclusively (>90%) on the keywords *binds*, *interacts* and *complex*. Most sentences did not provide any information on activation/inhibition, expression or directed interactions. This indicates that the curation protocol employed by HPRD is selective with regard to immediate protein-protein interactions and we could not expect to derive a balanced distribution of attribute labels this way.

Further curation thus focused on the second variant of our sentence selection protocol. In total, 269 sentences have been annotated yielding attribute labels for 1090 instances of candidate interactions. The overall distribution of labels is shown in Table 1. The *interacting* and *directed* attributes were most straightforward to annotate. Only few instances were labelled with moderate confidence (labels 2 and 4) whereas label 3 (not specified) was virtually absent. Table 1 also shows that certain attributes are less frequent in free text interactions especially striking for *inhibition*, but still noticeable in the case of *long-range* and *protein-gene* interaction.

3.2 Evaluation of classifiers

Evaluation of performance for different classifiers was carried out on a set of 1090 annotated training instances defined by a sentence identifier and both interaction partners. For training and prediction, both strong (labels 1 and 5) and moderate (labels 2 and 4) confidence annotations were included. A stratified 10-fold cross validation has been repeated 10 times (i.e. 10*10) for different random splits. The performance estimates (Table 2 and Figure 3) show that attributes

with a larger number of examples yield a better performance. On average, precison is higher than recall, so predicted interactions and interaction attributes are reliable while some annotations could not be recovered. We also compared the performance with regard to the different options for feature generation (Table 3). The performance increased significantly when specific features were generated for dependency paths. Table 3 also compares the influence of hints on the performance. Here, hints showed a significant increase in performance (+5.5% in f-measure) only if features from the RelEx paths were included. The influence of hints was hardly noticable if only the simple bag-of-words feature source has been used.

Classifier	Accuracy	Precision	Recall	F-measure
interacting	94.1	82.7	87.2	84.9
not directed	97.6	90.7	81.8	86.0
directed	95.1	84.6	67.2	74.9
inhibiting	99.1	75.4	63.6	69.0
activating	97.7	85.1	q.0	73.0
long range	97.4	79.0	49.2	60.6
immediate	94.8	83.3	78.6	80.9
protein-protein	95.5	86.9	81.2	83.9
protein-gene	97.8	86.1	65.3	74.3
overall	96.9	85.4	73.3	78.9

Table 2. Cross-validation performance on a data set of 1090 candidate interactions. Mean measures have been calculated via microaveraging. The *overall*-performance was calculated as the mean of all classifiers except *interacting.*

Protocol	Precision	Recall	F-measure
bag-of-words (BOW)	35.5	68.3	46.7
BOW + hints	36.1	69.0	47.4
BOW + path	78.2	82.3	79.4
BOW + path + hints	82.7	87.2	84.9

Table 3. The prediction performance of the classifier co-occurrence vs. interaction has been compared with regard to different feature sources and the utilization of hints.

In the following (see also Figure 4), a few examples will be mentioned where classification has been mislead by lexical subtleties or incorrect parse trees. In the

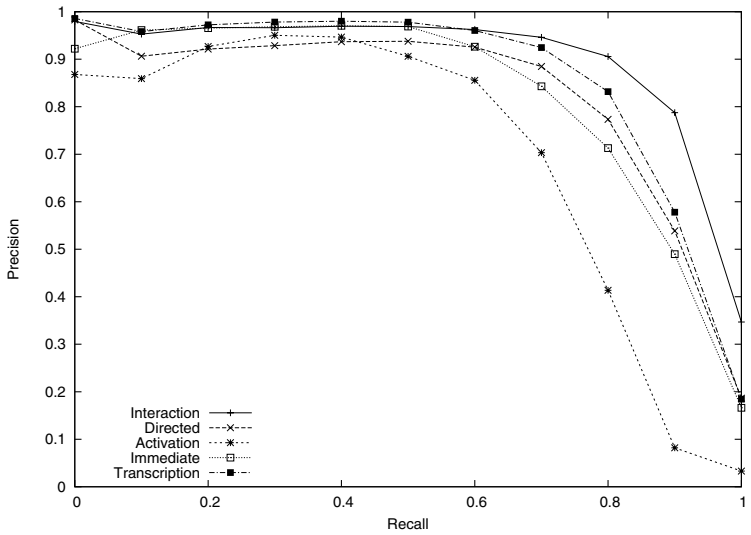
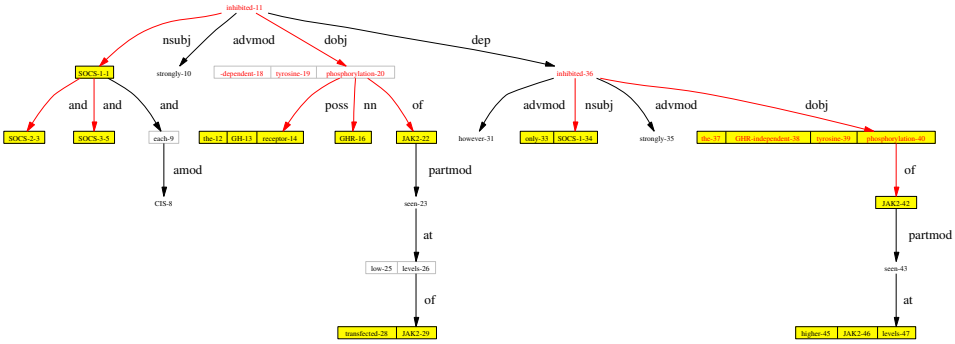


Fig. 3. Precision of attributes at 11 recall values. Performance estimates have been combined for the four attributes that require 2 classifiers, i.e. except for the simple attribute *interaction.*

sentence *A induces methylation of B* the word *induce* does not refer to induction of a gene as it would be the case if the term *methylation* would have been absent. This is different from the sentence *A induces methylating B.* As currently only word stems are considered for features and frequent stop words (such as 'of') are discarded both sentences yield the same set of features. Another difficult case is represented by *A inhibits signalling downstream of B* where a direct relationship (in the causal sense) between A and B is not necessarily implied. Some problems arise from incorrect dependency trees, e.g. in *A activates B but not C* the negation refers to B according to the parser [10]. Future improvements will also need to focus on multiple negations and to consider specific negations such as *A-null mice* or *A(-/-) mice.*

4 Discussion

The construction of advanced causal network models requires specific annotation (called attributes throughout this paper) on protein interactions such as activating vs. inhibiting or protein-protein vs. protein-gene. Such details on interactions are not available from current databases or text extraction approaches in a systematic and comprehensive way. We propose to alleviate this problem with a two step strategy for the extraction and characterization of molecular interactions from free texts. Starting from sentences we narrow down to the context or *path* comprising the actual assertions on a given candidate interaction. We presented



SOCS-1, SOCS-2, SOCS-3, and CIS each strongly inhibited the GH receptor (GHR)-dependent tyrosine phosphorylation of JAK2 seen at low levels of transfected JAK2; however, only SOCS-1 strongly inhibited the GHR-independent tyrosine phosphorylation of JAK2 seen at higher JAK2 levels.

Fig. 4. Dependency graph of a misclassified interaction. Here the interaction between each of the SOCS and GHR-16 are incorrectly classified as inhibiting. However, the text describes no direct inhibiting interaction between SOCS and GHR, but SOCS inhibits the GHR dependent phosphorylation of JAK2-22.

two major contributions: (1) a systematic and convenient curation protocol for the positional curation of candidate protein interactions including the manual annotation of a training set and (2) a protocol for training and evaluation of classifiers for the accurate prediction of interactions and four interaction attributes (Table 1).

Candidate interactions are annotated according to three levels of confidence: not specified, moderate and high confidence (Table 1). The introduction of the moderate confidence level helped to speed up the curation process as it was especially applicable to difficult examples. Without this level of confidence, several examples would have been annotated as *not specified,* so it also helped to improve recall during curation. We also introduced *hints*, i.e. labelling of special words essential for capturing a particular meaning of a given interaction. Hints are used to ensure that interaction paths can be excluded from classifier training if essential terms have been lost during preprocessing. We showed that the annotation of hints did not introduce a significant bias into classification (Table 3). As an additional advantage, hints capture information on why curators made particular decisions. In our experience the proposed curation protocol was simple to learn and use and categorized curator decisions appropriately.

We then constructed classifiers for the five attributes. These demonstrated good cross validation performance for predicting interactions (as opposed to mere co-occurrence of proteins) as well as other attributes. On average, precision was higher than recall, indicating that the manual annotation could not always be recovered automatically from the given sentences. At the same time we noticed that attribute performance was positively correlated with the abun-

dance of available annotation. This indicates that an enlargement of our current dataset, possibly selective with regard to the underpopulated attributes, will be beneficial. Our method itself is generic, so that an extension to accommodate additional attributes would be simple although additional manual annotation would be required to provide the necessary training data.

References

1. G. D. Bader, D. Betel, and C. W. Hogue, "Bind: the biomolecular interaction network database," *Nucleic Acids Res*, vol. 31, no. 1, pp. 248–50, 2003.
2. I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and D. Eisenberg, "Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions," *Nucleic Acids Res*, vol. 30, no. 1, pp. 303–5, 2002.
3. S. Peri, J. D. Navarro, T. Z. Kristiansen, R. Amanchy, V. Surendranath, B. Muthusamy, T. K. Gandhi, K. N. Chandrika, N. Deshpande, S. Suresh, B. P. Rashmi, K. Shanker, N. Padma, V. Niranjan, H. C. Harsha, N. Tal reja, B. M. Vrushabendra, M. A. Ramya, A. J. Yatish, M. Joy, H. N. Shivashankar, M. P. Kavitha, D. R. Menezes, M. a nd Choudhury, N. Ghosh, R. Saravana, S. Chandran, S. Mohan, C. K. Jonnalagadda, C. K. Prasad, C. Kumar-Sinha, K. S. Deshpande, and A. Pandey, "Human protein reference database as a discovery resource for proteomics," *Nucleic Acids Res*, vol. 32, no. Database issue, pp. D497–501, 2004.
4. M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. . Araki, and M. Hirakawa, "From genomics to chemical genomics: new developments in kegg," *Nucleic Acids Res*, vol. 34, no. Database issue, pp. D354–7, 2006.
5. M. Krull, S. Pistor, N. Voss, A. Kel, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapo v, C. Choi, O. Kel-Margoulis, and E. Wingender, "Transpath: an information resource for storing and visualizing signaling pathways and their pathological aberrations," *Nucleic Acids Res*, vol. 34, no. Database issue, pp. D546–51, 2006.
6. K. Fundel, R. Küffner, and R. Zimmer, "Relex - a new approach for relation extraction using dependency parse trees," *manuscript in preparation*, 2006.
7. C. Nedellec, "Learning language in logic - genic interaction extraction challenge," *Proceedings of the ICML05 workshop: Learning Language in Logic (LLL05)*, 2005.
8. D. Hanisch, K. Fundel, H. T. Mevissen, R. Zimmer, and J. Fluck, "Prominer: rule-based protein and gene entity recognition," *BMC Bioinformatics*, vol. 6 Suppl 1, p. S14, 2005.
9. R. Küffner, K. Fundel, and R. Zimmer, "Expert knowledge without the expert: integrated analysis of gene expression and literature to derive active functional contexts," *Bioinformatics*, vol. 21 Suppl 2, pp. ii259–ii267, 2005.
10. D. Klein and C. D. Manning, "Fast exact inference with a factored model for natural language parsing," *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, 2002.
11. M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14 (3), pp. 130–137, 2003.
12. T. Joachims, *Learning to Classify Text Using Support Vector Machines*. Dissertation, Kluwer.

Invited Talk

Docking protein domains using a contact map representation

Stefano Lise and David Jones

University College London, UK

In this talk I will discuss the possibility of predicting protein-protein interactions (docking) using a contact map representation. Rather than providing a full three dimensional model of the predicted complex, the method predicts contacting residues across the interface. A scoring function is used that combines structural, physicochemical and evolutionary information, where each potential residue contact is assigned a value according to the scoring function and the hypothesis is that the real configuration of contacts is the one that maximizes the score. The search is performed with a simulated annealing algorithm. The method has been tested on interacting domain pairs with encouraging results. Lastly, we find that predicted contacts can often discriminate the best model (or the native structure, if present) among a set of optimal solutions generated by a standard 3-D docking procedure.

Annotation-based Distance Measures for Patient Subgroup Discovery in Clinical Microarray Studies

Claudio Lottaz,* Joern Toedling,† Rainer Spang

Max Planck Institute for Molecular Genetics &
Berlin Center for Genome Based Bioinformatics,
Ihnestr. 73, D-14195 Berlin (Germany)

Abstract:

Background Clustering algorithms are widely used in the analysis of microarray data. In clinical studies, they are often applied to find groups of co-regulated genes. Clustering, however, can also stratify patients by similarity of their gene expression profiles, thereby defining novel disease entities based on molecular characteristics. Several distance-based cluster algorithms have been suggested, but little attention has been given to the choice of the distance measure between patients. Even with the Euclidean metric, including and excluding genes from the analysis leads to different distances between the same objects, and consequently different clustering results.

Methodology We describe a novel clustering algorithm, in which gene selection is used to derive biologically meaningful clusterings of samples. Our method combines expression data and functional annotation data. According to gene annotations, candidate gene sets with specific functional characterizations are generated. Each set defines a different distance measure between patients, and consequently different clusterings. These clusterings are filtered using a novel resampling based significance measure. Significant clusterings are reported together with the underlying gene sets and their functional definition.

Conclusions Our method reports clusterings defined by biologically focused sets of genes. In annotation driven clusterings, we have recovered clinically relevant patient subgroups through biologically plausible sets of genes, as well as novel subgroupings. We conjecture that our method has the potential to reveal so far unknown, clinically relevant classes of patients in an unsupervised manner.

1 Introduction

Gene expression profiling using whole genome microarrays has generated large amounts of data in various clinical contexts. One goal of these studies is the discovery of clinically relevant patient subgroups. Of interest are e.g. groups of patients which require a particular treatment.

*Corresponding author

†Current address: EMBL - European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD (UK)

An example from lymphoma research Alizadeh et al. [AED+00] define two new subtypes of diffuse large B-cell lymphoma based on a hierarchical clustering analysis using a functionally restricted set of genes. The two disease entities refer to distinct differentiation stages of B-cells. Monti et al. [MSK+05] postulate a different partitioning of diffuse large B-cell lymphomas supported by genes which have been excluded from the first analysis. Their disease entities reflect proliferation properties of the B-cell malignancies. None of the results can be easily proven wrong. In fact, they do not contradict each other. The two research groups had a priori different notions as to which genes are relevant. This led to two dissimilar but relevant clusterings of samples.

Different genes - different distances - different results In the context of class discovery, the objects that are to be clustered are patient samples. For clustering, pairwise distances between these objects are calculated. Using the Euclidean metric to do so, does not yet uniquely define these distances, though. Which genes to include in the analysis is very important. Using all measured genes as such is not a good choice. Several independent molecular characteristics of the patients like age, gender, and disease status will overlap and obscure the result. Gene selection is called for but certainly affects the clustering. Each choice of a gene set to use defines a particular distance between any two samples. Different gene sets lead to different distances between the same objects, although we always use the Euclidean metric to compute them. In many clinical studies, gene selection is used for unsupervised analysis, too. The intention is either to reduce noise in the expression data (e.g. [CSF+05]) or, in addition, to focus on reproducible features (e.g. [BRS+01, MSK+05]). However, little attention on the effect of gene discarding on the resulting disease class definition has been given.

The concept of our algorithm Instead of selecting genes according to purely statistical characteristics, we suggest a systematic approach to gene selection according to functional annotation. We describe an algorithm that produces a list of alternative clusterings using different gene sets for computing distances between samples. We derive candidate gene sets from functional annotation data, and filter the list by a novel significance measure for clustering strength.

Previous work Clustering of gene expression data is routine in bioinformatics. Several methods have been suggested in this field (for a review, see Chapter 4 of [Spe03]). Various approaches to score the quality of clusterings, and to determine the best number of clusters exist [DF02, KC01]. All these methods have in common that the underlying metrics need to be specified beforehand. Several authors also have suggested ways to judge stability and statistical significance of clusters [HBV01, LRBB04, MRF+02, MTMG03, MSS+05]. Semi-supervised clustering approaches include additional clinical information about patients. Bullinger et al. [BDB+04] as well as Bair and Tibshirani [BT04] suggest finding classes of patients using a clustering metric derived from the expression data and additional survival times. In a completely unsupervised setting, biclustering [CC00, TSKS04, MO04] and class-finding algorithms [vHHPV01, RL04, VS04] combine the gene selection process with the clustering. These methods produce alternative clusterings and characterize them by underlying gene sets. Unfortunately, such methods are rarely used in clinical studies. One reason might be that a large set of alternative clusterings is hard to interpret, unless the driving genes have a clear functional focus.

The role of functional annotations We believe that the major shortcoming of class discovery algorithms is that they treat gene expression levels as anonymous variables. For many genes, a lot is known about their function and their role in cellular processes. This knowledge is stored in databases like the Gene Ontology [ABB+00], Transpath [SCG+01], Biocarta (http://www.biocarta.com) or the Kyoto Encyclopedia of Genes and Genomes (KEGG) [Kan96]. Today, such annotations are routinely used to interpret results produced by statistical analysis. Several tools for such a-posteriori analysis are available [BS04, DSH+03, AS04, DSD+03, STM+05, GBRV06].

A-priori use of functional annotations Unlike a-posteriori methods, we propose using annotations *within* the statistical analysis of the expression data. In different contexts this a-priori use of functional annotations has already been investigated. Pavlidis et al. [PLN02] and Zien et al. [ZKZL00] use functional annotations to improve the sensitivity of algorithms for detecting differentially expressed genes. Rahnenführer et al. [RDML04] apply pathway annotations to investigate metabolic pathways. Subclass finding in complex clinical phenotypes using functional annotations is the topic of [LS05]. Here, we apply similar concepts to the problem of molecular class discovery in patients.

Outline of the paper In the next section, we describe the clustering procedure as well as the scoring of clustering results. In Section 3, we illustrate the usefulness of functional gene annotation for producing alternative clusterings of samples on a number of cancer related clinical microarray datasets. Finally, we discuss possible extensions of the method and interpret our observations from a biological perspective in Section 4.

2 Method

We present a novel algorithm for producing a list of alternative patient clusterings in clinical microarray studies. The key idea is to use meaningful gene sets for computing distances between samples. For practical use, it is desirable to have functional rationales characterizing clusterings, such as clusterings related to proliferation or apoptosis. To this end, we define candidate gene sets using functional annotations, and call the resulting clusterings *annotation driven*.

We use the k-means algorithm to generate clusterings based on candidate gene sets. The quality of these clusterings is assessed using the *diagonal linear discriminant* (DLD) score [vHHPV01]. In order to determine the statistical significance of scores, we also compute DLD scores for clusterings driven by randomly chosen gene sets. Empirical p-values are calculated and false discovery rates (FDR) computed according to Benjamini and Hochberg [BH95]. Finally, we filter the list of clusterings for minimal subgroup size and to control the FDR. In a nutshell, the algorithm consists of the following steps:

For each biological term / pathway of interest, denoted B_i:

1. Find all n_{B_i} genes annotated to B_i and discard all others.
2. Perform 2-means clustering on the reduced expression matrix. This yields an anno-

tation-driven clustering C_{B_i}.

3. Compute DLD score $S(C_{B_i})$ for this clustering.
4. Draw 10000 random gene sets of size n_{B_i} from the set of all measured genes. For each of them compute steps 2 and 3. This yields a vector $\mathbf{r}_{n_{B_i}}$ of 10000 scores.
5. Assign an empirical p-value to the original clustering, denoting the proportion of entries of $\mathbf{r}_{n_{B_i}}$ being greater or equal than $S(C_{B_i})$.

In the following, we provide more details on certain steps of the procedure.

2.1 Annotation data

We suggest the use of annotation data to generate candidate gene sets of interest. Genes in a candidate set have common involvement in biological processes or pathways. To generate such gene sets, pathway databases such as KEGG [Kan96] and Gene Ontology [ABB$^+$00] are particularly adequate.

Sets of genes collected for a particular application from literature or a biologist's experience are possible alternatives. Very small gene sets should not be considered, since clusterings supported by very few genes are unlikely to represent a clustering of biological interest. On the other hand, sets containing too many genes are prone to be very unspecific, and thus their results are of little explanatory power.

2.2 Distance metric

K-means clustering is based on pairwise object dissimilarities. Objects in our case are the samples' expression profiles. We obtain dissimilarity measures from the family of restricted Euclidean metrics, which we will define next.

Let $(x_i, x_{i'})$ be any two expression profiles, both containing measured expression values for p genes. Reducing the expression profiles to a limited set of genes before computing the distance, can also be interpreted as computing a Euclidean distance specific for gene set G between the original profiles

$$D_G(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{p} I_{j\in G} \cdot (x_{ij} - x_{i'j})^2}$$

where $I_{j\in G}$ is an indicator variable taking the value 1 if gene j is in set G and 0 otherwise. We call D_G a restricted Euclidean metric on patient space.

By selecting different gene sets before clustering, we choose different measures of distance between any two expression profiles. Since the choice of the distance measure affects the outcome of clustering stronger than the choice of the clustering algorithm (see Chapter 14 in [HTF01], clusterings of the same samples with different metrics disagree substantially.

2.3 K-means initialization

K-means clustering critically depends on its initialization step. We derive an initialization based on the first split of a divisive hierarchical clustering (Chapter 6 in [KR90]). Of the resulting two clusters, we compute centroids which provide the starting points for the k-means algorithm [Mac67]. This has been shown to outperform standard k-means with random starting points [MS80]. In other words, k-means is used to refine individual clusters and to correct inappropriate assignments made by the hierarchical method.

2.4 Scoring clusterings

For clustering evaluation, we employ the *diagonal linear discriminant* (DLD) score, adopted from [vHHPV01]. We briefly review it here.

Let $\mathbf{X}$ be the reduced expression matrix with rows containing the genes from the set of interest and columns representing the patient samples. Given a clustering C of samples, i.e. a binary vector of class labels for classes A and B, we are interested in those genes, whose expression levels reflect this class division best. A natural score for this purpose is Student's t-statistic. We discard all genes except those 50 genes with the highest absolute t-statistic. In case there are less than 50 genes in the functional group, all are kept. We avoid clusterings with very few supporting genes by discarding the top m genes with the highest absolute t-statistic to prevent the final DLD score from being strongly influenced by very few genes with extreme expression levels. This also makes results more robust against imprecise annotations. We chose $m = 5$. Discarding the respective rows (genes) from $\mathbf{X}$, yields a shortened expression matrix $\mathbf{X}^*$.

Now, the same projection method, which is used in the classification step of *diagonal linear discriminant analysis* [MKB79], is used to project the samples (columns) of $\mathbf{X}^*$ onto a one-dimensional space. The projection is defined by the vector

$$\mathbf{v} = \mathbf{S}^{-1} \left(\mu_A - \mu_B \right)$$

where μ_K denotes the centroid of all samples of class K and $\mathbf{S}$ is a diagonal matrix containing the weighted sums of within-class variances for each gene g:

$$\mathbf{S}_{gg} = (a-1)\sigma_{gA}^2 + (b-1)\sigma_{gB}^2$$

where A and B denote the two classes with cardinalities a and b respectively. Each patient sample, which is represented as a column of the shortened expression matrix $\mathbf{X}^*_{\bullet j}$, is projected onto the coordinate, given by the inner product $\mathbf{v}^\top \cdot \mathbf{X}^*_{\bullet j}$.

The DLD-Score S of a clustering C is the Student's t-statistic of the projected coordinates:

$$S(C) = \frac{\sqrt{\frac{a \cdot b}{a+b}} \cdot (\mu_{zA} - \mu_{zB})}{\sqrt{\frac{1}{a+b-2} \cdot ((a-1)\sigma_{zA}^2 + (b-1)\sigma_{zB}^2)}}$$

where z denotes the projected coordinates, μ_{zK} and σ^2_{zK} denote the mean and the variance of the projected coordinates of group K, while A and B denote the two groups with cardinalities a and b respectively.

