BIOINFORMATICS

Vol. 21 Suppl. 2 2005, pages ii42–ii46
doi:10.1093/bioinformatics/bti1107

Databases

SIMAP—The similarity matrix of proteins

Roland Arnold[1,†], Thomas Rattei[2,†,*], Patrick Tischler[1], Minh-Duc Truong[1], Volker Stümpflen[1] and Werner Mewes[1]

[1]Institute for Bioinformatics, GSF-National Research Center for Environment and Health, Ingolstädter Landstrasse 1, 85764 Neuherberg, Germany and [2]Department of Genome Oriented Bioinformatics, Technical University of Munich, Wissenschaftszentrum Weihenstephan, 85350 Freising, Germany

ABSTRACT

Motivation: Sequence similarity searches are of great importance in bioinformatics. Exhaustive searches for homologous proteins in databases are computationally expensive and can be replaced by a database of pre-calculated homologies in many cases. Retrieving similarities from an incrementally updated database instead of repeatedly recalculating them should provide homologs much faster and frees computational resources for other purposes.

Results: We have implemented SIMAP—a database containing the similarity space formed by almost all amino acid sequences from public databases and completely sequenced genomes. The database is capable of handling very large datasets and allows incremental updates. We have implemented a powerful backbone for similarity computation, which is based on FASTA heuristics. By providing WWW interfaces as well as web services, we make our data accessible to the worldwide community. We have also adapted procedures to detect putative orthologs as example applications.

Availability: The SIMAP portal page providing links to SIMAP services is publicly available: http://mips.gsf.de/services/analysis/simap/. The web services can be accessed under http://mips.gsf.de/proj/hobitws/services/RPCSimapService?wsdl and http://mips.gsf.de/proj/hobitws/services/DocSimapService?wsdl

Contact: t.rattei@wzw.tum.de

1 INTRODUCTION

Sequence similarity searches such as BLAST (Altschul *et al*., 1990) or FASTA (Pearson, 2000) are a basic tool for the *in silico* analysis of uncharacterized protein sequences since a high degree of similarity implies homology of a pair of sequences. These similarity searches are especially meaningful on protein sequences, since for functional genes the conservation is much higher on the level of the protein primary sequence than on the underlying DNA level (Gojobori *et al*., 1982).

Sequence similarity is represented as either a local or global pairwise sequence alignment depending on the mathematical optimization model employed (Smith and Waterman, 1981; Needleman and Wunsch, 1970). These sequence alignment and the database search algorithms to detect significant pairwise relationships in a large dataset are the basic tools in many bioinformatics tasks: in the process of annotating protein sequences, homology is used to infer knowledge from known to unknown sequences (Wilson *et al*., 2000; Mewes *et al*., 1997). The representation of protein families by a multiple alignment needs all pairwise comparisons in the first step (Higgins and Sharp, 1988) and in consequence, even the creation of sequence profiles and hidden Markov models is based on pairwise alignments (Eddy, 1998).

The potential increases when exhaustive all-against-all comparisons of a meaningful dataset are employed. These exhaustive searches result in the accumulation of knowledge of all detectable evolutionary relationships in a dataset and we refer to this as the 'protein similarity space', which can be subjected to a clustering analysis in order to get protein families and superfamilies in a sufficiently large dataset (Krause *et al*., 2002). With a pair of completely sequenced genomes, approaches can be used to detect ortholog and paralog relationships (O'Brien *et al*., 2005). With even more complete genomes, the creation of orthologous groups (Li *et al*., 2003) and methods for functional prediction such as phylogenetic profiling (Pellegrini *et al*., 1999), the Rosetta stone method (Marcotte and Marcotte, 2002) or the principle of conserved gene neighborhood (Rogozin *et al*., 2002) are applicable to the protein space.

A pairwise alignment is parameterized by its underlying substitution matrix, which models the exchange probabilities of the amino acids as the BLOSUM50 matrix (Henikoff and Henikoff, 1992), the costs of opening and extending gaps and the boundary condition whether the alignment should be optimized locally or globally.

The optimal solution for the local case is the Smith–Waterman algorithm (Smith and Waterman, 1981). A straightforward way to build up the similarity space would be to compute for each pair of proteins the Smith–Waterman alignment and to keep high-scoring hits for further processing. Although efficient implementations (Rognes and Seeberg, 2000) exist, the computational costs (i.e. the CPU time needed) are still high. So a number of heuristic approaches have been introduced, such as BLAST (Altschul *et al*., 1990) or FASTA (Pearson, 2000). These heuristics speed up the search for biologically meaningful hits in a database significantly and are therefore widely used by bioinformaticians and biologists alike.

