HIGH-THROUGHPUT FUNCTIONAL ANNOTATION OF NOVEL GENE PRODUCTS USING DOCUMENT CLUSTERING

A. Renner, A. Aszódi
Structural Bioinformatics Laboratory, Genetics Unit
Novartis Forschungsinstitut GmbH
Brunnerstrasse 59, A-1235 Vienna, Austria
Email: {Alexander.Renner, Andras.Aszodi}@pharma.novartis.com

Gene products differentially expressed in healthy vs. diseased tissues may be considered drug targets since the change in their expression level can be related to the cause and progression of the disease studied. A significant portion of the proteins produced by these genes will be unknown and consequently their function must be characterised. The experimental elucidation of biochemical function must be supported by computational tools which can help predicting the possible function of a given protein from its amino acid sequence. We have designed a high-throughput system which automatically analyses amino acid sequences deduced from differentially represented cDNA clones. The system attempts to assign a biological function to protein sequences by carrying out searches in sequence databanks and by locating functionally relevant motifs in the query sequences. The results delivered by the various prediction methods consist of the annotations of matching sequences and/or motifs, which are free-format texts written by humans and therefore may describe the same concept with synonymous words. It is desirable to present the results in such a way that the annotations describing the same biological function are grouped together. To this end we devised an algorithm that enables the hierarchical clustering of free-format documents based on their contents. The system is capable of detecting and flagging conflicting annotations, and will speed up the interpretation of the function prediction results.

Introduction

Protein function prediction methods are based on the observation that functional similarities can often be deduced from amino acid sequence similarities. Functionally related proteins usually show homology in their sequences that can be detected by various databank search methods employing sequence alignments. The biological function can also often be predicted from the presence of shorter “motifs” or “signatures” in the sequences consisting of conserved patterns of amino acid residues even if the overall homology is low. If a pattern matching method detects a similarity between the query sequence and another known sequence already annotated in a data bank (a “hit”), then one can transfer the function of the hit from its annotation to the query sequence. Intuitively, if several methods deliver the same functional annotation then we can have a higher confidence in the results. This approach is obviously not foolproof, as *e.g.* two proteins may share the same function in the absence of any detectable similarity, or similarities may be present even if the proteins have completely unrelated function. In spite of these difficulties the computational prediction of protein function is an extremely valuable tool that complements the experimental efforts geared towards the understanding of biological processes at the molecular level.

A project aimed at identifying genes differentially expressed in healthy and diseased tissues is currently underway in our Unit. The proteins coded for by these genes may serve as therapeutic targets as the changes in their expression levels could be related to the cause of the disease in question. In order to choose promising drug targets it is essential that information about the biological function of these proteins is available. Due to the large number of differentially expressed clones it would be impossible to perform detailed experimental studies on each of the corresponding gene products and a high-throughput computational prediction system is needed to assign probable function to as many of these gene products as possible.

We have constructed a protein function prediction pipeline that employs a number of independent pattern matching methods in parallel to increase the sensitivity and reliability of function predictions by delivering as many hits to a query as possible. If the annotations of these hits were simply presented in a list, then it would remain the task of a human expert studying the results to evaluate the annotations and come up with a consensus prediction. Given the large number of sequences processed by the pipeline, this procedure is tedious and error-prone. There

is a clear need for a post-processing step that can group those annotations together that describe essentially the same biological function with different words and at the same time highlights annotations that fall into different groups. Both tasks can be accomplished by a system that is capable of clustering free-format documents based on their content. This is a key problem in information retrieval applications extensively discussed in the literature (for an overview, see *e.g.* Schatz, 1997). We have built a prototype system that can cluster protein data bank annotations avoiding any human bias using an automatically constructed knowledge base of biomedical concepts.

Methodology

The Prediction Pipeline

The function prediction system was designed as a pipeline containing four stages, namely DNA similarity searches, DNA → protein translation, protein similarity searches and annotation (Figure 1). The input of the pipeline consists of sequences of cDNA clones identified as "differentially expressed". Every cDNA sequence entering the prediction pipeline is first scanned against the GenBank, EMBL and TAGS databanks using the gapped BLAST2 algorithm (Altschul *et al*, 1997) as supplied in the GCG package (Version 10). If a match with a probability less than 10^{-50} is detected, the gene is considered "not novel" and removed from the pipeline as such probabilities indicate a mismatch of a few bases only. Matches with probabilities between 10^{-10} and 10^{-50} occur for DNA sequences which are similar enough to some sequence(s) so that a direct functional prediction based on DNA-level similarity is possible but they are not identical to these hits and therefore may be considered "novel". Any other sequence for which no good matches with annotations were found enter the protein prediction stage.

