BIOINFORMATICS

Vol. 15 no. 3 1999
Pages 228–233

A novel method for automatic functional annotation of proteins

Wolfgang Fleischmann, Steffen Möller, Alain Gateau and Rolf Apweiler

The EMBL Outstation — The European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

Received on October 5, 1998; revised and accepted on December 21, 1998

Abstract

Motivation: *To cope with the increasing amount of sequence data, reliable automatic annotation tools are required. The TrEMBL database contains together with SWISS-PROT nearly all publicly available protein sequences, but in contrast to SWISS-PROT only limited functional annotation. To improve this situation, we had to develop a method of automatic annotation that produces highly reliable functional prediction using the language and the syntax of SWISS-PROT.*

Results: *An algorithm was developed and successfully used for the automatic annotation of a testset of unknown proteins. The predicted information included description, function, catalytic activity, cofactors, pathway, subcellular location, quaternary structure, similarity to other protein, active sites, and keywords. The algorithm showed a low coverage (10%), but a high specificity and reliability.*

Availability: *The results can be obtained by anonymous ftp from ftp.ebi.ac.uk/pub/databases/sp_tr_nrdb. The source code is available on request from the authors.*

Contact: *fleischmann@ebi.ac.uk*

Introduction

With the rapid growth of sequence-related databases, there is an increasing need for reliable functional annotation of newly predicted proteins. To cope with such large data volumes, faster and more effective means of creating annotation are required (Baker and Brass, 1998). One promising approach is automatic annotation, which is generated without human interaction (Apweiler *et al*., 1997; Gaasterland and Sensen, 1996).

Several solutions are based on high-level sequence similarity searches against known proteins (Fleischmann *et al*., 1995). Others collect the results of different prediction tools in a simple (Frishman and Mewes, 1997) or more elaborate (Scharf *et al*., 1994) manner. However, several pitfalls of these methods have been reported, e.g. using only the best database hit or ignoring the domain organisation of proteins (Galperin and Koonin, 1998; Bork and Koonin, 1998). An algorithm was designed that is based on multiple sequences and is independent of protein domains.

For the annotation of TrEMBL (Bairoch and Apweiler, 1998), a single sentence describing some properties of the unknown protein is not regarded as optimal annotation. Required is as much information as possible about properties like function(s) of the protein, domains and sites, catalytic activity, cofactors, regulation, induction, pathways, tissue specificity, developmental stages, subcellular location, quaternary structure, diseases associated with deficiencies in the protein, similarities to other proteins, etc.

To enhance the annotation of TrEMBL, we developed a novel method for the prediction of this information. The method tries to find groups of SWISS-PROT (Bairoch and Apweiler, 1998) entries similar to the unannotated protein, extracts the annotation shared by all entries of one group, and assigns this common annotation to the unannotated protein.

System and methods

The proposed algorithm requires four major components. First of all, a reference database serves as the source of annotation. It must be a protein sequence database containing highly reliable and well-curated information. Next, we need a list or better database of protein sequences that are to be annotated, called *target database*. Furthermore, an external database must supply the means to assign proteins to groups. For the nature of these groups a wide range is thinkable, as long as the members of a group are biologically related to each other. Finally, a database is necessary that stores and manages the developed rules, their sources and their usage.

The algorithm was developed for the automatic annotation of TrEMBL. Although it is applicable to a wide range of biological sequence databases, it is easier to follow the flow of information if we give detailed examples for actually used data sources. Since the TrEMBL database is used as a target, SWISS-PROT serves as the reference database and source of annotation. For this example, PROSITE patterns (Bairoch *et al*., 1997) have been chosen as external database and source of grouping information.