The result sets returned by database searches are typically filtered and sorted according to their biological relevance using statistics. The methods commonly applied are Z-score statistics (Bastien *et al*., 2004) and the statistics for high-scoring segment pairs relying on the extreme value distribution (Altschul and Gish, 1996), which is commonly used as an approximation for local alignment statistics. The expectation value (E-value) describes how many hits would be

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

formed randomly given a sequence of a certain length and a database to search against. According to the extreme value distribution model, it is computed by the well-known formula

$$E = mn\,2^{-S'} \quad (1)$$

It is parameterized by the query sequence length m and the length of the database n. This means that searches between differently sized databases result in different E-values.

The normalized bit-score S' represents the distance between the two sequences and is computed using the parameters K and λ by the following formula:

$$S' = \lambda S - \ln(K)/(\ln 2) \quad (2)$$

The two values K and λ rely on the database composition and the substitution matrix. They can be either estimated with each database search, as is the case for modern versions of the FASTA package (Pearson, 2000) or fixed estimates for an average database, and each substitution matrix can be used as in BLAST (Altschul *et al*., 1990) in order to gain speed.

Another widely used measurement to detect strongly related proteins is the overall sequence identity. There is a vivid discussion concerning the maximum thresholds at which annotation transfer can be done automatically (Wilson *et al*., 2000).

As described, similarity searches are a core application in computational biology. Furthermore, many analyses rely on exhaustive all-against-all comparisons. Typically, these comparisons are redone again and again since the available datasets change over time. In many analyses such as the detection of orthologous relationship (O'Brien *et al*., 2005), this re-computation is the most time-consuming step and makes the analysis intractable for large or many datasets. Therefore, a pre-calculated all-against-all matrix that stores the similarity space in a database and allows very rapid access to significant hits of interest becomes desirable.

Such a database should avoid redundancy and provide useful interfaces which allow for extraction of different subsets and the application of different cut-offs. It should be regularly updated and the saved values should therefore be independent of the database's size and composition in order to ensure compatibility between different versions.

The time complexity for an all-against-all comparison to produce a sequence similarity space is $O(n^2)$ in respect of the number of individual sequences since every protein must be compared with every other protein. In a good approximation, the alignments and the alignment raw scores are symmetrical (i.e. the score for an alignment formed by sequence A with sequence B is the same as that for B with A). This property has already been used to halve the amount of computation required (Dumontier and Hogue, 2002) and it also implies an incremental update process: every new sequence has to be computed against itself and against all sequences already in the database. The result is saved for the new sequence and the result lists of the old sequences can be updated without re-computation. As an implemented solution, we present SIMAP, the Similarity Matrix of Proteins.

2 METHODS

The central component of the SIMAP concept is the algorithm that precomputes the sequence similarities. As it was evaluated to be the best compromise between computational speed and sensitivity (Pearson, 1991) we have chosen FASTA (Pearson, 2000) for finding all putative hits. The FASTA parameter ktup = 1 and BLOSUM50 substitution matrix are used to adjust the calculations to optimal sensitivity. Before FASTA calculations, all low-complexity regions in the sequences are masked by seg (Wootton, 1994). In order to store exact alignment coordinates and scores in the hit database, every FASTA hit is recalculated without low-complexity filtering using the Smith–Waterman algorithm and BLOSUM50 substitution matrix. If the final Smith–Waterman-score is $\geq$80 the hit is accepted and stored. This score is independent of the query and database lengths, as is necessary in a growing database such as SIMAP. The score threshold of 80 is an optimal compromise between sensitivity and the amount of data to be handled.

3 IMPLEMENTATION

3.1 Import of data

SIMAP represents proteomes and sequence collections from very different data sources. For that reason we have implemented a flexible input layer which is based on the data access object (DAO) design pattern. DAO classes are available for files using multiple FASTA format and EMBL format, databases such as PEDANT (Riley *et al*., 2005) and web services as provided by the mips plantsDB and GenRE projects (Schoof *et al*., 2004). The imported data are separated into three entities:

- Proteome/project (describes the context of the proteins)
- Protein (describes a certain protein entry and references to proteome/project and sequence)
- Sequence (contains the non-redundant protein sequences, checksums and selfscores)

As all similarity calculations rely only on the pure sequences, the separation of protein and sequence information is necessary to avoid redundant calculations. All protein sequences are preprocessed for validation and low-complexity filtering. In order to avoid loss of information, low-complexity regions are not masked by 'X' but converted into lowercase letters.