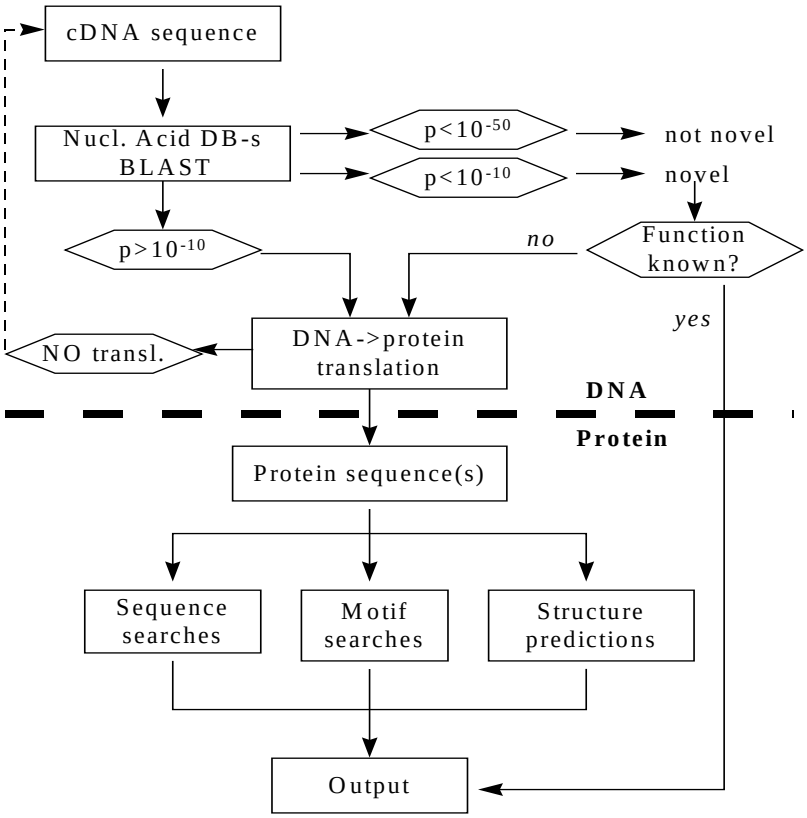


Figure 1. Schematic flow diagram of the function prediction pipeline. The cDNA sequences (possibly more than one per clone) enter the pipeline at the top and will be used as queries in a number of nucleic acid database searches with BLAST. Sequences with no significant matches or with matches to genes of unknown function will be translated into protein sequences and enter the protein prediction stage (below the thick broken line). The results of the various similarity searches and simple structure predictions are then merged into the output.

We perform FASTA searches (Pearson and Lipman, 1988) against the protein data banks SwissProt and PIR with our translated amino acid sequences to get matches possibly missed by the BLAST searches at the DNA level and to extract additional information from the protein databank annotations. The protein domain data bank PRODOM (Corpet *et al*, 1998) is also searched to simplify function assignment when the query sequence has FASTA matches against a number of different protein families. For queries with a few good databank matches we try to identify other sequences belonging to the same family by performing an iterative search inspired by the "SYSTERS" approach by Krause and Vingron (1998). This method is essentially a "walk in sequence space" (Holm, 1998) whereby the first search generates a few matches and then the closest match above a probability threshold (usually 10^{-30}) is selected to pull in more related sequences which had no easily detectable similarity to the original query. The process is repeated until a compact cluster of related sequences is found.

Functional clues can often be gained from detecting relatively short, characteristic sequence segments in proteins which occur in all members of the same functional family. Currently the following motif search methods are included in the prediction pipeline: the MOTIFS program that searches the PROSITE database of flexible patterns (Bairoch and Bucher, 1994), the PROFILESCAN method of Gribskov *et al* (1987), both implemented in the GCG package, and the hidden Markov model engine as implemented in the HMMER package (version 2.0, see

Eddy, 1996). These methods are complementary not only in the sense that they are based on different algorithms, but that they search different (albeit partially overlapping) motif collections. These methods are very flexible in that they enable the definition of new motifs. Work is currently under way to build an in-house motif library of protein families which might serve as interesting targets.