2.5 Assessing clustering significance

We introduce a new approach to address the question whether an annotation-driven clustering is statistically significant. To this aim, we observe clusterings based on randomly drawn gene sets, which have the same size as the set of functionally related genes but otherwise no restrictions on included genes. For each of these random gene sets, we find the optimal clustering and compute its DLD-Score as described above. The score derived from the annotation-driven clustering is compared with these random scores.

The DLD-Scores derived from random gene sets define a null-distribution of scores for gene sets of the given size. For each annotation-driven clustering C, we can compute an empirical p-value $\pi_E(C)$ denoting the proportion of random scores $\mathbf{r}$ being equal to or greater than the annotation-driven clustering's DLD-Score $S(C)$:

$$\pi_E(C) \;=\; \frac{1}{|\mathbf{r}|} \cdot \sum_{r \in \mathbf{r}} I_{r \geq S(C)}$$

where $I_{r \geq S(C)}$ is an indicator variable taking the value 1 if the random score r is bigger or equal than $S(C)$ and 0 otherwise, and $|\mathbf{r}|$ denotes the number of simulated random gene sets. This empirical p-value provides us with a measure of significance for clusterings.

2.6 Multiple testing

The algorithm described so far, determines an empirical p-value for each term we can find associated genes for. Depending on the employed annotation sources and the microarray at hand, hundreds of terms are considered to generate annotation-driven clusterings. Hence, the determination of empirical p-values is subject to multiple testing. A conservative approach to correct for the multiple testing problem is to determine false discovery rates according to Benjamini and Hochberg [BH95]. We employ this correction although its results are to be interpreted with care given the many dependencies between GO and KEGG terms which share commonly associated genes.

2.7 Implementation

We have implemented our clustering method in the statistical programming language R [IG96, R D05]. We employ the divisive hierarchical clustering method from the `cluster` package and the implementation of k-means clustering [HW79] from R's `stats` package. The implementation of the DLD score is taken from the `isis` package [vHHPV01].

We also use Bioconductor's [GCB+04] meta-data packages to retrieve gene annotations for GO and KEGG. Our code is available in the R package `adSplit` [LTS05] from http://compdiag.molgen.mpg.de/software. The package is also part of release 1.8 of the Bioconductor bundle of packages related to the life sciences.

3 Results

We show results of our method on several cancer related datasets from clinical gene expression studies. We focus on the use of Gene Ontology (GO) and the Kyoto Encyclopedia of Genes and Genomes (KEGG) for annotations.

3.1 Expression data

We have used 15 clinical microarray studies to investigate the behavior of our clustering procedure. These studies investigate diagnostic and prognostic issues in the context of brain tumors [FCVF+04, NMB+03, PTG+02, RBM+01], breast cancer [HCD+03, WBD+01], leukemia [ASS+02, CYP+03, RMO+04, WJS+04, YRS+02], lung cancer [BKH+02, BRS+01] and prostate tumors [SFR+02].

All 15 microarray studies are based on Affymetrix® GeneChip technology. Eight datasets were generated using the genome wide HG-U95Av2 microarray based on release 95 of UniGene [Sch97]. Four studies are based on the older HU6800 chip, and in [RMO+04] as well as [FCVF+04] the newer HG-U133A chip based on release 133 of UniGene was applied. Finally, Willenbrock et al. have worked with the HG-Focus chip, a microarray holding a subset of the probe-sets of the HG-U133A chip. Table 1 holds further information on the results obtained for these 15 studies.

For each of these datasets, gene expression profiles were background corrected and normalized on probe level using variance stabilization [HvHS+02] before summarizing the probes into probe-set expression levels using median polish [Tuk77] as suggested in the RMA method by Irizarry et al. [IHC+03]. Implementations of these methods were taken from Bioconductor [GCB+04].

3.2 Annotation data

For the systematic exploration of functional gene annotations, we suggest the use of the Gene Ontology (GO) and the Kyoto Encyclopedia of Genes and Genomes (KEGG). GO holds 17,601 biological terms, while KEGG comprises 231 pathways. For the considered Affymetrix® microarrays, Table 2 states the number of terms and pathways, which have more than 20 probe-sets but less than 10% of all probe-sets on the chip annotated.

Strikingly many GO terms have very few genes attributed: more than 75% of all terms

Author	Cancer type	Study topic	Chip	♯N	♯C	FDR
Freije	brain	survival	U133A	85	71	9.2
Nutt	brain	subtypes	U95Av2	50	8	9.1
Pomeroy	brain	outcome	HU6800	100	23	8.8
Rickman	brain	subtypes	HU6800	51	0	–
Huang	breast (lns)	risk groups	U95Av2	37	40	9.9
Huang	breast (rel)	outcome	U95Av2	52	0	–
West	breast (rel)	outcome	HU6800	49	0	–
Armstrong	leukemia	subtypes	U95Av2	72	18	9.2
Cheok	leukemia	treatment	U95Av2	31	0	–
Ross	leukemia	subtypes	U133A	142	133	10.0
Willenbrock	leukemia	outcome	Focus	45	11	9.6
Yeoh	leukemia	subtypes	U95Av2	327	179	9.6
Beer	lung	outcome	HU6800	96	2	8.3
Bhattacharjee	lung	survival	U95Av2	254	113	9.9
Singh	prostate	subtypes	U95Av2	102	40	8.8

Table 1: Cancer related datasets used for evaluation. In the column '♯C' contains the number of annotation driven clusterings with smallest group size at least 5 when false discovery rate is controlled at 10 %. The column ♯N holds the number of samples. *lns*=lymphnode status, *rel*=relapse.

	Probe-sets	GO	KEGG
HU6800	7129	4534 / 752	130 / 63
HG-U95Av2	12625	5000 / 962	132 / 77
HG-U133A	22283	5417 / 1223	132 / 92

Table 2: Gene sets defined by GO and KEGG per chip. Numbers of gene sets are given before/after filtering for gene sets holding more than 20 and less than 10% of all probe-sets on the chip.

hold less than 20 probe-sets. On the other hand, very few terms are too general holding more than 10% of the genes on the whole-genome microarrays. The KEGG database also defines some very small gene sets, but roughly two thirds hold more than 20 genes.

On commercial Affymetrix® oligonucleotide microarrays, many genes are represented by more than one probe-set, thus several rows in an expression matrix give measurements for the same gene. When extracting probe-sets with a common annotation, either all or none of the probe-sets representing the same gene are included. When drawing random sets of probe-sets, we mimic this fact, by actually drawing Entrezgene-IDs and including all probe-sets mapped to these in our random set. In this manner, we make sure that random scores actually correspond to random gene sets rather than random sets of probe-sets.

All data sets discussed in this article are based on Affymetrix microarrays. Thus, we can use BioConductor's meta-data packages to deduce associations of genes to GO terms and KEGG pathways. Our method, however, is not restricted to this chip technology. For other microarrays the needed annotation data can be extracted from corresponding databases.

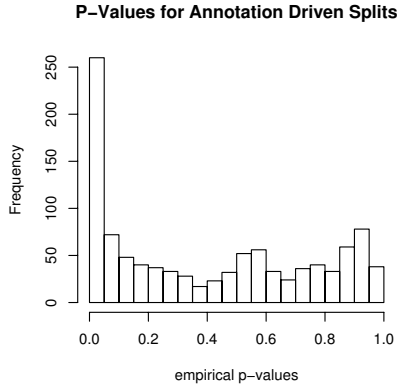


Figure 1: Distribution of empirical p-values of annotation driven clusterings on the gene expression study by Yeoh et al. on leukemia translocations.

3.3 Annotation driven clusterings

We observe that many annotation driven clusterings of patients obtain low empirical p-values. As illustrated in Figure 1 for the leukemia study by Yeoh et al. [YRS+02], the distribution of empirical p-values has a peak close to zero. Apparently, certain gene sets with common functional annotation provide a better basis for clustering samples than random sets of genes. Moreover, the clusterings corresponding to low p-values are of particular interest for the biological focus of their supporting genes.

Our second observation is that many clusterings with small p-values assign only few samples to one of the two clusters. In addition to a stringent p-value, we therefore also require a minimum group size of at least five samples for interesting clusterings. For the datasets analyzed, we thus obtain the number of interesting clusterings shown in the column 'Clusterings' of Table 1.

From the same table, we see that our clustering procedure behaves differently on different datasets. While it finds dozens of annotation-driven clusterings with false discovery rate lower than 10% and size of the small subgroup larger than 5 on most of our evaluation studies, it does not find any clustering in four datasets. In [YRS+02] very heterogeneous expression profiles caused by chromosomal aberrations are included, thus leading to a large number of significant annotation driven clusterings. We observe that our algorithm typically finds fewer annotation driven clusterings in small datasets. This may be caused by our second filtering criteria, which is more stringent on small datasets, given the absolute requirement of 5 samples per group in this criterion.

The set of annotation driven clusterings for one project may be quite heterogeneous. Figure 2 illustrates such a case occurring in the study on embryonic brain tumours by Pomeroy et al. [PTG+02]. Stratifying these tumors by morphological features is controversial. Hence, they present an interesting field of research for diagnosis on a molecular level. The

authors of this study acknowledge that the investigated tumours are very heterogeneous. In accordance with this observation, our method reports clearly differing annotation driven clusterings. Based on terms widely spread over the whole Gene Ontology, we determine 23 different gene sets justifying splits of samples into two groups on significantly better grounds than randomly picked genes.

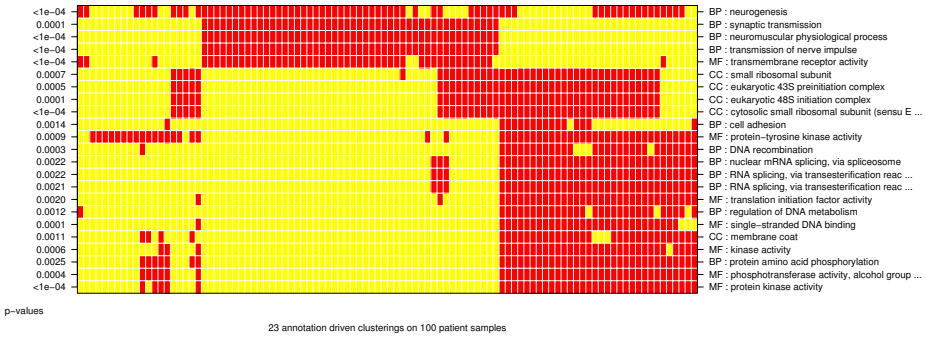
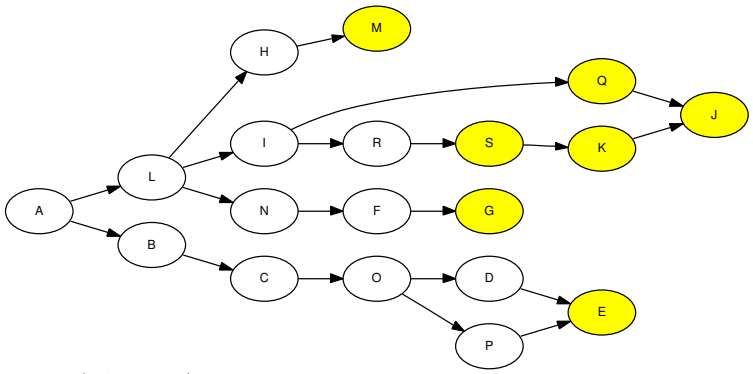


Figure 2: Annotation driven clusterings for the study by Pomeroy et al. Colors code the cluster to which a patient is attributed with respect to the corresponding gene set. In the gene set descriptions to the right of the image, the GO source ontologies of the annotations are indicated by *BP* for biological process *CC* for cellular component and *MF* for molecular function. Columns correspond to samples and rows to gene sets. The image is clustered in both directions in order to bring similar clusterings and similarly attributed samples close together. The depicted set of clusterings achieves a false discovery rate of 8.8%.

3.4 Coherence between clusterings and clinical parameters

The cited datasets from clinical microarray studies come with clinical information. For instance, in the lung-cancer study discussed in [BRS$^+$01], histologically defined subtype assignments are provided for the biopsies, while in [RMO$^+$04], cytogenetically determined translocations are given for each patient. In order to assess the clinical relevance of identified significant clusterings, we compare these with clinical parameters. We employ the χ^2-test to search for clusterings which are highly correlated with clinical parameters.

On several datasets, we observed clusterings of striking correlation with clinical parameters, thus supporting previous findings. For instance, on the acute myeloid leukemia (AML) data set of Ross et al. [RMO$^+$04], we found 11 patient splits for which the two groups correspond to some phenotypical separation of the samples. Less than 10 profiles are attributed inconsistently by these splits to the corresponding phenotypical separation and χ^2 contingency table tests yield p-values below 10^{-10}. Seven of these clusterings consistently separate the group of megakaryocytic leukemia profiles plus one other profile described as having an unspecified AML subtype from the other AML subtypes. The 7 clusterings stem from gene sets annotated to *blood coagulation* (GO:0007596) and related



A: Gene Ontology (GO:0003673)
B: cellular component (GO:0005575)
C: cell (GO:0005623)
D: plasma membrane (GO:0005886)
E: integral to plasma membrane (GO:0005887)
F: cell communication (GO:0007154)
G: cell adhesion (GO:0007155)
H: development (GO:0007275)
I: physiological process (GO:0007582)
J: blood coagulation (GO:0007596)
K: hemostasis (GO:0007599)
L: biological process (GO:0008150)
M: morphogenesis (GO:0009653)
N: cellular process (GO:0009987)
O: membrane (GO:0016020)
P: integral to membrane (GO:0016021)
Q: coagulation (GO:0050817)
R: organismal physiological process (GO:0050874)
S: regulation of body fluids (GO:0050878)

Figure 3: Clusterings driven by the gene sets associated to the 7 nodes colored in yellow identify acute megakaryocytic leukemia with just one conflicting class assignment in the dataset by Ross et al. The figure shows the GO subgraph induced by these nodes.

GO-terms. See Figure 3 for a display of the relationships between the 7 GO-terms and their ancestors within the Gene Ontology.

On the lung-cancer dataset by Bhattacharjee et al. [BRS+01], we identified 17 clusterings showing p-values $< 10^{-10}$ in the χ^2-test and differing by not more than 10 cluster assignments from the corresponding morphological classification of the tumors. 9 of these clusterings separate the group of 20 pulmonary *carcinoid tumors* from all other tumors. Five of the 9 clusterings also assign one or two other profiles to the cluster of carcinoid tumors. The 9 clusterings are derived from gene sets annotated to *central nervous system development* (GO:0007417), *ion channel activity* (GO:0005216) and related terms.

4 Discussion

An important goal of clinical microarray studies is the discovery of cohesive subgroups of patients according to molecular criteria. Commonly, unsupervised clustering is employed to this aim, although the evaluation of clustering results is notoriously difficult. One suggestion, to show whether a clustering is biologically meaningful, is to point out that functional annotation of the genes supporting the clustering are coherent or plausible.

In this paper, we propose an algorithm to use functional annotations stored in the Gene Ontology and the KEGG database of pathways directly to search for cohesive groups of

samples. By selecting genes sharing common annotation in GO or KEGG and limiting gene expression profiles to these, we define distinct distances between samples for each term or pathway. Consequently, different clusterings are found for each GO term or KEGG pathway. A notable difference to other approaches to select genes before clustering (e.g., [BDB+04]) is that the selection stems from independent data, which represent biological expert knowledge and are not affected by experimental variations.

The use of curated databases like GO and KEGG to extract functional annotations leads to the inclusion of some unreliable data. These databases, however, are always incomplete and the computationally derived annotations may contain errors. We expect our approach to be robust against such erroneous annotation data as long as the erroneous annotations do not dominate. Robustness is enhanced by the fact that clusterings are always supported by several genes with common annotation. Another characteristic of the Gene Ontology not taken into account by our method is its hierarchy. Genes annotated to a given GO term are also used to find clusterings for all parent terms. However, we do not very often observe that parents of children with significant clustering also have significant clusterings. The dependency between parents and children does not seem to be very strong.

We applied our method to a number of gene expression data sets (see Table 1) and found several significant annotation driven clusterings, which strongly correlate to patient stratifications based on clinical criteria and agree with previous reports on the biology behind tumor development. For instance, on the acute myeloid leukemia (AML) data set of [RMO+04], we found a large number of significant clusterings. AML is a heterogeneous disease, comprising abnormal proliferation of the precursors of granulocytes, monocytes, and thrombocytes [JHSV01]. Thus, it is not surprising to find many significant clusterings dividing one type of AML from the rest. For example, 7 clusterings that separate AML of the FAB-M7 type, i.e. acute megakaryocytic leukemia, from the other AML types, are based on gene sets attributed to *blood coagulation* (GO:0007596), *cell adhesion* (GO:0007155) and five related terms. Since megakaryocytes give rise to thrombocytes, whose primary function is to mediate cell adhesion to damaged endothelium and blood coagulation, they are bound to excel in the expression of genes involved in these processes. Remarkably, one patient profile that was clinically described as having an unspecified AML subtype is consistently assigned to the cluster of FAB-M7 samples. This sample seems to display molecular characteristics of the FAB-M7 subtype, although it would not be assigned to this subtype based on clinical criteria.

In accordance with other studies, Bhattacharjee et al. [BRS+01] have described lung cancer to be a general concept comprising very different tumor subtypes. We as well observe large biological differences between these subtypes in form of significant annotation driven clusterings. For example, 9 clusterings clearly separate pulmonary *carcinoid tumors* from all other types of lung cancer. These 9 clusterings are derived from gene sets annotated to *central nervous system development* (GO:0007417), *ion channel activity* (GO:0005216) and 7 related terms. Pulmonary carcinoid tumors have been previously reported to be of neuroendocrine origin and to be closely related to brain tumors [ATB+99]. Our finding of remarkable expression of nerve-cell associated genes by these tumors supports such reports.

In summary, the method presented in this paper has the potential to uncover clinically

relevant clusterings in gene expression studies. Moreover, such clusterings may be of particular interest due to the biological focus of their supporting genes.

Acknowledgments

The authors are grateful to Jochen Jäger, Dennis Kostka, Stefanie Scheid and Stefan Bentink from our work group as well as to our partners Renate Kirschner-Schwabe, Christian Hagemeier and Karl Seeger from the Charité Medical Center for fruitful discussions. This research has been supported by BMBF grants 01GS0445 and 01GR0455 of the German Federal Ministry of Education and the National Genome Research Network.

References

[ABB+00] M Ashburner, CA Ball, JA Blake, D Botstein, H Butler, JM Cherry, AP Davis, K Dolinski, SS Dwight, JT Eppig, MA Harris, DP Hill, L Issel-Tarver, A Kasarskis, S Lewis, JC Matese, JE Richardson, M Ringwald, GM Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.

[AED+00] AA Alizadeh, MB Eisen, RE Davis, C Ma, IS Lossos, A Rosenwald, JC Boldrick, H Sabet, T Tran, X Yu, JI Powell, L Yang, GE Marti, T Moore, J Hudson, L Lu, DB Lewis, R Tibshirani, G Sherlock, WC Chan, TC Greiner, DD Weisenburger, JO Armitage, R Warnke, and LM Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–11, 2000.

[AS04] B Adryan and R Schuh. Gene-Ontology-based clustering of gene expression data. *Bioinformatics*, 20(16):2851–2, 2004.

[ASS+02] SA Armstrong, JE Staunton, LB Silverman, R Pieters, ML den Boer, MD Minden, SE Sallan, ES Lander, TR Golub, and SJ Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat Genet*, 30(1):41–47, Jan 2002.

[ATB+99] R Anbazhagan, T Tihan, DM Bornman, JC Johnston, JH Saltz, A Weigering, S Piantadosi, and E Gabrielson. Classification of small cell lung cancer and pulmonary carcinoid by gene expression profiles. *Cancer Res*, 59(20):5119–5122, Oct 1999.

[BDB+04] L Bullinger, K Döhner, E Bair, S Fröhling, RF Schlenk, R Tibshirani, H Döhner, and JR Pollack. Use of Gene-Expression Profiling to Identify Prognostic Subclasses in Adult Acute Myeloid Leukemia. *N Engl J Med*, 350(16):1605–16, 2004.

[BH95] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995.

[BKH+02] David G Beer, Sharon LR Kardia, Chiang-Ching Huang, Thomas J Giordano, Albert M Levin, David E Misek, Lin Lin, Guoan Chen, Tarek G Gharib, Dafydd G Thomas, Michelle L Lizyness, Rork Kuick, Satoru Hayasaka, Jeremy MG Taylor, Mark D Iannettoni, Mark B Orringer, and Samir Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat Med*, 8(8):816–24, Aug 2002.

[BRS+01] A Bhattacharjee, WG Richards, J Staunton, C Li, S Monti, P Vasa, C Ladd, J Beheshti, R Bueno, M Gillette, M Loda, G Weber, EJ Mark, ES Lander, W Wong, BE Johnson, TR Golub, DJ Sugarbaker, and M Meyerson. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc Natl Acad Sci U S A*, 98(24):13790–5, Nov 2001.

[BS04] T Beissbarth and TP Speed. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics*, 20(9):1464–5, jun 2004.

[BT04] Eric Bair and Robert Tibshirani. Semi-supervised methods to predict patient survival from gene expression data. *PLoS Biol*, 2(4):E108, Apr 2004.

[CC00] Y Cheng and G Church. Biclustering of expression data. In *Intelligent System in Molecular Biology*, pages 93–103, aug 2000.

[CSF+05] G Cario, M Stanulla, BM Fine, O Teuffel, NV Neuhoff, A Schrauder, T Flohr, BW Schafer, CR Bartram, K Welte, B Schlegelberger, and M Schrappe. Distinct gene expression profiles determine molecular treatment response in childhood acute lymphoblastic leukemia. *Blood*, 105(2):821–826, Jan 2005.

[CYP+03] MH Cheok, W Yang, CH Pui, JR Downing, C Cheng, CW Naeve, MV Relling, and WE Evans. Treatment-specific changes in gene expression discriminate in vivo drug response in human leukemia cells. *Nat Genet*, 34(1):85–90, May 2003.

[DF02] S Dudoit and J Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3:R36, jun 2002.

[DSD+03] SW Doniger, N Salomonis, KD Dahlquist, K Vranizan, Lawlor SC, and B R Conklin. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biology*, 4(1):R7, jan 2003.

[DSH+03] G Jr. Dennis, BT Sherman, DA Hosack, J Yang, W Gao, HC Lane, and R A Lempicki. DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biology*, 4(5):P3, 2003.

[FCVF+04] WA Freije, FE Castro-Vargas, Z Fang, S Horvath, T Cloughesy, LM Liau, PS Mischel, and SF Nelson. Gene expression profiling of gliomas strongly predicts survival. *Cancer Res*, 64(18):6503–6510, Sep 2004.

[GBRV06] S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron. An Improved Statistic for Detecting Over-representated Gene Ontology Annotations in Gene Sets. In A Apostolico, C Guerra, S Istrail, P Pevzner, and M Waterman, editors, *Research in Computational Molecular Biology: 10th Annual International Conference, Proceedings of RECOMB 2006, Venice, Italy, April 2-5, 2006*, volume 3909 of *Lecture Notes in Computer Science*, pages 85–98. Springer, Heidelberg, 2006.

[GCB+04] RC Gentleman, VJ Carey, DM Bates, B Bolstad, M Dettling, S Dudoit, B Ellis, L Gautier, Y Ge, J Gentry, K Hornik, T Hothorn, W Huber, S Iacus, R Irizarry, F Leisch, C Li, M Maechler, AJ Rossini, G Sawitzki, C Smith, G Smyth, L Tierney, JY Yang, and J Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, 2004.

[HBV01] M Halkidi, Y Batistakis, and M Vazirgiannis. On Clustering Validation Techniques. *J. of Intell. Inform. Systems*, 17(2-3):107–45, 2001.

[HCD+03] E Huang, SH Cheng, H Dressman, J Pittman, MH Tsou, CF Horng, A Bild, ES Iversen, M Liao, CM Chen, M West, JR Nevins, and AT Huang. Gene expression predictors of breast cancer outcomes. *Lancet*, 361(9369):1590–1596, May 2003.

