An Environment for Consistent Sequence Annotation and its Application to Transmembrane Proteins

Steffen Möller

Selwyn College

Cambridge, United Kingdom

June, 2001

The dissertation is submitted for the degree of Doctor of Philosophy.

An Environment for Consistent Sequence Annotation and its Application to Transmembrane Proteins

Steffen Möller

Summary

This thesis describes my research leading to the development of a novel software environment to combine multiple tools for an automated prediction of the properties of protein sequences. The test case of the implementation of the tool lies on transmembrane proteins. Part of the environment is a conflict-resolution mechanism and respective rules for its application. This contributes to an improvement of the automated sequence annotation of transmembrane proteins.

The integrated tools include several membrane prediction methods. These are combined to provide an integrated method for both signal peptide prediction and membrane spanning region prediction. A database was created to describe the correlation between individual InterPro entries and transmembrane annotation. This led to the development of specialised predictors, constrained to individual protein families, and set the basis for an automated discovery of constraints for transmembrane topologies.

To facilitate an evaluation of membrane protein prediction, a collection of biochemically well-characterised transmembrane protein sequences was created. As a novelty, raw data from those experiments where added from which a protein’s topology was elucidated. This was applied for analysing aberrations of predictions from the experimental results.

The thesis closes with a novel application of the developed techniques to find determinants for the coupling of G protein-coupled receptors to G proteins and thereby facilitates a functional characterisation of these transmembrane receptors.

Publications

The following publications resulted from the research described in this thesis.

S. Möller, U. Leser, W. Fleischmann, R. Apweiler;
"EDITtoTrEMBL: A distributed approach to high-quality automated protein sequence annotation."
In: Proceedings of the German Conference on Bioinformatics (GCB'98), Zimmermann O., Schomburg D. (eds.); Köln, Germany (1998).

W. Fleischmann, S. Möller, A. Gateau, R. Apweiler;
"A novel method for automatic and reliable functional annotation."
In: Proceedings of the German Conference on Bioinformatics (GCB'98),
Zimmermann O., Schomburg D. (eds.); Köln, Germany (1998).

R. Apweiler, C. O'Donovan, M.J. Martin, W. Fleischmann, H. Hermjakob, S. Möller, S. Contrino;
"SWISS-PROT and its computer-annotated supplement TrEMBL: How to produce high quality automatic annotation."
In: Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (SCI '98 / ISAS '98), Callaos N., Holmes L., Osers R. (eds.), Orlando, FL, USA; 4:184-191 (1998).

S. Möller, U. Leser, W. Fleischmann, R. Apweiler
"EDITtoTrEMBL: a distributed approach to high-quality automated protein sequence annotation."
Bioinformatics 15 (3):219-27 (1999).

W. Fleischmann, S. Möller, A. Gateau, R. Apweiler
"A novel method for automatic functional annotation of proteins."
Bioinformatics 15 (3):228-33 (1999).

S. Möller, M. Schroeder
"Conflict-resolution for the automated annotation of transmembrane proteins."
In: Proceedings of the AISB 2000 in Birmingham, UK

S. Möller, M. Schroeder, R. Apweiler
"Conflict-resolution for the automated annotation of transmembrane proteins."
Computers and Chemistry 26 (1):41-49 (2001).

S. Möller, E. V. Kriventseva, R. Apweiler
"A collection of well characterised integral membrane proteins."
Bioinformatics 16 (12):1159-1160 (2000).

S. Möller, M. D. R. Croning, R. Apweiler
"Evaluation of methods for the prediction of membrane spanning regions in proteins."
Bioinformatics, 17 (7) 646-653 (2001)

S. Möller, J. Vilo, M. D. R. Croning
" Prediction of the coupling specificity of GPCRs to their G proteins."
In: Int. Sys. Mol. Biol., Copenhagen, 2001, appeared as
Bioinformatics 17 (Supplement 1) S174-S181.

Özgün Babur, Steffen Möller and Rolf Apweiler
"TransMotif - Linking sequence motifs with transmembrane annotation."
in preparation

Michael D. R. Croning, Steffen Möller, James Stalker, Arne Stubenau, Alex Kasprzyk,
Ewan Birney, Teresa K. Attwood
“Sequence mining reveals the full extent of the human 7tm receptor families.”
submitted

Posters that were presented on international conferences are shown in the appendix. For references to the location of source code to the programs, the collection of membrane proteins or the predictor or the GPCR coupling please contact Steffen Möller per email: moeller@ebi.ac.uk. On the Internet the page http://www.ebi.ac.uk/~moeller also provides respective links.

Contents

PUBLICATIONS......3

CONTENTS......5

LIST OF FIGURES......9

LIST OF TABLES......12

ABBREVIATIONS......14

PREFACE......15

Audience......16

Acknowledgements......16

II.INTRODUCTION......19

A.Genomic Sequencing......19

1.The Quest......*19*

2.Biological background......*19*

3.Genomic Sequencing......*22*

B.Bioinformatics and Molecular Biology Databases......22

1.Motivation......*22*

2.Current Applications of Protein Sequence Databases......*25*

C.Need for automated annotation......26

1.Background......*26*

III.AUTOMATED ANNOTATION OF PEPTIDE SEQUENCES......31

A.Data flow of protein annotation......31

1.Submission of nucleotide sequence to EMBL......*31*

2.Manual protein annotation......*33*

B.Concept of automated protein annotation......33

1.Direct transfer by sequence similarity......*34*

2.Indirect annotation by protein domains (InterPro)......*35*

3.Abstract sequence annotation (Algorithms)......*35*

4.Future sources......*36*

5.Performance of automated annotation......*36*

C.Evidences for information......38

D.How a biologist would use computer algorithms to annotate a protein sequence....................................38

1.Selection of programs..39

2.Integration of results...40

E.Potential alternatives to a new development...41

F.Reasons for the separate development of an environment for TrEMBL..................................44

1.Lack of understanding of existing SWISS-PROT/TrEMBL annotation..44

2.Lack of annotation evidence...45

3.Lack of conflict-detection and -resolution..45

4.Lack of context-dependent execution of prediction methods..45

IV.ENVIRONMENT FOR THE DISTRIBUTED ANNOTATION OF TREMBL................................46

A.Automation of the protein annotation process...46

1.Rewrite of results into SWISS-PROT format...47

2.Special predicates for information that can not be expressed in SWISS-PROT syntax 47

3.Properties of a system supporting a semantic integration of predictions..49

B.Problems of the automation...49

C.Concept of the Annotation System...51

1.Architecture..53

2.Environment..59

3.Multi-level conflict/redundancy treatment..67

4.Parallelism..68

5.Comparison of EDITtoTrEMBL with other initiatives...68

D.EDITtoTrEMBL as a Multi-Agent-System..70

1.Knowledge..70

2.Belief..71

3.Intention..71

4.Why a Multi-Agent-System (MAS) was not used as a basis to implement EDITtoTrEMBL.................71

E.Technique of conflict resolution..72

1.Introduction..72

2.Representing Knowledge and Uncertain Beliefs...74

3.Conclusion and Future Work...80

F.Implementation of EDITtoTrEMBL..81

1.Layers..81

2.Object Model.....82

G.Future development of EDITtoTrEMBL.....86

1.Integration of a relational database.....86

2.Syntax versus content.....86

3.Workflows.....87

4.Creation and validation of rules.....87

5.Intra-Entry Rules and Inter-Entry Relationships.....88

6.Summary.....88

V. TRANSMEMBRANE PROTEINS.....89

A.Introduction.....89

1.Function of transmembrane proteins.....89

2.Transmembrane topology.....90

3.Determination of membrane spanning regions.....94

4.The difficulty in experimentally determining the topology of transmembrane proteins.....96

5.Dependence on programs for transmembrane segment prediction.....97

6.Topology predictions for heptahelical receptors.....97

B.Test set of Transmembrane Proteins..... 98

1.Motivation.....99

2.Concept.....100

3.Results.....103

C.Evaluation of Predictors of Membrane Spanning Regions.....106

1.Evaluation.....106

2.Results.....108

D.Linking sequence motifs with transmembrane annotation.....121

1.Introduction.....123

2.Methods.....125

3.Results.....130

4.Summary and Conclusion.....137

E.Integration of Prediction methods for the automated annotation of transmembrane proteins.....139

1.Representation of the Biological Knowledge and the Analyser Beliefs.....139

2.Analysers for transmembrane topology prediction.....142

3.Application of the conflict resolution.....145

4.Summary...149

F.Specialisation and data-mining on 7TM proteins...149

1.Motivation and history.. 149

2.Prediction of Membrane-Spanning Regions in Heptahelical Receptors...151

3.Prediction of the coupling specificity of GPCRs to their G proteins..155

VI.SUMMARY AND DISCUSSION... 176

A.Overview...176

B.This work and the future of Automated Sequence Annotation...177

1.Distributed computing becomes routine..177

2.Importance of constraint-based techniques...178

3.Future sources to be integrated...179

VII.APPENDIX...180

A.Posters...180

1.EDITtoTrEMBL...180

2.Facilitating Semantic Queries..181

3.Test set of Transmembrane Proteins.. 182

4.Domain-finding with CluSTr: Re-occurring motifs determined with a database of mutual sequence similarity...183