In addition to the similarity search methods described above, our system contains a few simple structure prediction methods. These algorithms do not build complete three-dimensional structural models but only predict certain structural features. We wrote a program that implements von Heijne's transmembrane helix prediction program (von Heijne, 1992), adapted to the requirements of the high-throughput pipeline. Detection of coiled coils is accomplished by invoking the CoilScan program (Lupas *et al*, 1991) from the GCG package and by using a freely available version of the MultiCoil program (Wolf *et al*, 1997). Although these programs provide only indirect evidence to function, they can still be useful in characterising the protein in question.

Most of the programs described above generate copious output which quickly becomes totally incomprehensible if a large number of sequences are analysed. The raw output of the prediction programs are therefore parsed into a hierarchical scheme of HTML files. While this post-processing step enables easy visualisation via Web browsers, additional support is needed for condensing the raw information and presenting a useful summary. This can be achieved by document clustering as described below.

Overview of Annotation Clustering

Once we are able to measure the similarity of any two documents, then appropriate clustering algorithms are available that can cluster the documents efficiently according to the given similarity criteria. A straightforward way to measure document similarity would be to construct a list of "important" words, the *term list*, then for any two documents we would simply check which terms occur in them. If the same terms occur in both documents then they probably refer to the same concept.

This naïve algorithm would be sufficient if every term in the term list corresponded to only one distinct concept in the documents. However, biomedical texts often describe the same entity with several synonymous words. For example, the molecule Apo-3, a member of the tumor necrosis factor receptor family, is also known as DR3, WSL-1, TRAMP or LARD (Kitson *et al*, 1996; Marsters *et al*, 1996). Moreover, there is an intricate network of semantic relations between biological concepts. To compare biomedical documents or annotations efficiently, we would ideally like to grasp at least some of these conceptual relations. To use the above example, the multiply-named Apo-3 molecule plays a role in apoptosis and contains a feature called the "death domain". An expert will of course know that the terms "apoptosis" and "programmed cell death" describe the same concept, and that proteins containing the "death domain" play a role in apoptosis. Consequently, if one document contains the term "apoptosis" and another contains the term "death domain", the third "Apo-3" and the fourth "WSL-1", then they may very well be related, *even if they do not have any terms in common.*

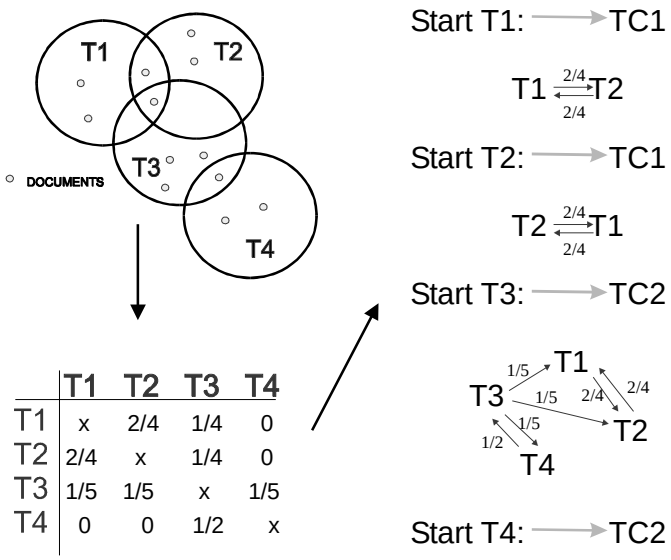
The appropriate similarity criterion between two documents can then be formulated as follows:- "Two documents are similar if they contain terms which are related to the same biomedical concept." Consequently a simple term list is not enough for the detection of document similarity; the terms in the list must be clustered first into *term clusters* representing biomedical concepts. These term

clusters need to be constructed only once. Two documents can then be compared by checking which terms occur in them, and to which term clusters these terms belong. If both documents contain terms belonging to the same term cluster, then the documents probably describe the same phenomenon and may be clustered together.

Automatic Construction of Term Clusters

Our primary term list was a compilation based on the SwissProt data bank keyword list (Bairoch and Apweiler, 1997), and the functional hierarchy of biomolecules in the InCyte data bank. The final term list contained about 900 multiword terms describing proteins, molecular and cellular processes. *Stopwords* having a very unspecific general meaning such as "protein" or "metabolism" that may occur in a wide variety of contexts were filtered out before the term list construction.