[HTF01] T Hastie, R Tibshirani, and J Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, 2001.

[HvHS+02] W Huber, A von Heydebreck, H Sültmann, A Poustka, and M Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(Suppl 1):96–104, 2002.

[HW79] JA Hartigan and M A Wong. A K-means clustering algorithm. *Applied Statistics*, 28:100–4, 1979.

[IG96] Ross Ihaka and Robert Gentleman. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[IHC+03] RA Irizarry, B Hobbs, F Collin, YD Beazer-Barclay, KJ Antonellis, U Scherf, and TP Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–64, Apr 2003.

[JHSV01] ES Jaffe, NL Harris, H Stein, and JW Vardiman, editors. *World Health Organization Classification of Tumours. Pathology and Genetics of Tumours of Haematopoietic and Lymphoid Tissues*, chapter 4. IARC Press, Lyon, France, 2001.

[Kan96] M Kanehisa. Toward pathway engineering: a new database of genetic and molecular pathways. *Sci & Tech Japan*, 59:34–8, 1996.

[KC01] MK Kerr and G A Churchill. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proc. Natl. Acad. Sci.*, 98(16):8961–5, jul 2001.

[KR90] L Kaufman and P J Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, 1990.

[LRBB04] Tilman Lange, Volker Roth, Mikio L Braun, and Joachim M Buhmann. Stability-based validation of clustering solutions. *Neural Comput*, 16(6):1299–323, Jun 2004.

[LS05] Claudio Lottaz and Rainer Spang. Molecular decomposition of complex clinical phenotypes using biologically structured analysis of microarray data. *Bioinformatics*, 21(9):1971–8, May 2005.

[LTS05] Claudio Lottaz, Joern Toedling, and Rainer Spang. Annotation-Driven Class Discovery. Technical Report 2005/02, Max Planck Institute for Molecular Genetics, Berlin (Germany), 2005.

[Mac67] J B MacQueen. Some Methods for classification and analysis of multivariate observations. In *Symposium on Math, Statistics, and Probability*, volume 1, pages 281–97, 1967.

[MKB79] K Mardia, J Kent, and J Bibby. *Multivariate Analysis*. Academic Press, San Diego, 1979.

[MO04] SC Madeira and A L Oliveira. Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1), Jan-Mar 2004.

[MRF+02] LM McShane, MD Radmacher, B Freidlin, R Yu, MC Li, and R Simon. Methods for assessing reproducibility of clustering patterns observed in analyses of microarray data. *Bioinformatics*, 18(11):1462–1469, Nov 2002.

[MS80] G Milligan and L Sokol. A Two Stage Clustering Algorithm with Robust Recovery Characteristics. *Educational and Psychological Measurement*, 40:755–9, 1980.

[MSK+05] S Monti, KJ Savage, JL Kutok, F Feuerhake, P Kurtin, M Mihm, B Wu, L Pasqualucci, D Neuberg, RC Aguiar, P Dal Cin, C Ladd, GS Pinkus, G Salles, NL Harris, R Dalla-Favera, TM Habermann, JC Aster, TR Golub, and MA Shipp. Molecular profiling of diffuse large B-cell lymphoma identifies robust subtypes including one characterized by host inflammatory response. *Blood*, 105(5):1851–1861, Mar 2005.

[MSS+05] Munneke, Schlauch, Simonsen, Beavis, and Doerge. Adding Confidence to Gene Expression Clustering. *Genetics*, Jun 2005.

[MTMG03] S Monti, P Tamayo, JP Mesirov, and T R Golub. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, 52(1-2):91–118, 2003.

[NMB+03] CL Nutt, DR Mani, RA Betensky, P Tamayo, JG Cairncross, C Ladd, U Pohl, C Hartmann, ME McLaughlin, TT Batchelor, PM Black, A von Deimling, SL Pomeroy, TR Golub, and DN Louis. Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Res*, 63(7):1602–1607, Apr 2003.

[PLN02] P Pavlidis, DP Lewis, and WS Noble. Exploring gene expression data with class scores. In *Proc Pacific Symposium on Biocomp*, pages 474–85, 2002.

[PTG+02] SL Pomeroy, P Tamayo, M Gaasenbeek, LM Sturla, M Angelo, ME McLaughlin, JY Kim, LC Goumnerova, PM Black, C Lau, JC Allen, D Zagzag, JM Olson, T Curran, C Wetmore, JA Biegel, T Poggio, S Mukherjee, R Rifkin, A Califano, G Stolovitzky, DN Louis, JP Mesirov, ES Lander, and TR Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–42, 2002.

[R D05] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

[RBM+01] DS Rickman, MP Bobek, DE Misek, R Kuick, M Blaivas, DM Kurnit, J Taylor, and S M Hanash. Distinctive molecular profiles of high-grade and low-grade gliomas based on oligonucleotide microarray analysis. *Cancer Res*, 61(18):6885–6891, Sep 2001.

[RDML04] J Rahnenführer, FS Domingues, J Maydt, and T Lengauer. Calculating the Statistical Significance of Changes in Pathway Activity From Gene Expression Data. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.

[RL04] Volker Roth and Tilman Lange. Feature Selection in Clustering Problems. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[RMO+04] ME Ross, R Mahfouz, M Onciu, HC Liu, X Zhou, G Song, SA Shurtleff, S Pounds, C Cheng, J Ma, RC Ribeiro, JE Rubnitz, K Girtman, WK Williams, SC Raimondi, DC Liang, LY Shih, CH Pui, and J R Downing. Gene expression profiling of pediatric acute myelogenous leukemia. *Blood*, 104(12):3679–87, Dec 2004.

[SCG+01] F Schacherer, C Choi, U Götze, M Krull, S Pistor, and E Wingender. The TRANSPATH signal transduction database: a knowledge base on signal transduction networks. *Bioinformatics*, 17:1053–7, 2001.

[Sch97] GD Schuler. Pieces of the puzzle: expressed sequence tags and the catalog of human genes. *Journal of Molecular Medicine*, 75(10):694–8, Oct 1997.

[SFR+02] D Singh, PG Febbo, K Ross, DG Jackson, J Manola, C Ladd, P Tamayo, AA Renshaw, AV D'Amico, JP Richie, ES Lander, M Loda, PW Kantoff, TR Golub, and WR Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–9, Mar 2002.

[Spe03] T Speed, editor. *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall/CRC, Florida, USA, 2003.

[STM+05] Subramanian, Tamayo, Mootha, Mukherjee, Ebert, Gillette, Paulovich, Pomeroy, Golub, Lander, and Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, Sep 2005.

[TSKS04] A Tanay, R Sharan, M Kupiec, and R Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci U S A*, 101(9):2981–6, Mar 2004.

[Tuk77] J W Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading Masschusetts, USA, 1977.

[vHHPV01] A von Heydebreck, W Huber, A Poustka, and M Vingron. Identifying splits with clear separation: a new class discovery method for gene expression data. *Bioinformatics*, 17(Suppl 1):S107–14, 2001.

[VS04] Sudhir Varma and Richard Simon. Iterative class discovery and feature selection using Minimal Spanning Trees. *BMC Bioinformatics*, 5(1):126, Sep 2004.

[WBD+01] M West, C Blanchette, H Dressman, E Huang, S Ishida, R Spang, H Zuzan, JA Olson, JR Marks, and JR Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc Natl Acad Sci U S A*, 98(20):11462–7, Sep 2001.

[WJS+04] H Willenbrock, AS Juncker, K Schmiegelow, S Knudsen, and L P Ryder. Prediction of immunophenotype, treatment response, and relapse in childhood acute lymphoblastic leukemia using DNA microarrays. *Leukemia*, 18(7):1270–1277, Jul 2004.

[YRS+02] EJ Yeoh, ME Ross, SA Shurtleff, WK Williams, D Patel, R Mahfouz, FG Behm, SC Raimondi, MV Relling, A Patel, C Cheng, D Campana, D Wilkins, X Zhou, J Li, H Liu, CH Pui, WE Evans, C Naeve, L Wong, and JR Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133–143, Mar 2002.

[ZKZL00] A Zien, R Kuffner, R Zimmer, and T Lengauer. Analysis of gene expression data with pathway scores. *Proc Int Conf Intell Syst Mol Biol*, 8:407–417, 2000.

Invited Talk

Imaging-Based Systems Biology

Gene Myers

Howard Hughes Medical Institute, Janelia Farms

Arguably the most significant contribution of the human genome project is that we can now build a recombinant construct of every gene and every promotor in C. elegans (worm), D. melanogaster (fly), M. musculus (mouse), and H. sapiens (human). These include fluorescent proteins and other markers that can be induced at controlled time points via a change in temperature, light, or chemistry. Combined with tremendous advances in light and electron microscopy in recent years, I believe we are now poised to visualize the meso-scale of the cell, and development and small organs (e.g. a fly's brain) at the resolution of individual cells.

Toward this end, my group is working on a number of preliminary imaging projects along these lines. These include (a) studies of development and gene expression in worms and flies, (b) the biophysics of mitosis, (c) neural patterning in flies and mice, and (d) the interpretation of signals from a new sub-wavelength resolution light microscope. We describe preliminary results on limited data sets and extrapolate on what we might be able to infer from such data. We further speculate on the potential implications of such work for the future of molecular biology.

Comparative Analysis of Cyclic Sequences: Viroids and other Small Circular RNAs

Axel Mosig[1,2], Ivo L. Hofacker[3] and Peter F. Stadler[4,3,5]
[1]Department of Combinatorics and Geometry (DCG),
MPG/CAS Partner Institute for Computational Biology (PICB),
Shanghai Institutes for Biological Sciences (SIBS) Campus, Shanghai, China
[2]Max Planck Insitute for Mathematics in the Sciences,
Inselstrasse 22, D-04103 Leipzig, Germany
[3]Institute for Theoretical Chemistry, University of Vienna,
Währingerstrasse 17, A-1090 Vienna, Austria
[4]Bioinformatics Group, Department of Computer Science,
and Interdisciplinary Center for Bioinformatics,
University of Leipzig, Härtelstrasse 16-18, D-04107 Leipzig, Germany.
[5]The Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, New Mexico

Abstract: The analysis of small circular sequences requires specialized tools. While the differences between linear and circular sequences can be neglected in the case of long molecules such as bacterial genomes since in practice all analysis is performed in sequence windows, this is not true for viroids and related sequences which are usually only a few hundred basepairs long. In this contribution we present basic algorithms and corresponding software for circular RNAs. In particular, we discuss the problem of pairwise and multiple cyclic sequence alignments with affine gap costs, and an extension of a recent approach to circular RNA folding to the computation of consensus structures.
Keywords: RNA secondary structure, circular RNA, dynamic programming, viroids

1 Introduction

Circular DNA is a common phenomenon in nature. Indeed, bacterial genomes as well as their plasmids are circular. Most organellar genomes of mitochondria and plastids are circular as well. In practice, however, the distinction between linear and circular sequences is irrelevant for bioinformatics at least in the case of long sequences, because the analysis will always focus on individual genes or on short sequence windows. Shorter sequences, on the other hand, with a length of, say, less than 10kb or 20kb, could be investigated as a whole. While mitochondrial genomes (with a length between 15 and 17kb for most metazoan animals and much longer for most other Eukaryote clades) have to be treated at the gene level due to rapid genomic rearrangements [BB98], this is not the case for most virus families. Proper virus genomes can be as short as 2kb, see Tab. 1. They form a heterogeneous group consisting of sequences from various viral families, two of which are retro-transcribing (Hepatitis B Virus and Caulimo-Virus), several have double-stranded

Table 1: Natural Short Circular Nucleic Acids Groups

Name	Type	Size(kb)	Seq.*	Remarks
rCarSV	ssRNA	0.2-0.3	13	viroid-like
Viroids	ssRNA	0.3-0.4	659	
Satellite RNAs	ssRNA	0.35	11	9 distinct groups
Cherry SCV	ssRNA	0.45	1	viroid-like
Deltavirus	ssRNA	1.7	51	viroid-like
Circoviridae	ssDNA virus	2	299	
Hepadnaviridae	dsDNA RT-virus	3	893	Hepatitis B
Parvoviridae	ssDNA virus	4-6	84	
Microviridae	ssDNA virus	4-6	92	
Polyomaviridae	dsDNA virus	5	447	
Geminiviridae	ssDNA virus	5	301	1 or 2 chromosomes
Nanovirus	ssDNA virus	6-9	2	4-6 chromosomes
Papillomaviridae	dsDNA virus	7-8	169	
Inoviridae	ssDNA virus	8	42	
Caulimoviridae	dsDNA RT-virus	8	57	
Corticoviridae	dsDNA virus	9	2	
Plasmaviridae	dsDNA virus	12	2	
Fuselloviridae	dsDNA virus	15	9	
Mitochondria	dsDNA virus	$\geq$13	$\sim$1000	rapid rearrangements
Plasmids $\leq$ 20kb	dsDNA		50	

NCBI Query: *Name*`[orgn] and ''complete genome''`, 2006-03-12

circular DNA genomes, and a few have short, single-stranded DNA genomes.

The overwhelming majority of single-stranded nucleic acid is RNA, and most RNA molecules are linear. The `Subviral RNA DB` [PRPP03, RP06], nevertheless lists more than 1000 circular RNA genomes of viroids and related objects. Viroids are important plant pathogens that induce symptoms similar to those accompanying virus infections. They are composed of a small, nonprotein-coding, single-stranded, circular RNA, with autonomous replication that proceeds through an RNA-based rolling-circle mechanism, see [FHMdA$^+$05] for a recent review. Several different classes of satellite RNAs are also circular [SR99]. In addition, there are three distinct classes of viroid-like sequences: cherry small circular viroid-like RNA [DSDRR97], carnation small viroid-like RNA (CarSV RNA), which is unique in that is has a DNA form and behaves similar to a retrovirus [HDB04], and Hepatitis delta virus [Tay06]. A phylogenetic analysis of viroid and viroid-like satellite RNAs can be found in [EDdlP$^+$01].

Recently, several additional (mutually unrelated) groups of circular RNAs have been discovered. In particular, alternative splicing may lead to circular RNAs from intronic sequences. This appears to be a general property of nuclear group I introns [NFB$^+$03] and was also observed during tRNA splicing in *H. volcanii* [SSGG03]. Circularized C/D box snoRNAs were recently reported in *Pyrococcus furiosus* [SMJ$^+$04]. Circular nucleic acids, furthermore, have been investigated in the context of *in vitro* selection experiments [KZG$^+$02].

Single-stranded nucleic acids, in particular, have to be treated with care when windowing techniques are used. For instance, secondary structure formation is well known to be an inherently global phenomenon. As demonstrated in [HS06], neglecting circularity can have quite dramatic effects. Substantial errors might furthermore result from the inconsistent treatment of "end-gaps" when circular sequences are aligned as if they were linear, even if the cut-point is chosen between homologous positions.

The wide variety of short circular nucleic acids listed above calls for the development of specific computational tools to deal with these exceptional sequences in a consistent way. In particular, for the case of single-stranded nucleic acids one will be interested in determining conserved RNA secondary structures, which have been demonstrated to exist in many viral RNA genomes [HSS04], in viroids [SHF$^+$84,RWR$^+$99], which were among the first RNAs for which secondary structures have been studied systematically, and also in some ssDNA viruses including parvoviridae, circoviridae, and geminiviridae [VCF05].

2 Cyclic Alignments

An alignment $\mathbb{A}$ of two strings x and y of length n and m, resp., is a sequence of pairs of the form (x_i, y_j), $(x_i, -)$, and $(-, y_j)$ that preserves the order of sequence positions in both x and y. A maximal sequence of $(x_i, -)$ pairs is called a *deletion*, while a maximal sequence of $(-, y_j)$ is called an *insertion*. We assume that the (similarity) score $S(\mathbb{A})$ of the alignment $\mathbb{A}$ is the sum of scores for individual substitutions, insertions, and deletions.

In the case of cyclic sequences, insertions and deletions may wrap around the ends, of course. Thus the cyclic score $S_C(\mathbb{A})$ is in general larger than the score $S(\mathbb{A})$ of the linear (representation of the) alignment $\mathbb{A}$. The cyclic shift operator σ that rotates a string or an alignment by one position: $\sigma(x) = (x_2, \ldots, x_{n-1}, x_n, x_1)$. The cyclic score of the alignment is thus

$$S_C(\mathbb{A}) = \max_k S(\sigma^k(\mathbb{A})) \tag{1}$$

under the above additivity assumption on the scoring model.

The cyclic string associated with an ordinary string x conveniently represented as the equivalence class $[x] = \{x, \sigma(x), \sigma^2(x), \ldots, \sigma^{n-1}(x)\}$. The cyclic alignment problem thus consists of finding the optimal linear alignment of $\sigma^p(x)$ and $\sigma^q(y)$ for the optimal choices of p and q [BB93,GT93,Mae90,MVC02]. This problem can be solved in polynomial time in full generality: One simply has to compute the optimal alignment $\mathbb{A}^{(p,q)}$ of $\sigma^p(x)$ and $\sigma^q(y)$, which can done in $\mathcal{O}(nm \max(n+m))$ time and $\mathcal{O}(nm)$ space [Dew01], for all nm combinations of p and q. This quintic algorithm is implemented in the `circal` approach to aligning mitochondrial genome arrangements [FSS06], which uses a complex gap cost function while having to deal with only short "sequences". This approach becomes impractical, however, for sequences with several hundred or thousand characters.

In the case of linear gap cost functions ($g(k) = k\delta$), [Mae90] introduced a $\mathcal{O}(n^2 \ln n)$ algorithm that is based on the fact that optimal alignment paths cannot cross in this case. The most widely used alignment programs, including `clustalW` [THG94], uses a scor-

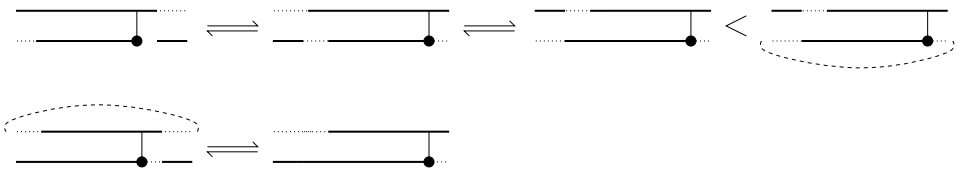


Figure 1: Alignments of x with $\sigma^q(y)$ in which the longer sequence x ends in a gap need not be considered. The "tail" of $\sigma^k y$ after the last matched position can always be rotated to the beginning of the alignment without decreasing the score. Of the three possible cases for the beginning of the alignment we show only the two non-trivial ones. If the alignment begins with a (mis)match we have the same situation as in the second case. $\leftrightharpoons$ indicates changes that do not influence the score of the alignment, $<$ indicates a possible increase in the score, and a brace means that the alignment is scored as circular alignment rather than as a linear one.

ing scheme with affine gap costs. In this model a sequence of k contiguous insertions (deletions) incurs a cost $g(k) = \delta_o + (k-1)\delta_e$ independent of the inserted or deleted characters. In the case of affine gap function, optimal alignment paths may cross, so that a direct generalization of Maes's idea does not seem to be feasible.

The alignment problem for linear strings with affine gap costs is solved by Gotoh's algorithm [Got82]: Let $S_{i,j}$ be the score of an optimal alignment of the prefixes $x[1 \ldots i]$ and $y[1 \ldots j]$; similarly, E_{ij} and F_{ij} are the optimal alignments on the prefixed subject to the constraint that they end with a deletion or an insertion, respectively. We have the following recursions

$$\begin{aligned} E_{i,j} &= \max\{E_{i-1,j} - \delta_e, S_{i-1,j} - \delta_o\} \\ F_{i,j} &= \max\{F_{i,j-1} - \delta_e, S_{i,j-1} - \delta_o\} \\ S_{i,j} &= \max\{E_{i,j}, F_{i,j}, S_{i-1,j-1} + \sigma(x_i, y_j)\} \end{aligned} \tag{2}$$

with initial conditions $S_{0,0} = 0$, $E_{0,0} = F_{0,0} = -\infty$. Combined with the understanding that terms with negative indices are set to $-\infty$, the above recursions are properly initialized.

This automatically yields a quartic algorithm for the cyclic case by simply considering all cyclic shifts of x and y. In the case of linear gap costs one easily sees that it is sufficient to use all rotations of the shorter string, which reduces the CPU requirements to $\mathcal{O}(nm \min(n,m))$, i.e. a cubic algorithm. This simplification does not work for more general gap costs, however, since gaps may "wrap around" the ends of the sequence.

In the case of *affine* gap functions we can also obtain a cubic algorithm using the fact that we have to distinguish only three cases: (1) The alignment of x with $\sigma^q(y)$ ends in a match $(x_n, (\sigma^q(y))_m)$, (2) it ends in a deletion $(x_n, -)$ or (3) it ends in an insertion $(-, (\sigma^q(y))_m)$. In the first case, the score of the cyclic alignment is the same as in the linear alignment. In case (2), we simply start the recursions with the initialization $S_{0,0} = F_{0,0} = -\infty$, $E_{0,0} = 0$, and again all terms involving negative indices considered to be $-\infty$. Consequently, we obtain the optimal score in $E_{n,m}$. Case (3) works analogously: we initialize $S_{0,0} = E_{0,0} = -\infty$, $F_{0,0} = 0$ (while again, all terms involving negative

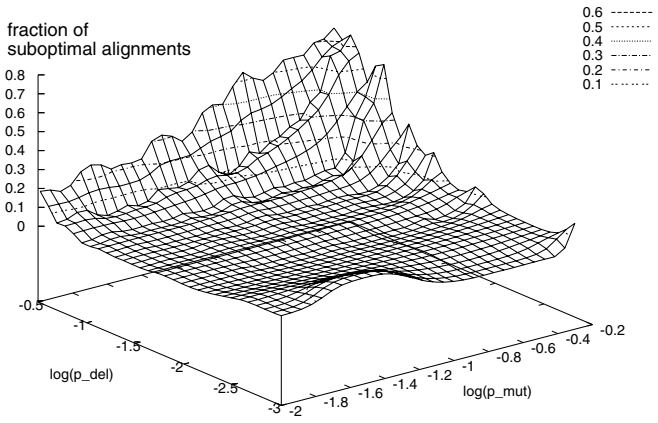


Figure 2: Fraction of non-optimal alignments computed using the best local heuristic as a function of the fraction of substitutions (0-50%) and indels (0-30%) for sequence of length $N = 100$. The average score of the heuristic alignments is above 97% of the optimal score in the entire range.

indices are considered to be $-\infty$) and obtain the optimal solution from $F_{n,m}$. Since we rotate the sequence y we do not need to consider case (3) where $(\sigma^q(y)_y)$ is unpaired since this situation can always be achieved by rotating the remaining tail after the last match in $(\sigma^q(y)_y)$ to the beginning of the string, see Fig. 1. This yields a simple $\mathcal{O}(nm \times \min(n, m))$ algorithm for the cyclic alignment problem with affine gap costs.

The close relationship between cyclic and linear alignment suggests a number of plausible heuristics. For instance, one may search for the best local alignment of x and y and use a central match from this local alignment to "anchor" the alignment of cyclic sequences. Fig. 2 summarizes the performance of this heuristic approach: as long as there is only a moderate fraction of insertions and deletions, it yields the correct solution in most cases, and only slightly sub-optimal alignments in the remaining cases.

The $\mathcal{O}(n^3)$ exact algorithm as well as the $\mathcal{O}(n^2)$ heuristic algorithms are implemented in the `cyclope` package[1].

3 Multiple Alignments and Phylogenetics

Progressive multiple alignments can be constructed by generalizing the pairwise algorithms described above to profiles in the same way as in `clustalW` [THG94]. To this end, we use the sum-of-pairs score to measure the similarity of two profiles and employ iterative clustering to construct the guide tree. After each step, similarity scores to the newly joined profile are computed explicitly rather than estimated by an averaging procedure such as WPGMA.

[1] Available from `www.bioinf.uni-leipzig.de/Software/cyclope/`.