New proteome/project entries are added manually because some additional information, such as the taxonomy node ID, is required. The protein import and update procedures run fully automatically and very efficiently. They can be scheduled in advance or run manually. New sequences are tagged and need to be processed by the calculation module.

3.2 Similarity calculation

The SIMAP calculation module was designed to run both in standalone mode and in grid environments. It is based on the FASTA source code and performs the two-step computational process as mentioned above: first it compares low-complexity masked proteins using FASTA heuristics and then it recalculates found hits using non-masked sequences and the Smith–Waterman algorithm.

The calculation client runs as a command-line program, e.g. in Sun Gridengine clusters (http://gridengine.sunsource.net), and also contains the BOINC core client to be used in BOINC-based grid systems (http://boinc.berkeley.edu). The results are validated by the SIMAP server and encoded into the binary hit format. Every hit consumes 19 bytes and contains

- Sequence IDs
- Smith–Waterman score

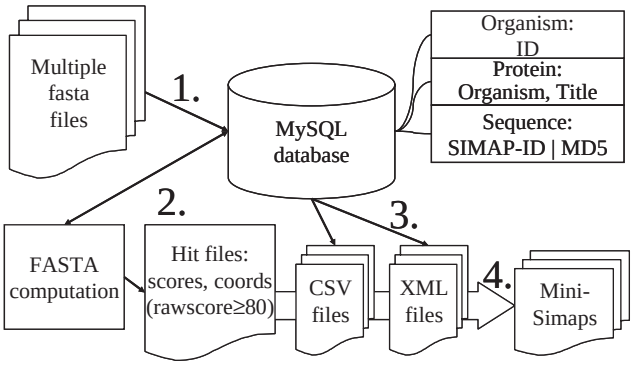

Fig. 1. Schematic data flow and data structure in SIMAP

- Identity
- Gapped Identity
- Overlap
- Start and stop coordinates of the alignment in both proteins

To provide retrieval-optimized data structures, all hits are sorted descending by score and organized in a hash-like structure that is stored in one binary hitfile per sequence:

- The key (sequence ID) is encoded by pathname and filename.
- The value (sorted list of hit data blocks as described above) is stored within the file content.

This approach trades time for disk space, so every hit is stored redundantly in two hitfiles according to the two sequences of the pair. Nevertheless, this turned out to be the only implementation that provides the necessary retrieval speed.

The whole SIMAP data can be exported using filtering conditions into custom 'mini-SIMAPs', which contain data from selected organisms, for example, or hits down to a specified score. The data flow and data structure in SIMAP are shown schematically in Figure 1.

3.3 Data access and retrieval

The most important use for SIMAP is the retrieval of homologs for a given protein according to a user-defined search space and filtering. We have implemented a flexible and fast infrastructure to get data out of SIMAP in order to realize this and other usages as well.

First, a PERL package has been implemented for integrating SIMAP into other bioinformatics applications. The package allows the programmer to specify the search space and filtering conditions and then to retrieve the homologs from SIMAP.

Additionally, we have implemented a server-based retrieval layer using Enterprise Java Beans (EJB). It serves as a database abstraction layer and hides the internal structure of SIMAP from users. Several EJBs have been implemented that represent the different SIMAP classes such as sequence, protein, organism and taxonomy. The EJBs are server-side components designed for distributed access and information management. They allow easy integration of SIMAP into any kind of application within the mips Genome Research Environment (GenRE) (http://mips.gsf.de/genre/proj/genre) used for our various genome and protein interaction databases. However, we the programmatic access is not restricted to internal applications but offers the same functionality also for web-wide external access. Therefore we developed additionally a HOBIT service layer (http://hobit.gsf.de) based on the web service technology to share SIMAP in a programming language independent and web-wide way with the public domain.