© Oxford University Press

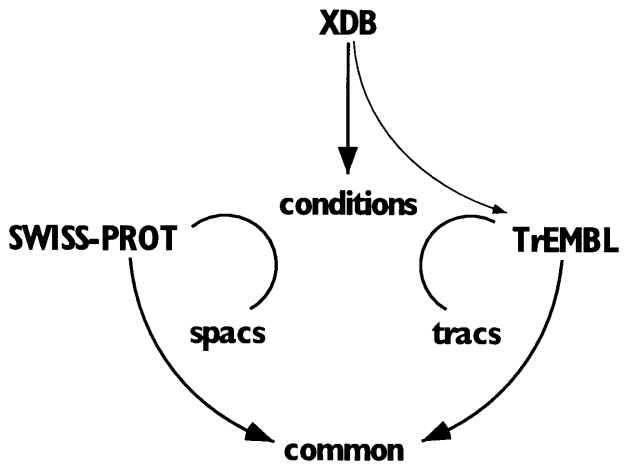

Fig. 1. Dataflow of the automatic annotion. XDB: external database; spacs: set of SWISS-PROT accession numbers; tracs: set of TREMBL accession number.

Algorithm

Figure 1 shows the flow of information during the automatic annotation. The algorithm can be divided into five major steps.

(1) Extract conditions from external database

To make use of the external database, we must translate its information into some kind of conditions that can be tested efficiently on a protein entry. In the PROSITE example, we translate the pattern from the proprietary syntax to a standard unix regular expression. As only the notation has to be changed, this can be done automatically without errors.

(2) Group SWISS-PROT by conditions

We test all conditions against all SWISS-PROT entries and store which entry fulfils which condition. This could be done by matching the translated patterns to whole SWISS-PROT — or in this special case — simply by looking at the links from SWISS-PROT to PROSITE given in the DR PROSITE-lines. For every condition, we have now a set of SWISS-PROT entries that fulfil the condition.

(3) Extract common annotation

For every condition, we take the stored list of SWISS-PROT entries and try to find any annotation that is common to all entries. For practical reasons we allow also annotation that is shared by nearly all entries, while the exact definition of 'nearly' depends on the type of annotation and the size of the set of entries. For instance, the subcellular location 'nuclear' is accepted, if 90% of more than 20 entries are annotated as nuclear protein. This common annotation is linked to the condition and to the set of SWISS-PROT entries and stored in a so called RuleBase. At this stage, the automatically created rules can be checked and improved manually.

(4) Group TrEMBL by conditions

Whenever there are new or changed entries in TrEMBL, or new or changed conditions, we test all conditions against all proteins in TrEMBL. For conditions derived by PROSITE, we have to match the translated patterns from step (1) against all TrEMBL protein sequences.

(5) Add common annotation to TrEMBL

The final step copies the annotation attached to a condition to all TrEMBL entries that meet this condition. Since more than one condition may provide the same information, or the information may be already known from other sources, we avoid collisions and apply the rules in a strictly sequential manner. No existing or already successfully added annotation is replaced.

Furthermore, the new annotation has to be flagged as derived 'BY SIMILARITY'.

Reliability of the conditions

As the reliability of the conditions is crucial to the reliability of the algorithm, a three-step procedure is used to reduce the number of false positive PROSITE hits.

Firstly, the taxonomic classification of the TrEMBL entry must be within the known taxonomic range of the PROSITE pattern. For instance, a match of an a-priori prokaryotic pattern against a human protein is regarded as false positive and filtered out.

Secondly, the significance of the PROSITE pattern match is checked. This is done by a second check of the TrEMBL sequence with a set of secondary patterns derived from the PROSITE pattern. These secondary patterns are computed with the eMotif algorithm (Nevill-Manning *et al*., 1997). The PROSITE database contains a list of all SWISS-PROT proteins that are true members of the relevant protein family. For each pattern, the true positive sequences are aligned and fed into eMotif, which computes a nearly optimal set of regular expressions based on statistical rather than biological evidence. We us a stringency of 10^{-9}, so that each eMotif pattern is expected to produce on random a false positive hit in 10^9 matches.

Thirdly, in cases where a protein family is characterized by more than one PROSITE signature, all signatures must be found in the entry. For instance, bacterial rhodopsins have a signature for a conserved region in helix C and another signature for the retinal binding lysine. If a TrEMBL entry

matches only the helix-C-pattern, but not the retinal-binding pattern, it will not be regarded as a bacterial rhodopsin.