Two terms were considered related if they were often found together in Medline abstracts (the "co-occurrence criterion"). Batch Medline searches were conducted with all entries in the term list and a maximum of 1000 documents were returned for each term, 711887 in total. The set of these Medline records comprised the document *training set*. Next, for each term a list was constructed that contained the other terms co-occurring with that given term, sorted such that the most often co-occurring term was the first in the list. Term cluster building began with the first term T_1: it became the first term cluster together with its most often co-occurring term T_2. Then the co-occurrence list of T_2 was investigated: if its most often co-occurring term was the first term T_1, then the cluster was finished. If, however, the term that most often occurred together with T_2 was T_3, then T_3 also became the member of the cluster and the search continued with its co-occurrence list until no appropriate new terms were found. If two terms had the same co-occurrence value then both were followed up the same way (Figure 2).



	T1	T2	T3	T4
T1	x	2/4	1/4	0
T2	2/4	x	1/4	0
T3	1/5	1/5	x	1/5
T4	0	0	1/2	x

Figure 2. The term clustering algorithm. The co-occurrence matrix contains the probabilities with which two terms are contained in the same document, e.g. the entry for the pair T1/T2 is 2/4, meaning that T2 occurs together with T1 in 2 out of 4 documents. For each term, another term with the highest co-occurrence probability is identified and term clusters are constructed such that terms often occurring together will end up in the same cluster. See text for more details.

Since a given term may carry a different meaning in a different context, terms were allowed to be members of different term clusters. The term clusters were "fuzzy", i.e. each term had a membership probability between 0 and 1.

The probability *Pr* of term T_x co-occuring with term T_y is:

$$\Pr(T_x, T_y) = \frac{N(T_x, T_y)}{N_{Tot}(T_x)}$$

whereby $N(T_x,T_y)$ is the number of co-occurring terms and $N_{Tot}(T_x)$ is how many times T_x occurs in all documents in the training set. We can define the probability $PC_k(T_x)$ of a term T_x occurring in term cluster k by finding out how many times the term T_x co-occurred with all other terms contained in the cluster, and normalising this to the total number of occurrences of T_x. Formally,

$$PC_k(T_x) = \frac{\sum_{\substack{T_y \in TC_k \\ T_y \neq T_x}} N(T_x, T_y)}{N_{Tot}(T_x)}$$

whereby $NC_k(T_x)$ is the number of term T_x occurring in term cluster k.

Measuring Document Dissimilarity

The comparison of two documents began with finding those terms that occurred in them. The words in the documents were stemmed before term matching using the Porter algorithm (Porter, 1980). The original algorithm was adapted so that words shorter than five characters or ending in numbers should not be stemmed, e.g. "CD40" should not be stemmed to "CD".

The match score $Match(k,d)$ for a term cluster k in document d was defined as the maximum of the $PC_k(T_x)$ scores of its individual terms matching the document. In the example of Figure 3, term cluster TC_1 matches the first document D_1 through term T_1, and document D_2 through terms T_2 and T_3. The score of TC_1 in D_1 is the probability of T_1 belonging to the cluster TC_1, while the score of TC_1 in D_2 is the larger of the probabilities of T_2 and T_3 in TC_2.

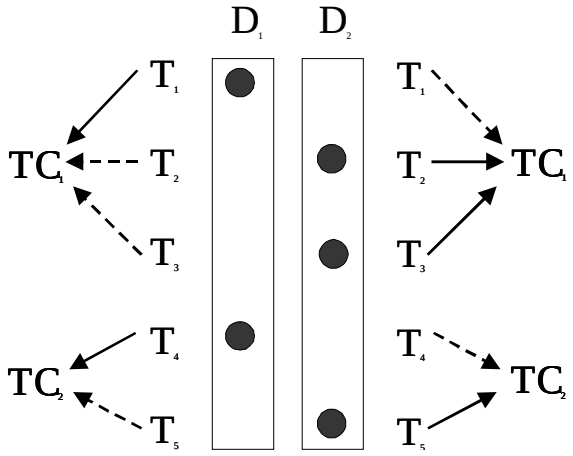


Figure 3. Document comparison with term clusters. Document D_1 contains term cluster TC_1 through term T_1 and term cluster TC_2 through term T_4, while document D_2 contains TC_1 through T_2 and T_3 and TC_2 through T_5 (term-document matches are indicated by the dots). The documents are considered similar even though there is no term that occurs in both. See text for details.