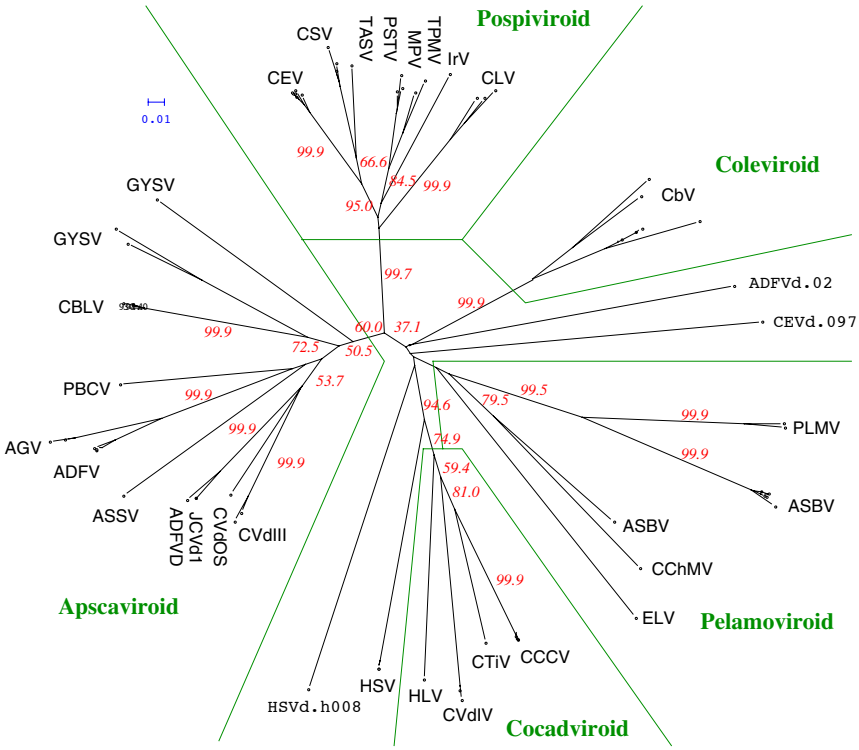


Figure 3: Neighbor-Joining tree obtained from an alignment of sequences from the `Subviral RNA DB`. Starting with 987 viroid sequences from 25 known classes of viroids, we applied `cyclope` to compute one alignment for the sequences of each of the 25 classes. These 25 alignments were used to derive 70 representative sequences, an alignment of which was taken as input to obtained the neighbor joining tree shown here. The circular multiple alignment was computed using `cyclope`, followed by a realignment step using `clustalW`. Major edges in the tree are shown along with their bootstrap values, as computed by the `SplitsTree` package.

An issue common to all progressive alignment methods, independent of whether linear or cyclic sequences are considered, is the assignment of appropriate weights to the individual sequences. Without a weighting scheme, groups of highly similar sequences tend to be overemphasized and dominate the alignment, while individual highly divergent sequences are neglected. Avoiding the costly computation of all distances in a neighbor joining tree, `cyclope` supports a simple, yet effective weighting scheme that works as follows: Initially, each individual sequence is assigned weight 1. After two alignments A and B with K and L many sequences, respectively, have been aligned to one alignment C with $M = K + L$ sequences, the weights are readjusted. Let $v_1, \ldots, v_K$ and $w_1, \ldots, w_L$ denote the original weights of the sequences in A and B. We compute the *alignment distance* (i.e., the relative number of of matching nucleotides in the alignment of A and B) $D := \mathbf{d}(A, B)$ between A and B, which then allows us to update the weights by setting $v_i := (v_i + D)/(2K)$ as well as $w_i := (w_i + D)/(2L)$. This essentially corresponds

to setting branches in the guide tree to the midpoint at each vertex, while the weights are adjusted such that small and highly divergent classes receive higher scores than large groups of similar sequences. During any alignment step, those weights are normalized such that the largest weight will be 1.

In order to utilize the much more sophisticated features of standard linear alignment packages such as `clustalW`, one can use `cyclope` to obtain a rough preliminary multiple alignment and realign those with linear alignment packages. Note that the heuristic for shifting implemented by `cyclope` is based on conserved blocks, so that the likeliness of gaps at starting positions – which linear alignment programs are not capable of handling – is kept minimal.

As a demonstration of `Cyclope` we constructed a multiple alignment of representative sequences from the `Subviral RNA DB`. Neighbor-joining was then used to infer the phylogenetic tree in Fig. 3. In general, the results are in good agreement with an earlier study [EDdlP+01]. A few details, however, deviate. For example, our data places Avocado Sunbloch Viroid ASB-Viroid within the Pelamoviroid group.

4 RNA Folding Algorithms for Circular Sequences

Now that we can compute alignments of circular RNAs, it makes sense to extend `RNAalifold` to folding of circular sequences. Approaches such as `RNAalifold` generalize single-sequence RNA folding to alignments. Michael Zuker's approach to computing both the minimum energy structure and a certain class of suboptimal folds for an RNA sequence is directly applicable to circular RNAs. In fact, `mfold` treats linear RNAs as exceptional variants of the circular ones [Zuk89, Zuk03]. In contrast, the `Vienna RNA Package`[2] [HFS+94, Hof03] optimizes the memory requirements for linear RNAs; this approach saves approximately a factor of 2 in memory as well as some CPU time. Circular RNAs can be treated as a kind of "post-processing" step of the forward recursion and as a corresponding "pre-processing" step for the the backtracking part of the folding algorithms without requiring significant additional resources or a re-design of the recursions that are optimized for the linear RNA case, see [HS06] for details.

Using the same algorithmic approach, it is straightforward to generalize `RNAalifold` [HFS02] to from linear to circular sequences. This option, which allows the computation of the consensus structure of an alignment of circular single-stranded RNA or DNA molecules, is implemented in the current version of `Vienna RNA Package`.

As an example we show the consensus structure for eggplant latent viroid in Fig. 4. The structure is very stable and moreover supported by several consistent and compensatory mutations. Unusual structural stability is sometimes used as a marker for *functional* non-coding RNAs. A well-tested measure for structural stability is the z-score comparing the consensus folding energy of an alignment with randomized alignments obtained by shuffling columns, which is computed by the `alifoldz` program [WH04]. Interestingly,

[2] Available at `http://www.tbi.univie.ac.at/RNA/`

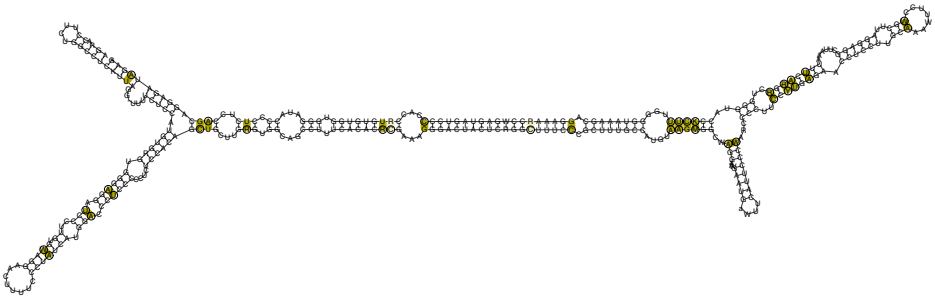


Figure 4: Consensus secondary structure for eggplant latent viroid as predicted from a `cyclope` alignment by `RNAalifold -circ`. The alignment comprises four sequence selected to have have less than 95% sequence identity. Base pairs shown in ochre are supported by a consistent or compensatory mutation with circles marking the sites of variation.

viroids exhibit the strongest signals for structural conservation of all RNAs investigated so far. While most "classical" ncRNA can be detected by `alifoldz` at a z-score cut-off of -4, the ELVd alignment exhibits a z-sore of -13.8.

5 Concluding Remarks

So far, circular RNA and DNA sequences have been considered as a rather exotic side-line that do not warrant specialized tools. With increasing sequence information becoming available, the (ab)use of methods that are designed for linear sequences becomes increasingly tedious as it requires manual corrections of both alignments and subsequent analysis. In this contribution we presented a dedicated alignment tool for (short) circular sequences, which is particularly geared towards viroids and small virus RNAs. While the time complexity is higher than classical alignment algorithms, it is efficient enough in practice for use with viroid and other subviral sequences. Based on the pairwise dynamic programming alignments, `cyclope` also features a `clustal`-like progressive alignment tool. These alignments can be used without further processing for phylogeny reconstruction or RNA secondary structure analysis.

In the case of viroid phylogeny, previous studies were essentially confirmed based on extended data sets. An RNA consensus folding program for circular RNAs, which combines the `RNAalifold` approach with a recent algorithm for folding circular RNAs, shows that viroids have exceptionally strong signals for structural stability when compared to other functional ncRNAs.

Acknowledgments. This work has been funded, in part, by the Austrian GEN-AU projects "bioinformatics integration network II" and "non coding RNA", and by the German *DFG* Bioinformatics Initiative BIZ-6/1-2. PFS thanks PICB for its hospitality during a visit in March 2006, during which

a significant part of this work was conceived.

References

[BB93] H. Bunke und U. Bühler. Applications of approximate string matching to 2D shape recognition. *Patt. Recogn.*, 26:1797–1812, 1993.

[BB98] J. L. Boore und W. M. Brown. Big trees from little genomes: mitochondrial gene order as a phylogenetic tool. *Curr. Opinion Gen. Devel.*, 8:668–674, 1998.

[Dew01] T. G. Dewey. A sequence alignment algorithm with an arbitrary gap penalty function. *J. Comp. Biol.*, 8:177–190, 2001.

[DSDRR97] F Di Serio, J A Daros, A Ragozzino und Flores R. A 451-nucleotide circular RNA from cherry with hammerhead ribozymes in its strands of both polarities. *J Virol.*, 71:6603–6610, 1997.

[EDdlP+01] Santiago F. Elena, Joaquín Dopazo, Marcos de la Peña, Ricardo Flores, Theodor O. Diener und Andrés Moya. Phylogenetic Analysis of Viroid and Viroid-Like Satellite RNAs from Plants: A Reassessment. *J. Mol. Evol.*, 53:155–159, 2001.

[FHMdA+05] R Flores, C Hernandez, A E Martinez de Alba, J A Daros und F. Di Serio. Viroids and viroid-host interactions. *Annu. Rev. Phytopathol.*, 43:117–139, 2005.

[FSS06] Guido Fritzsch, Martin Schlegel und Peter F. Stadler. Alignments of Mitochondrial Genome Arrangements: Applications to Metazoan Phylogeny. *J. Theor. Biol.*, 240:511–520, 2006.

[Got82] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162:705–708, 1982.

[GT93] J. Gregor und M. G. Thomason. Dynamic programming alignment of sequences representing cyclic patterns. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15:129–135, 1993.

[HDB04] K Hegedus, G Dallmann und E. Balazs. The DNA form of a retroviroid-like element is involved in recombination events with itself and with the plant genome. *Virology*, 325:277–286, 2004.

[HFS+94] Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, Sebastian Bonhoeffer, Manfred Tacker und Peter Schuster. Fast Folding and Comparison of RNA Secondary Structures. *Monatsh. Chemie*, 125:167–188, 1994.

[HFS02] Ivo L. Hofacker, Martin Fekete und Peter F. Stadler. Secondary Structure Prediction for Aligned RNA Sequences. *J. Mol. Biol.*, 319:1059–1066, 2002.

[Hof03] Ivo L. Hofacker. Vienna RNA secondary structure server. *Nucl. Acids Res.*, 31:3429–3431, 2003.

[HS06] Ivo L. Hofacker und Peter F. Stadler. Memory Efficient Folding Algorithms for Circular RNA Secondary Structures. *Bioinformatics*, 22:1172–1176, 2006. **See also:** Proceedings of the German Conference on Bioinformatics. GCB 2003 Torda, A., Kurtz, S. and Rarey, M., Lecture Notes in Informatics P-71, pp.15-25 (2005).

[HSS04] Ivo L. Hofacker, Roman Stocsits und Peter F. Stadler. Conserved RNA Secondary Structures in Viral Genomes: A Survey. *Bioinformatics*, 20:1495–1499, 2004. **See also:** Proceedings of the German Conference on Bioinformatics. GCB 2003 vol. **1**; Mewes, H.-W., Heun, V., Frishman, D. and Kramer, S. (eds.); belleville Verlag Michael Farin, München, D (2003), pp. 57-62.

[KZG+02] X. D. Kong, S. Z. Zhu, X. J. Gou, X. P. Wang, H. Y. Zhang und J. Zhang. A circular RNA-DNA enzyme obtained by in vitro selection. *Biochem. Biophys. Res. Commun.*, 292:1111–1115, 2002.

[Mae90] M. Maes. On a cyclic string-to-string correction problem. *Inform. Process. Lett.*, 35:73–78, 1990.

[MVC02] R. A. Mollineda, E. Vidal und F. Casacuberta. Cyclic sequence alignments: approximate versus optimal techniques. *Int. J. Pattern Rec. Artif. Intel.*, 16:291–299, 2002.

[NFB+03] H. Nielsen, T. Fiskaa, A. B Birgisdottir, P. Haugen, C. Einvik und S. Johansen. The ability to form full-length intron RNA circles is a general property of nuclear group I introns. *RNA*, 9:1464–1475, 2003.

[PRPP03] M. Pelchat, L. Rocheleau, J. Perreault und J-P Perreault. SubViral RNA: a database of the smallest known auto-replicable RNA species. *Nucleic Acids Res.*, 31:444–445, 2003.

[RP06] Lynda Rocheleau und Martin Pelchat. The Subviral RNA Database: a toolbox for viroids, the hepatitis delta virus and satellite RNAs research. *BMC Microbiology*, 6:24, 2006. doi:10.1186/1471-2180-6-24.

[RWR+99] D. Repsilber, S. Wiese, M. Rachen, A. W. Schroder, D. Riesner und G. Steger. Formation of metastable RNA structures by sequential folding during transcription: time-resolved structural analysis of potato spindle tuber viroid (-)-stranded RNA by temperature-gradient gel electrophoresis. *RNA*, 5:574–584, 1999.

[SHF+84] G. Steger, H. Hofmann, J. Fortsch, H. J. Gross, J W Randles, H L Sanger und D. Riesner. Conformational transitions in viroids and virusoids: comparison of results from energy minimization algorithm and from experimental data. *J. Biomol. Struct. Dyn.*, 2:543–571, 1984.

[SMJ+04] Natalia G. Starostina, Sarah Marshburn, L. Steven Johnson, Sean R. Eddy, Rebecca M. Terns, und Michael P. Terns. Circular box C/D RNAs in *Pyrococcus furiosus*. *Proc. Natl. Acad. Sci. USA*, 101:14097–14101, 2004.

[SR99] R H Symons und J W. Randles. Encapsidated circular viroid-like satellite RNAs (virusoids) of plants. *Curr. Top. Microbiol. Immunol.*, 239:81–105, 1999.

[SSGG03] Shilpa R. Salgia, Sanjay K. Singh, Priyatansh Gurha und Ramesh Gupta. Two reactions of *Haloferax volcanii* RNA splicing enzymes: Joining of exons and circularization of introns. *RNA*, 9:319–330, 2003.

[Tay06] J. M. Taylor. Hepatitis delta virus. *Virology*, 344:71–76, 2006.

[THG94] J. D. Thompson, D. G. Higgs und T. J. Gibson. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties, and weight matrix choice. *Nucl. Acids Res.*, 22:4673–4680, 1994.

[VCF05] Ramachandran Vanitharani, Padmanabhan Chellappan und Claude M. Fauquet. Geminiviruses and RNA silencing. *Trends Plant Sci.*, 10:144–151, 2005.

[WH04] Stefan Washietl und Ivo L. Hofacker. Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J. Mol. Biol.*, 342:19–39, 2004.

[Zuk89] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[Zuk03] Michael Zuker. `Mfold` web server for nucleic acid folding and hybridization prediction. *Nucl. Acids Res.*, 31:3406–3415, 2003.

Invited Talk

Combining Sequence Information with T-Coffee

Cedric Notredame
CNRS, Marseille, France

Well integrated biological data lends itself to the identification of biologically meaningful patterns. Multiple Sequence Alignments constitute one of the most powerful ways of carrying out such a task. In this context, the integration takes the form of simultaneously aligning related sequences in order to reveal evolutionary conserved patterns. Multiple Sequence Alignments have so many applications that they have become household items in biology and few data processing pipelines exist that do not require the assembly of an alignment. Yet, the wealth of available alternative methods means that the user is not only faced with the problem of selecting and aligning sequences, but also with the necessity of choosing one method or integrating the results delivered by many. In the course of this seminar I will discuss how various methods can be integrated into one. I will also go further and show that a multiple sequence alignments can be used to integrate much more than sequence information, as long as this information is properly mapped onto the sequences. This concept, named template-based multiple sequence alignment will be illustrated with a simple example: the combination of sequences and structures within multiple sequence alignments. I will finally discuss how multiple sequence alignment methods are currently validated and why I believe we need to challenge these procedures in order to take further our understanding of biological sequences. Most of the tools discussed in this talk are available from www.tcoffee.org.

References:

1. Armougom, F., S. Moreti, O. Poirot, S. Audic, V. Kedaus, and C. Notredame, *Expresso: Automatic Combination of Sequence and Structural Information.* Nucleic Acids Res, 2006. **In Press**.
2. Wallace, I.M., O. O'Sullivan, D.G. Higgins, and C. Notredame, *M-Coffee: combining multiple sequence alignment methods with T-Coffee.* Nucleic Acids Res, 2006. **34**(6): p. 1692-9.
3. O'Sullivan, O., K. Suhre, C. Abergel, D.G. Higgins, and C. Notredame, *3DCoffee: combining protein sequences and structures within multiple sequence alignments.* J Mol Biol, 2004. **340**(2): p. 385-95.
4. Notredame, C., D.G. Higgins, and J. Heringa, *T-Coffee: A novel method for fast and accurate multiple sequence alignment.* J Mol Biol, 2000. **302**(1): p. 205-17.

PALMA: Perfect Alignments using Large Margin Algorithms

G. Rätsch,[a] B. Hepp,[b] U. Schulze,[a] and C.S. Ong[a,c]

[a] Friedrich Miescher Lab., Max Planck Society, Spemannstr. 39, 72076 Tübingen, Germany,
[b] Fraunhofer FIRST, Kekulèstr. 7, 12489 Berlin, Germany,
[c] Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany

Abstract: Despite many years of research on how to properly align sequences in the presence of sequencing errors, alternative splicing and micro-exons, the correct alignment of mRNA sequences to genomic DNA is still a challenging task. We present a novel approach based on large margin learning that combines kernel based splice site predictions with common sequence alignment techniques. By solving a convex optimization problem, our algorithm – called PALMA – tunes the parameters of the model such that the true alignment scores higher than all other alignments. In an experimental study on the alignments of mRNAs containing artificially generated micro-exons, we show that our algorithm drastically outperforms all other methods: It perfectly aligns all 4358 sequences on an hold-out set, while the best other method misaligns at least 90 of them. Moreover, our algorithm is very robust against noise in the query sequence: when deleting, inserting, or mutating up to 50% of the query sequence, it still aligns 95% of all sequences correctly, while other methods achieve less than 36% accuracy. For datasets, additional results and a stand-alone alignment tool see `http://www.fml.mpg.de/raetsch/projects/palma`.

1 Introduction

Many genomes have been sequenced recently. This is only a first step to understand what the genome actually encodes. Fortunately, most of them also come with rather large amounts of expressed sequence tags (ESTs; sequenced parts of mRNA), which help to accurately recognize genes and to identify the exon/intron boundaries as well as alternative splice forms (see [ZG06] and references therein).

Many methods for aligning ESTs to genomic DNA have been proposed, including approaches based on *blast* [AGM$^+$90], *spliced alignments* [GMP96], *sim4* [FHZ$^+$98], *GeneSeqer* [UZB00], *Spidey* [WS01], *blat* [Ken02], an approach to find additional microexons [VHS03] and most recently *exalin* [ZG06]. The identification of exon/intron boundaries is important for finding the correct alignment. Therefore most approaches try to find an alignment that prefers splice site consensus signals in the identified introns (usually `GT/AG`, considerably less often `GC/AG` and in some organisms also `AT/AC`) that help to accurately identify these boundaries. This is done by employing either dynamic programming or sophisticated heuristics.

[ZG06] used an information theoretic approach to combine the two types of information available during alignment: the sequence similarity and splice site predictions. Given this model, dynamic programming is used to compute the maximum-log likelihood alignment. Our algorithm, called *PALMA*, is based on similar ideas as *exalin*. The main difference is

the modeling of splice sites using support vector machines, the modeling of intron lengths and the large margin based combination of the different types of information. Our approach does not include any probabilistic models and hence does not return probabilities for a particular alignment. It is, however, able to very accurately and robustly align sequences as will be seen in the experimental section where we consider the problem of aligning modified EST sequences to genomic DNA (here of the model organism *C. elegans*) using the most difficult setup: We consider artificially generated short internal exons (2-50nt) combined with small to large amounts of noise in the query sequence. We show that our method perfectly aligns all sequences while other methods fail as soon as the exons become too short or the amount of noise too large.

2 Method

The idea of our algorithm is to compute an alignment by dynamic programming that uses a scoring function. We tune the parameters of the scoring functions such that the true alignment does not only achieve a large score (to be "most likely"), but also that all other alignments score considerably lower than the true alignment (to obtain a "large margin between the alignments"). Similar ideas are used in other large margin algorithms such as Support Vector Machines [Vap95] and Boosting [FS97]. Also, a similar approach for aligning protein sequences (without intron related gaps) has been proposed independently by [JGE05]. The resulting scoring function can then be maximized using dynamic programming in order to obtain the optimal alignment. Our method consists of three independent parts: the splice site prediction model, the dynamic programming algorithm and the optimization of the scoring function which we describe in the following sections.

Training the splice site model and also the large margin combination requires separate sequence data sets. For the splice site model, we used genes that were EST confirmed but without full length cDNA support (set 1). We consider a random subset of 40% of all cDNA confirmed genes without evidence for alternative splicing for training the large margin combination (set 2). The remaining 20% and 40% were used for hyper-parameter tuning (set 3) and final evaluation (set 4) respectively.

2.1 Splice Site Predictions

From the set of EST sequences (set 1) we extracted sequences of confirmed donor (intron start) and acceptor (intron end) splice sites (see Appendix A for details). For acceptor splice sites we used a window of 80bp upstream to 60bp downstream of the site (on the DNA). For donor sites we used 60bp upstream and 80bp downstream. Also from these training sequences we extracted non-splice sites that are within an exon or intron of the sequence and have `AG` (acceptor) or `GT`/`GC` (donor) consensus. In order to recognize acceptor and donor splice sites, we trained two Support Vector Machine classifiers [Vap95] with soft-margin using the so-called "weighted degree" kernel [SRJM02, RSS06]. The kernel mainly takes positional information (relative to the splice site) about the appearance of certain motifs into account. It computes the scalar product between two sequences $\mathbf{x}$ and $\mathbf{x}'$:

$$\mathrm{k}(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{d} v_j \sum_{i=1}^{N-j} \mathbf{I}(x_{[i,i+j]} = x'_{[i,i+j]}), \tag{1}$$

where $N = 140$ is the length of the sequence and $x_{[a,b]}$ denotes the substring of x from position a to (excluding) b. The function $\mathbf{I}$ is the indicator function, $\mathbf{I}(\text{true}) = 1$, $\mathbf{I}(\text{false}) = 0$ and the weights $v_j := d - j + 1$. We used a normalization of the kernel $\tilde{k}(s_1, s_2) = \frac{k(s1,s_2)}{\sqrt{k(s_1,s_1)k(s_2,s_2)}}$ and $d = 22$ for the recognition of splice sites. Additionally, the regularization parameter of the Support Vector Machine was set to be $C = 2$ and $C = 3$ for acceptor and donor sites, respectively. All parameters (including the window size, regularization parameters etc) have been tuned on data set 2 (cf. [RSS05]).

Given a DNA sequence as target of an alignment we can now use the two SVMs to compute scores for each position with corresponding consensus[1] for being a splice acceptor or donor site, respectively. Since we consider *C. elegans* where U12 splicing is extremely rare or not present, we exclude the `AT/AC` splice sites from our splice site model.