The raw PROSITE hits and all results of the confirmation steps are stored in a hidden section of the TrEMBL entry, but only those hits that satisfy all confirmation conditions are made publicly visible in a DR PROSITE-line.

Modelling of rules

The conditions and the blocks of common annotation are modelled according to the way scientists do the manual annotation of SWISS-PROT. We use the concept of a rule, and demand that every rule has one or more conditions and one or more actions associated to it. If the conditions hold for an unknown TrEMBL protein, all the actions are applied to it.

Formal language for the rules

SWISS-PROT contains controlled vocabulary fields (e.g. keywords), structured simple sentences (e.g. subcellular location), and free text fields (e.g. function). Keeping the TrEMBL objectives in mind, to add rather none than wrong annotation, but to include as much reliable information as quickly as possible, we developed a simple but effective method to process the conditions and the common annotation text.

For every condition we want to test on a protein, we have to define a *condition type* and describe the testing routine in a programming language. For every annotation we want to add or change, we have to define an *action type* and implement the necessary steps.

Now we can express the rules in a formal language that is independent of the used platform and programming language. As an example, Figure 2 shows the rule for ribulose bisphosphate carboxylase proteins. The rules are available online at http://tonic.ebi.ac.uk:8889/rulebase/plsql/ruledemo.exe_query?rule_no=RU*nnnnnn*.

For bookkeeping and version control, we have line tags starting with '#'. Condition lines can be recognised by the question mark followed by a four-letter code for the condition type (e.g. '?PSAC *x*': does the protein sequence contain a PROSITE match with accession number *x*). Similar, the annotation to be added is written with an exclamation mark followed by the action code (e.g. '!SPDE *y*': replace the description line with *y*).

```
#RULE   RU000183
#DATE   1998-05-05
#USER   wfl
#PACK   PROSITE
?PSAC   PS00157
?EMOT   PS00157
!SPDE   RIBULOSE BISPHOSPHATE CARBOXYLASE LARGE CHAIN
!ECNO   4.1.1.39
!CCFU   RUBISCO CATALYSES TWO REACTIONS: THE
        CARBOXYLATION OF D-RIBULOSE 1,5-BISPHOSPHATE,
        THE PRIMARY EVENT IN PHOTOSYNTHETIC CARBON
        DIOXIDE FIXATION, AS WELL AS THE OXIDATIVE
        FRAGMENTATION OF THE PENTOSE SUBSTRATE IN THE
        PHOTORESPIRATION PROCESS. BOTH REACTIONS OCCUR
        SIMULTANEOUSLY AND IN COMPETITION AT THE SAME
        ACTIVE SITE
!CCCA   D-RIBULOSE 1,5-BISPHOSPHATE + CO(2) =
        2 3-PHOSPHO-D-GLYCERATE
!CCCA   D-RIBULOSE 1,5-BISPHOSPHATE + O(2) =
        3-PHOSPHO-D-GLYCERATE + 2-PHOSPHOGLYCOLATE
!CCSU   8 LARGE CHAINS + 8 SMALL CHAINS
!CCLO   CHLOROPLAST
!SPKW   PHOTOSYNTHESIS; CARBON DIOXIDE FIXATION;
        PHOTORESPIRATION; LYASE; OXIDOREDUCTASE;
        MONOOXYGENASE
!FTPS   PS00157: ACT_SITE +6 +6 BINDING OF CO(2)
        ACTIVATES THE ENZYME
```

Fig. 2. An example rule for ribulose bisphosphate carboxylase proteins.

Implementation

The rules, conditions, and actions are stored in a relational database. The schema uses a main table for the rules, which is linked to an action table in a (1:*n*)-relationship. The condition table is linked to the rules using a separate table to model the (*n*:*m*)-relationship. Since this table contains fields for a negation flag and a disjunctive set number, we can easily express any Boolean term with this linkage table. Every row in these tables contains information about the data source, be it manual entry or automatically derived by a certain program. The necessary version control was implemented using the methods used by the EMBL nucleotide sequence database. For every table a so called *audit table* was created that stores deleted or changed rows together with a timestamp and a comment. Thus we can query how a certain rule looked like at any stage of its history.