Formally,

$$Match(k,D) = \begin{cases} \max\limits_{T_x \in TC_k} PC_k(T_x) & if \quad T_x \in D \\ 0 & if \quad T_x \notin D \end{cases}$$

The distance between two documents D_1 and D_2 is then defined as:

$$Dist(D_i, D_j) = \frac{1}{N_{match}} \sum_{k=1}^{N} \left(Match(k, D_i) - Match(k, D_j) \right)^2$$

where N is the number of all term clusters, and N_{match} is the number of term clusters that had a match in at least one of the documents. We used the normalization factor N_{match} because most documents contained hits to only a few term clusters and the fact that lots of term clusters were not present in either of the documents did not imply that the documents were similar.

Hierarchical Clustering of Documents

In order to cluster a collection of documents, a distance matrix was constructed that described the pairwise dissimilarities between the individual documents. Standard algorithms are available that, given a distance matrix for a set of objects, can cluster them into hierarchical groups, such as single linkage, complete linkage, average, McQuitty's method, centroid etc. After initial tests we decided to use Ward's algorithm (Ward, 1963) as implemented in the R statistical package (http://lib.stat.cmu.edu/R/CRAN) that constructs clusters in such a way that the intercluster variance is maximised, while the intracluster variance is minimised. This ensures the construction of compact, well-separated clusters.

Results and Discussion

The Term Clusters

The clustering of our term list provided a set of 400 term clusters. Inspection of the results confirmed that the method was indeed capable of grouping the terms such that these represented general biological concepts. A few example clusters are listed below (Table 1).

Table 1. Three representative term clusters. The individual terms are shown together with their respective cluster membership probabilities PC(T).

Cluster 306	PC(T)
RIBOSOMAL+PROTEIN	0.408
TRANSLATION+REGULATION	0.140
Ribosome	0.278
INITIATION+FACTOR	0.513
Translation+initiation+factors	0.410

Cluster 332	PC(T)
PHOSPHOLIPID+BIOSYNTHESIS	0.244
GM2+GANGLIOSIDOSIS	0.038
PHOSPHOLIPID+DEGRADATION	0.410
LIPID+METABOLISM	0.845
LIPID+DEGRADATION	0.918
SPHINGOLIPID+METABOLISM	0.129
LIPID+BINDING	0.527

Cluster 385	PC(T)
Chromatin	0.355
SPLICEOSOME	0.132
MRNA+SPLICING	0.649
RIBONUCLEOPROTEIN	0.269
ALTERNATIVE+SPLICING	0.527
Splicing+factors	0.187
NUCLEOPROTEIN	0.549

Document Clustering

The performance of the document clustering method was tested as follows. We chose 23 Medline abstracts on the topic of “hemoglobin” (the “circle” documents), 46 documents (“arrows”) on the topic “oxygen transport”, which is obviously related to “hemoglobin” and 10 abstracts about the unrelated topic of “tyrosine kinase” (the “diamond” documents). These documents were then clustered using our method and the clusters were displayed graphically (Figure 4). The documents were separated into two distinct clusters: one containing exclusively “diamond”, and another one containing a mixture of “circle” and “arrow” documents. The appearance of the mixed cluster was expected due to the relatedness of the concepts “hemoglobin” and “oxygen transport”. The second cluster is splitting again in two clusters: one mixed “circle/arrow” and one “clean arrow” cluster reflecting the fact that the concept “hemoglobin” implies “oxygen transport” but the opposite is not true.

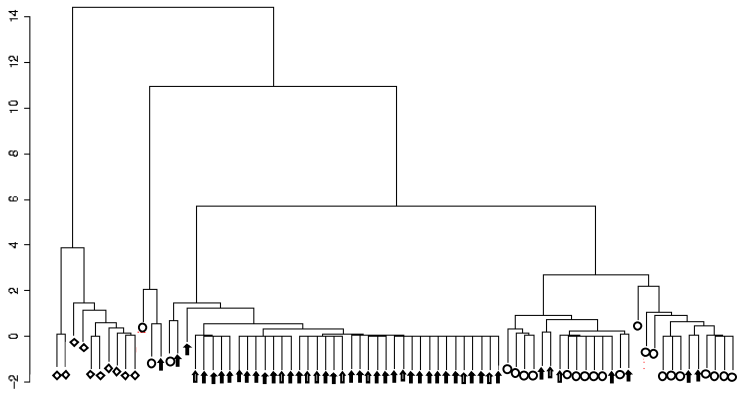


Figure 4. Clustering of documents containing the concepts "tyrosine kinase", "hemoglobin" and "oxygen transport" (diamonds, circles and arrows respectively). The left bar indicates the dissimilarity score.