2.2 Needleman-Wunsch Alignments with Intron Model

The classical deterministic and exact alignment algorithm is the Needleman-Wunsch algorithm and is based on dynamic programming. Its running time is $\mathcal{O}(m \cdot n)$, where m is the length of the EST sequence S_E, and n is the length of the DNA sequence S_D. It builds up a $m \cdot n$ matrix and hence has the same space complexity.

The main idea of the algorithm is to compute an overall alignment by determining the maximum over all alignments of all prefixes $S_E(1:i) := (S_E(1), \ldots, S_E(i))$ and $S_D(1:j) := (S_D(1), \ldots, S_D(j))$ of the two sequences S_E and S_D, respectively. An alignment is given by a sequence of pairs (a_r, b_r), $r = 1, \ldots, R$, where $R \leq m + n$ depends on the alignment and $a_r, b_r \in \Sigma := \{A, C, G, T, N, -\}$. A pair consists either of the two corresponding letters of the two sequences or a single letter in one sequence paired with a gap in the other sequence. The alignment is scored using a substitution matrix M, which we interpret as a function $M : \Sigma \times \Sigma \to \mathbb{R}$. Then the score for the alignment $\mathcal{A} = \{(a_r, b_r)\}_r$ is simply $\sum_r M(a_r, b_r)$.

We define $V(i, j)$ to be the score of the best possible alignment of prefixes $S_E(1:i)$ and $S_D(1:j)$. Then $V(n, m)$ can be computed using the following recurrence formula (for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$):

$$V(i,j) = \max \begin{cases} V(i-1, j-1) + M(S_E(i), S_D(j)) \\ V(i-1, j) + M(S_E(i), '-') \\ V(i, j-1) + M('-', S_D(j)) \end{cases} \tag{2}$$

The recurrence is initialized with $V(0,0) := 0$, $V(i,0) := 0$ and $V(0,j) := 0$ for all $i = 1 \ldots m$ and $j = 1 \ldots n$. There are three possibilities:

- $S_E(i)$ and $S_D(j)$ are aligned to each other (possibly a mismatch).
- $S_E(i)$ is aligned to a gap in the DNA sequence.
- $S_D(j)$ is aligned to a gap in the EST sequence.

In the original setting there are only these three possibilities and one can straightforwardly fill the matrix from left to right and top to bottom to finally compute $V(n, m)$. The optimal alignment can then be obtained by backtracking [DEKM98].

[1] `AG` for acceptor sites and `GT` or `GC` for donor sites.

The Needleman-Wunsch algorithm only aligns the single bases of two sequences and does not distinguish between exons and introns – it essentially treats everything as exons. We therefore propose to extend the Needleman-Wunsch algorithm to better model introns: We allow an additional "intron transition" that is separately scored based on its length and the scores of splice sites at its ends. We denote by $f_I(i_1, i_2)$ the intron scoring function for an intron starting at i_1 and ending at i_2. The intron scoring function $f_I(i_1, i_2)$ is computed based on the intron length $i_2 - i_1$, the donor SVM output $g_{don}(i_1)$ for position i_1 and the acceptor SVM output $g_{acc}(i_2)$ for position i_2. During learning we determine three functions f_ℓ, f_{acc} and $f_{don} : \mathbb{R} \to \mathbb{R}$ to combine these three values:

$$f_I(i_1, i_2) = f_\ell(i_2 - i_1) + f_{don}(g_{don}(i_1)) + f_{acc}(g_{acc}(i_2)). \tag{3}$$

When there is no donor consensus at position i_1, then we define $f_{don}(g_{don}(i_1)) := -\infty$ (similarly for $f_{acc}(g_{acc}(i_2))$). Given the intron scoring function f_I we can now restate the recurrence formula (for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$):

$$V(i,j) = \max \begin{cases} V(i-1, j-1) + M(S_E(i), S_D(j)) \\ V(i-1, j) + M(S_E(i), '-') \\ V(i, j-1) + M('-', S_D(j)) \\ \max_{1 \le k \le j-1}(V(i,k) + f_I(k,j)) \end{cases} \tag{4}$$

where we consider the additional possibility of an intron starting at position k and ending at j. Please note that the above recurrence formula is considerably more computationally expensive than the previous one: every step involves finding the optimal intron start ($\mathcal{O}(n)$). However, one only needs to consider those positions where the intron start and end exhibit the corresponding splice consensus sites and also the splice site predictors output large enough scores. Additional strategies for speeding up such algorithms are discussed in [ZG06].

For completeness we need to extend our notation for alignments with introns. We use again alignment pairs $\mathcal{A} = \{(a_r, b_r)\}_r$, but extend the alphabet for a_r to $\Sigma \cup \{+\}$ ("intron sequence missing") and for b_r to $\Sigma \cup \{*\}$ ("intron sequence"). Note that b_r should only contain strings of length greater than one if $a_r = '+'$. Then the score $f(\mathcal{A})$ for an alignment $\mathcal{A}$ with intron is computed as before, i.e. $\sum_r M(a_r, b_r)$, except when $a_r = +$: In this case the intron score function is used to score the corresponding intron.

2.3 Large Margin Combination

In the previous section we assumed that the functions f_{acc}, f_{don} and f_ℓ as well as the substitution matrix M were given. We now describe a algorithm to determine these parameters based on the training set of sequences and their true alignments.

Two methods based on a similar idea have been independently proposed in [JGE05] and [KK06]. They both present a simpler algorithm for learning the substitution matrix required for aligning protein sequences. [KK06] presents an algorithm–based on the method from [GBN94]–that can learn hundreds of parameters simultaneously and is able to model affine gap-costs. [JGE05] propose an algorithm related to support vector machines. However, both approaches do not model introns or splice sites explicitly and are therefore expected to fail in identifying microexons.

Note that our proposed algorithm is two-layered: First one learns the splice site model

and then how to combine the different pieces of information. In principle these two steps can be combined to one step. Then the piecewise linear functions can be replaced with linear combinations of kernel elements as similarly done in [ATH03]. However, this makes training much slower and is not expected to improve the results in our case.

Since the three functions are one-dimensional, it suffices to use a simple piecewise linear model: Let s be the number of supporting points x_i (satisfying $x_i < x_{i+1}$) and y_i their values, then the piecewise linear function is defined by

$$f(x) = \begin{cases} y_1 & x \leq x_1 \\ \frac{y_i(x_{i+1}-x)+y_{i+1}(x-x_i)}{x_{i+1}-x_i} & x_i \leq x \leq x_{i+1} \\ y_s & x \geq x_s \end{cases} . \tag{5}$$

After having appropriately chosen supporting points on the x-axis we only need to optimize the corresponding y-values. For f_{acc} and f_{don} we use 30 supporting points uniformly sampled between -5 and $+5$ (our SVM outputs are typically not larger). For f_ℓ we use 30 log-uniformly sampled supporting points between 30nt and 1000nt.[2] Given the three functions and the substitution matrix, the alignment scoring function $f(\mathcal{A})$ is fully specified. Moreover, for a given alignment $\mathcal{A}$, it can be verified that $f(\mathcal{A})$ is linear in all parameters that we denote by $\boldsymbol{\theta}$, i.e. in the values of the substitution matrix and the y-values of the three piecewise linear functions, $\boldsymbol{\theta} := (\boldsymbol{\theta}_{acc}, \boldsymbol{\theta}_{don}, \boldsymbol{\theta}_\ell, \boldsymbol{\theta}_M)$.

2.3.1 Optimization

For training we are given a set of N true alignments $\mathcal{A}_i^+$, $i = 1, \ldots, N$. The goal is to find the parameters $\boldsymbol{\theta}$ of the alignment scoring function f such that the difference of scores $f_\theta(\mathcal{A}_i^+) - f_\theta(\mathcal{A}^-)$ is large for *all* wrong alignments $\mathcal{A}^- \neq \mathcal{A}_i^+$. This can be done by solving the following convex optimization problem:

$$\min_{\boldsymbol{\xi}\geq\mathbf{0},\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N}\xi_i \qquad \text{s.t.} \qquad f_\theta(\mathcal{A}_i^+) - f_\theta(\mathcal{A}^-) \geq 1 - \xi_i \quad \forall i \text{ and } \mathcal{A}^- \neq \mathcal{A}_i^+. \tag{6}$$

Here we introduced so-called slack-variables ξ_i to implement a soft-margin [CV95]. The above optimization problem has exponentially many constraints and cannot be easily solved directly. Instead one adopts a column generation technique (cf. [HK93] and references therein) and for every true alignment one maintains a set of wrong alignments $\mathcal{A}_{i,j}^- \neq \mathcal{A}_i^+$, for all j. Initially this set is empty but it can easily be filled by running the dynamic programming algorithm discussed in the previous section to generate wrong alignments (based on some arbitrary initial parameters). Then one solves the following optimization problem

$$\min_{\boldsymbol{\xi}\geq\mathbf{0},\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N}\xi_i \qquad \text{s.t.} \qquad f_\theta(\mathcal{A}_i^+) - f_\theta(\mathcal{A}_{i,j}^-) \geq 1 - \xi_i \quad \forall i, j \tag{7}$$

Given a set of wrong alignments one can now determine the intermediate optimal parameters $\boldsymbol{\theta}$, and further use the dynamic programming algorithm to find other wrong alignments to be included in the set of wrong alignments. The procedure is iterated and provably converges to the solution of (6) in a finite number of steps (in our application often not more than 10 iterations).

[2] For other organisms one might want to choose a larger range.

2.3.2 Regularization

In empirical inference it is common to regularize the parameters in order not to overfit. We implement this by adding a regularization term $C\mathbf{P}(\boldsymbol{\theta})$ to (6), where C is the regularization constant and $\mathbf{P}$ a regularization function. Recall that the parameter vector consists of four parts, and we define the regularization term as follows:

$$\mathbf{P}(\boldsymbol{\theta}) = \sum_{i=1}^{n-1}(\theta_{acc,i+1}-\theta_{acc,i})^2+\sum_{i=1}^{n-1}(\theta_{don,i+1}-\theta_{don,i})^2+\sum_{i=1}^{n-1}(\theta_{\ell,i+1}-\theta_{\ell,i})^2+\sum_{a,b} M(a,b)^2.$$

It implements the idea that the piecewise-linear functions should be smooth and the values in the substitution matrix small.

3 Results and Discussion

Most alignment algorithms work very well for aligning mRNA sequences against genomic DNA when query and target perfectly match and the matching blocks are long enough. In our experimental study we are interested in the most difficult cases, where most algorithms start to fail. If an algorithm works in such case we expect that it will also return correct alignments for easier cases.

We evaluate our proposed method, *PALMA*, and other methods such as (*exalin*, *sim4* & *blat*). We consider the alignment of mRNA sequence fragments containing three exons where we artificially shortened the middle exon (final length of 2-50nt, see Appendix B for details). Artificially generating the data has the benefit of knowing exactly what the correct alignment has to be. Additionally, we add considerable amounts of noise ($p = 0\%, 1\%, 10\%, 20\%, 50\%$ of random mutations, deletions and insertions) to the query sequence. We then measure how often the methods find the middle exons and the whole alignment correctly. The evaluation is done on a separate test set which was not used during training of our method (set 4, cf. Appendix B).

The model selection for the splice site predictors have been performed on separate validation sets (set 2). Model selection of regularization parameter C in our method (cf. Section 2.3.2) was done by simple validation on a separate validation set (set 3). While the method was trained on noise-free data, we applied it to the noisy versions during validation since otherwise the validation error rate was always zero, almost independently of the choice of C. We determined $C = 0.01$ as optimal regularization constant. To analyze the importance of the splice site predictions relative to the sequence similarity for correct alignments, we additionally trained a second model that does not use splice site information (but only intron lengths and the substitution matrix). We call it *PALMA* without splice sites (SS).

Figure 1 shows the alignment error rate for different methods on the 4358 test sequences. Here we counted an alignment as a mistake if the exon boundaries deviated by at least one nucleotide. [3] We also looked at how often the middle exon has been correctly identified.

[3] For *exalin* we noticed that the alignment is very often off by 2nt. We assume that this is a fixable bug in the *exalin* implementation. For fairness we therefore allowed deviations of $\pm$2nt for exalin only. The problem often occurs for high noise levels. For instance at $p = 20\%$ we find 20% error rate for the strict evaluation, while only 6% error when using the relaxed criterion.

We observed that in most cases an alignment error is induced by the inaccurate identification of the middle exon.[4] From our results in Figure 1 we observe that there are drastic differences between the methods. Almost all methods perform reasonably well when the query perfectly matches the target – with the exception of *sim4* which has problems aligning at least 18% of the sequences. For *blat* and *sim4* the error rates drastically increase when adding noise to the query sequence. Only *exalin* and *PALMA* (with and without splice site information) have low error rates for noise levels of at most 20%. When deleting, inserting or mutating up to 50% of the query sequence, PALMA (with splice sites) still aligns 95% of all sequences correctly, while the other methods achieve less than 36% accuracy. For high noise levels the splice site information helps to reduce the error rate considerably. But also in the low noise cases the splice site predictions help to accurately identify very short exons that can be found ambiguously in the intronic regions (0.4% of the test sequences).

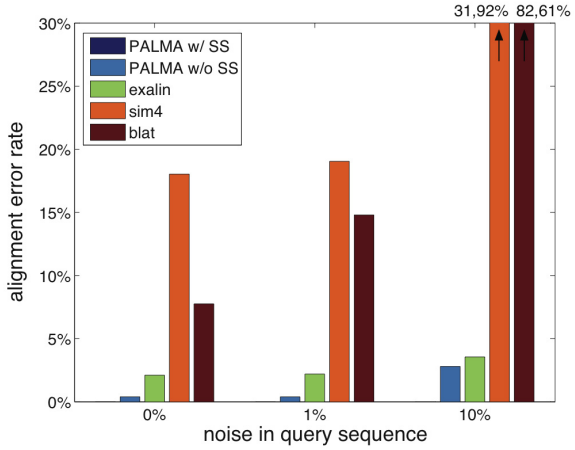


Figure 1: Comparison of different methods for aligning mRNAs to genomic DNA: We considered the particularly difficult task of aligning exon triples with short middle exons (2-50nt) in the presence of noise. Although an alignment is already declared as true if the intron boundaries are correct, only PALMA (with splice sites) achieves 0% error rate for aligning queries with up to 10% noise.

Figures 2-4 show the optimized parameters determined by our algorithm. For the piecewise linear functions in 2 we obtain smooth sigmoid-shaped functions ("differences between large score values do not matter"). Comparing with Figure 4 we observe that the difference between a weak and a strong splice site is worth about 3-4 matches, since the substitution matrix contains values between -0.4 and $+0.4$. Figure 3 illustrates the piecewise linear function for scoring intron lengths. We observe that the maximum coincides with the most frequent intron length of around 50nt. The optimized substitution matrix is essentially diagonal, which is not surprising as there was no preference for substitutions in our data.

4 Conclusion

We have proposed a new alignment algorithm that computes the optimal alignment of mRNA sequences to genomic DNA while exploiting existing very accurate kernel-based splice site predictions. In a simulation study on aligning sequences with very short exons and considerable amounts of noise we have shown that our method achieves significantly

[4] Since it gives a very similar figure, we omitted it from the manuscript.

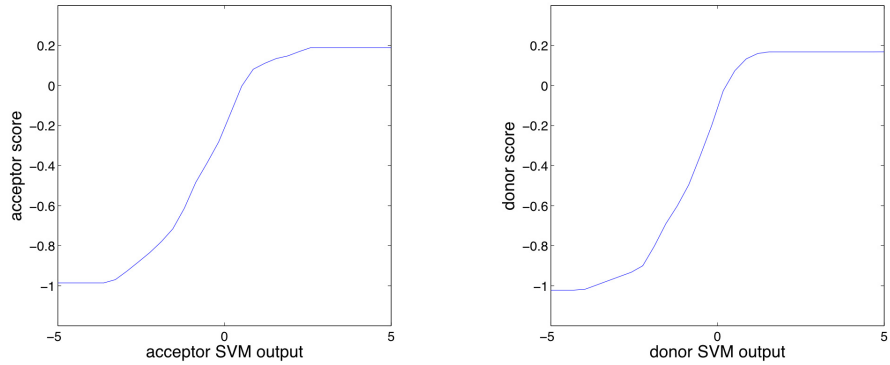


Figure 2: *PALMA*'s optimized functions f_{acc} and f_{don} scoring acceptor and donor SVM outputs.

lower error rates than other methods. This indicates that the proposed method would be more effective than current approaches for discovering microexons, i.e. exons between 2-25nt in length. This is especially true in the presence of sequencing errors or mutations which may render current approaches and heuristics inaccurate. In addition, by combining it with other methods such as *blast* we can reduce the computational cost in order to apply our method for alignments of ESTs to whole-genomes.

Acknowledgments

We thank G. Schweikert and N. Toussaint for proofreading the manuscript.

A Processing of Sequence Databases

We collected all known *C. elegans* ESTs from Wormbase [HCC$^+$04] (release WS120; 236,893 sequences) and dbEST [BT93] (as of February 22, 2004; 231,096 sequences). Using *blat* [Ken02] we aligned them against the genomic DNA (release WS120). The alignment was used to confirm exons and introns. We refined the alignment by correcting typical sequencing errors, for instance by removing minor insertions and deletions. If an intron did not exhibit the consensus `GT`/`AG` or `GC`/`AG` at the 5' and 3' ends, then we tried to achieve this by shifting the boundaries up to 2 base pairs (bp). If this still did not lead to the consensus, we split the sequence into two parts and considered each subsequence separately. In a next step we merged consistent alignments, if they shared at least one complete exon or intron. This lead to a set of 124,442 unique EST-based sequences.

We repeated the above procedure with all known cDNAs from Wormbase (release WS120; 4,855 sequences). These sequences only contain the coding part of the mRNA. We used their ends as annotation for start and stop codons.

We clustered the sequences in order to obtain independent training, validation and test sets. In the beginning each of the above EST and cDNA sequences were in a separate cluster. We iteratively joined clusters, if any two sequences from distinct clusters match to the same genomic location (this includes many forms of alternative splicing). We obtained

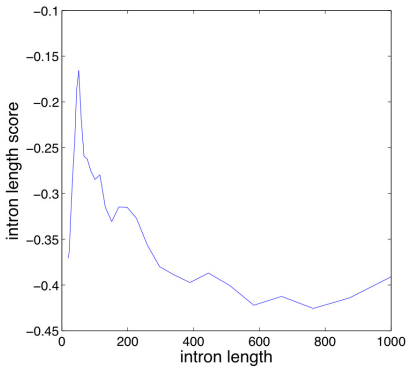


Figure 3: Shown is the optimized intron length scoring function f_ℓ: The maximum is located at around 50nt, which is also the most frequent intron length in *C. elegans*.

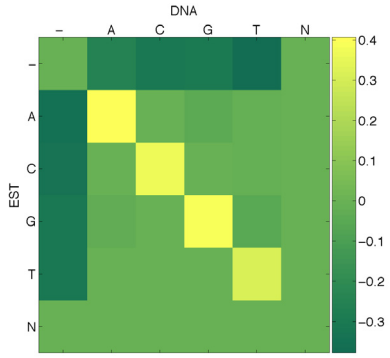


Figure 4: Shown is the optimized substitution matrix: matches score high and gaps low. Mismatch scores are all close to zero and do not contribute much.

21,086 clusters, while 4072 clusters contained at least one cDNA.

For *set 1* we chose all clusters not containing a cDNA (17215), for *set 2* we chose 40% of the clusters containing at least one cDNA (1536). For *set 3* we used 20% of clusters with cDNA (775). The remaining 40% of clusters with at least one cDNA (1,560) were used as *set 4*. Sets 2-4 were filtered to remove confirmed alternative splice forms. This left 1,177 cDNA sequences for *testing* in set 4 with an average of 4.8 exons per gene and 2,313bp from the 5' to the 3' end.

B Artificial Microexon Dataset

Based on sets 2-4 described in the last section we created sets of consecutive exon triples from the confirmed transcripts in these sets. This lead to 4604, 2257 and 4358 triples. In a first processing step we shortened the middle exons to a random length between 2nt and 50nt (uniformly distributed). To do so, we removed the correct number of nucleotides from the center of the middle exon – from the query as well as the DNA. This leaves the splice sites mostly functional. In a second step we added varying amounts of noise. For a given noise level p and a query sequence of length L, we first replaced $p \cdot L/3$ positions with a random letter ($\Sigma = \{A, C, G, T, N\}$). Then we deleted the same number of non-overlapping positions in the sequence and added the same number of random nucleotides at random positions. We used $p = 0\%, 1\%, 10\%, 20\%, 50\%$.

References

[AGM+90] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal Molecular Biology*, 215(3):403–10, 1990.

[ATH03] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov Support Vector Machines. In *Proc. 20th International Conference on Machine Learning*, 2003.

[BT93] M.S. Boguski and T.M. Lowe C.M. Tolstoshev. dbEST–Database for "Expressed Sequence Tags". *Nat Genet.*, 4(4):332–3, 1993.

[CV95] C. Cortes and V.N. Vapnik. Support Vector Networks. *Machine Learning*, 20:273–297, 1995.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of protein and nucleic acids*. Cambridge University Press, 1998. 7th edition.

[FHZ+98] L. Florea, G. Hartzell, Z. Zhang, G.M. Rubin, and W. Miller. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research*, 8:967–974, 1998.

[FS97] Y. Freund and R.E. Schapire. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[GBN94] D. Gusfield, K. Balasubramanian, and D. Naor. Parametric Optimization of Sequence Alignment. *Algorithmica*, 12:312–326, 1994.

[GMP96] M.S. Gelfand, A.A. Mironov, and P.A. Pevzner. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci.*, 93(17):9061–6, 1996.

[HCC+04] T.W. Harris, N. Chen, F. Cunningham, et al. WormBase: a multi-species resource for nematode biology and genomics. *Nucl. Acids Res.*, 32, 2004. Database issue:D411-7.

[HK93] R. Hettich and K.O. Kortanek. Semi-Infinite Programming: Theory, Methods and Applications. *SIAM Review*, 3:380–429, September 1993.

[JGE05] T. Joachims, T. Galor, and R. Elber. Learning to Align Sequences: A Maximum-Margin Approach. In B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, and A. Mark, editors, *New Algorithms for Macromolecular Simulation*, number 49 in LNCS, pages 57–71. Springer, 2005.

[Ken02] W. J. Kent. BLAT–the BLAST-like alignment tool. *Genome Res*, 12(4):656–664, April 2002.

[KK06] J. D. Kececioglu and E. Kim. Simple and Fast Inverse Alignment. In *RECOMB*, pages 441–455, 2006.

[RSS05] G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: Recognition of Alternatively Spliced Exons in *C. elegans*. *Bioinformatics*, 21(Suppl. 1):i369–i377, June 2005.

[RSS06] G. Rätsch, S. Sonnenburg, and C. Schäfer. Learning Interpretable SVMs for Biological Sequence Classification. *BMC Bioinformatics*, 7(Suppl 1):S9, February 2006.

[SRJM02] S. Sonnenburg, G. Rätsch, A. Jagota, and K.-R. Müller. New Methods for Splice-Site Recognition. In *Proc. International Conference on Artificial Neural Networks*, 2002.

[UZB00] J. Usuka, W. Zhu, and V. Brendel. Optimal spliced alignment of homologous cDNA to a genomic DNA template. *Bioinformatics*, 16(3):203–211, 2000.

[Vap95] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.

[VHS03] N. Volfovsky, B.J. Haas, and S.L. Salzberg. Computational Discovery of Internal Micro-Exons. *Genome Research*, 13:1216–1221, 2003.

[WS01] Ostell JM. Wheelan SJ, Church DM. Spidey: a tool for mRNA-to-genomic alignments. *Genome Research*, 11(11):1952–7, 2001.

[ZG06] M. Zhang and W. Gish. Improved spliced alignment from an information theoretic approach. *Bioinformatics*, 22(1):13–20, January 2006.