VIII.REFERENCES..184

List of Figures

FIGURE 1: THE GROWTH OF SWISS-PROT AND TREMBL OVER TIME......................................24

FIGURE 2: A TREMBL ENTRY..29

FIGURE 3: PHASES OF PROTEIN ANNOTATION..31

FIGURE 4: ENTRY AFTER TRANSLATION FROM EMBL PRIOR TO AUTOMATED ANNOTATION..37

FIGURE 5: DISASSEMBLY OF A SWISS-PROT ENTRY FOR SEMANTIC REASONING.............48

FIGURE 6: FLOW OF DATA INSIDE THE FRAMEWORK..52

FIGURE 7: THE ARCHITECTURE OF EDITTOTREMBL..52

FIGURE 8: THE ANNOTATION ENVIRONMENT'S TREE STRUCTURE..53

FIGURE 9: ALGORITHM FOR ANNOTATION...62

FIGURE 10: EXAMPLE FOR DESCRIPTIONS OF TWO ANALYSERS A1 AND A2........................64

FIGURE 11: PROLOG CODE TO DETERMINE A DEPENDENCY BETWEEN TWO ANALYSERS. 65

FIGURE 12: IMPLEMENTATION OF A PLANNER IN PROLOG..65

FIGURE 13: JAVA CODE TO CALL THE PROLOG-IMPLEMENTED PLANNER............................66

FIGURE 14: EXPLAINING CONFLICT RESOLUTION IN A COMMUNICATIONS NETWORK..77

FIGURE 15: GRAPHICAL VISUALISATION OF A SMALL REVISION TREE..................................80

FIGURE 16: CONCEPTIONAL LAYERS OF THE ANNOTATION ENVIRONMENT.......................81

FIGURE 17: UML CLASS DIAGRAM FOR THE JAVA SWISS-KNIFE CLASSES USED WITHIN EDITTOTREMBL...82

FIGURE 18: UML CLASS DIAGRAM OF EDITTOTREMBL..83

FIGURE 19: CLASS INTERACTION DIAGRAM FOR THE ANNOTATION PROCESS..................84

FIGURE 20: SCHEMATIC DRAWING OF A TRANSMEMBRANE PROTEIN WITH FOUR TRANSMEMBRANE REGIONS...91

FIGURE 21: SCHEMATIC TOP-VIEW OF A TRANSMEMBRANE PROTEIN....................................92

FIGURE 22: VISUALISATION OF EXPERIMENTS FOR THE DETERMINATION OF TRANSMEMBRANE TOPOLOGY... 94

FIGURE 23: A TYPICAL ENTRY IN THE ANNOTATION FILE...101

FIGURE 24: FORMAT OF STORED EXPERIMENTAL DATA FOR TRANSMEMBRANE TOPOLOGY..102

FIGURE 25: EXAMPLE ENTRY DESCRIBING EXPERIMENTAL EVIDENCE FOR A TRANSMEMBRANE PROTEIN'S TOPOLOGY..103

FIGURE 26: EXCERPT FROM A MANUAL FORMALISATION OF PROSITEDOC.......................123

FIGURE 27: RELATIVE DISTANCES BETWEEN MOTIF AND TRANSMEMBRANE SEGMENT... 126

FIGURE 28: MSRS FROM MULTIPLE SEQUENCES RELATIVE TO A COMMON MOTIF.......127

FIGURE 29: REPRESENTATION OF CONSTRAINTS...129

FIGURE 30: MOTIFS WITH REGION-BASED CONSTRAINTS..131

FIGURE 31: REGION-BASED CONSTRAINTS..132

FIGURE 32: MOTIFS WITH RESIDUE-BASED CONSTRAINTS...133

FIGURE 33: RESIDUE-BASED CONSTRAINTS...133

FIGURE 34: COVERAGE OF BY REGION-BASED CONSTRAINTS...134

FIGURE 35: COVERAGE BY RESIDUE-BASED CONSTRAINTS...134

FIGURE 36: THE PERCENTAGE OF CONSTRAINTS NOT BE FULFILLED BY ANY OF THE TRANSMEMBRANE PREDICTION TOOLS...137

FIGURE 37: VISUALISATION OF ANALYSERS AND THE DATAFLOW OF ENTRIES..............145

FIGURE 38: REVISABLES PRESENTED TO REVISE...146

FIGURE 39:GRAPHICAL INTERPRETATION OF THE REVISION PROCESS..............................148

FIGURE 40: CONSERVED TOPOLOGY AND RESIDUES OF GPCRS IN FAMILY ONE.............150

FIGURE 41: GRAPHICAL VISUALISATION OF HIDDEN MARKOV MODELS TOWARDS THE DEVELOPMENT OF 7TMHMM...152

FIGURE 42: AN ACTIVATED G PROTEIN APPROACHING ITS EFFECTOR ENZYME.............157

FIGURE 43: VISUALISATION OF THE SEARCH SPACE FOR COUPLING-SPECIFIC PATTERNS OF AMINO ACIDS...159

FIGURE 44: 7TMHMM FOR THE PREDICTION OF GPCR TOPOLOGY.......................................160

FIGURE 45: VISUALISATION OF THE INTRACELLULAR POSITIONS OF PATTERN MATCHES...165

FIGURE 46: DISTRIBUTION OF THE NUMBER OF COMBINATIONS...167

FIGURE 47: VISUALISATION OF MATCHES ON THE PROTEIN SEQUENCE............................169

FIGURE 48: 3D VISUALISATION OF PATTERN SIMILARITIES...171

FIGURE 49: DENDROGRAM OF CYTOPLASMIC RESIDUES OF GPCRS....................................174

FIGURE 50: POSTER FOR THE PRESENTATION OF EDITTOTREMBL......................................180

FIGURE 51: PRESENTATION OF IDEA TO FACILITATE SEMANTIC QUERIES WITH PROLOG...181

FIGURE 52: PRESENTATION OF THE COLLECTION OF WELL-ANNOTATED TRANSMEMBRANE PROTEINS..182

FIGURE 53: DOMAIN FINDING WITH CLUSTR...183

List of Tables

TABLE 1: SIZES OF GENOMES........21

TABLE 2: CONSENSUS OF CLASSIFICATIONS FOR TRANSMEMBRANE PROTEINS........93

TABLE 3: COMMONLY USED INDICATOR PROTEINS FOR TOPOLOGICAL ANALYSIS........95

TABLE 4: CATEGORIES OF TRUST FOR THE TRANSMEMBRANE PROTEINS IN THE TEST SET........100

TABLE 5: INCORPORATED TEST SETS........104

TABLE 6: PERFORMANCE ON KNOWN TRANSMEMBRANE REGIONS........108

TABLE 7: PERFORMANCE ON PROTEINS WITH CHARACTERISED MSRS........109

TABLE 8: PERFORMANCE ON KNOWN MSRS NOT USED IN THE TRAINING SETS OF THE METHOD........110

TABLE 9: PERFORMANCE ON PROTEINS WITH CHARACTERISED MSRS NOT KNOWN TO THE METHOD 111

TABLE 10: COMPARISON OF PERFORMANCE ON AN IDENTICAL SET OF PROTEINS UNKNOWN TO METHODS........112

TABLE 11: DISCRIMINATIVE PERFORMANCE ON SIGNAL AND TRANSIT PEPTIDES........113

TABLE 12: MEMBRANE PROTEINS WHOSE MSRS WERE NOT CORRECTLY PREDICTED BY ANY PROGRAM........114

TABLE 13: MEMBRANE PROTEINS FOR WHICH ONLY SIDEDNESS WAS NOT CORRECTLY PREDICTED........115

TABLE 14: CORRELATION BETWEEN TRANSMEMBRANE PREDICTION METHODS........116

TABLE 15: PERFORMANCE ON G-PROTEIN COUPLED RECEPTORS........118

TABLE 16: PERFORMANCE ON A SET OF SOLUBLE PROTEINS........119

TABLE 17: TRANSMEMBRANE REGIONS OF SWISS-PROT ENTRY Q47825 RELATIVE TO MOTIF PS50286........127

TABLE 18: DESCRIPTION OF DATA SERVING AS INPUT FOR THE ANALYSIS........130

TABLE 19: PERFORMANCE OF INDIVIDUAL MEMBER DATABASES TO DESCRIBE THE POSITIONS OF MSRS........135

TABLE 20: AVERAGE LENGTH OF CONSTRAINTS IN SEQUENCES OF SWISS-PROT AND TREMBL........135

TABLE 21: EFFECT OF THE CONSTRAINTS ON THE AUTOMATED PREDICTION OF MEMBRANE HELICES..136

TABLE 22: LIST OF GENERATED PATTERNS FOUND TO BEST REPRESENT A SPECIFIC COUPLING MECHANISM..164

TABLE 23: PREDICTIONS FOR 10 SEQUENCES THAT ARE UNRELATED TO THE TRAINING SET...166

TABLE 24: PATTERNS AND THE SUBSEQUENCES THESE MATCH...168

TABLE 25: THE NUMBER OF MATCHES IN DEPENDENCE ON THE NUMBER OF TRANSMEMBRANE RESIDUES...170

TABLE 26: DISTRIBUTION OF PATTERNS OVER INTRACELLULAR DOMAINS.....................172

Abbreviations

bp	Basepair
CluSTr	Clustering of SWISS-PROT and TrEMBL
CDS	Coding sequence
cDNA	complementary-DNA, reverse-transcribed RNA
DNA	Deoxyribonucleicacid
EBI	European Bioinformatics Institute, Hinxton, Cambridge, UK
EDITtoTrEMBL	Environment for the Distributed Information Transfer to TrEMBL
EMBL	European Molecular Biology Laboratory, Heidelberg, Germany
EST	Expressed sequence tag
G Protein	GTP-binding protein
GNU	GNU's not UNIX
GPRC	G Protein-Coupled Receptor
GTP	Guanosine triphosphate
MAS	Multi Agent System
MSR	Membrane spanning region
PAGE	Polyacrylamide gel electrophoresis
RMI	Remote Method Invocation
RNA	Ribonucleic acid
SDS	Sodium dodecyl sulfate
TM	Transmembrane
TrEMBL	Automated translation of CDSs from the EMBL nucleotide sequence database

Preface

The demand for an automated conflict-resolution grows with every additional protein sequence discovered, and especially with every additional tool integrated into the automated annotation process (Fleischmann, Möller et al. 1999; Möller, Leser et al. 1999). With the increased performance of sequencing technology and reduced cost, the submission rate of DNA sequences to databases continously increases. Although the DNA of the first higher organisms are now completely sequenced, the number of submissions for new protein sequences is still growing (Figure 1). The discovery of an increasing number of variants of proteins and the addressing of many more organisms will see an enormous increase in the number of sequences to be held in databases over the next few decades. And with every well-described protein sequence available in molecular database, the automated annotation steadily becomes stronger. However, the manual annotation of protein sequences will continue to be required as will the biochemical experiment.

Over the last few years, the newly developed biological techniques became largely intertwined with bioinformatics. The generated data is accessible for computational analysis, the computation is even required for the interpretation of the data. With protein-protein interaction, microarray and 2D gel databases with mass spectrometry now becoming available as a primary resource of most relevant information, tomorrow's biological research will have a chance to be by a much greater extend initiated by computational analysis. While working on this kind of data during my time in Cambridge, the concept of deriving sequence annotation from experimental data set is addressed in this thesis. The formalisation of experimental evidence for transmembrane protein topology may take a role as an extension to the actual sequence annotation (Möller, Kriventseva et al. 2000). It is expected that future database annotation will use results from experimental studies in proteomics directly. The conflict resolution, a main theme throughout this thesis, will then be applied on the original experimental data, besides the conflict-resolution in the final sequence annotation that is addressed in this work.

The thesis's final section, the prediction of patterns for the coupling of GPCRs to their G proteins (Möller, Vilo et al. 2001), represents an effort to close the loop from the sequence analysis back to the biological laboratory. The immediate feedback from both industry and academia was most positive and both local and international contacts are being established to achieve an experimental verification of the patterns determined.

Audience

As a computational biologist it was my intention to make this thesis accessible both to the interested computer scientists and biologists. Therefore, this thesis became slightly more elaborate than it would have become if it were only due to being spread among bioinformaticians.

Acknowledgements

Many people have contributed to this work. Michael Ashburner, Rolf Apweiler, Terri Attwood and Thure Etzold formed my supervising committee. I am thankful for their open door policy and their constructive comments in the yearly seminars. Especially there is to mention Rolf Apweiler as my main supervisor who proposed the task to create an environment for automated sequence annotation in the first place and directed me towards an application on membrane proteins.

The EBI in general, and the SWISS-PROT group at the EBI in particular, is a very friendly working environment. My thanks go to all my colleagues at the EBI who contributed to making it so. Michael Croning, Wolfgang Fleischmann, David Kreil (all EBI), Jong Park (formerly EBI, now MRC, Cambridge), Michael Schroeder (City University London) and Garnet Suck (Medical University of Lübeck, Germany) have been the most influential on my development and I deeply wish to thank them for this.

Special thanks also go to Amos Bairoch, Özgün Babur, Rita Casadio, Sabine Dietmann, David Gilbert, Julia Henckel, Anders Krogh, Ulf Leser, Henrik Nielsen, Burkhard Rost, Oliver Schön, Walther Traut and Dieter Weichenhan. In addition, I have not forgotten about my most influential teachers for my first degree at the University of Hildesheim, Germany, namely Eike Best (now University of Oldenburg), Michael Kolonko (now University of Clausthal-Zellerfeld) and Andreas Pfitzmann (now University of Dresden) who introduced me to what science is alike.

Credits for reviewing this thesis go to Rolf Apweiler and Michael Croning and special thanks go Soren Brunak (DTU, Copenhagen) and Christos Ouzounis (EBI) for their most constructive comments in their role as official examiners during my viva.

Very special thanks and love go to my wife Claudia and our son Maik.

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration unless explicitly stated.

II. Introduction

A. Genomic Sequencing

1. The Quest

From our earliest experiences, we humans know that our bodies can only act within certain limits. These limits vary from person to person. Some strengths and weaknesses are acquired after birth, but many are determined by one’s molecular machinery in the cells of one’s body. The understanding of these molecular processes, helps us to understand these limits, and eventually leads to solutions to overcome these, i.e. to fight disease. Less pragmatically, one may feel a moral duty to reach the best possible understanding of the world which includes the understanding of the machinery of live (Ridley 1999).

2. Biological background

Living organisms can be divided into three kingdoms, i.e. archea, bacteria and eukaryota. All these organisms use a string of deoxyribonucleic acid (DNA) to pass information from the current generation to the next. DNA codes for the RNA, which in turn may code for protein. Some RNA however has functional properties by themselves and is not translated to protein, e.g. the RNA that performes the translation.

The rule is that DNA is only passed from parents to their progeny. However, there are exceptions that demonstrate how dynamic genomic DNA is. Some bacteria are known to be competent to exchange DNA between individuals. A virus can't replicate without using the cellular machinery of its host, they may even integrate with the human genome and drop/pick up genes for their genome. Mostly studied in lower eukaryotic organisms such as the fruit fly, DNA features mobile elements that may move within the genome and take genes with it.Since virusses cannot replicate and hence reproduce themselves, these are not accepted as living organisms. Cells of multicellular organisms (metazoa, which are all eukaryota while some bacteria show properties of multicellular organisms) generally all have the same information in their DNA. External impact as from virusses leads to execeptions to this rule, so do errors during replication, the cloning of organisms from mature (and differentiated) cells proves that the DNA was not lost.

Human cells share with cells of other eukaryotes that these have the DNA separated in the nucleus. This cellular compartment exists except for a limited time during cellular division. Eukaryotic cells also have additional compartments, especially to mention are mitochondria, and, for plants only, the chloroplasts. These contain their own DNA and have their own separate mechanism to read it out and to form RNA and protein. These are now accepted to be evolutionary derived from symbiotically living bacteria that have no nucleus (prokaryotes) (Alberts, Bray et al. 1994).

Not all information is coded in the DNA, e.g. the initial cellular structure. This in particular includes the cell's membranes that must be present to give DNA its context (Alberts, Bray et al. 1994). It is the easiness by which information on proteins can be derived by nucleotide analysis that first raised interest in DNA sequencing. In today’s post-genome area the regulation of RNA transcription, and subsequently of the translation of the transcript to protein, is a major focus of investigations. Defects at any of these stages are known to cause disease, and with human, mouse and chimp DNA being >95% identical one considers the regulation of genes the major difference between species.

Defects in mitochondrial DNA (mtDNA) are also known to be responsible for diseases, e.g. forms of muscular dystrophy. In addition, they are presumed to play an imminent role in ageing (Grey 2000). Nevertheless, the term genomic DNA is usually taken to be nuclear DNA. Genes have frequently been exchanged between mitochondrial and nuclear DNA during evolution. This has resulted in a loss of genes in the mitochondria, shrinking their genome considerably and making it much smaller than the ones of bacterial counterparts (Table 1).

Genome	**Size** (bp)
Human papillomavirus type 18	7857

Human immunodeficiency virus type 1 *	9181
Hepatitis C virus *	9413
Human mitochondria *	16569
Fruit Fly mitochondria *	19517
Human herpesvirus 6 *	159321
Chlamydia trachomatis *	1042519
Methanococcus jannaschii *	1664970
Mycobacterium tuberculosis *	4411529
Escherichia coli K12 *	4639221
Yeast *	12000000
*Arabidopsis thaliana**	118000000
Fruitfly*	1360000000
Zebrafish	1700000000
Mouse	3059000000
Human*	3286000000

Table 1: Sizes of genomes

*The information was collected from the Internet (Dahm 2001; NCBI 2001; Sterk 2001), marked with * are organisms that are fully sequenced.*

Other molecular factors besides the DNA of an individuum are known to influence its development (Alberts, Bray et al. 1994; Ohlsson, Tycko et al. 1998), e.g. cells helping in egg development lead to effects on the child that are not in its genes (maternal effect) and so do modifications to DNA by methylation (imprinting). However, the genomic DNA is most influential and helps to explain many of the observable human properties (phenotypes) (Shriver, Beaudet et al. 1995). Most importantly, the nuclear DNA represents a major element in the control of the development of different tissues (cellular differentiation) and it determines the reaction on extracellular stimuli.

The process to determine the sequence of nucleotides of the DNA of an organism is called "genomic sequencing". The description of the position of regions within DNA that code for RNA (transcription) and regions of its control is referred to as "genome annotation". Hypothetical proteins are derived by an automated translation from such predicted coding sequences. This thesis describes novel mechanisms to predict the properties of proteins by their sequence of amino acids.

3. Genomic Sequencing

DNA sequencing is the process to determine the sequence of nucleic acids daisy-chained by a sugar-phosphate backbone. The process has in its principles not changed since Fred Sanger and colleagues introduced it in 1977 (Sanger, Nicklen et al. 1977; Sanger, Nicklen et al. 1992).

In the big genomic sequencing alliances (Bentley 2000; Consortium 2001) and in commercial ventures (Venter 2001) it is the whole of a specific (or multiple) genome that is sequenced. A certain gene of interest will then be contained because the genome contains all the genes. A researcher in a small lab, especially while the respective genome has not been fully sequenced, will need partial knowledge about a gene to fetch the RNA of a gene and to then have it reverse-transcribed back to DNA which is then in turn sequenced. Otherwise the determination of a gene related to a specific illness is very cumbersome (NHGRI 2001).

There are countless other applications of DNA sequencing. The reverse-transcription of RNA yields the then called complementary DNA (cDNA). This process should be remembered since it gives information on how exactly the original DNA sequence is read out. With the insertion of the yielded cDNA in e.g. bacteria for multiplication, the process of molecular cloning is completed.

B. Bioinformatics and Molecular Biology Databases

1. Motivation

The price for DNA sequencing has dropped massively such that a small whole genome of e.g. a pathogenic bacterium is now within the scope of a well-equipped research laboratory. This was much different 20 years ago when a single fully sequenced gene was eventually submitted as a doctoral thesis. To avoid redundancy of work in biological research it was necessary then, as it is now, to distribute knowledge of nucleotide (RNA and DNA) sequence within the scientific community. Japan, the USA and for Europe the EMBL in

Heidelberg (Stoesser, Baker et al. 2001) created nucleotide databases to which researchers submitted their findings.

Since the early 1990s many specialised databases have been created. Bioinformatic went hand in hand with the sequencing of genomes (Ouzounis, Bork et al. 1995; Ouzounis, Casari et al. 1996) and prospects and challenges of computational biology were perceived at the same time (Casari, Andrade et al. 1995; Casari, Daruvar et al. 1996). Tools like SRS (Kreil and Etzold 1999) and recently EnsEMBL (Hubbard, Barker et al. 2002) have been developed to facilitate access to these. Information on the expression of RNA is stored in expressed-sequence-tags (EST) databases and is i.e. used to verify the existence of otherwise unconfirmed of genes and knowledge about a the expression of a gene in a specific tissue at a specifc developmental stage is of great value per se. Those sub-sequences of a gene that determine the sequence of the RNA which read from it, i.e. not the parts that control the gene's expression, are called coding sequences (CDSs). Their transcription to RNA and a later translation of mRNA to protein follow strict rules that can be performed automatically. However, the automated detection of genes is problematic, splicing and RNA editing (Bass 2001) lead to a variety mRNA for an input to the translation and with the introduction of selenocysteine (Bock, Forchhammer et al. 1991) and recently, for methanogen archea, pyrrolysine as 22nd amino acid (Hao, Gong et al. 2002) was found to increase the difficulty of a complete automation.

Current knowledge in proteins is stored in the protein database SWISS-PROT (Bairoch 2000; Bairoch and Apweiler 2000), carefully manually maintained by several dozens of biologists based on publications and submissions from researchers. PIR and GenPep also need . With the increase of protein sequences that are derived from nucleotide analysis, the number of sequences increased for which no biochemical evidence was yet available. Those were not included in the manually curated database. More importantly, the SWISS-PROT group does not have the capacity to annotate all sequences for which there is a biochemical characterisation. To avoid losing these sequences for comparative studies and to provide a reference and repository for these, the database TrEMBL was introduced to augment SWISS-PROT. This stands for "automated translation of coding sequences from the EMBL nucleotide database". With the high number of new CDSs derived today, TrEMBL gains more and more importance (Figure 1). TrEMBL NEW is an interim database that contains direct translations from EMBL that are not yet integrated within SWISS-PROT or TrEMBL, due with every new release of TrEMBL and SWISS-PROT.