The rule storage database may also be viewed over the WWW. This access was implemented using Oracle Webserver and PL/SQL-based procedures.

Since SWISS-PROT and TrEMBL are currently stored as structured flat files, all condition and action types have been implemented as Perl subroutines. Figure 3 shows an example of the code for condition type 'protein has a certain PROSITE match' and a simplified example of the code for action type 'add enzyme code to description line'. Furthermore, an implementation of the syntactical constraint for this action type is shown. This constraint ensures that only valid enzyme numbers are added to the TrEMBL entries.

In the current implementation, a wrapper module uses Perl::DBI to access the relational database and hides the details of its relational schema. Only Rule-objects with attached Condition- and Action-Objects are visible to the Perl script doing the automatic annotation.

```
sub condition_PSAC {
  my $psac = shift;
  return $ac =~ /^PS\d{5}$/ ? 1 : 0;
}

sub action_ECNO {
  my $ecno = shift;
  s/^DE.*$/$& (EC $ecno)/m;
}

sub constraint_ECNO {
  my $ecno = shift;
  return $ecno =~ /^\d+(\.(\d+|-)){0,3}$/ ? 1 : 0;
}
```

Fig. 3. Examples of the implementation of conditions, actions, and constraints in Perl.

Results

The implemented procedures have been used for the automatic annotation of TrEMBL releases 5 and 6. To create rules, we selected 295 reliable PROSITE patterns and the 12 105 SWISS-PROT proteins that are known to be true positive matches of these patterns. We chose 25% of the available PROSITE patterns to be able to verify the automatically generated rules manually and to compare them with the underlying protein entries.

Table 1. Results of the automatic annotation showing how many lines of annotation have been added or updated

Database	Type	Added	Updated
TREMBL 5, old entries	DE	9862	
	CC	20 024	6740
	KW	8979	3941
	FT	2476	2443
TREMBL 5, new entries	DE	2199	
	CC	8863	
	KW	3618	1808
	FT	367	247
TREMBL 6, old entries	DE	2136	
	CC	850	175
	KW	384	124
	FT	67	16
TREMBL 6, new entries	DE	960	763
	CC	5328	
	KW	1947	794
	FT	385	6

DE: description lines; CC: comments; KW: keyword lines; FT: sequence features.

Approximately 35% of all TrEMBL entries can be characterized by a PROSITE signature but only around 30% of all TrEMBL entries are true positive matches. The characterization based only on PROSITE would lead to 10–20% of false positive assignments. The confirmation steps reduced the level of characterization by nearly a third to 25%. At this stage, we achieve a level of less than 0.07% of false positive assignments.

The RuleBase was filled with 262 rules using 597 conditions and 1099 actions. The actions have been derived by extracting the common annotation, therefore their distribution reflects the capability of the extraction procedure.

Table 2. The action types stored in the RuleBase

Count	Code	Type	Description
151	SPDE	DE	Replace description line
93	ECNO	DE	Add enzyme number to description line
6	SPGN	GN	Add gene name to gene line
88	CCFU	CC	Add comment on function
93	CCCA	CC	Add comment on catalytic activity
27	CCCO	CC	Add comment on cofactor
40	CCPA	CC	Add comment on pathway
42	CCSU	CC	Add comment on subunit
43	CCLO	CC	Add comment on subcellular location
3	CCTI	CC	Add comment on tissue specificity
1	CCIN	CC	Add comment on induction
145	CCSI	CC	Add comment on similarity
4	CCCC	CC	Add other comment
260	SPKW	KW	Add one or more keywords
55	FTPS	FT	Add feature related to PROSITE match

Count: number of actions; Code: identifier of the action type; Type: first two letters of the affected TrEMBL lines.