We have also measured the distribution of the number of annotation clusters per sequence processed by the function prediction pipeline. To this end, we post-processed the raw annotation results for an in-house collection of differentially expressed genes. Out of 7943 sequences, 2177 had no hits at all, 3024 could be annotated directly because the annotations could be grouped into one cluster corresponding to a single concept. The rest of the sequences gave 2 to 8 annotation clusters, indicating the presence of multiple concepts in the hit annotations (Figure 5). In some cases this was due to conflicting annotations, but the majority of multicluster annotations simply reflected the fact that the sequence in question could be described by several biochemical concepts. In particular, covalent modification sites (phosphorylation signals etc.) tended to build a separate annotation cluster on their own but these of course were not in conflict with the "main" function annotation cluster.

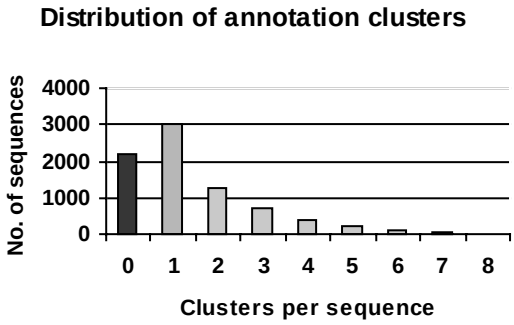


Figure 5. Distribution of annotation clusters. 2177 sequences had no hits in the prediction pipeline (column "0"). Of the 5766 sequences that produced at least one hit, the majority (3024, 52%) could be annotated unambiguously because the annotations could be grouped into a single cluster (column "1"). The number of multiple annotation clusters then fell off rapidly.

Comparison to Other Methods

The key to reliable protein function prediction from sequence information is to combine the results of various methods designed to detect sequence similarities. Our

prediction pipeline also implements this paradigm and in this respect it is similar to several other bioinformatics packages such as GENEQUIZ (see *e.g.* Casari *et al*, 1996) that have been developed over the past few years. While the prediction quality ultimately depends on the underlying sequence analysis algorithms, we felt that usability and the clear presentation of the results are equally important aspects, especially in a multidisciplinary environment like Novartis. This need prompted us to provide an automatic annotation analysis stage before the end user receives the prediction results.

Automatic processing of annotations by computer is a challenging task due to the seemingly irregular nature of free-format texts created by human experts. Several research groups have felt the need to bring some order in this apparent chaos: Fukuda *et al* (1998) have designed the algorithm PROPER to extract synonymous protein names from biological texts. Our term-cluster building approach performs this sub-task automatically. Andrade and Valencia (1998) extracted keywords associated with protein families from annotations based on statistical considerations. Their prototype system, however, needs to be initialised manually, while our approach is fully automatic and the generation of the term clusters are not linked to protein families; in fact, given an appropriate primary term list and an example document database, term clusters can be generated for *any* kind of knowledge domain. We feel that the term-cluster based document similarity measurement technique represents an important step towards the construction of intelligent biological knowledge data bases (Craven and Kumlien, 1999) by automatically assigning terms to meaningful biological concepts: a task that until now had to be performed manually *e.g.* in the construction of the UMLS system (http://www.nlm.nih.gov/research/umls/umlsmain.html).

Outlook

The high-throughput function prediction engine presented here can currently process about a thousand sequences a day. Its flexible architecture enables us to incorporate new components if necessary, thus keeping up with the latest developments in algorithm design. The raw search results are post-processed by detecting annotation similarities based on term clusters which corresponded to known biomedical concepts rediscovered without any human guidance whatsoever. The query sequence can simply be annotated to have the function indicated by the hits belonging to the largest document cluster. If the clustering procedure found only one annotation cluster then the predicted function for the query is unequivocal. If the annotations can be clustered into several independent groups then this may indicate that conflicting matches have been found and a human expert must decide which annotation group is to be preferred, or that more distinct concepts are needed to describe the predicted function of the query. Needless to say, the system does not "understand" biology the same way a human expert would, and due to the fuzziness of the biomedical concepts its performance cannot be expected to be perfect.