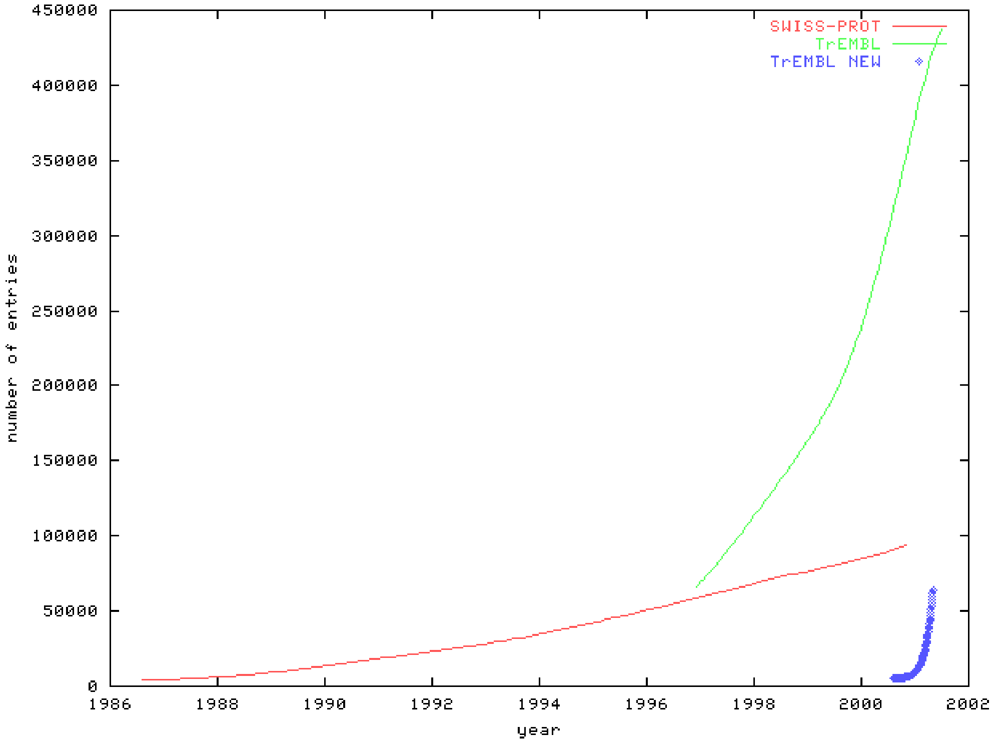


Figure 1: The growth of SWISS-PROT and TrEMBL over time.

The red line shows that SWISS-PROT's growth is linear over the last six years with a steady increase of ~10000 entries per year. TrEMBL, represented with a green line, grows much faster. It is expected to gain 25% in 2001.

Together with the development of SWISS-PROT went the development of a database to store recurrent patterns in protein sequences. Aside this early database PROSITE, today many other databases are available that all chose different methods to represent patterns. Often enough the development of such patterns leads to new biological insights that are published on their own in important journals. Recently these protein domain databases have been integrated in the database InterPro (Apweiler, Attwood et al. 2001) to provide means for centralised access, classification and documentation. The documentation of a pattern contains a summary of known common functionality of proteins. This is essential for a sequence-derived description (annotation) of proteins.

2. Current Applications of Protein Sequence Databases

Nucleotide sequence databases, EST (subsequence of DNA transcript) databases, protein sequence or protein structure databases can all be considered as primary databases. They serve as input for the construction of other secondary databases as for the description of protein domains, metabolism or the transcriptome.

The bookkeeping of information by molecular databases is essential, so essential that the whole of bioinformatics research is often mistaken for this task alone. The most important roles of protein databases are:

- Identification and reference of proteins and their sequences
- Direction of biochemical research, as it is performed by the summary of a protein's properties and involvement in disease, the summary of the amino acid chain's modifications and references to literature.
- As a basis for the creation of secondary databases as for protein domains or for specialisation in protein families or organisms.
- The linking to other databases with related information.

While many additional molecular databases exist, the protein sequence databases are perceived by many as central repositories to understand cellular processes.

Wet-lab work and bioinformatics are getting more and more intertwined. One reason for this is that genomic sequencing and the DNA analysis, and recently also the growing of protein crystals and their structural analysis, became highly automated. Conversely researchers in the lab became very familiar with protein domain information or the notion of sequence similarity. Yet bioinformatics is by many perceived as a separate entity from biological research, which is certainly due to change.

C. Need for automated annotation

1. Background

SWISS-PROT, a high-quality database for protein sequence data, is annotated manually by a team of professional annotators (Bairoch and Apweiler 2000). Also direct submission of protein sequences to SWISS-PROT is possible and SWISS-PROT curators trace publications of sequences in the scientific literature. Most sequences are derived from a semi-automatic search of the public nucleotide sequences for potential genes that have been submitted, e.g. the EMBL nucleotide sequence database (Stoesser, Baker et al. 2001). However, the ever-increasing amount of data creates the need for new techniques to complement manual curation as submissions from large sequencing projects do not offer any biochemical characterisation of proteins.

TrEMBL was introduced 1996 to complete SWISS-PROT with the protein sequences that could be derived from the nucleotide sequence, but that the human curators of SWISS-PROT could not yet fully annotate (SP-TrEMBL), and those peptides that not covered by SWISS-PROT (REM-TrEMBL) like immunoglobulins, synthetic or very short peptides. The concept of SWISS-PROT + TrEMBL allows the provision of a comprehensive protein sequence database without lowering the editorial standards of SWISS-PROT. Every entry in TrEMBL is enriched by automated annotation. This means that every TrEMBL entry is analysed by a set of programs, and from their output new or improved annotation is derived, in order to facilitate an easier pre-selection of sequences for further studies.

It is widely recognised that SWISS-PROT and TrEMBL provide the best possible short summary of the proteins's functions and sequence properties. Biochemical research is not equally done on all proteins, since a focus is given to those that seem most essential to metabolism, known to be involved in diseases or are easy to study. This influences the range of papers that are available for the SWISS-PROT annotators. With an additional pressure to finish the annotation of completely sequences genomes, SWISS-PROT must be regarded as intrinsically biased, i.e. it is not a description of a random selection of proteins and as a consequence the sequences are not randomly selected either.

The annotation of TrEMBL is performed in a semi-automated manner, which will be eluded in more detail later, but it should be stressed, that an automation of the annotation process does not mean that the automated annotation would not be biased. Simple reasons for this are that the physiology of plants is much less known than of the fruit fly, the bacterium *E. coli* and other model organisms. Rules for automated annotation are not complete or perfect. Presumably, this will never be achieved. Also these are derived from SWISS-PROT that is biased. Consequently the automation of annotation will introduce another bias of annotation towards proteins that can be described with the respective current rule set. This again means a preference towards better-understood proteins or protein-families as recurrent patterns in well studied protein sequences or those with an experimentally defined protein structure, tend to have a better description of the pattern's role and therefore have a bigger impact on the annotation of otherwise undescribed proteins.

Data representation in SWISS-PROT and TrEMBL

Both TrEMBL and SWISS-PROT are internally maintained in a relational database. The databases are distributed in *flat-files*, which is a textual representation of the database in a format that is shown in Figure 2 (Bairoch and Apweiler 1999). They consist of a large number of structurally homogeneous *entries*, each representing one protein sequence together with its annotation. The biologist can access the data via the Internet, e.g. browsing entries in SRS (Etzold, Ulyanov et al. 1996; Zdobnov, Lopez et al. 2000; Zdobnov, Lopez et al. 2002) or NiceProt (Gasteiger 2001), or can download the whole data for its inspection in a text editor. The annotation describes the function of the protein, post-translational modifications (phosphorylation, acetylation...), domains and sites, secondary and quaternary structure, similarities to other proteins, diseases associated with deficiencies, sequence conflicts, variants and further information when considered most relevant.

```
ID   Q12618      PRELIMINARY;      PRT;   476 AA.
AC   Q12618;
DT   01-NOV-1996 (TrEMBLrel. 01, Created)
DT   01-NOV-1996 (TrEMBLrel. 01, Last sequence update)
DT   01-JUN-2001 (TrEMBLrel. 17, Last annotation update)
DE   Acyl-COA desaturase (EC 1.14.99.5) (Stearoyl-COA desaturase)
(Fatty
DE   acid desaturase) (Delta(9)-desaturase){EA2}.
GN   OLE1.
OS   Ajellomyces capsulata (Histoplasma capsulatum).
OC   Eukaryota; Fungi; Ascomycota; Pezizomycotina; Eurotiomycetes;
OC   Onygenales; Onygenaceae; Ajellomyces.
OX   NCBI_TaxID=5037;
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=DOWNS;
RX   MEDLINE=96122844; PubMed=8538376;
RA   Gargano S., Di Lallo G., Kobayashi G.S., Maresca B.;
RT   "A temperature-sensitive strain of Histoplasma capsulatum has an
RT   altered delta 9-fatty acid desaturase gene.";
RL   Lipids 30:899-906(1995).
CC   -!- CATALYTIC ACTIVITY: STEAROYL-COA+AH(2)+O(2)=OLEOYL-COA+A+
CC       2 H(2)O{EA2}.
CC   -!- COFACTOR: IRON{EA2}.
CC   -!- SUBCELLULAR LOCATION: INTEGRAL MEMBRANE PROTEIN. ENDOPLASMIC
CC       RETICULUM (BY SIMILARITY){EA2}.
CC   -!- DOMAIN: THE HISTIDINE BOX DOMAINS MAY CONTAIN THE ACTIVE SITE
CC       AND/OR BE INVOLVED IN METAL ION BINDING (BY SIMILARITY){EA2}.
CC   -!- SIMILARITY: TO CYTOCHROME B5 DOMAIN{EA1}.
CC   -!- SIMILARITY: TO OTHER FATTY ACID DESATURASES{EA2}.
DR   EMBL; X85963; CAA59939.1; -.
DR   InterPro; IPR001199; Cyt_B5.
DR   InterPro; IPR001522; Desaturase.
DR   Pfam; PF00173; heme_1; 1.
DR   Pfam; PF01069; Desaturase; 1.
DR   ProDom; PD002221; Desaturase; 1.
DR   PROSITE; PS00191; CYTOCHROME_B5_1; UNKNOWN_1.
DR   PROSITE; PS50255; CYTOCHROME_B5_2; 1.
DR   PROSITE; PS00476; FATTY_ACID_DESATUR_1; 1.
KW   Endoplasmic reticulum{EA2}; Fatty acid biosynthesis{EA2}
KW   Heme{EA1}; Iron{EA2}; Membrane; Oxidoreductase{EA2};
KW   Transmembrane{EA2}.
**
**   ################     SOURCE SECTION     #################
**   H.capsulatum Ole1 gene (strain DOWNS)
**   [1]
**   MEDLINE; 96122844.
**   Gargano S., Di Lallo G., Kobayashi G.S., Maresca B.;
**   "A temperature-senstive strain of Histoplasma capsulatum has
**   an altered D9-fatty acid desaturase gene";
**   Lipids 30:899-906(1995).
**   [2]
**   1-1590
**   Gargano S.;
**
**   Submitted (28-MAR-1995) to the EMBL/GenBank/DDBJ databases.
**   S. Gargano, International Institute of Genetics, &
**   Biophysics, CNR,
**   Via Marconi 10, 80125 Naples, ITALY
**   source          1..1590
**                   /organism="Histoplasma capsulatum"
**                   /variety="capsulatum"
**                   /strain="DOWNS"
**   CDS             join(1..295,390..1525)
**                   /gene="Ole1"
**                   /product="delta-9 fatty acid desaturase"
**                   /EC_number="1.14.99.5"
**                   /product="stearoyl-CoA desaturase"
**                   /db_xref="PID:g757860"
**   CDS_1_OUT_OF_1
**   15-MAR-1996 (Rel. 47, Last updated, Version 2)
**   #################    INTERNAL SECTION    ##################
**EV EA1; Rulebase; -; RU000446; 09-JUN-2001.
**EV EA2; Rulebase; -; RU000581; 09-JUN-2001.
**GO GO:0004768; stearoyl-CoA desaturase;
**GO GO:0005506; iron binding;
**GO GO:0005624; membrane fraction;
**GO GO:0005783; endoplasmic reticulum;
**GO GO:0006118; electron transport;
**GO GO:0006633; fatty acid biosynthesis;
**ID XXXX_AJECA
**PM ProDom; PD002221; Desaturase; 53; 316; T; 29-MAR-2001;
**PM Pfam; PF00173; heme_1; 356; 430; T; 16-OCT-2000;
**PM Pfam; PF01069; Desaturase; 59; 298; T; 16-OCT-2000;
**PM PROSITE; PS00191; CYTOCHROME_B5_1; 380; 387; ?; 28-SEP-2000;
**PM PROSITE; PS00476; FATTY_ACID_DESATUR_1; 271; 285; T; 02-MAY-2000;
**PM PROSITE; PS50255; CYTOCHROME_B5_2; 349; 427; T; 28-JAN-2000;
**RU RU000086; 26-MAI-1999.
SQ   SEQUENCE   476 AA;  53790 MW;  A91A9CE2A865CADB CRC64;
     MALNEAPTAS PVAETAAGGK DVVTDAARRP NSEPKKVHIT DTPITLANWH KHISWLNVTL
     IIAIPIYGLV QAYWVPLHLK TALWAVVYYF MTGLGITAGY HRLWAHCSYS ATLPLKIYLA
     AVGGGAVEGS IRWWARGHRA HHRYTDTDKD PYSVRKGLLY SHIGWMVMKQ NPKRIGRTEI
     TDLNEDPVVV WQHRNYLKVV IFMGIVFPML VSGLGWGDWF GGFIYAGILR IFFVQQATFC
     VNSLAHWLGD QPFDDRNSPR DHIVTALVTL GEGYHNFHHE FPSDYRNAIE WHQYDPTKWT
     IWIWKQLGLA YDLKQFRANE IEKGRVQQLQ KKIDQRRAKL DWGIPLEQLP VIEWDDYVDQ
     AKNGRGLIAI AGVVHDVTDF IKDHPGGKAM INSGIGKDAT AMFNGGVYNH SNAAHNQLST
     MRVGVIRGGC EVEIWKRAQK ENKEVESVRD EYGNRIVRAG AQVTKIPEPI TTADAA
//
```

Figure 2: A TrEMBL entry

*Lines that start with ** are not visible in the distribution of the entry and serve internal purposes only, often a preparation for parts becoming public. This includes evidences for annotation, a specification of matches to protein domain databases and the EMBL entry the coding sequence was derived from. Parts derived from automated protein annotation are tagged with EAX evidence tags.*

Every entry consists of a number of lines, each starting with a two-letter identifier, the *line tag* (see Figure 2). The line tag identifies the content or type of the line. Important line types are comment lines (tag `CC`), the feature table (`FT`) and keyword lines (`KW`). Each entry has a name, which is stored in the `ID` line, and a unique identifier, the accession number, stored in the `AC` line. The content of most line types follows fixed rules and employs a controlled vocabulary. This facilitates searching of the text and it is absolutely crucial for an automated handling of the data in TrEMBL. Although SWISS-PROT does this better than any other protein database, it is still a long way towards a completely formal storage of biological information.

III.Automated Annotation of Peptide Sequences

A. Data flow of protein annotation

It should first be explained in what context the automated protein annotation is understood in the SWISS-PROT group. The protein sequences undergo three major phases of annotation (Figure 3).

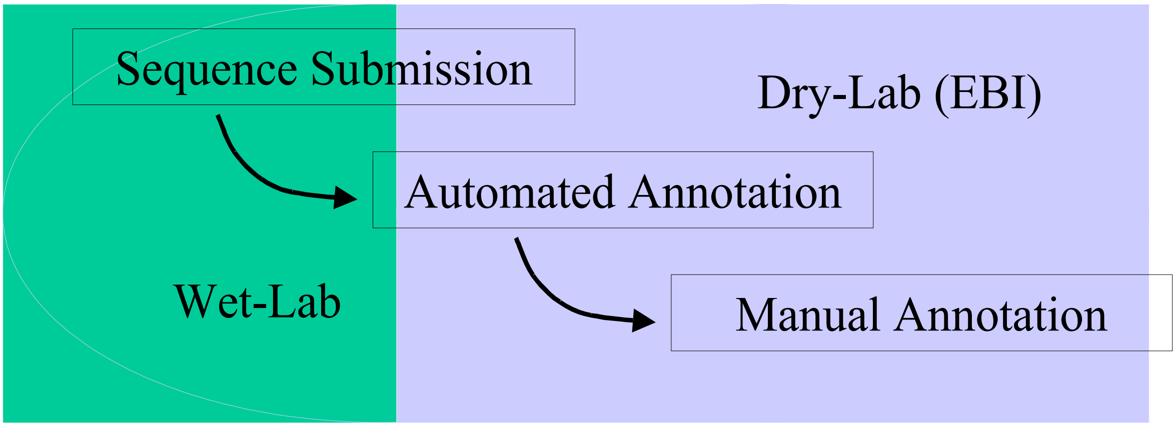


Figure 3: Phases of protein annotation

Automated annotation connects the wet-lab with the dry lab. The larger wet-labs and the dry-labs perform an automated annotation of their data.