Invited Talk

Pushing details into interaction networks

Rob Russell

EMBL, Heidelberg, Germany

Many experiments suggest that pairs of proteins are involved in physical interactions, though few give any insights as to the details of how they are mediated. We have worked on inferrring details at various levels from interaction networks. I will discuss our attempts to: infer details of interaction strength from purification data (and use this to deduce complexes, ref. 1), model interactions within complexes using three-dimensional structures (2), and identify new modes of domain/peptide recognition involved in mediating interactions (3).

References:

1. Gavin AC, Aloy P, et al, Russell RB, Superti-Furga G. Proteome survey reveals modularity of the yeast cell machinery. Nature. 2006 440(7084):631-6.

2. Aloy P, Russell RB. Structural systems biology: modelling protein interactions. Nat Rev Mol Cell Biol. 2006 7(3):188-197.

3. Neduva V, Linding R, Su-Angrand I, Stark A, de Masi F, Gibson TJ, Lewis J, Serrano L, Russell RB. Systematic discovery of new recognition peptides mediating protein interaction networks. PLoS Biol.2005, 3(12):e405.

Functional evaluation of domain-domain interactions and human protein interaction networks

Andreas Schlicker[1], Carola Huthmacher, Fidel Ramírez,
Thomas Lengauer, and Mario Albrecht[2]

Department of Computational Biology and Applied Algorithmics
Max Planck Institute for Informatics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
andreas.schlicker@mpi-inf.mpg.de
mario.albrecht@mpi-inf.mpg.de

Abstract: Large amounts of protein and domain interaction data are being produced by experimental high-throughput techniques and computational approaches. To gain insight into the value of the provided data, we used our new similarity measure based on the Gene Ontology to evaluate the molecular functions and biological processes of interacting proteins or domains. The applied measure particularly addresses the frequent annotation of proteins or domains with multiple Gene Ontology terms. Using our similarity measure, we compare predicted domain-domain and human protein-protein interactions with experimentally derived interactions. The results show that our similarity measure is of significant benefit in quality assessment and confidence ranking of domain and protein networks. We also derive useful confidence score thresholds for dividing domain interaction predictions into subsets of low and high confidence.

1 Introduction

Experimental high-throughput techniques have produced enormous amounts of protein-protein interaction (PPI) data for different species [1]. These data can now be mined for new information on the functions and interrelationships of proteins [2]. In particular, different bioinformatics methods, mainly based on the homology of protein sequences, have supported the large-scale prediction of human protein networks [3-8]. Additionally, manually curated literature data and four large-scale yeast-2-hybrid maps have recently become available for the human interactome [9-13]. However, in contrast to predicted data, the experimental coverage of the human interactome is still low. To predict protein interaction networks, domain-domain interactions (DDIs) are often taken into account [8, 14-16]. For this purpose, different sets of DDIs have been predicted using bioinformatics methods [16-18] and supplement experimental DDI sets derived from 3D structure data [19, 20].

---

[1]Presenting author at GCB
[2]Corresponding author

The Gene Ontology (GO) consortium provides a standardized vocabulary that is commonly used to annotate genes and their products with biological processes and molecular functions [21]. This annotation particularly allows for assessing the functional similarity of genes or their products. Resnik [22] and Lin [23] introduced semantic similarity measures for the comparison of single terms in "is-a" ontologies. Both measures are based on the information content of ontology terms. Based on these semantic similarity measures, several methods for the functional comparison of gene products have been introduced. Lu *et al*. [24] and Lin *et al.* [25] evaluated the usefulness of different features, ranging from expression profiles to functional relationships between genes, for the prediction of PPIs. They concluded that functional similarity based on GO annotation leads to high accuracy in predicting PPIs. Wu *et al.* also introduced new similarity measures between GO terms and proteins [26]. Their measures were used to create a predicted network of PPIs and to evaluate genome-scale datasets. Very recently, Guo *et al*. assessed the applicability of GO-based similarity measures to human regulatory pathways [27]. They showed that the functional similarity between two proteins decreases as their distance within the same regulatory pathway increases.

One problem with existing GO-based similarity measures is that they do not account for the frequent annotations of gene products or protein domains with multiple GO terms or that they simply average over all annotations. To address this problem, we use our novel GO similarity measure that explicitly deals with this functional multiplicity [28]. The measure is applied to ranking the interaction networks and the corresponding prediction methods based on the overall functional similarity of the interacting proteins or domains. The comparison of experimentally derived sets with predicted sets of DDIs using our GO similarity measure results in confidence score thresholds separating low- and high-confidence subsets of predicted DDIs. In addition, we utilize our measure to analyze experimental and predicted networks of human protein interactions.[3]

2 Materials and Methods

2.1 Experimental and predicted datasets

Two experimental sets of DDIs were taken from iPfam [19] and the database of 3D interacting domains (3did) [20] and compared to three sets of predicted interactions between Pfam-A domains [29]. The first predicted set is InterDom, a database of putatively interacting domains compiled from different data sources [17]. The other two sets are taken from two recent publications by Liu, Liu, and Zhao (LLZ) [16] and by Riley *et al.* (domain pair exclusion analysis, DPEA). Their bioinformatics approaches are methodological extensions of an expectation-maximization algorithm first applied to the prediction of domain interactions by Deng *et al.* in 2002 [15]. The DDI prediction methods assign a confidence score (CS) to each DDI and rank the predicted DDIs

[3]Abbreviations: ATX, ataxin; BP, biological process; CS, confidence score; DDI, domain-domain interaction; GO, Gene Ontology; HTT, huntingtin; MF, molecular function; PPI, protein-protein interaction; Y2H, yeast two-hybrid.

according to the score. InterDom uses different data sources to infer DDIs and calculates the CS based on the support from each source [17]. LLZ and DPEA compute maximum-likelihood estimates to derive a CS, and we use the probability λ and the log-odds score E as CS from LLZ and DPEA, respectively [16, 18]. The pfam2go file from the GO web site (http://www.geneontology.org/external2go/pfam2go) contains a mapping of Pfam-A domains to GO terms. This file (downloaded on July 7, 2005) was used to annotate the Pfam-A domains with GO terms. Table 1 summarizes the number of DDIs in each dataset.

Table 1: Total number of Pfam-A domains in the different datasets of DDIs (column 'Total'). The columns for biological process ('BP') and molecular function ('MF') contain the fraction of interactions whose interacting domains are both annotated with GO.

Dataset	Total	BP (%)	MF (%)
iPfam	3,046	52.07	56.30
3did	3,034	49.51	54.19
InterDom	29,957	27.07	37.64
LLZ	9,160	17.75	19.64
DPEA	3,005	22.40	24.19

We also analyzed six predicted sets of human PPIs named Bioverse [6], HiMAP [8], HomoMINT [7], Sanger [4], OPHID [5], and POINT [3]. Additionally, subsets of core interactions with high confidence were derived from Bioverse, HiMAP and Sanger. The Bioverse-core set contains very reliable interactions based on a sequence similarity threshold of at least 80% between human and the homolog of the source species [30], HiMAP-core interactions have a large likelihood ratio [8], and Sanger-core comprises only predictions with the greatest experimental support [4]. Additionally, we assembled five consensus sets named ConSet***n*** that consist of protein interactions contained in at least ***n*** predicted interactomes, with ***n*** ranging from 2 to 6.

As experimental datasets, we downloaded the manually curated human protein reference database (HPRD) [13], release of 13 September 2005, and two yeast two-hybrid (Y2H) maps that we named 'Vidal' [10] and 'Wanker' [11] after the senior authors. We also merged the two Y2H maps into the combined dataset Vidal & Wanker. Both Y2H maps and the HPRD data became available after the six predicted human networks were published. Further experimental PPIs were extracted from the published networks of direct and indirect interaction partners for ataxins (ATX) [12] and huntingtin (HTT) [9]. These networks include Y2H and literature-derived datasets, which we call ATX-/HTT-Y2H and ATX-/HTT-literature, respectively. The ATX-interologs set comprises interactions from the ATX network that have been derived by mapping interologs [12], and thus we regard it as another predicted set of PPIs. Generally, the diverse gene and protein accession numbers of the PPI sets were mapped to NCBI Entrez gene identifiers [31]. The mapping of Entrez gene identifiers to GO annotations was obtained from NCBI (ftp://ftp.ncbi.nih.gov/gene/DATA/gene2go.gz). Furthermore, we compiled another set of PPIs using the interacting proteins that underlie iPfam DDIs with both domains belonging to different proteins. This set was annotated from two different sources, that is, with the GO annotation from the UniProt release 5.4 (IUP-set) and with GO terms from the pfam2go file (IPG-set).

2.2 Functional similarity measure

The GO controlled vocabulary consists of three different ontologies: biological process (BP), molecular function (MF), and cellular component. The ontologies are organized as directed acyclic graphs with terms being represented as nodes and parent-child relationships as edges. There are two types of edges: "is-a" links, indicating that the child is an instance of its parent, and "part-of", used if the child is a component of its parent. Each node may have several parents and children.

Our semantic similarity measure is an extension of previous measures by Resnik and Lin [22, 23]. As suggested by Resnik, we defined the probability of a term as its relative frequency of occurrence in a set of annotated gene products. The root node of each ontology has the probability 1. We used the GO annotation of all proteins in the UniProt release 5.4 for the calculation of term frequencies. The semantic similarity of two terms is defined as follows:

$$sim(t_1, t_2) = \max_{a \in CA} \left(\frac{2 * \log p(a)}{\log p(t_1) + \log p(t_2)} * (1 - p(a)) \right),$$

where t_1 and t_2 are GO terms, $p(t_1)$ and $p(t_2)$ their probabilities, and CA is the set of their common ancestors in the graph. This similarity measure takes into account how similar and detailed both terms t_1 and t_2 are, and it ranges from 0 (for terms that only have the root node in common) to 1.

This semantic similarity measure for single GO terms can be expanded to a functional similarity measure of gene products. Let g_1 and g_2 be two gene products annotated with the GO term sets GO^1 and GO^2 of size N and M, respectively. The similarity matrix S containing all pair-wise similarity values is computed as

$$s_{ij} = sim(GO_i^1, GO_j^2), \forall i \in \{0,..., N\}, \forall j \in \{0,..., M\}.$$

The row vectors and column vectors of matrix S represent the two possible directions of comparing g_1 and g_2. While the similarity computed from g_1 to g_2 (*rowScore*) is defined as the average over the row maxima, the similarity from g_2 to g_1 (*columnScore*) is defined as the average over the column maxima:

$$rowScore = \frac{1}{N} \sum_{i=1}^{N} \max_{1 \le j \le M} s_{ij} \; ; \; columnScore = \frac{1}{M} \sum_{j=1}^{M} \max_{1 \le i \le N} s_{ij} \; .$$

The *rowScore* and the *columnScore* are always between 0 and 1. Furthermore, we define the functional similarity of two gene products with respect to one ontology as

$$GOscore(g_1, g_2) = \max\{rowScore(g_1, g_2), columnScore(g_1, g_2)\}.$$

We refer to this *GOscore* as *MFscore* for MF and *BPscore* for BP. One important aspect of this score is that it allows for comparing gene products with multiple functions. This

property is especially important when comparing GO annotations of domains because they occur in diverse proteins involved in different processes. For more details on our GO similarity measure, see Schlicker *et al.* [28].

3 Results and Discussion

3.1 Comparing confidence scores for domain interactions

The predictions of DDIs by InterDom, LLZ and DPEA are compiled from diverse data sources using different bioinformatics methods. To gain insight into the similarity and the quality of the predictions, we compared the predicted sets of DDIs with each other and to the experimentally derived sets iPfam and 3did. The overlap of the datasets InterDom, LLZ and DPEA regarding Pfam-A domains as well as regarding their predicted interactions are given in Table 2. LLZ and DPEA share many Pfam-A domains and predicted DDIs with InterDom, while the overlap between LLZ and DPEA is much smaller.

Table 2: Overlap of the InterDom, LLZ and DPEA datasets with regard to Pfam-A domains and predicted domain interactions. Each number refers to the percentage of domains or interactions in the row datasets that are also contained in the respective column dataset. Percentages in parentheses give the number of DDIs shared between two datasets in ratio to the overall number of DDIs with interacting domains contained in both datasets.

	Pfam-A domains (%)			Domain-domain interactions (%)		
Dataset	**InterDom**	**LLZ**	**DPEA**	**InterDom**	**LLZ**	**DPEA**
InterDom	100.0	44.4	25.1	100.0 (100.0)	11.4 (19.3)	4.8 (23.2)
LLZ	79.3	100.0	26.9	58.8 (72.7)	100.0 (100.0)	10.6 (60.8)
DPEA	86.5	51.9	100.0	78.9 (89.3)	32.9 (62.2)	100.0 (100.0)

Figure 1 and Table S1 give an overview of the overlap of the experimental interactions contained in iPfam and 3did and the three sets of predicted interactions InterDom, LLZ and DPEA. 11.8% of the DDIs predicted by DPEA are confirmed by iPfam or 3did, whereas only 7.4% and 3.0% of the DDIs predicted by InterDom and LLZ, respectively, are in common with iPfam or 3did. Thus, DPEA appears to be the best of the three prediction methods.

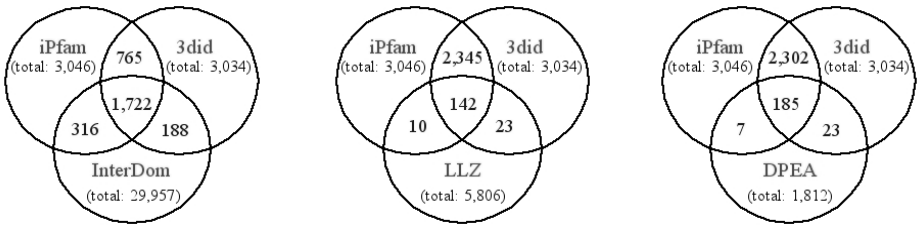


Fig. 1: Overlap of the datasets containing predicted or experimental Pfam-A domain interactions.

Other criteria for prediction quality are the CS and the rank assigned to domain interactions observed experimentally. The distributions of CSs show that many interactions in iPfam and 3did receive a high CS by LLZ and a low CS by InterDom and DPEA (Figure S1). However, DDIs contained in iPfam and 3did are assigned top ranks by all three prediction methods (Figure S2). Surprisingly, further analyses indicate only weak correlations between CSs and ranks of different prediction methods (Figures S3-S5). However, DDIs from iPfam that are predicted by two different computational methods are assigned a good rank by at least one method. This suggests that all methods are able to detect correct domain interactions. Further details on the results are described in the online supplement.

3.2 Background distribution and randomized domain networks

In order to obtain a background distribution, all available Pfam-A domains (release 17.0) were mapped to BP and MF terms of GO, and the distributions of the *MFscore* and *BPscore* for all pairs of Pfam-A domains were calculated (Figure S6). Apparently, most domain pairs have very low *MFscore*, which indicates that the molecular functions of the domains are generally quite distinct. The mean is about 0.1 and the median is 0. The *BPscore* is distributed similarly, but there are fewer domain pairs with *BPscore* below 0.1. This finding is also reflected by the higher mean and median of 0.23 and 0.17, respectively. These results indicate that the *BPscore* should generally be higher than the *MFscore*.

Subsequently, we randomized all DDI networks in our analysis to determine a possible bias towards specific functions or processes. This was accomplished by keeping one of the two nodes of the interaction edges fixed while randomly shuffling the other nodes of the edges. The obtained distributions are all very similar and closely resemble the background distribution for BP and MF (Figures S7 and S8). The distributions of the randomized experimental iPfam and 3did networks for BP contain more DDIs with *BPscore* below 0.1, but fewer with *BPscore* between 0.1 and 0.2 in contrast to the predicted datasets. The means and medians of all randomized experimental and predicted networks are similar, suggesting that neither of the networks is biased towards specific processes or functions.

3.3 Computing and analyzing *GOscore* distributions

The *BPscore* distributions for iPfam and 3did (Figure 2) show that most DDIs have a very high similarity score exceeding 0.8, which means that the corresponding interacting domains are part of the same process or closely related processes. This is supported by high means of about 0.9 and medians of almost 1. The distributions for the predicted sets InterDom or DPEA look alike. Interestingly, only one third of the predicted interactions have a *BPscore* above 0.8. Furthermore, both datasets include a large fraction of interactions with *BPscore* below 0.4, indicating almost no functional similarity between the domains. The mean is 0.51 for both datasets and the medians 0.39 and 0.41 for InterDom and DPEA, respectively. The LLZ predictions contain substantially fewer interactions with high *BPscore*, and many more interactions with very low *BPscore*. This is reflected by the relatively low mean of 0.35 and the median of 0.2. In summary,

DPEA performs slightly better than InterDom, and both show a better performance than LLZ.

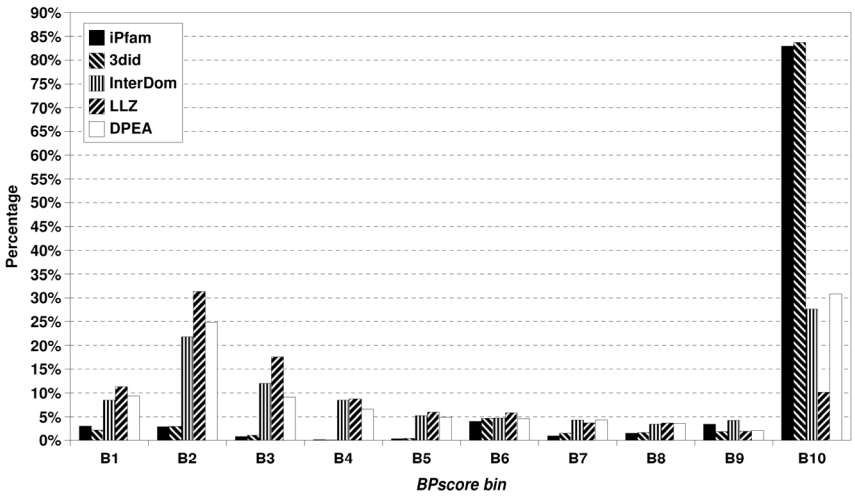


Fig. 2: BPscore distribution for the different datasets of experimental DDIs (iPfam and 3did) and predicted DDIs (InterDom, LLZ and DPEA). The BPscore bins correspond to the following intervals: B1: [0.0, 0.1[; B2: [0.1, 0.2[; B3: [0.2, 0.3[; B4: [0.3, 0.4[; B5: [0.4, 0.5[; B6: [0.5, 0.6[; B7: [0.6, 0.7[; B8: [0.7, 0.8[; B9: [0.8, 0.9[; B10: [0.9, 1.0].

Figure S9 contains the *MFscore* distributions of all datasets. Interestingly, the distributions for iPfam and 3did are quite distinct from the other distributions. Almost 80% of the domain interactions in iPfam or 3did have an *MFscore* above 0.8, which indicates related molecular functions annotated to the interacting domains. In addition, both sets contain very few interactions with very low *MFscore*. The means of over 0.8 and the medians of almost 1 corroborate this interpretation. The predictions made by InterDom and DPEA show similar distributions, but rather low means and medians. Similar to the findings for the *BPscore* distribution, predictions made by LLZ show a lower *MFscore*. As in the case of the *BPscore* distribution, InterDom and DPEA have similar performance and both perform significantly better than LLZ.

3.4 Deriving confidence score thresholds

The methods InterDom, LLZ and DPEA all provide CSs for the prediction of DDIs. However, in order to utilize sets of predicted interactions in practice, it is important to derive reasonable thresholds for low- and high-confidence sets of DDIs. It is to be expected that the functional similarity of domains predicted to interact increases as the confidence in these predictions rises. To verify this expectation, we used different CS thresholds to calculate the *GOscore* means and medians of all interactions with a CS larger than the respective threshold. We also calculated the overlap of these interactions with iPfam and 3did.

Figure 3 shows the change in *BPscore* mean and median, and the change in dataset size with varying CS threshold for the DPEA dataset. When raising the DPEA CS threshold

from 3 to 6, the *BPscore* median increases from slightly over 0.4 to almost 1, and the mean raises from 0.51 to approximately 0.7. The *MFscore* median and the overlap with iPfam and 3did show a steep increase in this CS range (Figures S10 and S11). Consequently, we suggest assigning predictions with a CS between 3 and 6, and above 6 to DPEA subsets of low- and high-confidence DDIs, respectively.

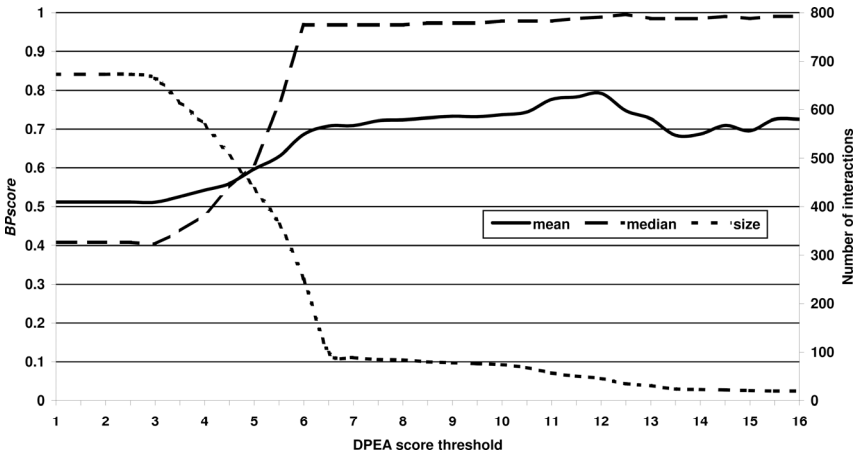


Fig. 3: Change in BPscore mean and median, and in dataset size with varying confidence score threshold for DPEA. Size refers to the number of DDIs with confidence score above the threshold.

The analysis of the InterDom set reveals that the *BPscore* median reaches 0.98 with a CS threshold of 30 (Figure S12). The *BPscore* mean is 0.68 at this point and increases with higher thresholds. The same score development holds true for *MFscore*, but it is shifted slightly towards higher thresholds (Figure S13). At a threshold of 60, the dataset consists of 1,888 interactions and the median increase diminishes. The overlap with iPfam and 3did increases with rising InterDom score and is about 27% for a threshold of 60 (Figure S14). Altogether, these results suggest a threshold of 60 for InterDom predictions of high confidence.

The analysis of LLZ predictions reveals that the *BPscore* mean and median, and the overlap with iPfam and 3did are very low over the whole CS range (Figures S15-S17). These results do not allow deriving any reasonable CS threshold for some LLZ subset of DDIs.

3.5 Comparing human protein interaction networks

We calculated the *BPscore* for all datasets of PPIs. Table 3 summarizes the results ranked by the average *BPscore*. The *BPscore* means range from 0.82 for Bioverse-core to 0.37 for Wanker PPI set. While the average *BPscores* for the predicted datasets vary significantly, the experimental Y2H datasets have rather low mean *BPscore*. In contrast, predicted datasets such as both HiMAP datasets and Bioverse-core as well as the manually curated sets HPRD and HTT-literature receive high mean scores. The different results for the HTT and ATX networks also indicate that literature-curated, carefully

validated, PPIs reach a higher *BPscore* than PPIs derived by high-throughput experiments.

Table 3: Ranking of predicted and experimental protein networks based on BPscore. The column 'Scored' contains the fraction of PPIs with an assigned BPscore. The two rightmost columns give the percentages of PPIs contained in HPRD or the combined Y2H set Vidal & Wanker.