Taken together, the annotation clustering approach provides a very efficient way of information compression as the expert evaluating the prediction results needs to check the annotation groups first and he would inspect the individual annotations only in the case of doubt. This will greatly simplify the tedious work of analysing annotations and will speed up the identification of novel genes that can serve as potential drug targets.

Acknowledgements

The authors wish to thank G. Werner and H. Lapp for their valuable comments and suggestions.

References

Altschul, S. F., Madden, T.L., Schaffer, A.A.,
Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. (1997):
Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.
Nucl. Acids Res. **25**: 3389-3402.

Andrade, M.A., Valencia, A. (1998):
Automatic extraction of keywords from scientific text: Application to the knowledge domain of protein families.
Bioinformatics **14**: 600-607.

Bairoch, A., Apweiler, R. (1997):
The SWISS-PROT protein sequence data bank and its supplement TREMBL.
Nucleic Acids Res. **25**:(31-36).

Bairoch, A., Bucher, P. (1994):
PROSITE: Recent developments.
Nucl. Acids Res. **22**: 3583-3589.

Casari, G., Ouzonis, C., Valencia, A., Sander, C. (1996):
GeneQuiz II: Automatic function assignment for genome sequence analysis.
In: Proceedings of the First Annual Pacific Symposium on Biocomputing, pp. 707-709. World Scientific, Hawaii, USA.

Corpet, F., Gouzy, J., Kahn, D. (1998):
The ProDom database of protein domain families.
Nucl. Acids Res. **26**: 323-326.

Craven, M., Kumlien, J. (1999):
Constructing biological knowledge bases by extracting information from text sources.
In: Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology. AAAI Press, Menlo Park, California, USA.

Eddy, S. R. (1996):
Hidden Markov models.
Curr. Opin. Struct. Biol. **6**: 361-365.

Fukuda, K., Tamura, A., Tsunoda, T., Takagi, T. (1998):
Toward information extraction: Identifying protein names from biological papers.
In: Proceedings of the 1998 Pacific Symposium on Biocomputing, pp. 707-718. World Scientific, Hawaii, USA.

Gribskov, M., McLachlan, A. D., Eisenberg, D. (1987):
Profile analysis: Detection of distantly related proteins.
Proc. Natl. Acad. Sci. USA **84**: 4355-4358.

von Heijne, G. (1992):
Membrane protein structure prediction: Hydrophobicity analysis and the positive-inside rule.
J. Mol. Biol. **225**: 487-494.

Holm, L. (1998):
Unification of protein families.
Curr. Op. Struct. Biol. **8**: 372-379.

Kitson, J., Raven, T., Jiang, Y. P., Goeddel, D. V., Giles, K. M., Pun, K. T., Grinham, C. J., Brown, R., Farrow, S. N. (1996):
A death domain containing receptor that mediates apoptosis.
Nature **384**: 372-375.

Krause, A., Vingron, M. (1998):
A set-theoretic approach to database searching and clustering.
Bioinformatics **14**, 430-438.

Lupas, A., Van Dyke, M., Stock, J. (1991):
Predicting coiled coils from protein sequences.
Science **252**: 1162-1164.

Marsters, S. A., Sheridan, J. P., Donahue, C. J., Pitti, R. M., Gray, C. L.,Goddard, A. D., Bauer, K. D., Ashkenazi, A. (1996):
Apo-3, a new member of the tumor necrosis factor receptor family, contains a death domain and activates apoptosis and NF-κB.
Curr. Biol. **6**: 1669-1676.

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

Porter, M.F. (1980):
An algorithm for suffix stripping.
Program **14** (3): 130-137.

Schatz, B.R. (1997):
Information retrieval in digital libraries: Bringing search to the Net.
Science **275**: 327-334.

Ward, J.H. (1963):
Hierachical grouping to optimize an objective function.
J. Amer. Statist. Assoc. **58**: 236-244.

Wolf, E., Kim, P. S., Berger, B. (1997):
MultiCoil: A program for predicting two- and three-stranded coiled coils.
Protein Sci. **6**: 1179-1189.