1. Submission of nucleotide sequence to EMBL

Methods of bioinformatics are applied at different stages in the protein sequence annotation. The process starts with the wet-lab researcher who submits a sequence to EMBL. It should be expected that a similarity analysis, presumably including search for protein domains, will be performed prior to sequence submission. Also the coding sequence will be determined manually or with the aid of a computer. But the results of these tools are not interpreted in an automated fashion. Hence, methods of the automated annotation are used by single researchers in the wet-lab.

Larger sequencing efforts like of the Sanger Centre next to the EBI have their own bioinformatics research groups. These apply bioinformatics to their DNA analysis prior to submission. The results of such efforts can’t be understood without computers, which led to initiatives like Ensembl (Butler 2000; Birney 2001). This efforts represents the strongest integration of wet-lab work and automated annotation. Its automated protein annotation is performed in collaboration with the SWISS-PROT group at the EBI.

The initial information available for every sequence in TrEMBLis derived from either large sequencing efforts or individual's submissions. The minimal information transferred from the EMBL entry to a TrEMBL entry is:

1. Protein's amino acids sequence together with the information if the sequence is complete or fragmentary.
2. Organism name and classification
3. Organelle (if applicable)
4. References

A submitted protein sequence would be added to TrEMBL directly. Today, most protein sequences first appear as direct translations from nucleotide sequences, which are produced in large-scale genome projects and are available through the DDBJ/EMBL/Genbank nucleotide sequence databases. All coding sequences in the nucleotide sequence databases are first translated into preliminary TrEMBL protein sequence entries.

A major step in the production of TrEMBL is the removal of redundancy from EMBL submissions. In here lies a strong manual verification process. Identical non-fragmented protein sequences as derived from EMBL do appear as a combined single entry in TrEMBL. In EMBL, these may appear multiple times due to multiple submissions or a very strong homology throughout a part of the respective animal kingdom. In addition, variants of the same gene are summarised in TrEMBL and SWISS-PROT in a single entry. These are due to the sequencing of the gene in multiple individuals or to multiple occurrences of respective coding sequences in the genome.

Novel sequences are then annotated automatically and stored in TrEMBL. The TrEMBL entries undergo the same automated protein-domain assignment procedure as SWISS-PROT, except that these are not manually verified.

2. Manual protein annotation

TrEMBL entries are gradually moved into SWISS-PROT after their manual curation by biologists and passing the internal quality control. Thereby they are enriched with information extracted from publications, improved by expert knowledge and enriched by sequence analysis. SWISS-PROT entries are regularly updated with data found in research or review articles. Additionally external experts provide considerable help. The production of TrEMBL is explained elsewhere in detail (Apweiler, O'Donovan et al. 1998).

Manual work is not only performed for the initial submission to SWISS-PROT but also for a continuous updating of the SWISS-PROT entries. With the increasing number of entries in SWISS-PROT this is of increasing importance to keep the information stored in sync with new biochemical discoveries. The manual work is the biggest bottleneck in the creation of a protein database with the high quality of SWISS-PROT.

B. Concept of automated protein annotation

A variety of different methods have been created to augment information on a protein sequence in an automated manner. For every source of information, rules determine its interpretation and reformulation to fit the syntax and semantics of SWISS-PROT (Bairoch and Apweiler 1999) (see Figure 6). Any transfer of information to TrEMBL is made according to sequence similarity to entries in SWISS-PROT, directly or indirectly, since the sequence is the only information that is available.

Programs can be implemented in multiple ways. Prior to the use of the “environment for the distributed information transfer to TrEMBL” (EDITtoTrEMBL) that is described in chapter IV, the programs used for the automatic annotation exchanged data via files and were controlled by a UNIX Makefile. Makefiles allow for every program to determine its input and output files and dependencies of programs on these files. The program make then executes these programs in parallel and in the correct order. While useful, this approach is only sufficient while the interdependencies between programs are simple and can be modelled statically. EDITtoTrEMBL follows a different approach. It dynamically determines the right procedure to be employed for a specific sequence, based upon a declarative description of the analysis programs. As a result, different sequences are generally subjected to different combinations of analysers, in different orders.

The annotation process treats entries individually. This means that every single sequence plus relevant parameters for each of the methods that could be derived from the current annotation, are fed into the respective algorithm and additional annotation is eventually added to the entry. The concept for this approach stems from the representation of SWISS-PROT and TrEMBL in flat files. Now with the storage of SWISS-PROT and TrEMBL in a relational database, the entries could be selected that share applicable methods and parameters. But this is only true while the methods selected are either very much dependent on each other and the parameters invariant or the number of tools incorporated very small. With an increased complexity of the annotation process any preselection of homogenous entries is expected to become impracticable as it would lead to very many small groups of complete homogeneity and hence become equivalent to an individual treatment of entries.

The SWISS-PROT group now has implemented methods for automated protein sequence annotation that work on three different levels of abstraction as explained below.

1. Direct transfer by sequence similarity

The CluSTr project at the EBI (Kriventseva, Fleischmann et al. 2001) provides a matrix of protein-protein similarities. A clustering on the basis of these similarity scores determines groups of proteins. Groups with a homogenous annotation in SWISS-PROT can be assumed to share this annotation with otherwise unannotated proteins in TrEMBL. The respective annotation common to SWISS-PROT entries of the group is transferred during the process of automated annotation. This approach can equivalently be used to find new areas of local similarity for the finding and description of new protein domains (appendix). Many methods to determine sequence similarity have been implemented. For CluSTr the method of Smith-Waterman was chosen (Smith and Waterman 1981). The maximum sequence similarity of multiple sequences goes together with an optimal alignment of these sequences (Jeanmougin, Thompson et al. 1998). This alignment can be the basis to transfer annotation from a "master entry" in SWISS-PROT to unannotated entries in TrEMBL, a process called feature propagation (Velds 1999).

2. Indirect annotation by protein domains (InterPro)

InterPro (Apweiler, Attwood et al. 2001) is an effort to represent a uniform integration of protein domain databases. Matches to domains of a sequence trigger the annotation of this sequence. The annotation transferred is determined by the annotation common to all entries in SWISS-PROT that match the same entries in InterPro (Fleischmann, Möller et al. 1999; Babur, Möller et al. 2001). An extension to this work is presented in section D.

3. Abstract sequence annotation (Algorithms)

For an algorithmic protein sequence annotation, there must be a model for sequences available to reflect distinct properties of the mature protein. Domain databases can be seen as a collection of models for a certain algorithm, i.e. regular expressions, profiles and Hidden Markov Models.

The prediction of membrane spanning regions or molecular modelling use abstract models on proteins mainly derived from theoretical considerations. If enough data is available then this model can alternatively be induced from sequences rather than a-priori knowledge being applied. This then leads to e.g. artificial neural networks as an underlying model. With the existence of a model an algorithm can be developed to act as a classifier, which is a decider if a sequence belongs to a certain group of sequences and if a subsequence has a certain property, by checking if the model can be applied to a given sequence.

4. Future sources

With proteomics, the analysis of all proteins in an organism, being more and more established the available experimental data on proteins increases dramatically. Future development will link this data with the annotation of current protein databases. There is no intermediate layer, the literature, in between that would have to be manually processed. Examples are the protein sequences that are confirmed by mass spectroscopy and the subcellular localisation of proteins. The spots of 2D gels give evidence on the protein expression levels and DNA microarrays give direct evidence on RNA expression levels of gene. This information will be merged with existing nucleotide and protein sequence databases to yield a quantitative integration of theses databases.

A continuous effort is made to determine variants in protein sequences and their imposed change in the protein's function.

5. Performance of automated annotation

Historically, the performance of the automated annotation of TrEMBL was evaluated by the number of lines or keywords added to the blank entry or to the database as a whole. Today with an increased specialisation of the rules for annotation this flawed, since e.g. the substitution of a GO term with a more specific GO term is invariant to the number of lines or items.

Leaving the hidden section of the TrEMBL entry aside, Figure 4 displays the TrEMBL entry from Figure 2 prior to its automated annotation. The rules that lead to its annotation are described in a work of Wolfgang Fleischmann that extends an effort of Alain Gateau (Fleischmann, Möller et al. 1999). The paper gives numbers on the lines added to the automated annotation, however, for prior mentioned reasons no actual recalulation of these values is presented at this place. One should instead compare the development of protein domain databases on which most rules for automated annotation rely. At that time (late 1998) the coverage of SWISS-PROT (and TrEMBL) by domain databases was around

40%, today it is 82% (72% for TrEMBL) with sequences in SWISS-PROT having doubled (113 thousand) and at least tripled for TrEMBL (670 thousand). Rulesfor annotation have also dramatically improved with now several people being dedicated to rule creation (Kretschmann, Fleischmann et al. 2001).

```
ID   Q12618      PRELIMINARY;      PRT;   476 AA.
AC   Q12618;
DT   01-NOV-1996 (TrEMBLrel. 01, Created)
DT   01-NOV-1996 (TrEMBLrel. 01, Last sequence update)
DT   01-NOV-1996 (TrEMBLrel. 01, Last annotation update)
DE   H.capsulatum Ole1 gene (strain DOWNS).
OS   Ajellomyces capsulata (Histoplasma capsulatum).
OC   Eukaryota; Fungi; Ascomycota; Pezizomycotina; Eurotiomycetes;
OC   Onygenales; Onygenaceae; Ajellomyces.
OX   NCBI_TaxID=5037;
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=DOWNS;
RX   MEDLINE=96122844; PubMed=8538376;
RA   Gargano S., Di Lallo G., Kobayashi G.S., Maresca B.;
RT   "A temperature-sensitive strain of Histoplasma capsulatum has an
RT   altered delta 9-fatty acid desaturase gene.";
RL   Lipids 30:899-906(1995).
DR   EMBL; X85963; CAA59939.1; -.
SQ   SEQUENCE   476 AA;  53790 MW;  A91A9CE2A865CADB CRC64;
     MALNEAPTAS PVAETAAGGK DVVTDAARRP NSEPKKVHIT DTPITLANWH KHISWLNVTL
     IIAIPIYGLV QAYWVPLHLK TALWAVVYYF MTGLGITAGY HRLWAHCSYS ATLPLKIYLA
     AVGGGAVEGS IRWWARGHRA HHRYTDTDKD PYSVRKGLLY SHIGWMVMKQ NPKRIGRTEI
     TDLNEDPVVV WQHRNYLKVV IFMGIVFPML VSGLGWGDWF GGFIYAGILR IFFVQQATFC
     VNSLAHWLGD QPFDDRNSPR DHIVTALVTL GEGYHNFHHE FPSDYRNAIE WHQYDPTKWT
     IWIWKQLGLA YDLKQFRANE IEKGRVQQLQ KKIDQRRAKL DWGIPLEQLP VIEWDDYVDQ
     AKNGRGLIAI AGVVHDVTDF IKDHPGGKAM INSGIGKDAT AMFNGGVYNH SNAAHNQLST
     MRVGVIRGGC EVEIWKRAQK ENKEVESVRD EYGNRIVRAG AQVTKIPEPI TTADAA
//
```

Figure 4: Entry after translation from EMBL prior to automated annotation

The TrEMBL entry from Figure 2 is shown before it was further described by the automated protein annotation process. It is the state that entries in TrEMBL NEW are in. Prior to the annotation a decision is made if the EMBL entry translates to a new protein, to a variant of another protein or if it is identical to a sequence already stored in SWISS-PROT or TrEMBL..

Many more examples of unannotated entries are found in the TrEMBL NEW database. Before these are annotated, first a decision is made, if the sequence is already found in TrEMBL or SWISS-PROT and only afterwards the TrEMBL NEW entry can become a new TrEMBL entry or be merged with an existing entry, respectively. The process of integrating novel transcribed entries into the-TrEMBL is described in a paper of Claire O'Donovan (O'Donovan, Martin et al. 1999).

C. Evidences for information

The automated translation from EMBL entries to TrEMBL should be perceived as the first step of automated annotation, with additional annotation added as previously described. It is necessary, that the information provided to individual protein sequences can be selectively revoked if additional evidence hints that it was not completely reliable. But the information should only be deleted if there is no other additional confirmation for it beyond the source now declared as unreliable.

These difficulties are overcome by the introduction of evidence tags to accompany the annotation. These tell both the human reader and the program for the automated annotation, what the source of every individual piece of information is and, consequently, if certain information can be updated.

D. How a biologist would use computer algorithms to annotate a protein sequence

This section describes an intuitive design process for protein sequences with computer-implemented algorithms. It was a major design goal for the environment for automated protein annotation presented in chapter IV that the process is similar to an approach a biologist would choose. This should facilitate the tracking of errors. Also a human annotator has a plan for the execution of programs to use computational resources efficiently.

A trivial point in this context is that if a biochemical analysis of a certain property is available, then a tool which has only the purpose to confirm this property should not be executed. But if the first source of information has no such high reliability, then it may well be requested to start a second tool to ask for a confirmation of the first's result, and potentially even more tools if the second result seems surprising.

1. Selection of programs

An entry's full annotation involves many programs, not only for confirmations, but especially since different tools were created for different biological properties, each of which with different strengths and weaknesses. The experience of the biologist together with the annotation already gathered from previous results will result in an execution-plan for the applications the annotator understands and has access to. The output of the first program run will be used to determine the next programs to be applied.

Some constraints and heuristics apply:

- Only those tools feasible for annotation are executed

  Programs that are known not to be valuable for a specific kind of protein will not be run.

- Computationally cheap tools first

  If only a single protein would be annotated, a human would probably select only very few and very good tools. But for a larger set of proteins CPU time becomes an issue and the biologist would plan a sequence of program invocation. If a computationally cheap tool with a low specificity but a high sensitivity can exclude certain properties of a protein, this information can be used to decide not to investigate further in this direction and hence save many hours of CPU time.

- Reflect dependencies between programs

  The output of one program may serve as input for another. These dependencies should be reflected in the execution plan of the applications. Eventually the same tool should be executed twice when additional evidence is introduced to change a parameter.

2. Integration of results

To get a comprehensive overview about a protein‘s properties the output of multiple programs need to be analysed. Usually different programs have different output formats. Hence, a comparison can be performed directly on the returned values but must be syntactically analysed prior to a comparison.

It is important to look for semantic identity. If two programs predict the same properties, then these achieved a consensus. The annotation should be assumed as correct with the always-present caveat that no prediction can be better than the underlying model. To verify the predicted annotation, the sequence annotation must be seen as a whole. The following questions will be addressed:

1. Is the annotation consistent?

   Even if all programs responsible for a certain protein feature agree on this annotation, they all may be in conflict with other annotation previously determined. If a program‘s function is well understood it may be possible to determine what part of the annotation should be regarded as wrong and hence removed.

2. What is the probability of a false prediction?

   Many programs offer an estimate of the probability at which the program‘s authors assume a prediction to be reliable. Though it is very hard to determine to what degree the results of two programs can be assumed as independent. Only from independly derived results, the redundancy of the results can give additional confidence. If both programs use very different approaches, though, it may be justified to regard both results as independent. Experience together with an in-depth analysis of the participating programs according to a reference annotation would be used for an evaluation of the program‘s reliability.

3. Error propagation

   With the source of an annotation not being closely linked to the annotation of the sequence, the retraction of the wrong annocation becomes hard to remove once the evidence for the annotation is considered erroneous. Automated annotation means the transfer from what is known to the unknown. To correct all entries to which th annotation was (directly e.g. by sequence comparison or indirectly by e.g. protein domain databases) transferred is even harder. Hence, the automated annotation should be most careful.

Although the process described above is quite straightforward, the process is very hard to formalise.

E. Potential alternatives to a new development

There are also other groups trying to widen the bottleneck in the manual annotation of protein sequences.

The following paragraphs summarise the most prominent environments that were considered to have brought new impulses to the field. The following paragraphs, short summaries are presented of the concepts of other annotation systems with respect to the research described in this thesis.