Dataset	Interactions	Scored (%)	*mean BPscore*	HPRD (%)	Vidal & Wanker (%)
Bioverse-core	3,266	83.2	0.823	28.9	1.1
IPG-set	1,931	45.9	0.815	15.9	0.7
HiMAP-core	8,832	84.6	0.813	9.1	0.6
HiMAP	38,378	89.4	0.799	3.8	0.2
IUP-set	1,931	22.8	0.764	15.9	0.7
ConSet6	484	77.5	0.709	21.3	1.2
HPRD	20,121	86.1	0.662	100.0	0.6
HTT-literature	428	97.4	0.643	90.2	0.2
ConSet5	1,565	73.2	0.642	16.1	1.3
Bioverse	233,941	81.4	0.572	1.5	0.1
ConSet3	10,844	66.5	0.561	9.2	0.8
ConSet4	4,747	67.1	0.559	10.2	0.9
ConSet2	38,258	69.3	0.556	6.0	0.4
Sanger-core	11,131	65.3	0.551	4.5	0.6
ATX-literature	4,796	67.5	0.537	46.9	39.1
HomoMINT	10,870	57.5	0.510	5.6	0.7
OPHID	28,255	62.6	0.499	4.4	0.2
Vidal	2,754	40.2	0.471	3.5	100.0
HTT-Y2H	164	62.2	0.456	3.8	5.1
POINT	98,528	56.9	0.451	2.6	0.2
Sanger	67,518	62.3	0.427	1.3	0.1
ATX-interologs	1,527	62.0	0.418	6.8	1.2
ATX-Y2H	770	39.9	0.394	1.4	1.0
Wanker	2,033	54.8	0.370	1.2	100.0

The *BPscore* means of the iPfam-derived IUP- and IPG-sets with the same PPIs, but distinct GO annotations, are 0.76 and 0.81, respectively. These values are lower than the mean of the corresponding DDIs in iPfam, which may be partly due to the fact that we excluded self-interactions in the two PPI sets. The score distributions for the IUP- and IPG-sets show that using the GO annotation of proteins or Pfam domains leads to different results (Figure S18). In contrast to the small increase in mean *BPscore*, the distributions of the IUP- and IPG-sets differ significantly. In comparison, the manually curated HPRD set has a mean similarity measure of 0.66. The distribution of this set shows that over 50% of the interactions have a *BPscore* above 0.7 (Figure S20). However, 10% of the interactions have a score between 0.1 and 0.2. The consensus PPI

sets ConSet1-4 show a similar mean *BPscore*, and ConSet5 and ConSet6 score higher, but they constitute small interaction sets only.

Especially on the lower ranks, the *BPscore* ranking of the datasets is similar to rankings resulting from the computed HPRD or Y2H verification rate (Table 3), that is, the percentage of interactions contained in HPRD or the combined Y2H dataset Vidal & Wanker. The predicted Bioverse-core set and the consensus sets have the best verification rates with respect to HPRD. The fact that the Vidal and Wanker sets have published validation rates of 78% and 62-66%, respectively, agrees well with the slightly higher mean *BPscore* 0.47 of Vidal in contrast to the mean 0.36 of Wanker [10, 11]. The lower mean *BPscore* of Wanker may also be due to the use of many protein fragments in contrast to full-length proteins employed by Vidal [10, 11].

4 Conclusions

Following the idea that interacting domains or proteins should have highly similar biological process (BP) annotation and, to a smaller degree, similar molecular function (MF) annotation, we evaluated the functional similarity of three predicted and two experimental domain-domain interaction (DDI) networks as well as several predicted and experimental human protein-protein interaction (PPI) networks. Furthermore, we analyzed to which extent predicted DDIs or PPIs overlap with experimentally derived interactions.

We demonstrated that the application of functional similarity measures is not restricted to the validation of PPIs [27], but also useful for DDIs. Our analysis of DDIs revealed that the BP similarity of interacting domains is generally higher than the corresponding MF similarity. This observed difference between BP and MF similarity agrees well with findings by Guo *et al*. for PPIs using other GO similarity measures [27]. The difference may be partly due to the fact that interacting domains or proteins may perform different functions though they act in similar processes. Another reason may be that GO terms are more densely connected in the top levels of the BP ontology than of the MF ontology.

The iPfam-derived IUP- and IPG-sets encompass the same PPIs, but the IUP-set is annotated with the GO terms of the proteins in UniProt and the IPG-set with the GO terms of the Pfam domains. The comparison of these two sets revealed that the *BPscore* results depend on the annotation used. This indicates that the choice of the annotation source contributes to the differing findings for DDIs and PPIs. Moreover, a higher number of proteins annotated with diverse BPs may decrease the mean *BPscore* of protein networks in contrast to sets of DDIs annotated with more generic GO terms.

In agreement with our results on human protein interaction networks, Reguly *et* al. observed for yeast interaction datasets that the GO annotation of literature-curated PPI sets is more coherent than the GO annotation of high-throughput PPI sets [32]. Since manually curated datasets of PPIs taken from scientific literature have a higher mean *BPscore* than most predicted and high-throughput sets, the latter sets may contain a significant number of false interactions or a large amount of proteins involved in novel processes. This can lead to a considerable decrease in *BPScore*. Furthermore, proteins described in the literature may be annotated particularly well with GO. Therefore, a

more thorough analysis of the PPI results using alternative measures will be required to explain differences between predicted and experimental datasets.

Our functional similarity analysis in conjunction with an evaluation of the overlap between experimentally derived and predicted DDIs allowed the definition of confidence score thresholds for DDI prediction results. These thresholds are useful for improving PPI predictions based on DDIs as well as for assessing the confidence of PPIs derived by high-throughput experiments. In the future, incorporating other similarity criteria besides GO may improve the confidence assessment of predicted interactions further. As the coverage and quality of GO annotations improves, the importance of approaches that use functional similarity for the validation and prediction of PPIs and DDIs will increase.

Acknowledgements

We are grateful to Francisco S. Domingues and the anonymous reviewers for useful comments on the manuscript. Part of this study was financially supported by the German National Genome Research Network (NGFN) and by the German Research Foundation (DFG), contract number KFO 129/1-1. This work was conducted in the context of the BioSapiens Network of Excellence funded by the European Commission under grant number LSHG-CT-2003-503265.

References

1. Sharan, R., Ideker, T.: Modeling cellular machinery through biological network comparison. Nat Biotechnol **24** (2006) 427-433
2. Bork, P., Jensen, L.J., von Mering, C., Ramani, A.K., Lee, I., Marcotte, E.M.: Protein interaction networks from yeast to human. Curr Opin Struct Biol **14** (2004) 292-299
3. Huang, T.W., Tien, A.C., Huang, W.S., Lee, Y.C., Peng, C.L., Tseng, H.H., Kao, C.Y., Huang, C.Y.: POINT: a database for the prediction of protein-protein interactions based on the orthologous interactome. Bioinformatics **20** (2004) 3273-3276
4. Lehner, B., Fraser, A.G.: A first-draft human protein-interaction map. Genome Biol **5** (2004) R63
5. Brown, K.R., Jurisica, I.: Online predicted human interaction database. Bioinformatics **21** (2005) 2076-2082
6. McDermott, J., Bumgarner, R., Samudrala, R.: Functional annotation from predicted protein interaction networks. Bioinformatics **21** (2005) 3217-3226
7. Persico, M., Ceol, A., Gavrila, C., Hoffmann, R., Florio, A., Cesareni, G.: HomoMINT: an inferred human network based on orthology mapping of protein interactions discovered in model organisms. BMC Bioinformatics **6 Suppl 4** (2005) S21
8. Rhodes, D.R., Tomlins, S.A., Varambally, S., Mahavisno, V., Barrette, T., Kalyana-Sundaram, S., Ghosh, D., Pandey, A., Chinnaiyan, A.M.: Probabilistic model of the human protein-protein interaction network. Nat Biotechnol **23** (2005) 951-959
9. Goehler, H., Lalowski, M., Stelzl, U., Waelter, S., Stroedicke, M., Worm, U., Droege, A., Lindenberg, K.S., Knoblich, M., Haenig, C., et al.: A protein interaction network links GIT1, an enhancer of huntingtin aggregation, to Huntington's disease. Mol Cell **15** (2004) 853-865
10. Rual, J.F., Venkatesan, K., Hao, T., Hirozane-Kishikawa, T., Dricot, A., Li, N., Berriz, G.F., Gibbons, F.D., Dreze, M., Ayivi-Guedehoussou, N., et al.: Towards a proteome-scale map of the human protein-protein interaction network. Nature **437** (2005) 1173-1178

11. Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F.H., Goehler, H., Stroedicke, M., Zenkner, M., Schoenherr, A., Koeppen, S., et al.: A human protein-protein interaction network: a resource for annotating the proteome. Cell **122** (2005) 957-968
12. Lim, J., Hao, T., Shaw, C., Patel, A.J., Szabo, G., Rual, J.F., Fisk, C.J., Li, N., Smolyar, A., Hill, D.E., et al.: A protein-protein interaction network for human inherited ataxias and disorders of Purkinje cell degeneration. Cell **125** (2006) 801-814
13. Mishra, G.R., Suresh, M., Kumaran, K., Kannabiran, N., Suresh, S., Bala, P., Shivakumar, K., Anuradha, N., Reddy, R., Raghavan, T.M., et al.: Human protein reference database - 2006 update. Nucleic Acids Res **34** (2006) D411-414
14. Wojcik, J., Schachter, V.: Protein-protein interaction map inference using interacting domain profile pairs. Bioinformatics **17 Suppl 1** (2001) S296-305
15. Deng, M., Mehta, S., Sun, F., Chen, T.: Inferring domain-domain interactions from protein-protein interactions. Genome Res **12** (2002) 1540-1548
16. Liu, Y., Liu, N., Zhao, H.: Inferring protein-protein interactions through high-throughput interaction data from diverse organisms. Bioinformatics **21** (2005) 3279-3285
17. Ng, S.K., Zhang, Z., Tan, S.H., Lin, K.: InterDom: a database of putative interacting protein domains for validating predicted protein interactions and complexes. Nucleic Acids Res **31** (2003) 251-254
18. Riley, R., Lee, C., Sabatti, C., Eisenberg, D.: Inferring protein domain interactions from databases of interacting proteins. Genome Biol **6** (2005) R89
19. Finn, R.D., Marshall, M., Bateman, A.: iPfam: visualization of protein-protein interactions in PDB at domain and amino acid resolutions. Bioinformatics **21** (2005) 410-412
20. Stein, A., Russell, R.B., Aloy, P.: 3did: interacting protein domains of known three-dimensional structure. Nucleic Acids Res **33** (2005) D413-417
21. Gene Ontology Consortium: The Gene Ontology (GO) project in 2006. Nucleic Acids Res **34** (2006) D322-326
22. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence (1995) 448-453
23. Lin, D.: An information-theoretic definition of similarity. Proceedings of the Fifteenth International Conference on Machine Learning (1998) 296-304
24. Lu, L.J., Xia, Y., Paccanaro, A., Yu, H., Gerstein, M.: Assessing the limits of genomic data integration for predicting protein networks. Genome Res **15** (2005) 945-953
25. Lin, N., Wu, B., Jansen, R., Gerstein, M., Zhao, H.: Information assessment on predicting protein-protein interactions. BMC Bioinformatics **5** (2004) 154
26. Wu, X., Zhu, L., Guo, J., Zhang, D.Y., Lin, K.: Prediction of yeast protein-protein interaction network: insights from the Gene Ontology and annotations. Nucleic Acids Res **34** (2006) 2137-2150
27. Guo, X., Liu, R., Shriver, C.D., Hu, H., Liebman, M.N.: Assessing semantic similarity measures for the characterization of human regulatory pathways. Bioinformatics **22** (2006) 967-973
28. Schlicker, A., Domingues, F.S., Rahnenführer, J., Lengauer, T.: A new measure for functional similarity of gene products based on Gene Ontology. BMC Bioinformatics **7** (2006) 302
29. Finn, R.D., Mistry, J., Schuster-Bockler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., et al.: Pfam: clans, web tools and services. Nucleic Acids Res **34** (2006) D247-251
30. Yu, H., Luscombe, N.M., Lu, H.X., Zhu, X., Xia, Y., Han, J.D., Bertin, N., Chung, S., Vidal, M., Gerstein, M.: Annotation transfer between genomes: protein-protein interologs and protein-DNA regulogs. Genome Res **14** (2004) 1107-1118
31. Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez Gene: gene-centered information at NCBI. Nucleic Acids Res **33** (2005) D54-58
32. Reguly, T., Breitkreutz, A., Boucher, L., Breitkreutz, B.J., Hon, G.C., Myers, C.L., Parsons, A., Friesen, H., Oughtred, R., Tong, A., et al.: Comprehensive curation and analysis of global interaction networks in *Saccharomyces cerevisiae*. J Biol **5** (2006) 11

Invited Talk

Encoding evolvability: The hierarchical language of polyketide synthase protein interactions

Mukund Thattai

National Centre for Biological Sciences, Bangalore, India

Polyketide synthases use an assembly-line mechanism to catalyse the synthesis of antibiotics and other natural products. Each member of a multi-protein complex adds a particular building block to a growing polyketide chain, so the order of the proteins determines the order of the product. In the laboratory, this property has been used to drive combinatorial chemistry; in the bacterial world, polyketide synthase pathways have been repeatedly shuffled in an arms race to generate novel poisons. I will show that the language of polyketide synthase protein interactions has been designed to facilitate this kind of innovation. I will present the interaction code in detail, and emphasize the elements it shares with high-level computer languages, including modularity, hierarchical organization, and abstraction.

Invited Talk

Genomic Variation and Incipient Speciation in Arabidopsis thaliana

Detlef Weigel

Max Planck Institute for Developmental Biology, Germany
and Salk Institute for Biological Studies, La Jolla, USA.

Comprehensive polymorphism data are a prerequisite for the systematic identification of sequence variants affecting phenotypes. In the first part of my talk, I will discuss our efforts to provide a whole genome resource for the study of population level evolutionary processes in an experimentally tractable, multicellular organism, Arabidopsis thaliana. To this end, we have collaborated with Kelly Frazer and colleagues at Perlegen Sciences, and hybridized genomic DNA of 20 strains to custom microarrays that tile all possible single nucleotide polymorphisms (SNPs) along the entire genome with close to one billion (109) different oligonucleotides. The analysis of SNP distribution and haplotype maps is being carried out in collaboration with the groups of Bernhard Schölkopf (MPI for Biological Cybernetics), Gunnar Rätsch (Friedrich Miescher Laboratory), Daniel Huson (University Tübingen), Joe Ecker (Salk Institute), and Magnus Nordborg (USC). Using novel analysis methods, we identified up to 1.1 million non-redundant SNPs at various levels of precision. In addition, we predicted nearly 5% of the genome to be highly polymorphic or deleted in at least one strain. These data allow for the first time a systematic description of the types of genes that harbor major changes (e.g., stop codons or whole gene deletions) in wild populations. Although major changes are frequent, allele frequency patterns indicate that they are often associated with a fitness cost. Disease resistance (R) genes are found to be the most polymorphic class of genes.

Through our work on natural variation, we have also become involved in more general questions of species-wide evolution. It has long been suggested that post-zygotic hybrid incompatibility between closely related species arises as a by-product of deleterious interactions between genes that have diverged since the most recent common ancestor. In animals, several such gene pairs have been identified in interspecies crosses, but it is not yet known whether they play only a role in maintaining species boundaries, or whether they are also important in establishing barriers to gene flow. To understand the mechanisms underlying nascent incompatibilities, we performed an extensive survey for hybrid incompatibilities within A. thaliana. We identified numerous independent F1 incompatibilities with a range of phenotypically related abnormalities. Each case is attributable to two to three epistatic loci. A common autoimmune mechanism--activation of pathogen responses in the absence of pathogens--underlies the majority of incompatibilities. Moreover, in a collaboration with Jeff Dangl (UNC), we have found that higher disease resistance correlates with incompatibility phenotypes, suggesting a fitness trade-off. Detailed characterization of one hybrid interaction identified a disease resistance (R) gene variant as causal for the incompatibility phenotype. R genes constitute the fastest evolving gene family in plants, suggesting that such incompatibilities arise frequently as a by-product of natural selection.

A novel, comprehensive method to detect and predict protein–protein interactions applied to the study of vesicular trafficking

Christof Winter, Thorsten Baust, Bernard Hoflack and Michael Schroeder
Biotechnological Centre, Dresden University of Technology, 01307 Dresden, Germany

Abstract: *Motivation.* Computational methods to predict protein–protein interactions are of great need. They can help to formulate hypotheses, guide experimental research and serve as additional measures to assess the quality of data obtained in high-throughput interaction experiments. Here, we describe a fully automated three-step procedure to predict and confirm protein–protein interactions. By maximising the information from text mining of the biomedical literature, data from interaction databases, and from available protein structures, we aim at generating a comprehensive picture of known and novel potential interactions between a given set of proteins. *Results.* A recent proteomics assay to identify the protein machinery involved in vesicular trafficking between the biosynthetic and the endosomal compartments revealed 35 proteins that were found as part of membrane coats on liposomes. When applying our method to this data set, we are able to reconstruct most of the interactions known to the molecular biologist. In addition, we predict novel interactions, among these potential linkers of the AP-1 and the Arp2/3 complex to membrane-bound proteins as well as a potential GTPase–GTPase effector interaction. *Conclusions.* Our method allows for a comprehensive network reconstruction that can assist the molecular biologist. Predicted interactions are backed up by structural or experimental evidence and can be inferred at varying levels of confidence. Our method pinpoints existing key interactions and can facilitate the generation of hypotheses.

Keywords: Protein interaction, text mining, protein structure, interaction prediction, membrane traffic.

Introduction

Protein interactions. Protein–protein interactions are fundamental to almost all cellular processes. In addition, nearly every major process in a cell is believed to be carried out by assemblies of ten or more protein molecules [1]. Identification of putative binding partners of a protein is therefore a desirable ambition. It can contribute to understand how such complex molecular machines are organised and how their parts work together. While much effort has been put in large-scale experiments to identify protein–protein interactions in yeast, worm, fly, and human on a genome-wide extent [2–9], the false positive rates of such approaches are estimated to be as high as 50% [10, 11]. Moreover, the intersection of large-scale interaction data sets with those derived from the literature is surprisingly small [8]. By predicting and assessing protein interactions, computational

approaches can help in separating false positive from true positive ones. Sequence-based methods for the prediction of protein–protein interactions include gene context conservation [12], synthetic lethality [13], phylogenetic profiling [14, 15] or co-evolution of gene expression [16]. Structural approaches have focused on the study of protein complexes of known structure [17]. Various databases of binding sites and interfaces between proteins and their domains exist [18–20]. The modelling of interactions using structural templates of sufficient similarity was employed by Aloy and colleagues to successfully model the yeast exosome and some 100 yeast complexes [21, 22].

Additionally, much knowledge on interactions is stored in abstracts that are publicly available in literature databases such as PubMed. While the expert normally is aware of the relevant literature concerning his field, this knowledge remains hidden to the non-specialist who just encounters a couple of genes, for example as a result of a microarray experiment. Text mining can provide access to theoretically all protein interactions hidden in the biomedical literature [23].

We will apply the above methods and others to reconstruct networks relevant for vesicular trafficking. While several recent studies made use of homologous interactions in other species to assess and predict protein interactions [24–27], no study has so far used sequence similarity to literature interactions obtained from text mining to our knowledge.

Vesicle coats and adaptor proteins. Vesicles are small, membrane-enclosed containers that mediate transport between cellular compartments. The formation of vesicles and the selective incorporation of cargo molecules are both mediated by protein coats, which are recruited onto the cytosolic side of the forming vesicle. In the case of clathrin-coated vesicles, the cargo transmembrane proteins are linked via adaptor proteins (APs) with structural coats such as clathrin [28]. AP-1 mediates the transport of selected transmembrane proteins that cycle between the trans-Golgi network, endosomes and the plasma membrane. How APs interact with other molecular components of the complex coat machinery remains largely unclear.

Materials and Methods

As resources, we combine protein–protein interactions derived from text mining of the biomedical literature, available protein interaction databases, and structural domain interactions. Aim of our method is to obtain all known interactions between a given set of proteins, and to further predict new interactions that are likely to be present within this set. The overall approach is a procedure of three steps. It is summarised in Figure 1 and described in detail below.

Collection of literature-based interactions. In the first step, we start collecting known interactions from the literature. We use NetPro, an expert curated and annotated database containing $\sim$100,000 protein–protein interactions [29]. These were extracted from PubMed abstracts by a semi-automated method and then cross-checked by human experts. For every

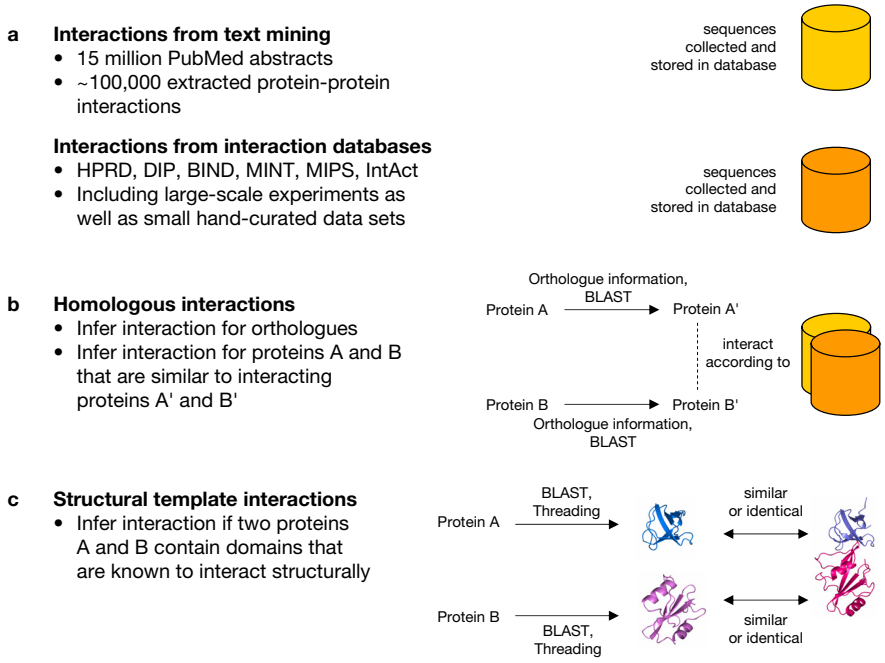


Figure 1. A three-step approach to gather all knowledge about protein–protein interactions. **a.** First, text mining is used to collect interactions from the literature. Sequences of the involved proteins are stored in a database. In a similar manner, available interaction databases are integrated and their protein sequences stored. **b.** Second, for a given protein pair A and B, a search for homologous interactions is performed using orthologue information and BLAST. Thus, interactions between homologous proteins in other species or similar proteins can serve as templates to predict an interaction between the original proteins A and B. **c.** Third, similarity to structural templates of interacting domains is used to predict interactions. To this end, we employ both sequence similarity (BLAST) and structural similarity (Threading) measures.

interaction, NetPro lists the two involved gene identifiers, species, the abstract sentences documenting the interaction, an interaction verb and an interaction nature. The interaction nature can be direct or indirect, where interactions verbs such as *binds to* classify an interaction as direct, verbs such as *colocalises with* as indirect. For every protein in NetPro, we collect its sequence from the NCBI Protein Database (Figure 1a).

Collection of interactions from interaction databases. We complement our collected literature-derived interactions with data from various interaction databases. Protein–protein interaction sets are obtained from HPRD [30], DIP [31], BIND [32], MINT [33], MIPS [34], and IntAct [35]. Again, sequences of the interacting proteins are collected and stored (Figure 1a). We are aware of potential false positives introduced by high-throughput interactions screens [10]. Since our approach aims at maximum sensitivity, we do not apply

any filter, but rather record the experimental origin of an interaction finding as confidence criterion.

Identification of interactions for a given data set. To identify known interaction in a given data set, we simply query our databases of collected interactions. The result is a protein–protein interaction network where an interaction either was described in the literature or stored in an interaction database. In a second step, we expand this network. Orthologues, i.e. homologues in other species, of proteins in our given data set are identified using the NCBI HomoloGene Database (Release 46.1). Our collected interactions are again checked for the orthologues. Following the idea of *interologs* described in [36], we predict a putative interaction between two proteins if they have interacting homologues in another species. We further extend this idea of homologous interactions by performing a BLAST search with our data set against the collected interaction sequences. Thereby, we are able to find a template protein pair A' and B' known to interact, where A' and B' are similar to two proteins A and B from our data set (Figure 1b). If the similarity (e.g. measured in sequence identity) is sufficient, we infer a putative interaction between A and B. In order to obtain a score reflecting the reliability of the prediction, we calculate the joint percentage identity. For a protein pair (A, B), this score is defined as $min(I_A, I_B)$ where I_A, I_B are the sequence percentage identities between the protein pair and the template. In this study, we require a joint percentage identity of at least 40%.