Table 3. Comparision of the annotation content of the three databases SWISS-PROT, TrEMBL, and TrEMBLNEW

Database	Lines	Entries	Lines/entry
TrEMBLNEW, (no automatic annotation)	104 548	34 428	3.0
TrEMBL, (automatic annotation)	646 081	150 491	4.3
SWISS-PROT, (manually curated)	1 208 797	73 348	16.5

There is a clear bias to line types with a controlled vocabulary. For almost all rules (260 of 262) at least one keyword has been found. Line types with a restricted syntax (e.g. similarity comment) are found quite often as well. A similarity comment is available for more than half of the rules, an enzyme number and a catalytic activity for more than a third of the rules.

These rules provided annotation for 2951 of the 29 330 new entries in TrEMBL 5, 1443 of the 15 078 new entries in TrEMBL 6, 9658 of the 106 330 entries already stored in TrEMBL 5, and 3254 of the 140 635 entries already stored in TrEMBL 6.

The resulting annotation (Table 1), especially of the new entries, resembles the distribution of line types in the RuleBase (Table 2). For the old entries, the number of annotated lines is significantly lower, which is due to annotation added already to these entries. Since the rules have not been changed between the releases, any annotation added to the old TrEMBL 6 entries is only due to sequence updates and an improved PROSITE pattern matching procedure.

The information content of TrEMBL was significantly increased. Only 32% of the entries in the TrEMBLNEW database, which is not subjected to these automatic annotation methods, contain one or more keywords. After the last automatic annotation run, more than 51% of the TrEMBL database entries contain at least one keyword. However, this must be compared to the manually curated SWISS-PROT database, where 97% of the entries are annotated with keywords.

To compare the coverage of the automatic annotation, a simple measure is the total number of relevant annotation lines. Disregarding information about the organism, its classification, bibliographic references, and cross-links to other databases, there is a small but significant increase in the total annotation content of TrEMBLNEW and TrEMBL (Table 3).

Discussion

The concept of propagating annotation from a reference database to a target database based on the annotation common to certain groups was successfully applied to the automatic annotation of the TrEMBL protein database. Using only a mere quarter of the available PROSITE patterns, more than 68 000 lines of reliable annotation have been added and 17 000 lines updated or improved. It was shown that the amount of added annotation clearly depends on the usage of restricted or simple language in the reference database. The implemented algorithm worked well for controlled vocabulary fields. However, there is room for further improvement in the natural language processing of free text fields (Eisenhaber and Bork, 1998).

Because the algorithm extracts common annotation it is also used to check the consistency of annotation in SWISS-PROT protein families. Since most TrEMBL entries will be moved eventually into SWISS-PROT, the result of the automatic annotation tends to smooth its own template in the long run. Because of the triple-checking of entries incorporated into SWISS-PROT, this scheme could be seen as a beautiful feedback loop of automatically suggested annotation that is checked manually by scientists and used as a template for even more or better automatic annotation. The implemented version control and history function helps to optimise this feedback and allows to spot the TrEMBL entries touched by a certain rule if the involved scientists do not agree with an automatically assigned annotation.

It must be emphasised that the developed mechanism responds to updates of the reference database SWISS-PROT and the external databases. Both are constantly growing, and this additional information is fed into the data flow. The formal language for the rules is independent of the programming language and was designed to be used by imperative languages (‘for every entry: if entry matches condition, add annotation to this entry’) as well as by set-oriented languages like SQL (‘add annotation to all entries that matches the condition’).

Annotating features that are position-specific, like active sites, was possible, but was based mainly on information that was directly extracted from external databases. To deduce such position specific information from groups of SWISS-PROT entries, the protein sequences have to be aligned, and the conservation profile as well as the overall alignment quality assessed.

The procedures have been found to be stable and reliable, therefore we are planning to add more rules to the RuleBase, f.i. all PROSITE patterns and profiles, and use other external databases, f.i. Pfam (Sonnhammer *et al.*, 1997). Since the devised method works with any external database that assigns proteins to groups, we might also use cluster databases for this approach.