- MAGPIE

  MAGPIE (Gaasterland, Maltsev et al. 1994; Gaasterland and Sensen 1996; Gaasterland and Sensen 1996; Gaasterland and Lobo 1997) is a tool for the assistance of manual annotation of a whole genome. The following description of the system was extracted from (Gaasterland, Sczyrba et al. 2000): *The microbial MAGPIE genome annotation system accepts assembled, unannotated contiguous genome sequence data as input. For finished genome sequence data, the system performs three phases of analysis. Phase 1 identifies coding regions, builds DNA-level and protein-level analysis requests for the coding regions, manages the execution of the requests on remote or local machines, and parses the output data into local relational facts [...]. Included in the phase 1 data collection are comparisons of each protein sequence encoded in the query* *genome with the proteins from each available complete genome or chromosome. In phase 2,* MAGPIE *generates a functional report for each coding region by synthesizing all overlapping functional evidence into a single view according to user-specified preferences* (Gaasterland and Lobo 1997)*. A series of decision rules generate one or more suggested functions for the gene product of the coding region. Alignments with proteins from other genomes are used to determine potential boundaries between protein domains. [...]. The system also suggests one or more functional categories for the protein based on categories of similar functions in Escherichia coli, yeast, Synechocystis sp., and other complete genomes with assigned function categorization. [...] The synthesis of evidence overlays PROSITE* (Hofmann, Bucher et al. 1999)*, Blocks* (Henikoff, Henikoff et al. 1999)*, and PRINTS* (Attwood, Croning et al. 2000) *functional motifs with sequence alignments so that a biologist user can easily see whether motif information is consistent with suggested enzyme functions. In phase 2, biologist users are expected to confirm or edit the annotations of individual gene products through interactive forms. [...] In phase 3, the* MAGPIE *system generates a series of whole-genome reports.*

Looking back it is not clear, to what extend the system described in this thesis, EDITtoTrEMBL, was influenced by the existence. While Terry Gaasterland of MAGPIE has the background to create a system like EDITtoTrEMBL herself, network distribution and consistency checks are features of MAGPIE, the two fields of application (genome vs. protein annotation) were too different to think of an extension of MAGPIE, if we would be allowed access to the source in the first place. The abstraction of the annotation system towards EDITtoTrEMBL that would in principle also allow genomic annotation, was achieved at a later stage.

- PEDANT

PEDANT (Frishman and Mewes 1997) assists human annotation with a run of a static set of selected programs. While it gathers information from different tools, the semantic integration of tools is left to the biologist using the platform.

- GeneQuiz

GeneQuiz (Scharf, Schneider et al. 1994; Casari, Ouzounis et al. 1996) performs a similarity analysis of all hypothetical genes in a genome. These are manually edited. This approach yields a conditional transfer of protein descriptions of well-annotated entries to unannotated entries. From the project's web site (http://www.ebi.ac.uk/research/cgg) the following description was extracted:

GeneQuiz is an integrated system for large-scale biological sequence analysis, that goes from a protein sequence to a biochemical function, using a variety of search and analysis methods and up-to-date protein and DNA databases. Applying an "expert system" module to the results of the different methods, GeneQuiz creates a compact summary of findings. It focuses on deriving a predicted protein function, based on the available evidence, including the evaluation of the similarity to the closest homologue in the database (identical, clear, tentative, or marginal). The analysis yields everything that can possibly be extracted from the current databases, including three-dimensional models by homology, when the structure can be reliably calculated.

- PSORT

PSORT (Nakai and Horton 1999) may be the approach most similar to the one chosen in this thesis. Like PEDANT, it tries to give a picture of the whole protein with no manual interference. The selection of programs for annotation is chosen dynamically, but not on an abstract level as presented in this work.

- BioScout

BioScout is a commercial product of the company LION AG, Heidelberg, Germany. Its main selling points are rules for biological inference and the high number of external sources syntactically integrated, rather than the semantic integration of multiple tools. It is also strong in the interaction with users, which is no intention for the automated annotation of TrEMBL.

- GeneWeaver

GeneWeaver (Bryson, Joy et al. 1999; Bryson, Luck et al. 2000) was the first agent system for genome annotation that presented itself as such. It features many of the concepts presented in the following section, especially the dynamic integration of tools.

- EnsEMBL

EnsEMBL (Hubbard, Barker et al. 2002) is a project that aims to present a unique entry point to the human genome. Genes and their location on chromosomes plus the most essential tools are visualised to the biologist and accessible as a database to computational analysis. Most recently a distributed annotation system (DAS) was introduced to facilitate an integration of prediction methods (Dowell, Jokerst et al. 2001).

F. Reasons for the separate development of an environment for TrEMBL

The potential gain of shared code with other groups, if this could be agreed, was considered low in comparison of the additions necessary to fulfil our needs. GeneWeaver was created not before 1999 and hence not available. MAGPIE's expertise is primarily on the nucleotide level and full genome annotation, also performed by GeneQuiz. Especially at that time there seemed to be no real alternative to a totally new system.

1. Lack of understanding of existing SWISS-PROT/TrEMBL annotation

Technically it may be possible to adapt the previously listed environments for the annotation of TrEMBL. However, the environments would not understand the annotation that is currently present in the database. These would have to be rephrased and the resulting information would have to be merged with the information available.

The gain of an integration of this complex environment would then be fairly limited. Requiring similar technical efforts, it should be preferred to integrate underlying tools with no additional interface rather than performing an integration of environments. The alternative would be a loss of control and an increase of redundancy with no or only little gains.

2. Lack of annotation evidence

SWISS-PROT and TrEMBL introduced evidence tags to explain how and on what basis the annotation was created. To use external environments makes this process less transparent, both for the annotation system of TrEMBL (important for automated conflict resolution) and for the reader of the annotation.

3. Lack of conflict-detection and -resolution

The emphasis of this work lies on the semantic integration of tools for the automated sequence analysis. This is apparently not addressed at all by any of the tools currently existing and in bioinformatics at large.

4. Lack of context-dependent execution of prediction methods

None of the frameworks has an abstract notion to describe the pre- and post-conditions of tools that are integrated in their framework. This forbids any execution planning as required to limit execution time and to ensure that the system scales with the number of tools and sequences to be annotated.

IV.Environment for the Distributed Annotation of TrEMBL

In the introduction it was stressed how important automated protein sequence annotation is and that there is currently no approach to achieve a comprehensive, scaling and reliable annotation for all protein sequences. This chapter introduces the system that implements a framework for sequence annotation close to the principles outlined in section D.

A. Automation of the protein annotation process

It was necessary, as explained below, to formalise SWISS-PROT entries in order to facilitate allowing semantic reasoning and an implementation of this concept is presented. In this section, the different levels of formalisms are explained in detail.

- *Description of tools*

  Every tool used in the annotation process must be known to the system. This means that the programs name and information on how it is accessed must be stored in a computer-understandable format.

- *Dependencies of tools on each other*

  Dependencies of tools on each other could be directly stored. However, this would be hard to maintain. Instead, a description of the tools performance gets formalised to let the environment compute the dependencies from this information.

- *Analysis of results*

  The program's output must be interpreted. Any interpretation should be performed on a common syntax to avoid the need to create semantically identical rules working with different syntaxes.

1. Rewrite of results into SWISS-PROT format

An ontological unity (common semantics) can be achieved by a translation of any program output to the nomenclature used in SWISS-PROT. This works fine for SWISS-PROT keywords and most of the FT lines, but is problematic for lines that contain much free text as SWISS-PROT and TrEMBL CC lines or the DE line.

However, this is still computationally feasible since any information created by programs can be expressed in a formal manner. It must be ensured that also the rules that provide the automation of the transfer from SWISS-PROT avoid informal knowledge representation, i.e. that these don't store CC lines that can not be understood in a later run of the same annotation machinery. Admittedly, this is not fully possible at this stage since there is yet no complete ontology for molecular biology. Some rules have to be updated in this respect.

2. Special predicates for information that can not be expressed in SWISS-PROT syntax

SWISS-PROT is designed to express known facts about proteins. It has weaknesses though in expressing constraints, or more abstract, partial knowledge about proteins, or even that certain earlier assumptions about the respective protein under scrutiny have now proven wrong.

More vague information on a protein that is harder to formalise can be derived from protein-protein-interaction experiments that are not fully reliable (Lappe, Park et al. 2001) or, as emphasised in this work, from protein domain databases. A match from a protein sequence to a protein domain database increases the knowledge about the protein. While this rarely yields a complete description of the protein, some information can be derived, even if this applies only to a few residues. Such information is most valuable for the evaluation of predicted sequence annotation.

The challenge was to extend the SWISS-PROT syntax such that semantically weaker expressions can be made, in order to use these for constraints of semantically stronger expressions. For this thesis, the introduction of predicates was chosen for an implementation of statements derived from SWISS-PROT entries. These are hidden from the final TrEMBL annotation and serve as means for communication between the different programs for sequence annotation. The predicates are directly accessible from the programming language PROLOG and the conflict resolution implemented in this language. For this purpose, a translation of SWISS-PROT entries to facts in PROLOG was performed. This disassembly of a SWISS-PROT or TrEMBL entry into atomic statements still very much resembles the original SWISS-PROT entry and keeps its nomenclature (Figure 5). Such individual statements are later referred to as “annotation elements”. These can be either right or wrong, meaning that they are individually revisable. The TrEMBL entry displayed in Figure 2 would appear as the following collection of facts:

```
id(q12618,'q12618').
de(q12618,'acyl-coa desaturase (delta(9)-desaturase) ').
de(q12618,ec,'(ec 1.14.99.5)').
os(q12618,['ajellomyces capsulata (histoplasma capsulatum)']).
oc(q12618,
[eukaryota,fungi,ascomycota,pezizomycotina,eurotiomycetes,onygenales,on
ygenaceae,ajellomyces]).
gn(q12618,'OLE1').
cc(q12618,'subcellular location','integral membrane protein.
endoplasmic reticulum (by similarity)').
cc(q12618,cofactor,iron).
cc(q12618,domain,'the histidine box domains may contain the active site
and/or be involved in metal ion binding (by similarity)').
cc(q12618,'catalytic activity',[stearoyl-coa,'ah(2)','o(2)']=[oleoyl-
coa,a,'2 h(2)o']).
cc(q12618,similarity,'to cytochrome b5 domain').
cc(q12618,similarity,'to other fatty acid desaturases').
% no feature
kw(q12618,['endoplasmic reticulum','fatty acid
biosynthesis',’heme’,’iron’,’membrane’,’oxidoreductase’,’transmembrane’
)
sq
(q12618,"MALNEAPTASPVAETAAGGKDVVTDAARRPNSEPKKVHITDTPITLANWHKHISWLNVTLII
AIPIYGLVQAYWVPLHLKTALWAVVYYFMTGLGITAGYHRLWAHCSYSATLPLKIYLAAVGGGAVEGSIRW
WARGHRAHHRYTDTDKDPYSVRKGLLYSHIGWMVMKQNPKRIGRTEITDLNEDPVVVWQHRNYLKVVIFMG
IVFPMLVSGLGWGDWFGGFIYAGILRIFFVQQATFCVNSLAHWLGDQPFDDRNSPRDHIVTALVTLGEGYH
NFHHEFPSDYRNAIEWHQYDPTKWTIWIWKQLGLAYDLKQFRANEIEKGRVQQLQKKIDQRRAKLDWGIPL
EQLPVIEWDDYVDQAKNGRGLIAIAGVVHDVTDFIKDHPGGKAMINSGIGKDATAMFNGGVYNHSNAAHNQ
LSTMRVGVIRGGCEVEIWKRAQKENKEVESVRDEYGNRIVRAGAQVTKIPEPITTADAA").
'//'(q12618).
```

Figure 5: Disassembly of a SWISS-PROT entry for semantic reasoning

The name of the predicate reflect the two letter line code of SWISS-PROT and TrEMBL. The first argument is the accession number of the transformed SWISS-PROT entry. It is *a key that allows individual predicates to stand on their own, i.e. with no context assigning them to a sequence.*

The representation of the feature table is explained in more detail in section E.

3. Properties of a system supporting a semantic integration of predictions

With the introduction of a controlled vocabulary for all tools the following should be achieved or achievable, respectively

- Identity

  The restricted usage of words allows reducing much of semantic checks to a syntactical level, e.g. literal identity.

- Conflict

  With a syntactical reference to biological properties, these can be included in other formalisms. One such application may be the storage of rules for biological inference. Another is the expression of contradictions.

A complete coverage of all possible conflicts is not possible – at least within the time available for this project. For specialised domains though the knowledge required allowing a qualified decision on conflicts can be formalised.

B. Problems of the automation

Many different potential sources can be used for protein sequence annotation. This heterogeneity of data impedes to consistently estimate the retrieved data‘s reliability. Also the semantic interpretation of the data is a real challenge.

An automated system of sequence annotation must overcome these problems. For the systems performance, the objectives are very similar to the ones eluded in section D of a human co-ordinator of annotation tools. These are:

- Minimisation of CPU time
- Ordered execution of programs
- Selection of applicable methods

- Reliability of results

One should not rely on the potential of the system to detect conflicts. Instead, the focus on correctness of any statement made during the annotation procedure is most essential. Since the annotation process is a sequential process with later added annotation partially depending on annotation added earlier, any error will be propagated and eventually be discovered only very late in the annotation process. Any such error may have consequences on the selection of programs, hence a wrong annotation may lead to certain programs never being executed.

The aim is to provide a stable framework where different analysing programs can be integrated in a plug-and-play manner. Since both the number of such programs and the amount of raw sequence data or outdated earlier annotation are increasing rapidly, certain issues for such frameworks are getting increasingly important:

- The performance of a tool contributing annotation must be carefully evaluated.
- The integration of arbitrary analysis programs should be possible at ease, allowing dynamic reconfiguration and recovery in case of a failing module. Additionally, it be possible to integrate remote services, which could be offered by third parties via the Internet.

The availability of a mechanism for the distribution of processes, which is also suitable for a dynamic load balancing in a farm of workstations, facilitates the inclusion of heterogeneous hardware platforms and operating systems. This permits the integration of programs that are only available on specific platforms.

The system should treat sequences individually to avoid semantically inappropriate or redundant processes. Most programs for sequence analysis require certain conditions to be met by a TrEMBL entry. If these are not fulfilled then the program should not be applied to the respective entry.

Interdependencies between analysis programs should be taken into account. The order in which those are applied is important, as the output of one program may be necessary as the input for another. In the presence of cyclic dependencies programs might have to be started more than once.

As some programs use the output of other programs as their input, it is necessary that data exchanged follows defined syntactical standards and a common ontology to represent semantics to ensure consistency.

The results of different analysing programs may be redundant as they may compute the same type of information. In these cases, the redundancy in the output should be removed.

In the following a concept of an implementation of a system is introduced, that considers the issues just listed. The application of the system on the annotation of transmembrane proteins is described in section E.

C. Concept of the Annotation System

Figure 6 presents a graphical overview on the annotation system. The system treats the automation of annotation as a workflow problem (Georgskopoulos, Hornick et al. 1995; Casati, Grefen et al. 1996). It provides a flexible software framework for arbitrary analysis programs.

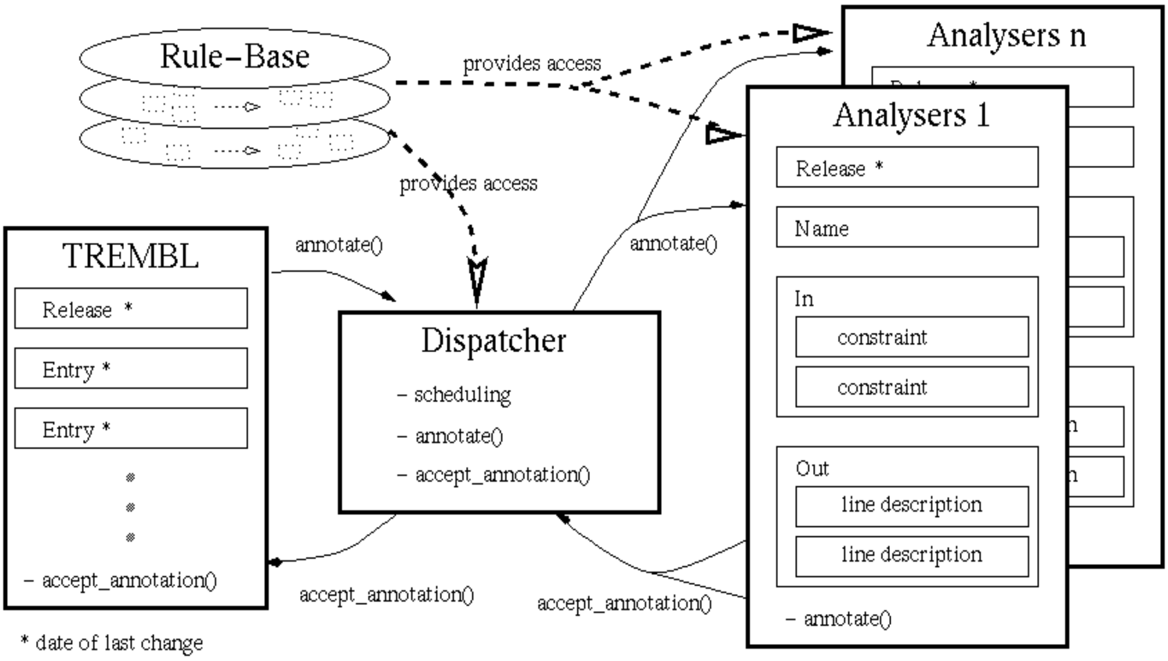


Figure 6: Flow of data inside the framework

A basic distinction is made between Analysers (units that wrap an individual source of information) and Dispatchers (units that control the data flow and that preserve semantic consistency). TrEMBL entries due for annotation are submitted to the dispatcher only, who is transparently performing the annotation by further submission to analysers and the integration of results.