Users who want to retrieve SIMAP data directly can use a command-line client for the EJBs or the SIMAP web server. This public server offers three entry points for users:

(1) ProtInfo (protein information system)

(2) SimpleSIMAP (simple SIMAP retrieval using a predefined set of parameters)

(3) AdvancedSIMAP (flexible SIMAP retrieval that provides a wide variety of parameters, sorting and filtering capabilities)

The ProtInfo system allows searching for sequences and proteins in SIMAP by sequence fragments and keywords. The query sequences are searched within the SIMAP sequences using an indexing structure that allows fast searches for similar or partial sequences in large databases. Keyword queries are evaluated using GISE (http://mips.gsf.de/genre/proj/gise). Each ProtInfo query yields a result list of the identical, containing, contained and most similar SIMAP sequences and their related protein entries. Using ProtInfo SIMAP can serve as a huge protein information system that provides very quickly all proteins that share the same or very similar sequences. Links to the SimpleSIMAP and AdvancedSIMAP systems are provided for every sequence.

SimpleSIMAP and AdvancedSIMAP retrieve homologs for given protein sequences that need to be contained in the SIMAP database. Whereas SimpleSIMAP provides only selected parameters and preconfigured search spaces, AdvancedSIMAP allows the user to specify search space, filtering and sorting parameters in a flexible manner. Both types of query result in lists of homologs that are linked in turn to their homologs. So the web interfaces allow users to explore the protein world by homology, starting with a user-defined protein sequence.

4 APPLICATIONS

4.1 Integration into genome databases

SIMAP is already integrated into several genome databases such as the Comprehensive Yeast Genome Database (CYGD) (Guldener *et al*., 2005) and the genome database of the Parachlamydia-related symbiont UWE25 (Horn *et al*., 2004). Here, it serves as a substitute for pre-calculated BLAST searches. The user can choose subsets of interest by taxonomy and can, for example, display only hits in other chlamydiae. In CYGD, SIMAP is also used to display yeast homologs which exist exclusively in certain taxa.

4.2 Speeding up ortholog detection

In order to gain knowledge rapidly using the SIMAP data and to show the increase in performance we adapted some standard applications into the SIMAP system. In these standard applications, the main focus is the detection of orthologous sequences between different species. We implemented a pipeline for the exhaustive detection of bidirectional best hits (BBHs), which is

a standard approach to detecting putative orthologs in prokaryotic genomes.

We have implemented a data repository of these BBHs which is automatically updated whenever a proteome set changes. We provide web service access to these data.

4.3 SIMAP-integrated Inparanoid

Another widely used approach is the Inparanoid procedure (O'Brien *et al*., 2005), which detects orthologs and inparalogs in a pairwise comparison between two genome datasets. We adapted the original program to use SIMAP instead of BLAST searches. Since the BLAST queries are the costly step in this analysis, this adaptation brings down the computation time from ∼42 to ∼3 h when making a comparison between *Drosophila melanogaster* and *Caenorhabditis elegans*, for example. This makes an all-against-all approach tractable, and we provide a couple of comparisons on the SIMAP website. The computation is based on the same system as the BBH tool, so updates in the genome data are instantly taken into account and the data are updated automatically.

4.4 Phylogenetic profiling

Phylogenetic profiling is a well-established method for predicting functional relationships and physical interactions between proteins. Classical phylogenetic profiling computes occurrence profiles from orthologous relationships between proteins and is computationally expensive (Pellegrini *et al*., 1999). For a recent comparison between the newly developed profiling method DIMA and classical profiling, SIMAP was used (Pagel *et al*., 2004) to speed up the ortholog detection step.

5 RESULTS

Data from important public protein databases and completely sequences genomes have been imported into SIMAP over the past two years. At the moment SIMAP contains the recent version of these databases:

- UNIPROT TrEMBL
- UNIPROT SwissProt
- mips nonredH
- PFAM
- PDB
- All genomes from http://pedant.gsf.de
- All genome databases at mips, e.g. CYGD and MatDB
- Many project-specific databases

The total number of ∼7 million protein entries corresponds to ∼3.5 million non-redundant protein sequences. The hit files contain ∼10 billion single hits.

Most of the databases (UNIPROT, PFAM, PDB and PEDANT) are checked weekly for updated entries. The updates are performed using a fully automated procedure that also triggers the FASTA calculations for new sequences.

Comparing the speed of the FASTA binary and SIMAP we were able to demonstrate the enormous advantage of SIMAP, which is up to 800 times faster than FASTA calculations, depending on the lengths of query and database sequences (Table 1).