An advantage of the common annotation approach is that it may be used not only with protein families, but also with conditions aiming at a higher level in the protein family hierarchy. Only the annotation common to all members of this f.i. superfamily will be copied over. Also it is independent of multi-domain organisation of proteins. If a certain condition aims at a single domain that occurs with other domains, it can be expected that only the annotation referring to this single domain will be found in all relevant SWISS-PROT entries. On the other hand, if the single domain occurs always with another domain, the information for the other domain will be picked up as well.

By using the annotation of multiple entries, the implemented algorithm produces more reliable predicted annotation than methods based on the best hit of a sequence similarity search.

The community annotation approach (Harger *et al*., 1997), where multiple research groups are asked to annotate sequences, may speed up the annotation as well. However, it tends to lack consistent use of nomenclature and annotation rules, which are necessary to any successful querying of the resulting database.

Using this algorithm 10% of the TrEMBL entries have been annotated. To avoid overprediction, we generated rules based on a small testset of only 200 reliable PROSITE patterns. Furthermore, we rejected pattern matches when we expected more than 20 false positive hits in the whole TrEMBL database. It is easily possible to yield a higher coverage, if more patterns and improved conditions are used.

Acknowledgements

This work was supported in part by grant number BIO4-CT97-2099 of the European Commission.

References

Apweiler,R., Gateau,A., Contrino,S., Martin,M.J., Junker,V., O'Donovan,C., Lang,F., Mitaritonna,N., Kappus,S. and Bairoch,A. (1997) Protein sequence annotation in the genome era: the annotation concept of SWISS-PROT + TREMBL. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, **5,** 33–44. AAAI Press, Menlo Park, CA.

Bairoch,A. and Apweiler,A. (1998) The SWISS-PROT protein sequence data bank and its supplement TREMBL in 1998. *Nucleic Acids Res*., **26,** 38–42.

Bairoch,A., Bucher,P. and Hofmann,K. (1997) The PROSITE database, its status in 1997. *Nucleic Acids Res*., **25,** 217–221.

Baker,P.G. and Brass,A. (1998) Recent developments in biological sequence databases. *Curr. Opin. Biotechnol*., **9,** 54–58.

Bork,P. and Koonin,E.V. (1998) Predicting functions from protein sequences — where are the bottlenecks. *Nature Genet*., **18,** 313–318.

Eisenhaber,F. and Bork,P. (1998) Wanted: subcellular localization of proteins based on sequence. *Trends Cell Biol*., **8,** 169–170.

Fleischmann,R.D., Adams,M.D., White,O. *et al*. (1995) Whole-genome random sequencing and assembly of *Haemophilus influenza* Rd. *Science*, **269,** 496–512.

Frishman,D. and Mewes,H.W. (1997) PEDANTic genome analysis. *Trends Genet*., **13,** 415–416.

Gaasterland,T. and Sensen,C.W. (1996) MAGPIE — automated genome interpretation. *Trends Genet*., **12,** 76–78.

Galperin,M.Y. and Koonin,E.V. (1998) Sources of systematic error in functional annotation of genomes: domain rearrangement, non-orthologous gene displacement, and operon disruption. *In Silico Biol*., **1,** 0007. http://www.bioinfo.de/isb/1998/01/0007/

Harger,C., Skupski,M., Allen,E. *et al*. (1997) The Genome Sequence database version 1.0 (GSDB); from low pass sequences to complete genomes. *Nucleic Acids Res*., **25,** 18–23.

Nevill-Manning,C.G., Sethi,K.S., Wu,T.D. and Brutlag,D.L. (1997) Enumerating and ranking discrete motifs. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, **5,** 202–209. AAAI Press, Menlo Park, CA.

Scharf,M., Schneider,R., Casari,G., Bork,P., Valencia,A., Ouzounis,C. and Sander,C. (1994) Gene-Quiz: a workbench for sequence analysis. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, **5,** 348–353. AAAI Press, Menlo Park, CA.

Sonnhammer,E.L.L., Eddy,S.R. and Durbin,R. (1997) Pfam: a comprehensive database of protein domain families base on seed alignments. *Proteins*, **28,** 405–420.