To achieve an appropriate treatment of individual sequences, the execution of these programs is controlled by high-level descriptions of the conditions that must be fulfilled to make their application meaningful. Using these descriptions, a sequence of analyses is deduced dynamically at runtime.

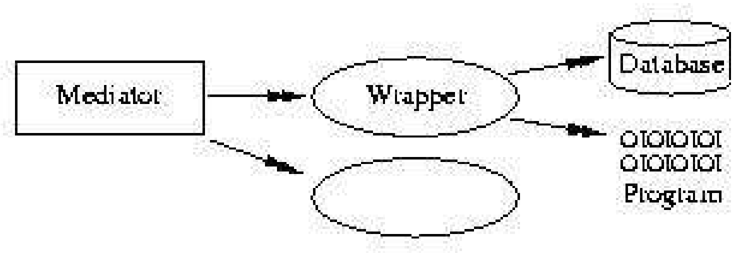


Figure 7: The architecture of EDITtoTrEMBL.

Dispatchers act as mediators, analysers as wrappers

1. Architecture

The environment comprises two kinds of agents, *dispatchers* and *analysers*. *Dispatcher*s act as a combination of mediator and facilitator. *Analysers* function as wrappers around the incorporated heterogeneous data sources to provide a homogenous environment. The analyser's responsibilities are to provide a consistent use of vocabulary and an interpretation of its content to assess its quality.

Figure 8 shows the system's tree structure. A subtree represents a problem domain. The entries are sent to a set of programs and the integration is performed by the respective dispatcher that is responsible for the problem domain. Depending on the workload multiple instances of a specific dispatcher and eventually its tools can be created. This ensures the scalability of the approach.

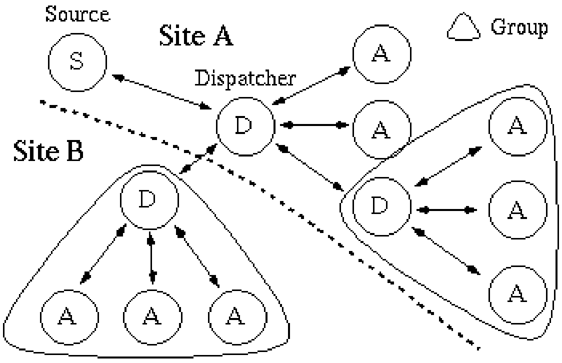


Figure 8: The annotation environment's tree structure.

The figure shows how a well-designed distribution of dispatchers among multiple sites can reduce data flow through the Internet, acting as mediators and collector of annotation.

Dispatchers may find the information provided by an analyser *inconsistent*. In section E it is explained how the dispatchers's capabilities are enhanced by inconsistency management. This means that dispatchers can identify semantic inconsistencies among the annotations provided by the analysers and revise them appropriately.

To reduce complexity a dispatcher assumes that entries sent to it are always consistent and hence only cares about inconsistency introduced by analysers under its control. Again, this ensures scalability.

Sources of Information

Information is either collected from applications that work on SWISS-PROT entries or that request information on the entry from other databases. The databases incorporated for the system's implementation are protein domain databases. The protein domain dispatcher retrieves matches to these external databases. A domain becomes a rule for annotation by induction from the annotation that is associated in the protein database SWISS-PROT. The constraint-induction process is described in section D. Technically, any additional programs whose output can be automatically rephrased in the SWISS-PROT format can be integrated.

Sequential annotation of protein sequence data

From a technical standpoint, dispatchers are analysers with the special ability of workflow management and summarisation. An entry comes with a stack of addresses of analysers to which the annotated result should be returned. This stack is incremented by dispatchers and read by analysers and determines to which analyser the annotated entry should be returned. Figure 8 shows, how analysers and dispatchers are organised in a tree-like structure to resemble problem domains and/or network infrastructure, in order to minimise network traffic and for easier maintenance.

The dispatcher creates a summary of the results of individual agents. This is the moment when the dispatcher may find the provided information *inconsistent* and the techniques of conflict resolution are applied.

Although there is a certain difference between adding information from databases and adding information from sequence analysis programs, since databases are queried while applications are started, the system does not distinguish between the two kinds of sources.

In both cases, one needs to provide wrappers written in Java to support the physical distribution of annotation processes. These wrappers solve three tasks:

- Reformatting of a TrEMBL entry to a valid input for a program or a query. For programs, this is usually easy since most programs either accept TrEMBL entries directly or use FASTA format. For queries, the wrapper extracts certain parts of the TrEMBL entry, which is then sent to the database, like the EC number from the description line (`DE`) for the ENZYME database.
- Setting of parameters: Every wrapper tries to choose the optimal setting of parameters for each individual entry. For most entries additional flags can be determined *a priori*, like for the dependency of SignalP on the entry's `OC` line. The wrapper could decide *a posteriori*, if an optimising function is known, to run a program several times with different parameters and then choose the best result according to this function.
- Output rephrasing: To ensure consistency with the controlled vocabulary of SWISS-PROT, the raw program output is transformed according to a manually curated set of rules and is not accessible from other components in the environment.

In the following, the unit of a wrapper with its associated program or database query is referred to as an analyser. From the outside, an analyser can be regarded as a black box, which is fed with entries, and subsequently returns them with additional annotation.

Interdependencies

Analysers are often highly specific. The correctness of their results depends partially on certain conditions, such as the taxonomic specification or certain keywords. Annotation, which was added by an analyser, is in turn often exploited by other analysers executed at a later stage of the annotation process.

Such analyser interdependencies can be rather simple or complex. An example for the importance of the order in which analysers are called is NNPSL (Reinhardt and Hubbard 1998), which is used to predict the subcellular location of a protein. Before starting

NNPSL, it is necessary to assure that the protein is not a transmembrane protein. Hence, the prediction of transmembrane proteins needs to be started before NNPSL is executed.
The output of an analyser may be used only to decide if the invocation of a second analyser might be worth the computational resources. A computationally cheap analyser performing a hydrophobicity analysis might precede the analyser for the transmembrane prediction.
Analysers may have cyclic interdependencies. For instance, the annotation that can be added following the identification of certain PROSITE patterns depends on the compartment where the protein is located. However, to predict this subcellular location using NNPSL, it first has to be assured that the protein is not transmembrane. This in turn in some cases can be achieved by identifying certain PROSITE patterns. Hence it is reasonable to start first PROSITE, then NNPSL, and then PROSITE again, assuming that a) the protein was found to be not transmembrane and that b) NNPSL could infer the compartment more precisely than known before.

It is clear that such situations are beyond the scope of a Makefile-based approach. EDITtoTrEMBL instead uses high-level descriptions of preconditions for the execution of analysers. These conditions are evaluated by *dispatchers*.

Dispatcher

Dispatchers are programs that co-ordinate the flow of entries between different analysers. Whenever a new analyser is introduced into the framework, it is registered with a dispatcher. The dispatcher stores the name of the analyser together with a description of its preconditions and potential output. These are later used to determine dynamically the execution order of analysers for each entry.

Dispatchers can use other dispatchers to delegate tasks, which means that dispatchers can also act as analysers (see Figure 8). This has several advantages:

- *Better maintenance*: The annotation process is broken into smaller subtasks, which are easier to administer. We envisage that specialised sites will arise that treat certain aspects of the annotation autonomously.
- *Higher efficiency*: In general, a dispatcher will send an entry to each analyser whose preconditions are satisfied. If the dispatcher and the analysers reside on different hosts, this will create significant network traffic. In contrast, if a dedicated dispatcher for a group of analysers residing on one host is installed on the same machine, an entry needs to be send via the network only once for the whole group.

Descriptions

Every analyser is characterised by its preconditions, the lines of an entry it uses for analysis and the potential result of its execution.

Preconditions are defined by two sets. Currently both sets consist of pairs comprised of a line tag and a regular expression. Each pair is interpreted as a condition, which is fulfilled by an entry if the corresponding lines match the regular expression. Regular expressions were chosen since they are simple and, due to the controlled vocabulary of SWISS-PROT and TrEMBL, they are semantically very descriptive, too. It could be any Boolean function on entries as described in section 2, which presents the algorithm.

The meaning of the two sets is the following: all conditions of the first set are *mandatory* preconditions. Their conjunction must be fulfilled by an entry before the analyser can be executed. Note that each condition can individually contain negation or disjunction. The second set describes which data the analyser uses for its work. These are named the *optional* constraints. Those lines of an entry, which match the optional constraints of an analyser, are referred to as *active lines*.

The biological meaning of the conditions can be rather technical, such as stating that a protein must have an enzyme classification number to be treated as an enzyme. It may also carry more semantics, such as expressing that the entry must not describe a transmembrane protein.

The lines possibly added to an entry by an analyser determine its output description. Dispatchers use this to determine the preference of one analyser over another. A description is stored as a string with the syntax and vocabulary of an incomplete entry such that the previously described preconditions can be directly tested on them.

A description could be a list of possibly resulting keywords. As it is in general not possible to specify all potential results, in some cases the output descriptions need to be fine-tuned with respect to existing preconditions. The description of a dispatcher is composed of the description of its registered analysers: input descriptions are logically connected by disjunction, whereas the output description is the union of all output descriptions of analysers.

Workflow Planning

It is not possible to compute the optimal sequence of analysers in advance, since output descriptions only give *potential* results. It is not known in advance, which of these results will emerge from an execution on an arbitrary entry. The workflow planning can prevent sequences to be chosen against better knowledge, which is expressed in the descriptions of the analysers. Upon retrieving an entry for annotation, a dispatcher proceeds as follows:

1. It first determines all *active analysers*. An analyser is active if two conditions are met:
   - Its preconditions must be fulfilled.
   - There must have been a change in one of its active line tags after its last execution. If it was not executed before, the second condition is obsolete.
2. The dispatcher selects one of the active analysers, using a heuristic explained below, and executes it.

3. It repeats this procedure until no more analysers are active or the annotation has reached a certain size.

The intuition behind this algorithm (see Figure 9) is the following: In general, analysers work the better the more precise knowledge they have. At the beginning, only little knowledge can be directly derived from the entry's annotation, and hence analysers will necessarily make vague and conservative decisions. During the annotation process, more and more knowledge is accumulated. If new evidence is found, a subset of analysers may be required to perform again, in order to provide the best possible annotation.

2. Environment

EDITtoTrEMBL is programmed in Java. For inter-process communication and the distribution of workload it uses the language's mechanism for remote method invocation (RMI).

The system follows a multi-level client-server architecture. Clients request annotation by sending an entry, or a set of entries, to a root dispatcher. The entire procedure of analyser selection and process distribution is completely transparent for the client. A dispatcher uses its registered analysers for annotation, but can also create further instances if necessary by starting remote shells. Asynchronous communication was implemented by letting all participating analysers serve as both servers (awaiting entries for annotation or their return from analysers) and clients (requesting the annotation or returning an entry) at the same time.

Analysers, respectively their wrappers, are in general started and stopped independently from the dispatcher. Upon start-up, analysers first register with a dispatcher and then persist in memory to wait for entries to annotate. This does not imply that the actual analysing program itself also stays in memory; this depends on the implementation of the program, which cannot be influenced by the framework, and is independent of the wrapper.

We currently have implemented two applications. One for the annotation of the entire TrEMBL database, and a second provides a graphical user interface for the annotation of single entries.

Distribution

The distribution of the annotation process could also be achieved with standard Makefiles using a queuing system. EDITtoTrEMBL conceptually expands the scope of distribution to every remote host connected via the Internet. This is possible due to the usage of Java's RMI mechanism.

Load balancing is performed inside every dispatcher. Naturally, it applies only if several instances of one analyser are available. An entry is then sent to that instance with the least number of entries in procession. The dispatcher might also decide to recruit more analysers on demand, which has not yet been annotated.

The Annotation Process

In this section, the annotation process is explained in detail. We first introduce a formal description and then explain the algorithm. Finally, potential pitfalls in the annotation process are eluded and ways are explained how they can be avoided.

Let A be the set of analysers. Let e be an entry, E the set of all possible entries. Let $F_a\ (e) : E \rightarrow E$ be the function that transforms an entry into another entry by applying analyser a.

We call ($a_1,\ldots,a_k$) with $a_i \in A$, $1 \leq i \leq k$ a path; it describes an ordered series of analyser executions. The annotation achieved by this path is $F_{a_1,\ldots,a_k} := F_{a_k} \circ F_{a_{k-1}} \circ \ldots \circ F_{a_1}$. Note that analysers can appear more than once in a path.

For every analyser $a \in A$ the pre-conditions are defined as a function $C_a : E \rightarrow \{true, false\}$. C_a is evaluated on an entry e by testing all mandatory constraints and returns *true* if they all evaluate to true. $C^+_a : E \rightarrow \{true, false\}$ is a second function, built from the optional constraints. C^+_a returns true if *any* of the optional constraints evaluates to true. O_a is the description of the potential output of *a*. An analyser *a* is completely characterised by the triple $\{C_a, C^{+}_{a}\ ,\ \mathrm{O_a}\}$.

We also introduce a subset relation on entries. $e_i \le e_j$; e_i, $e_j \in E$ means that $\mathrm{e_i}$ has a subset of the annotation of e_j. With $e^{C^{+}_{a}}$ the entry *e* is denoted, that is restricted to those lines contained in C^{+}_{a} .

It is not allowed by any analyser to remove any line of annotation. Hence, analysers are *monotonous* in the sense that $e \le F_a(e)\ \forall\ a \in A,\ e \in E.$

Algorithm

```
FUN annotate (e:E, T: $\mathbf{N}$) : E
VAR
  candidates : $\wp$(A) = $\emptyset$;
  history: ARRAY A .. E;
  count : ARRAY A .. $\mathbf{N}$;
  e : E;
BEGIN
  FOREACH  $a \in A$ DO
    history[a] := "";
    count[a] := 0;
  OD
  DO
    candidates := CALL active(history,e);
    a := CALL choose(candidates);
    e := CALL F_a(e);
    IF NOT CALL addhistory(history[a],e)
      THEN STOP;
    FI
    count[a]++;
  OD(#candidates = 0 OR count[a] = T)
  RETURN CALL summarize(e);
END
WHERE
FUN active (history: ARRAY A to E, e:E) $\wp$(A)
VAR
  diff : keyword{string} := "";
  analysers : powerset(A) := emptyset;
BEGIN
  FOREACH $a \in A$ DO
    IF CALL $C_a$(e)
    THEN
      IF history[a] = "" THEN
         analysers += a;
      ELSE
        diff:=e-history[a];
        IF CALL changed(a,diff) THEN
          analysers += a;
        FI
      FI
    FI
  OD
  RETURN analysers;
END
```

Figure 9: Algorithm for annotation.

It describes the selection of analysers that depends on the annotation previously performed for an entry. The functions summarise, choose, changed and addhistory are explained in the text.

The algorithm for incremental annotation of a single entry proceeds as follows (Figure 9). If a client wants an entry *e* to be annotated it will call the function *annotate* of the root dispatcher *D*. *A* is the set of analysers known to *D*. *D* then basically performs a loop. In each iteration first all active analysers are computed (function *active*. Of those, one is chosen using a heuristic (function *choose*, see below). The entry is annotated by this analyser and the loop starts again. If no more annotation can be added, the entry undergoes a final post-processing to reduce redundancy and increase consistency (*summarize*, see below).