Table 1. Benchmark results for the FASTA binary and SIMAP (Intel Pentium 4 CPU, 2.4 GHz) for searching the 20 best hits for sample UNIPROT sequences in UNIPROT-TrEMBL

Query ID	Query length	FASTA run time	SIMAP retrieval time (s)	Speed gain by SIMAP
AB020210_1	100	3 min 15.5 s	0.6	325
AB037127_1	200	5 min 07.5 s	0.7	440
AB035325_4	300	7 min 01.8 s	0.9	469
AB026669_3	400	8 min 49.7 s	0.8	662
AB016537_3	500	10 min 50.4 s	0.8	812
CCA6246_1	600	11 min 51.9 s	0.9	791

6 CONCLUSION

We have implemented SIMAP, a database containing the similarity space formed by ∼3.5 million amino acid sequences from >400 organisms by exhaustive similarity searches using the Fasta34 algorithm. The database is capable of handling very large datasets and contains >10 billion data points at the moment. We have implemented a powerful backbone for computation, which employs standard Grid systems as well as BOINC clients. This backbone, in addition to the FASTA heuristic and the incremental update process, enables us to keep up with the ever increasing amount of data by using our in-house hardware in an efficient way. By providing WWW interfaces as well as web services, we make our data accessible to the worldwide community. We have also adapted procedures to detect putative orthologs as example applications. SIMAP is a shortcut whenever all-against-all comparisons of protein sequences are needed and therefore speeds up many analysis steps used in genome projects.

ACKNOWLEDGEMENTS

We are grateful for the many helpful suggestions and comments from the research groups at mips (GSF-Institute for Bioinformatics). We would like to thank Drmitrij Frishman for critical readings of the manuscript.

Conflict of Interest: none declared.

REFERENCES

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

Altschul,S.F. and Gish,W. (1996) Local alignment statistics. *Methods Enzymol.*, **266**, 460–480.

Bastien,O. *et al*. (2004) Fundamentals of massive automatic pairwise alignments of protein sequences: theoretical significance of *Z*-value statistics. *Bioinformatics*, **20**, 534–537.

Dumontier,M. and Hogue,C.W. (2002) NBLAST: a cluster variant of BLAST for NxN comparisons. *BMC Bioinformatics*, **3**, 13.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Gojobori,T. *et al*. (1982) Patterns of nucleotide substitution in pseudogenes and functional genes. *J. Mol. Evol.*, **18**, 360–369.

Guldener,U. *et al*. (2005) CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Res.*, **33**, D364–D368.

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

Higgins,D.G. and Sharp,P.M. (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, **73**, 237–244.

Horn,M. *et al*. (2004) Illuminating the evolutionary history of chlamydiae. *Science*, **304**, 728–730.

Krause,A. *et al*. (2002) SYSTERS, GeneNest, SpliceNest: exploring sequence space from genome to protein. *Nucleic Acids Res.*, **30**, 299–300.

Li,L. *et al*. (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.*, **13**, 2178–2189.

Marcotte,C.J. and Marcotte,E.M. (2002) Predicting functional linkages from gene fusions with confidence. *Appl. Bioinformatics.*, **1**, 93–100.

Mewes,H.W. *et al*. (1997) Overview of the yeast genome. *Nature*, **387**, 7–65.

Needleman,S. and Wunsch,C. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

O'Brien,K.P. *et al*. (2005) Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Res.*, **33**, D476–D480.

Pagel,P. *et al*. (2004) A domain interaction map based on phylogenetic profiling. *J. Mol. Biol.*, **344**, 1331–1346.

Pearson,W.R. (1991) Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith–Waterman and FASTA algorithms. *Genomics*, **11**, 635–650.

Pearson,W.R. (2000) Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol. Biol.*, **132**, 185–219.

Pellegrini,M. *et al*. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl Acad. Sci. USA*, **96**, 4285–4288.

Riley,M.L. *et al*. (2005) The PEDANT genome database in 2005. *Nucleic Acids Res.*, **33**, D308–D310

Rognes,T. and Seeberg,E. (2000) Six-fold speed-up of Smith–Waterman sequence database searches using parallel processing on common microprocessors. *Bioinformatics*, **16**, 699–706.

Rogozin,I.B. *et al*. (2002) Connected gene neighborhoods in prokaryotic genomes. *Nucleic Acids Res.*, **30**, 2212–2223.

Schoof,H. *et al*. (2004) MIPS *Arabidopsis thaliana* Database (MAtDB): an integrated biological knowledge resource for plant genomics. *Nucleic Acids Res.*, **32**, D373–D376.

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

Wilson,C.A. *et al*. (2000) Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *J. Mol. Biol.*, **297**, 233–249.

Wootton,J.C. (1994) Non-globular domains in protein sequences: automated segmentation using complexity measures. *Comput. Chem.*, **18**, 269–285.