Structure-based interaction prediction. The third step of our approach is shown in Figure 1c. To predict interactions on the basis of known structures, we use SCOPPI [18], a database of domain–domain interactions and their interfaces derived from all multi-domain proteins in the Protein Data Bank [37]. Domains are defined by SCOP, the Structural Classification of Proteins. Domain residues within a distance of 5 Å to another domain are considered interacting, thus being in accord to other interface definitions [38]. As structural interaction templates for our predictions, we use a subset of SCOPPI obeying the following filter criteria: 1) interacting domains are required to be on different polypeptide chains, 2) interface size (defined as change in accessible surface area, ΔASA, calculated with Naccess) $\geq$ 600 Å^2 to filter out unspecific interfaces, 3) exclusion of homo-dimers to avoid false positive predictions between highly similar proteins. Two proteins are predicted to potentially interact if they contain domains that are known to interact structurally, according to the SCOPPI subset described above. To assign these domains to a given protein sequence, we employ both sequence similarity and structure prediction methods: First, we perform a BLAST search against a database containing the defined SCOPPI subset. By using a sequence identity cut-off at 40%, we ensure that the assigned domains have a similar structure to our structural interaction templates. Second, all protein sequences are threaded with GenTHREADER [39], considering hits with p-values $<$ 0.001 only. This procedure assigns probable SCOP domains being part of GenTHREADER's fold library to each protein sequence. As above, we assign a score of $min(I_A, I_B)$ for the sequence identities between the protein pair and the template, and $max(p_A, p_B)$ where p_A, p_B are the threading p-values, respectively.

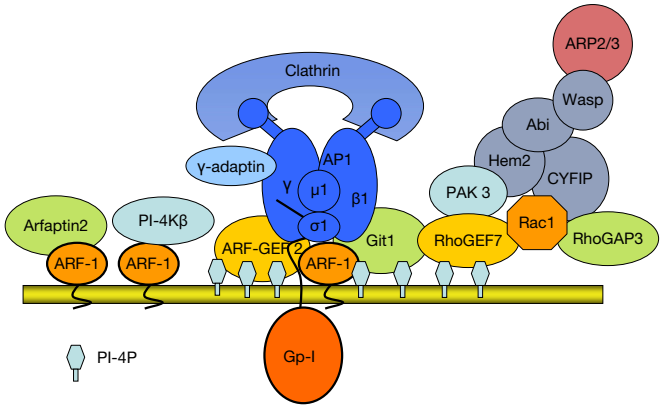


Figure 2. Theoretical model of the adaptor protein 1 (AP-1) related machinery. Several small GTPases along with their GAP and GEF effectors are involved. Some of the depicted interactions are known, while others are still presumptions [41].

Data set origin: Experimental identification of coat proteins. The data set of proteins used in our study was previously obtained by a collaborating group [40]. Aim of this study was to identify cytosolic proteins that are involved in the adaptor protein 1 (AP-1) coat assembly. The result comprises 35 murine proteins that could be selectively recruited onto liposomes that exhibit cytoplasmic domains of AP-1 cargo molecules. Among these, the AP-1 complex, clathrin, several GTPases and their effectors as well as an actin nucleation machinery was found. Table 1 shows the 35 proteins identified by mass spectrometry. How these proteins spatially arrange on liposome membranes is still speculative. Here, our method can help to suggest possible interactions and thus aid to formulate hypotheses concerning the recruited molecular machinery.

Results and discussion

Construction of an interaction network for proteins from clathrin-coated vesicles. As a use case for our method, we choose the data set described above. It contains 35 proteins shown in Table 1. Figure 2 shows the putative spatial arrangement of a subset drawn by the collaborating expert biologist.

Figure 3 shows the resulting network after taking the steps described in detail in Materials and Methods. Interactions that could be found from the literature or interaction databases are shown in red. Homologous interactions in close species (human, rat) are also red. Predicted interactions, inferred by orthology in remote species or by sequence similarity, are depicted in yellow. In addition, indirect literature interactions are dashed, whereas direct ones are solid lines. Blue lines indicate predictions based on structural templates.

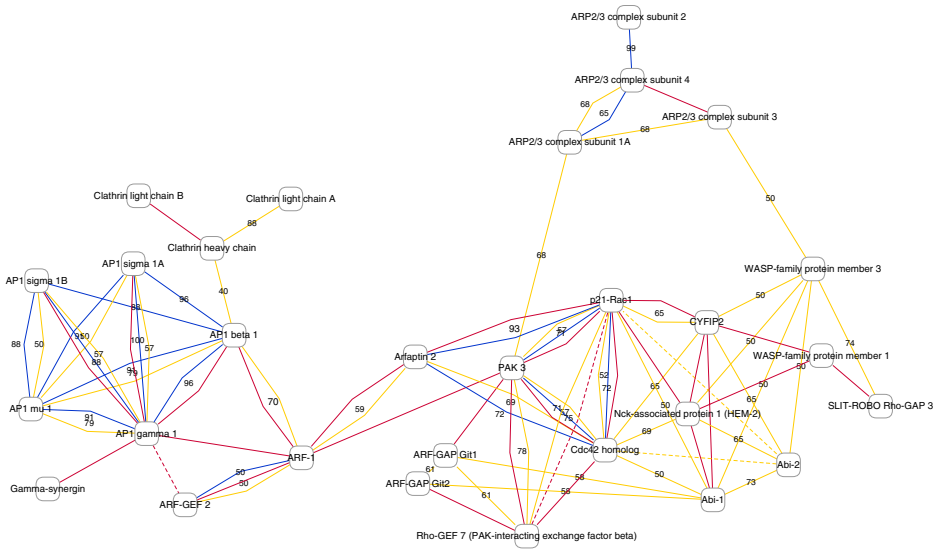


Figure 3. Constructed interaction network for the AP-1 complex. Interactions known by literature or interaction database are depicted in red, predictions based on these are yellow, and blue lines represent predictions based on 3D structural templates. Numbers indicate the sequence identity in percent that the predicted interaction shares with the template. The higher this number, the more reliable the prediction. For the sake of clarity, the cut-off is set at 50% here, except for the discussed example of the AP-1–Clatrhin example.

In case of yellow and blue lines, numbers specify the joint sequence identity percentage score of the prediction. The result allows for the generation of several new hypotheses about the molecular machinery around AP-1. In the following, three predicted examples will be discussed in detail.

Literature- and interaction databases approach. The ability to blast against a multitude of literature interactions represents a quite powerful tool. First, it allows for the detection of seamless grades of similarity. Second, it deals with the problem of synonyms in a very elegant way. Once text mining matched a protein name to a protein entity, we do not further rely on synonyms but on the sequence which—in combination with the species—unambigously described the protein entity.

PAK3 is a potential interactor of the Arp2/3 complex subunit 1A. The literature-derived search predicts, for example, the interaction between murine p21-activated kinase 3 (PAK3) and the Actin-related protein 2/3 complex subunit 1A. Basis of this prediction is a literature-documented interaction between human p21-activated kinase 1 (PAK1) and human Arp2/3 subunit 1B, reported in [42]. Overall sequence identity is 69% between the kinases, and 83% between the Arp2/3 subunits. We cannot be not sure if this interaction

is indeed true, but our method provides evidence that an interaction is likely. This would suggest that PAK3 functions in the given data set of mouse proteins in a similar manner as PAK1, namely by phosphorylating the Arp2/3 complex, thus influencing vesicle motility. Further inspection of this example reveals additional support for the prediction. In the abstract of [42], we learn that PAK1 phosphorylates p41-Arc (another name for the Arp2/3 subunit) on threonine 21. As we check the alignment, we find threonine 21 present in a well-conserved region in both proteins.

Text mining challenges: Clathrin should be linked to the AP-1 complex subunit beta. General problems of text mining still affect our approach. If interactions are not extracted in the first place, we lack this information and hence cannot infer any similar predictions. In our study, this occurs in case of the AP-1—Clathrin interaction. It is has long been known that Clathrin is associated with adaptor proteins on clathrin-coated vesicles that mediate traffic of between intracellular compartments. The physical interaction between Clathrin and the beta 1 and beta 2 subunits of the AP complexes was first described 1993 in [43]. However, neither the incorporated interaction databases, nor the literature-based NetPro database contain this particular interaction. For NetPro, the reason seems obvious. The relevant sentence in the abstract of [43] states: *"It was found that, in the absence of all the other AP subunits, beta 1 and beta 2 interact with clathrin."*. The fact that the interaction partners are just described as "beta1" and "beta2" makes it extremely hard for an algorithm to reason that these two are actually AP subunits. As we are lacking this interaction, and since no such interaction could be inferred from structural templates, we cannot connect Clathrin with the AP-1 complex in the interaction network. There is, however, a homologous interaction our method detects: According to interaction database DIP [31], yeast beta-adaptin homolog APL2 interacts with yeast clathrin heavy chain 1 [44]. The BLAST search of our method picks up the similarity between the yeast and mouse orthologues of AP-1 beta subunit (40 % identities, 62 % positives) and Clathrin heavy chain (49 % identities, 70 % positives). On this basis, our approach predicts a potential interaction between the two murine proteins. In this case, it turns out that the prediction is correct, with the known, but missing interaction validating our prediction.

Structural template-based approach. Structures are available for a considerable number of the AP1-related proteins (Table 1). The first stage easily detects these templates by sequence identity search via BLAST. If at 40% sequence identity cut-off no domain structures are found, we employ threading to assign domains. For every protein pair, we check if the SCOPPI database lists any of the assigned domains as interacting. If so, we mark these two proteins as potentially interacting (for details, see Materials and Methods).

Since crystal structures are available for the AP-1 and the Arp2/3 complex, the structural templates-based approach connects the subunits according to their contacts (blue lines in Figure 3). One interesting candidate is the predicted connection between the Rho GTPase CDC42 and Arfaptin, an effector of the Arf GTPase.

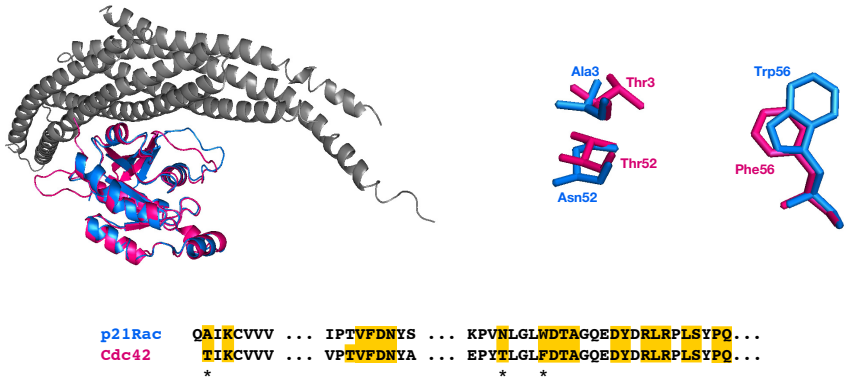


Figure 4. Structural modelling of the CDC42–Arfaptin 1 interaction. Left: The known complex structure p21Rac1 (blue) interacting with Arfaptin 1 (grey) serves as a template. CDC42 (red) and Rac1 (blue) structurally align reasonably well (RMSD 0.8). Bottom: Residues contributing to the binding site in the complex. Interacting residues are marked yellow, mismatching interacting residues are marked with a star. Right: Interface mismatches (Ala/Thr, Asn/Thr, Trp/Phe) after structural superposition of CDC42 and Rac1. Although being different amino acids, their structural side chain arrangement is similar.

The small Rho GTPase CDC42 is a potential interactor of Arfaptin 1. The small GTPases Rho, Rac and CDC42 are regulators of actin structures, cell adhesion and motility. Here, we predict the interaction between CDC42 and Arfaptin 1. As template, we use the crystal structure of RAC1-GDP in complex with Arfaptin (PDB ID 1i4l, [45]). Figure 4 shows the superposition of CDC42 and Rac1 with an RMSD of 0.8. The interfacial residues of both GTPases, as defined to be within 5 Angstrom distance to the Arfaptin, are aligned and highlighted in yellow. Closer examination of the three mismatches (Ala/Thr, Asn/Thr, and Trp/Phe) in the interface reveals that all three residues align reasonably well in the superposition of CDC42 and Rac1. We therefore have reason to believe that at least from a steric point of view the interaction is feasible.

Reliability scores and evidence. Our approach generates reliability scores as well as supporting evidence for the protein–protein interactions predicted. For structure-based predictions, we provide a structural template as well as confidence scores. These are sequence identity percentages and/or threading p-values. For literature-derived predictions, we provide statements from PubMed articles which explicitly document details of the interaction. In addition, joint sequence identity scores are available for every predicted interaction. These are 100% for known interactions described in the literature and lower for decreasing degrees of potential homology. An introduction of various cut-off levels could account for the different interaction nature of proteins, e.g. a stricter level for GTPases, and a more relaxed level for unspecific protein interactions. If the binding site is known, the matches of interfacial residues serve as additional parameters for the quality of the predicted inter-

action. The more conserved the interface, the more likely the interaction.

Limitations of method. Our method shares the common limitations of interaction prediction methods. Technical false positives (i.e. those due to the method) are likely, especially for predictions with low joint sequence percentage. Biological false positives (i.e. interactions that could be observed in vitro, but have no biological relevance, because the two proteins are not expressed in the same tissues or compartments, or not at the same time) can at least in this study be ruled out due to the experimental setup used to produce the data set. It ensures that the tested proteins are within close proximity, thus displaying a considerable potential to form interactions.

Another problem is that of the specificity of our predictions. Small GTPases and their effectors (such as GTPase activating and guanine nucleotide exchanging proteins) are good examples for forming specific interactions [46]. The problem can be addressed by a rigorous sequence identity threshold, as suggested above.

Although a considerable number of proteins in our study are known by structure, and although our method has access to $\sim$25,000 different domain interaction templates, we are not able to link the AP-1 or the Arp2/3 complex to any other protein in the data set by merely using structural information. This points at the problem that there are still comparatively few multi-domain structures available that can serve as modelling templates for interactions. As a positive outlook, we observe a supra-linear growth of these templates.

Evaluation of method. It is difficult to assess predicted interactions by other means than the biological experiment. The main problem is estimation of a false positive rate. How can one be sure that two proteins do *not* interact? Simple absence of the interaction in reliable data sources is not sufficient—the interaction might just not have been discovered yet. The *closed-world assumption*, i.e. interactions not known are also not true, does not hold for biology. Our predicted interactions are currently being tested by our collaborating group. The result of these experiments will allow for a thorough evaluation of our method.

Summary

We propose a fully automated method for the retrieval and prediction of protein–protein interactions. By merging information from literature abstracts stored in PubMed, interaction databases, and structures in the Protein Data Bank PDB we obtain a comprehensive picture on documented interactions. On the basis of this knowledge, we can construct an interaction network for any given data set, and further extend it predicted interactions at various confidence levels based on sequence or structural similarity to known interaction templates.

Applied to a data set of proteins that form coat complexes on vesicle membranes, our method identifies almost all relevant interactions. Further interactions are predicted, among them potential linkers for the AP1- and the Arp2/3 complex. Therewith, we provide po-

tential interaction candidates for further experimental testing. By incorporating the whole spectrum of text mining interactions described in the biomedical literature, data stored in interaction databases, and all structurally known domain–domain interactions, our method ensures a comprehensive network reconstruction that can assist the molecular biologist. Applying it on a genome-wide scale, we can further scale up this network to a systems biology level that provides a view on a whole interactome, thus providing a valuable tool for the life sciences.

Acknowledgements. Funding by EFRE project CODI no. 4212/04-07 is kindly acknowledged.

References

[1] Alberts, B. The cell as a collection of protein machines: preparing the next generation of molecular biologists. *Cell* **92**, 291–294 (1998).

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

[3] Ito, T. *et al.* A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U S A* **98**, 4569–4574 (2001).

[4] Gavin, A.-C. *et al.* Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415**, 141–147 (2002).

[5] Ho, Y. *et al.* Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature* **415**, 180–183 (2002).

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

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

[8] Rual, J.-F. *et al.* Towards a proteome-scale map of the human protein-protein interaction network. *Nature* **437**, 1173–1178 (2005).

[9] Gavin, A. *et al.* Proteome survey reveals modularity of the yeast cell machinery. *Nature* (2006).

[10] von Mering, C. *et al.* Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* **417**, 399–403 (2002).

[11] Sprinzak, E., Sattath, S. & Margalit, H. How reliable are experimental protein-protein interaction data? *J Mol Biol* **327**, 919–923 (2003).

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

[13] Tong, A. H. Y. *et al.* Global mapping of the yeast genetic interaction network. *Science* **303**, 808–813 (2004).

[14] Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D. & Yeates, T. O. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci U S A* **96**, 4285–4288 (1999).

[15] Sun, J. *et al.* Refined phylogenetic profiles method for predicting protein-protein interactions. *Bioinformatics* **21**, 3409–3415 (2005). Evaluation Studies.

[16] Fraser, H. B., Hirsh, A. E., Wall, D. P. & Eisen, M. B. Coevolution of gene expression among interacting proteins. *Proc Natl Acad Sci U S A* **101**, 9033–9038 (2004).

[17] Aloy, P. & Russell, R. Structural systems biology: modelling protein interactions. *Nat Rev Mol Cell Biol* **7**, 188–197 (2006).

[18] Winter, C., Henschel, A., Kim, W. K. & Schroeder, M. SCOPPI: A Structural Classification of Protein–Protein Interfaces. *Nucleic Acids Res* **34**, 310–314 (2006).

[19] Davis, F. & Sali, A. PIBASE: a comprehensive database of structurally defined protein interfaces. *Bioinformatics* (2005).

[20] Stein, A., Russell, R. B. & Aloy, P. 3DID: interacting protein domains of known three-dimensional structure. *Nucleic Acids Res* **33**, 413–417 (2005).

[21] Aloy, P. *et al.* A complex prediction: three-dimensional model of the yeast exosome. *EMBO Rep* **3**, 628–635 (2002).

[22] Aloy, P. *et al.* Structure-based assembly of protein complexes in yeast. *Science* **303**, 2026–2029 (2004).

[23] Jensen, L. J., Saric, J. & Bork, P. Literature mining for the biologist: from information retrieval to biological discovery. *Nat Rev Genet* **7**, 119–129 (2006).

[24] Ben-Hur, A. & Noble, W. S. Kernel methods for predicting protein-protein interactions. *Bioinformatics* **21 Suppl 1**, i38–46 (2005).

[25] Espadaler, J., Romero-Isart, O., Jackson, R. M. & Oliva, B. Prediction of protein-protein interactions using distant conservation of sequence patterns and structure relationships. *Bioinformatics* **21**, 3360–3368 (2005). Evaluation Studies.

[26] Kim, W. K., Bolser, D. M. & Park, J. H. Large-scale co-evolution analysis of protein structural interlogues using the global protein structural interactome map (PSIMAP). *Bioinformatics* **20**, 1138–1150 (2004).

[27] Han, D.-S., Kim, H.-S., Jang, W.-H., Lee, S.-D. & Suh, J.-K. Prespi: a domain combination based prediction system for protein-protein interaction. *Nucleic Acids Res* **32**, 6312–6320 (2004).

[28] Owen, D. J., Collins, B. M. & Evans, P. R. Adaptors for clathrin coats: structure and function. *Annu Rev Cell Dev Biol* **20**, 153–191 (2004).

[29] `http://www.molecularconnections.com`.

[30] Mishra, G. R. *et al.* Human protein reference database–2006 update. *Nucleic Acids Res* **34**, 411–414 (2006).

[31] Salwinski, L. *et al.* The database of interacting proteins: 2004 update. *Nucleic Acids Res* **32**, 449–451 (2004).

[32] Bader, G. & Hogue, C. Bind–a data specification for storing and describing biomolecular interactions, molecular complexes and pathways. *Bioinformatics* **16**, 465–77 (2000).

[33] Zanzoni, A. *et al.* Mint: a molecular interaction database. *FEBS Lett* **513**, 135–140 (2002).

[34] Pagel, P. *et al.* The mips mammalian protein-protein interaction database. *Bioinformatics* (2004). JOURNAL ARTICLE.

[35] Hermjakob, H. *et al.* IntAct: an open source molecular interaction database. *Nucleic Acids Res* **32**, 452–455 (2004).

[36] Walhout, A. J. *et al.* Protein interaction mapping in c. elegans using proteins involved in vulval development. *Science* **287**, 116–122 (2000).

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

[38] Tsai, C. J., Lin, S. L., Wolfson, H. J. & Nussinov, R. A dataset of protein-protein interfaces generated with a sequence-order-independent comparison technique. *J Mol Biol* **260**, 604–620 (1996).

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

[40] Baust, T., Czupalla, C., Krause, E., Bourel-Bonnet, L. & Hoflack, B. Proteomic analysis of adaptor protein 1a coats selectively assembled on liposomes. *Proc Natl Acad Sci U S A* (2006).

[41] Hoflack, B. Personal communication.

[42] Vadlamudi, R. K., Li, F., Barnes, C. J., Bagheri-Yarmand, R. & Kumar, R. p41-Arc subunit of human Arp2/3 complex is a p21-activated kinase-1-interacting substrate. *EMBO Rep* **5**, 154–160 (2004).

[43] Gallusser, A. & Kirchhausen, T. The beta 1 and beta 2 subunits of the AP complexes are the clathrin coat assembly components. *EMBO J* **12**, 5237–5244 (1993).

[44] Yeung, B. G., Phan, H. L. & Payne, G. S. Adaptor complex-independent clathrin function in yeast. *Mol Biol Cell* **10**, 3643–3659 (1999).

[45] Tarricone, C. *et al.* The structural basis of Arfaptin-mediated cross-talk between Rac and Arf signalling pathways. *Nature* **411**, 215–219 (2001).

[46] Hall, A. (ed.) *GTPases* (Oxford University Press, 2000).

Protein	NCBI gi	Known structure
COAT COMPONENTS		
Clathrin heavy chain	66773801	1xi5, 1xi4
Clathrin light chain B	62510439	1xi5, 1xi4
Clathrin light chain A	2493731	1xi5, 1xi4
AP1 beta 1	21541948	1w63
AP1 gamma 1	113349	1w63
AP1 mu 1	543817	1w63
AP1 sigma 1B	21541960	1w63
AP1 sigma 1A	48428720	1w63
AP1 gamma subunit binding protein 1 (gamma-synergin)	34996507	
ARF-1/ARF-3	51316986 47117658	1r8q
ARF-GEF 2 (Brefeldin A-inhibited)	63492672	
G protein-coupled receptor kinase-interactor 1 (ARF-GAP Git1)	58864889	
G protein-coupled receptor kinase-interactor 2 (ARF-GAP Git2)	18203126	
Arfaptin 1	63501125	
Arfaptin 2	67460562	
ACTIN POLYMERIZATION		
Nck-associated protein 1 (HEM-2)	26986194	
SH3 adapter protein SPIN90	57015413	
WASP-family protein member 1	16877274	
WASP-family protein member 3	20071942	
CYFIP2	19526988	
Abi-1	50400517	
Abi-2	50400259	
ARP2/3 complex subunit 1A	59797974	1u2v
ARP2/3 complex subunit 2	23621467	1u2v
ARP2/3 complex subunit 4	38372626	1u2v
ARP2/3 complex subunit 3	62899893	1u2v
p21-Rac1	51702788	1i4l
Cdc42 homolog	46397379	1grn
SLIT-ROBO Rho-GAP 3 (WAVE-associated Rac-GAP)	48428625	
Rho-GEF 7 (PAK-interacting exchange factor beta)	18202873	
PAK 3	47117898	1yhw
MEMBRANE FUSION		
Rab-11B	1172815	1oiv
Rab-14	46577103	2aed
Rab-4	15986733	2bme

Table 1. Mass spectrometric analysis of AP-1A-coated liposomes (data from [40]). For proteins which have a known structure, the Protein Data Bank identifer is given.

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER - Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods - Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahnič (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 - Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement - Erfahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometric and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 - Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 D. Huson, O. Kohlbacher, A. Lupas, K. Nieselt, A. Zell (Hrsg.): German Conference on Bioinformatics 2006

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14
53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de