An analyser *a* needs to fulfil two conditions to be considered as active for an entry *e*. First, all its pre-conditions must be true (function $C_a(e)$). Second, it is checked if the execution of *a* will produce any additional results. It is reasonable to presume that all analysers are idempotent, which means that always $F_a \circ F_a = F_a$. Hence, *a* should not be executed if nothing has changed in an entry it was applied to before. Using C^{+}_{a} , one can formulate more stringent conditions, demanding that something has changed in *e* at a positions that affects the analysis of *a*, that is there must have been a change in $e^{C^{+}_{a}}$.

To evaluate the second condition, *D* keeps a history list for every analyser, storing the result of its last application on *e*. *active* uses this to compute the difference between the current entry and the entry stored in the history. It then checks if the difference matches with one of the conditions of C^{+}_{a} (function *changed*). The history is furthermore used to check the consistency of each analyser inside the function *addhistory* (see below).

For sets of entries, such as a complete database, the algorithm is started once for every entry. Note that an execution of an analyser can actually mean that the entry is passed to another dispatcher, which takes the role of an analyser.

Properties of the Algorithm

Termination:

Using the monotonicity stipulated in the previous section, one can follow: let $e_n := F_a(e_{n-1})$, $n \geq 1$ for some $a \in A$, e_0 being the unannotated entry. Then $e_j \leq e_k$, $\forall j \leq k$. Hence, the total amount of annotation is never getting smaller, but either stays unchanged or grows with every iteration of the main loop in *annotate*.

In general, it is observed that analysers have only a limited 'knowledge capacity'. Multiple revisions of annotation created earlier occur rarely. After a few rounds the list of active analysers is empty and the algorithm terminates. However, one can imagine pathological cases in which for instance two analysers mutually add data into the others C^+-lines, provoking an infinite loop. This should not happen since the analysers would not be biologically sound then, which was postulated in the introduction of the environment. The limit on the number of times an analyser may be executed (variable *T*) serves as an indicator for a failure only.

In the following it is assumed that the algorithm stops naturally when all analysers have been fully exploited, the annotation process then reaches a fix point.

Negative Preconditions:

Without allowing negative constraints, the order in which analysers are executed would not matter. In this case, all possible paths would result in the same annotation - the annotation process has a unique fix point.

However, with negative constraints this does not hold any more. Following a different path will lead to different annotations. Consider the following example (Figure 10):

```
a₁:={{KW: ¬'DNA-BINDING'}, {}, {CC: 'SUBCELLULAR LOCATION'}},
a₂:={{CC: ¬'SUBCELLULAR LOCATION'}, {},{KW: 'DNA-BINDING'}}.
```

Figure 10: Example for descriptions of two analysers a_1 and a_2

Calling a_1 first might inhibit the execution of a_2 and vice versa. If the analysers are sound then one can think of a pathological case where two such mutually inhibiting analyser would yield two different annotations. To resolve the problem, one needs either to add an additional analyser for further information or to improve the analysers's descriptions. Note that this problem is closely related to the problem of negation in deductive databases, see for instance (Ullman 1988).

Heuristics:

In the last section, it was argued that a reliable decision could not be made, in order to determine which of the possible paths results in the best annotation. Therefore a *heuristic* (function *choose*) is used to select one of the active analysers at each iteration of *annotate*. Several strategies are possible:

- Prefer analysers with tight constraints since they are highly specific and use more knowledge.
- Prefer analysers with loose constraints since those are more general and should be executable with minimal influence on other analysers.
- Try to prevent as long as possible changes to lines that could block analysers having negative constraints.
- To prevent potentially wrong annotation call those analysers first that make changes to lines on which other analysers have negative constraints.
- Use additional priority information specified by the administrator of the system.

It is implemented as a small *planner* in the computer language PROLOG that gets static information on mutual dependencies among analysers. This is prepared at runtime by the Java code according to the analysers known to the system. Figure 11 and Figure 12 show the relevant lines to implement a planner for the sequence of visits to analysers for every entry.

```
dependency(Candidate,Active) ←
    dependsOn(Candidate,X),
    member(X,Active).
```

Figure 11: PROLOG code to determine a dependency between two analysers

The predicate dependency holds if the analyser Candidate *has a static dependency on the active analyser* X, *where* X *is currently allowed to execute according to its preconditions and whether it has been previously executed. The calling predicate (on the Java side) provides the list of active analysers. The predicate* dependsOn *is supplied once to the PROLOG engine since it is static information while analysers do not change.*

```
planner(Active,Next) ←
    member(Next,Active),
    \+ dependency(Next,Active).
```

Figure 12: Implementation of a planner in PROLOG

This is the predicate actually queried from the Java side. planner(+Active,-Next) returns the analyser Next as a member of the Active analysers that could be started next without violating any constraints.

In its current implementation a subset of active analysers is not executed while they mutually or circular depend on each other. This is not a bad solution while there is at least one analyser that can be executed. Its additional annotation will give a basis on how to break the deadlock among the interdependent analysers if the system is well designed. For those cases where this cannot be done, all possible paths through the analysers would have to be performed and subsequently analysed on identity and eventually a conflict resolution performed on these scenarios. For the current analysers this has been circumvented by a careful selection of analysers and definition of dependencies, i.e. the system's design. But for a later stage of the annotation environment this has to be addressed. Figure 13 shows the Java code to call the planner.

```
Javalog.PlEngine whosnext_prolog = null;   // prolog engine
AnalyserInterface decide_who_is_next(AnnotationBlock ab)
   throws AnnotationException, java.rmi.RemoteException
{
  if (null == whosnext_prolog) {
    whosnext_prolog = new JavaLog.PlEngine();
            [ ... consult code in Figure 11 and Figure 12 ... ]
    getStaticDependencies(whosnext_prolog);
  }
  StringQueue active = ab.whichOfTheseAreActive(Analysers);
  AnalyserInterface next = null;
  if (whosnext_prolog.call(sb.toString())){
    PlClause clause = whosnext_prolog.goal();
    String s = clause.stateOf("Next").toString();
    if (null != s) {
      String s3 = SWISS.util.String.deleteChars(s,'\'');
      next = Analysers.getAnalyser(s3);
    }
  }
  return next;
}
```

Figure 13: Java code to call the prolog-implemented planner

For the sake of clarity the code was shortened from debug messages and error checking. The code is executed within dispatchers where the variable Analysers *is a container for all the analysers known to the dispatcher that might accept an entry for annotation. The entry to be annotated is part of the* AnnotationBlock *that contains the full annotation history of an entry until the integration of the knowledge with an eventual conflict resolution is performed. The addition of facts that represent the mutual static dependencies of analysers is implemented in the function* getStaticDependencies*. This is implemented as an all against all of analysers where the output description of one analyser is successfully matched by another analyser's input description for a mutual dependency to be found. The code is not shown. The analyser to be perfoming next is returned with the variable* next*. A return of* null *is interpreted as a request to send the annotation block back to the calling superior dispatcher.*

Summarisation:

A final processing of an entry has two general aims:

1. It tries to reduce redundancy.
2. It spots possible inconsistencies.

Regarding consistency, each analyser is required to be *consistent with his own, previously drawn decisions*. To test this, it needs to be ensured that the new annotation of an analyser does not contradict its previous (function *addhistory*). If this happens, the process of annotation for this entry should not progress any further. In such cases, the specific analyser and its description need to be carefully checked by the administrator. Note that this only assures the consistency of a single analyser with itself.

Regarding redundancy, a similar approach was chosen. All the annotation is temporarily stored in a hidden section of the TrEMBL entry. As the last step of the algorithm, all annotation made by an analyser is deleted except for it's last, which is then shifted into the visible section of the entry. The above described consistency check assures that no information is lost through this clean up.

The removal of redundancy and the check on consistency is currently mainly possible on the formal parts of TrEMBL entries only - such as the FT and KW lines. We therefore try to find means to formalise as much of an entry as possible to allow the application of manually created rules. This set of rules is necessarily incomplete and evolves with the errors or redundancy that are found.

Many algorithms evaluate the confidence, e.g. most predictors of membrane spanning regions. Also if this does not always reflects a probability, this could be estimated by a correlation of the confidence value given and the success rate on a test set, e.g. on SWISS-PROT.

The analysers use such values very conservatively. This is explained in more detail in (Fleischmann, Möller et al. 1999).

3. Multi-level conflict/redundancy treatment

The resolution of conflicts/redundancy is performed on multiple levels. Every single tool only returns the information that it is certain to be correct, which is dependent on the information presented as input. On the lowest level, this may eventually be no more than the sequence and eventually the organism.

Dispatchers that retrieve results from multiple tools can summarise the results to groups of different scenarios, thereby reducing the redundancy. Also, on this or an even higher level, information from different kind of information sources is combined and a decision for a specific scenario made.

4. Parallelism

The increasing rate of new sequences submitted to the databases, imposes difficulties to cope with the flood of data. The computational resources need to grow proportionally. This becomes even more severe due to the steadily increasing number of applications that are available for the process of sequence annotation, created by bioinformatics research groups all over the world.

To ensure scalability of the approach it is essential that computational resources can be added incrementally to the annotation system. With close to 700,000 entries in SWISS-PROT and TrEMBL that can all be treated independently it is most promising to distribute the load within a local area network or even all over the Internet. EDITtoTrEMBL is capable of distributing entries throughout the Internet. Multiple dispatchers can be configured to share a common set of analysers. This may be desirable if the maintenance of a certain analyser is difficult or if only one site has access to dedicated hardware involved in the annotation.

The technical possibility exists to increase parallelisation such that the same entry may be annotated by different analysers at the same time and the resulting annotation gets merged subsequently. While this is technically of interest, the overhead of pathway *planning* to co-ordinate the parallel coordination of clones of *entries* would yield only minimal gains. The reason for this is that so many entries would have to be annotated and the time for a single entry to be annotated is negligible (between 0.1 and 10min).

5. Comparison of EDITtoTrEMBL with other initiatives

When EDITtoTrEMBL was first presented in 1998, it was a new approach towards the automatic annotation of protein sequences. EDITtoTrEMBL is a step forward to fulfil the requirements for sequence annotation frameworks postulated in the introduction. By using

Java wrappers, language and platform independence are achieved. In addition, Java RMI is available for a comfortable implementation of process communication. Dispatchers automatically distribute processes using a simple load-balancing algorithm.

One of the main differences to other approaches to automatic annotation, such as PEDANT (Frishman and Mewes 1997) or GeneQuiz (Casari, Ouzounis et al. 1996) is the dynamic, data-driven workflow planning. It allows us to treat every single entry in a way that is tailored to its data. This saves CPU time since only well-suited programs are run, and it improves the quality of the annotation. While traditional approaches tend to use only a few means of annotation on all entries, in this approach a preference is given to integrate many different programs. These are applied only on those entries, for which they are best suited and manually proven to be correct. The difference in the presentation of the system to the user and EDITtoTrEMBL's capacity of explicit negation aside and while being understood as less dynamic, MAGPIE seems is very similar EDITtoTrEMBL.

PSORT (Nakai and Kanehisa 1992) integrates a collection of algorithms that all use the same formal description such that no rephrasing is necessary. EDITtoTrEMBL has its focus on the integration of programs that previously stood alone and therefore needs the workflow planning as an additional flexibility to control the data flow.

It is seen as advantageous to separate the sequence annotation into distinct *modules*. The most important is a stable framework module, responsible for communication, process planning and intelligent combination of results. For instance, the GAIA project (L. Charles Bailey, Fisher et al. 1998) handles communication through a relational database serving as a black board. Analysers, called *sensors* in their environment, write results into that database. Later, other modules read from the database, trying to integrate results. GAIA does not include any planning facilities but simply gives each entry once to each sensor.

The annotation environment GeneWeaver (Bryson, Luck et al. 2000) strongly advocates that it represents a Multi-Agent System, an issue addressed in the following paragraph.

D. EDITtoTrEMBL as a Multi-Agent-System

For computer scientists a system like EDITtoTrEMBL is understood as a Multiple-Agent-System (MAS). This section is used to explain the more abstract character of this system and introduce terms that will be referred to in the explanation of the conflict resolution technique.

Information agents are computational software systems that have access to multiple, heterogeneous and geographically distributed information sources (Wiederhold and Genesereth 1996; Wiederhold, Genesereth et al. 1997). One of their main tasks is to perform active searches for relevant information in non-local domains on behalf of their users or other agents. Information from multiple autonomous sources is retrieved, analysed, manipulated and integrated to finally provide a high-level access to information that is otherwise not efficiently usable.

A common architecture for information agents consists of information providers (programs for the annotation of sequences), wrappers (analysers), facilitators (dispatchers), and mediators (integrated in both dispatchers and analysers) (Flores-Mendez 1999). A wrapper is associated with each information provider to prepare retrieved data for the mediator. The mediator is the point of contact for a user (human or agent); it uses the facilitator to get in touch with the wrappers and knows what kind of information the wrappers can provide. Given a user query it will then contact the wrappers, integrate the results, and return them to the user.

Agents are accepted as programs that have, and in a MAS share, knowledge, belief and intention. These terms should be explained in the context of EDITtoTrEMBL

1. Knowledge

Knowledge is represented as a set of facts. A seed of such are initially understood as axioms by the agent. In addition, Java code that is responsible for the mediation of the wrapped program's output should be understood as knowledge, although this does not have other properties of knowledge, e.g. that this can be distributed between agents.

2. Belief

Any information derived from knowledge that involved agent-external input represents belief. This belief can be temporary, just as the analysers work on a specific protein sequence and forget about this sequence when it is returned.

A dispatcher may be interested in the performance of analysers in the system and use the time measurement to decide where to send entries to, in the belief that this would be rational behaviour.

3. Intention

The efficient use of resources is an intention of analysers in EDITtoTrEMBL. The intention, i.e. a complete and efficient annotation of protein sequences, is the driving force of all analysers.

4. Why a Multi-Agent-System (MAS) was not used as a basis to implement EDITtoTrEMBL

Many reasons contributed to the decision not to use a MAS for the annotation of TrEMBL. The most important one is that no system was available at that time that seemed free, fast and flexible. Evaluated was the system ideas (Klusch 1998) and considered far too slow. Today, more agent systems are available and the recent comparison (Cogan, Gomoluch et al. 2001) with the system Voyager (ObjectSpace 2001) demonstrated, that agent systems may outperform this implementation. In late 1999 the MAS "Mozart" (Roy and Haridi 1999) introduced an integration of constraint technology with distributed agents. Still, their mechanism does not allow the revision of facts as presented in the following section E. Very recently new agent environments have been created like DECAF (Graham and Decker 2000) which has been applied to bioinformatics (Decker, Khan et al. 2001; Decker, Zheng et al. 2001). Another initiative is BioAgent (http://www.bioagent.net). One may argue that the implementation of these agent systems is not too distinct from the functionality of federated databases as implemented by IBM (http://www7b.software.ibm.com/dmdd/library/techarticle/0203haas/0203haas.html). However, the emphasis is not on the integration of remote databases but on the semantics of the data statically (database) or dynamically (application) available.

Source code of any environment selected had to be available since with such a high number of entries passing through the system, one should expect hidden problems to surface, be it due to memory handling or because of mere inefficiencies. When the system was developed in 1997, within the programming language Java, the basic skeleton was straightforward to implement and from there the system evolved.

Today the situation is much different. MAS research is a hot topic in AI research and many free and efficient systems are now available. Still, EDITtoTrEMBL has features that are not found as such in other environments, especially the integration of a conflict resolution scheme that was adapted for EDITtoTrEMBL with Michael Schroeder from the City University London.

E. Technique of conflict resolution

1. Introduction

Information agents integrate multiple distributed heterogeneous information sources. The challenging, yet unsolved, problem remains to ensure the semantic consistency of the integrated data. The task was to develop a general approach towards inconsistency management for information agents. It is implemented as part of the EDITtoTrEMBL system and applied on a real-world problem in the domain of bioinformatics, the annotation of transmembrane proteins described in chapter V.

The incorporation of conflict-resolution in EDITtoTrEMBL was a direct consequence of the idea to find errors in the annotation. Here it is presented how during the process of integration potential inconsistencies can both be revealed and removed. These techniques are implemented in EDITtoTrEMBL, for which the integration of data while preserving consistency is a special challenge due to the inherent uncertainty and incompleteness of provided data.

The SWISS-PROT syntax checker, developed by Elisabeth Gasteiger, checks more than only the syntax of an annotation, it also finds semantic problems. This is very helpful for both database curators and programmers. However, it was not extendable towards a revision of statements in SWISS-PROT to allow an error correction for TrEMBL.

To allow a revision it is first necessary to have an abstract notion of an *annotation element*, which is a statement in the syntax (or its precursor during the automated annotation process), that could be made and eventually reversed.

The initial work on conflict resolution tried to find errors in SWISS-PROT and TrEMBL not as a direct consequence of what was stated, but by adding rules for inference, especially for the already atomic SWISS-PROT keywords and FT lines. However, this was problematic. The main difficulties were that

- Too many exceptions made it difficult to come up with non-trivial rules that hold for the whole of SWISS-PROT.
- SWISS-PROT is not formal such that the more verbose comment lines can not be understood by the computer.
- The automated annotation in TrEMBL was mainly done by rules adding annotation in dependence of SWISS-PROT entries - which again is not formal.

A formal representation of biological knowledge is not existing. The GeneOntology (GO) (Ashburner, Ball et al. 2000; Ashburner 2001) consortium maintains and develops a representation of biological entities as a directed acyclic graph, where edges show “is-a” or “part-of” relations. At this stage, it is very much oriented towards an augmentation of model-organism databases, where this effort has its origin. While not completely formal, it is of great use for the database creators.

With more and more annotation being available in a formal description, i.e. the augmentation of SWISS-PROT annotation with GO terms is currently prepared, the verification of SWISS-PROT entries on a semantic level becomes more and more feasible.

2. Representing Knowledge and Uncertain Beliefs

A consensus from different databases and prediction method for the protein sequence annotation can be achieved in different ways. One might look for the annotation supported by the majority of tools (Cuff, Clamp et al. 1998). However, with extra knowledge from other databases the majority may be proven wrong. Integration can not be achieved without an interpretation of the data on a *semantic level.*

In this section *extended logic programming* is introduced as a formalism to represent the potentially inconsistent biological domain knowledge. This enables the revision of the minimal and least reliable information that contributed to a conflict.

Extended Logic Programming

Well-founded semantics with explicit negation, short WFSX, provides a semantics for extended logic programs, i.e. logic programs, which are extended by a second kind of negation. This powerful language is appropriate for a spate of knowledge representation and reasoning forms (Alferes and Pereira 1996). Formally, an extended logic program is defined as follows:

Definition: An *extended logic program* is a (possibly infinite) set of rules of the form $L_0 \leftarrow L_1,...,L_m, not\ L_{m+1},...,not\ L_n$ where each (L_i) is an objective literal ($0 \leq i \leq n$). An objective literal is either an atom A, or it is its explicit negation $\neg A$. Literals of the form *not L* are called *default literals*. Literals are either objective or default ones.

Example: Consider two predicates *domain* and *ft* for a feature table entry of the databases SWISS-PROT or TrEMBL. The *ft* predicate contains the start and end position of a given region such as *transmembrane*. The derived *domain* predicate states that all positions between these two boundaries are transmembrane. This relation can be captured by the rule below:

```
domain(Agent,Pos,transmem) ←
                    ft(Agent,transmem,Pos1,Pos2),Pos1≤Pos,Pos≤Pos2
```

Besides facts and rules, one can specify integrity constraints.

Definition: An *integrity constraint* has the form $\bot \leftarrow L_1, \ldots, L_m,\ not\ L_{m+1},\ \ldots,\ not\ L_n$ with $0 \leq m \leq n$ where each L_i with $0 \leq i \leq n$ is an objective literal, and $\bot$ stands for false. Syntactically, the only difference between the program rules and the integrity constraints is the head. A rule's head is an objective literal, whereas the constraint's head is $\bot$, the symbol for false. Semantically the difference is that program rules open the solution space, whereas constraints limit it.

The constraint below states that transmembrane regions have to be longer than 16:

```
 ⊥ ← ft(Agent,Acc,transmem,Pos1,Pos2), X is Pos2-Pos1, X ≤ 15.
```

When defining integrity constraints the first objective is to detect violations, the next step is to remove the violations. Since by definition it is not possible to change a fact, the *revisables* were introduced. Revisables are assumptions that may be changed when inconsistencies arise.

Definition: The revisables *R* of a program *P* are a subset of the (possibly default negated) literals, which do not occur as rule heads in *P*.

Example: Predictions of transmembrane regions are formalised in the feature table. For this application, these must not be taken for granted and hence they are defined as revisables rather than facts. By default, entries are set to true, but should inconsistencies arise, the may be withdrawn, i.e. set to false by the following statement:

```
  revisable(ft(tmhmm,p12345,transmem,6,26), true).
```

Similarly, it is possible to revise assumptions from false to true.

For many cases it is useful to specify how easily a revisable can be changed or, in other words, how reliable an assumption is.

Example: The probabilities below state that TMHMM's assumption about first transmembrane region is not reliable (0.5), while is its assumption about the second region is (0.1), stating a revision is expected with 10% probability.

```
probability(ft(tmhmm,p12345,transmem,6,26),  0.5).
probability(ft(tmhmm,p12345,transmem,27,50), 0.1).
```

Probabilities can also be used to rate competing *ft* entries that are generated by a single analyser. This is often useful for a wrapper, which uses a neural network, to represent the most active neurone with a higher probability than the second most active one.

To summarise, domain knowledge is modelled by facts, rules, and integrity constraints and beliefs of the agents by *revisables*. The certainty of the beliefs may be qualified by a probability to indicate the degree of reliability.

Graphical visualisation

To support the understanding of the prior described revision process it may be of help to visualise this process graphically. Figure 14 has its roots in a classic application of REVISE (Damasio, Pereira et al. 1997), which is the location of errors in larger electronic networks.

The figure shows lines of communication. A conflict in the model represents a communication breakdown between any two ends of the communication network, which are represented as lines in Figure 14. Errors may occur in both lines and nodes. When multiple faults are detected at the same time, then it should be assumed that these faults have a common cause. This information should be used to locate the error of the system rather than sequentially exchanging all combinations of components until the system is back up.

For any failure of a combination of lines and nodes, a set of end-to-end connections will be dysfunctional. Vice-versa for every disturbance a different possibility exists for a specific combination of failure sites. Assuming independence of failure occurrences, a probability of a certain failure scenario can be calculated by a mere multiplication. REVISE finds the most plausible scenario.

In Figure 14, the two circles that are coloured red are sufficient to inhibit communication from any end of the system that is labelled A to any end labelled B.

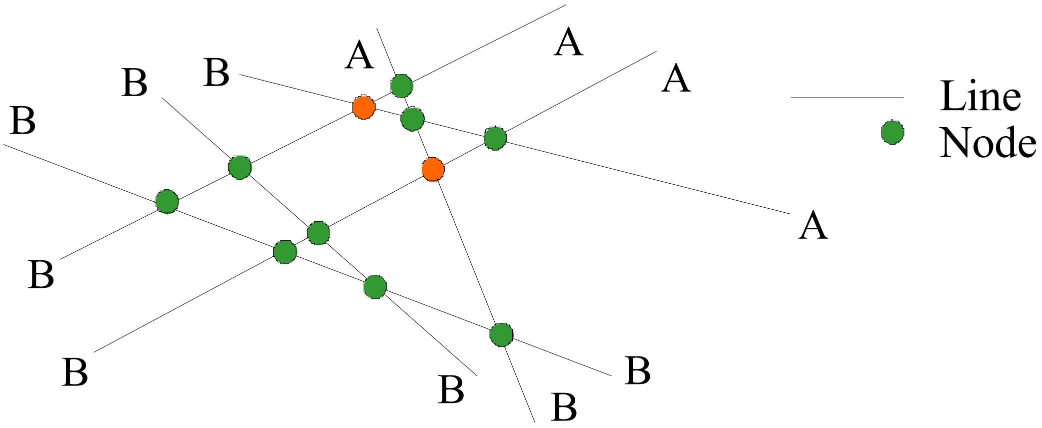


Figure 14: Explaining conflict resolution in a communications network

The two red-colored nodes are sufficient to fail communication between the sets A and B of line ends.

With this picture in mind, the understanding of the formal presentation that is presented in the following section should become easier. This has not been developed within this thesis and is presented here to give a complete picture.

Revising Inconsistent Domain Knowledge and Agent Beliefs

Our objective is to detect violations of the integrity constraints and to revise the assumptions involved as little as possible to repair them. Formally, such as revision is defined as follows:

Definition: Let P be a program and R a set of revisables. The set $R' \subseteq \{L \mid not\ L \in R\} \cup \{\neg L \mid L \in R\}$ is called a *revision* if it is a minimal set such that $P \cup R'$ is free of contradiction, i.e.

$P \cup R'\neg \vdash_{WFSX} \bot$. For details on the definition of the inference operator $\vdash_{WFSX}$ see e.g. (Alferes and Pereira 1996).

Before the process leading to the revisions is eluded in more detail, some definitions are needed.

Definition: Conflicts are sets of revisables that lead to a contradiction.

Definition: Let *P* be an extended logic program with revisables *R*. Then $R_\bot \subseteq R$ is a conflict iff $P \cup R_\bot \vdash \bot$.

To compute revisions, revisables need to be changed such that all conflicts are covered. Such a cover is called a *hitting set*, since all conflicts involved are hit.

Definition: A *hitting set* for a collection of sets *C* is a set $H \subseteq \cup_{S \in C} S$ such that $H \cap S \neq \emptyset$ for each $S \in C$. A hitting set is *minimal* iff no proper subset of it is a hitting set for *C*.

Theorem: Let *P* be a program. Then *R* is a revision of *P* iff *R* is a minimal hitting set for the collection of conflicts for *P*.

In (Raymond Reiter 1987) it is stated that revisions can be computed from conflicts and hitting sets which can be obtained from hitting set trees:

Definition: Let *C* be a collection of sets. An HS-tree for *C,* call it *T*, is a smallest edge-labelled and node-labelled tree with the following properties:

The root is labelled √ if *C* is empty. Otherwise the root is labelled by an arbitrary set of *C*

For each node *n* of *T*, let *H(n)* be the set of edge labels on the path in *T* from the root node to *n*. The label for *n* is any set $\Sigma \in C$ such that $\Sigma \cap H(n)=\emptyset$, if such a set Σ exists. Otherwise, the label for *n* is √.

If *n* is labelled by the set Σ , then for each $\sigma \in \Sigma$, *n* has a successor n_σ joined to *n* by an edge labelled by σ.

The remainder of this section informally explains the algorithm that was proposed in (Raymond Reiter 1987) and corrected in (Greiner, Smith et al. 1989) with its adaptation to extended logic programs.

To compute conflicts, the REVISE engine uses SLXA, a proof-procedure, which returns the revisables involved in the proof. It is based on the SLX proof procedure for WFSX (Alferes and Pereira 1996).

The calls to SLXA are driven by the REVISE engine. Its main data structure is the hitting-set tree. The construction of the hitting-set tree is started on candidate ∅, meaning that the revisables initially have their default value.

The node ∅ is expanded when the SLXA procedure is called to determine one conflict. If there is none, then the program is non-contradictory and the revision process is finished. Otherwise, the REVISE engine computes all the minimal ways of satisfying the conflicted integrity constraint returned by SLXA, i.e. the sets of revisables which have to be added to the program in order to remove that particular conflict.

For each of these sets of revisables, a child node of ∅ is created. If there is no way to satisfy the conflicted integrity then the program is contradictory. Otherwise the Revise engine selects a node to expand according to some preference criterion and cycles: it determines a new conflict, it expands that node with the revisables which remove the conflict. This continues until there is no further conflict remaining and hence a solution is found.

The solution is kept in a table in order to prune the revision tree by removing those nodes that contain some solution and have been selected according to the preference criterion. The order in which the nodes of the revision tree are expanded is important to obtain minimal solutions first. Minimality should here be understood as “minimal imposed change” and therefore “most plausible”. In the current implementation, it is catered for minimality by set-inclusion, cardinality and probability (Damasio, Pereira et al. 1997).

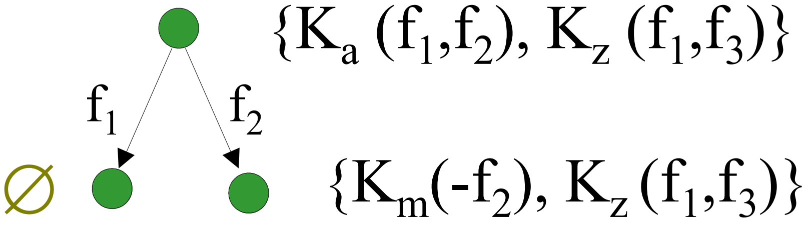


Figure 15: Graphical visualisation of a small revision tree

The figure shows nodes of the revision tree together with a potential labelling with conflicts. Starting from the root node with the conflicts K_a and K_z, a revision of the fact f_1 is successful and leaves no remaining conflicts. The alternative to revise f_2 and later f_3 seems not appropriate since (unforeseen) a conflict K_m was in need of f_2 to be false.

3. Conclusion and Future Work

With this work, it was demonstrated that the integration of heterogeneous data sources can have a symbiotic effect on the overall quality of the information provided. For the automated annotation of protein sequences, this is vital and similar approaches will be implemented for other domains in the future.

It was demonstrated how extended logic programming and program revision can be used to represent domain knowledge and agent beliefs in distributed information agent systems. In particular, it was demonstrated how to deal with different degrees of reliability and how to remove inconsistencies by using various options to define minimality.

While a revision is ideal for binary statements, it is not practical to use REVISE to allow a fact's refinement. This refers to the adaptation of a fact, instead of its removal. The possibility to allow refinements may have been beneficial for our application for which a domain's boundaries could have been changed to fulfil a constraint. This will be addressed in future work.

It may be of value to note that although the process of a revision leads to a centralisation of processing this does not represent a bottleneck. Any agent in the system can be individually cloned and thereby duplicate the bandwidth.

F. Implementation of EDITtoTrEMBL

1. Layers

The concept describes three different layers. On the lowest level there is a program run to predict a protein's feature. This is translated to a common vocabulary. This translation is verified and the next program for annotation chosen.

- *Prediction*

  Any program accepting sequences for annotation can be chosen. Its output is translated by a wrapper written in Java.

- *Semantics*

  Checks for consistency and redundancy are implemented in PROLOG.

- *Control*

  The determination of the workflow is implemented in Java. It is possible to include semantic queries in the determination of the workflow. The biological constraints imposed on the annotation environment are used in all stages.

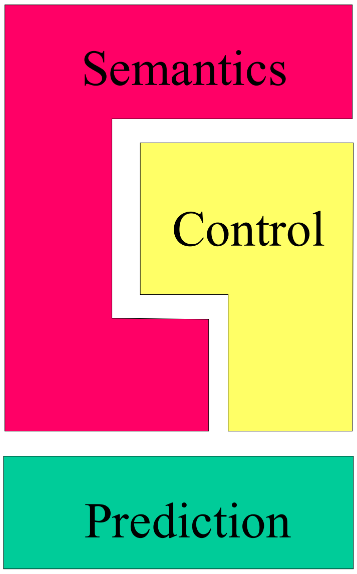


Figure 16: Conceptional layers of the annotation environment

Semantics represent background knowledge, e.g. the GeneOntology, but is also used to decide on the correctness of predictions and to impose constraints on the workflow.

2. Object Model

Java implementation of SWISS-KNIFE

SWISS-KNIFE is an API for the interaction with SWISS-PROT entries, based on the programming language Perl (Hermjakob, Fleischmann et al. 1999). Once the decision was made to use the language Java, in favour of the facilitated communication between objects, the functionality of SWISS-KNIFE had to be re-implemented in this language. This is a prerequisite for the automated annotation. The functionality includes reading, writing and modifying SWISS-PROT and TrEMBL entries, from both the flat file representation and the relational database. For this purpose, a set of classes was created to represent and connect entities within SWISS-PROT and TrEMBL. These also facilitate the translation of information contained in SWISS-PROT/TrEMBL entries to a format accessible to PROLOG. A central class diagram of the Java SWISS-KNIFE is shown in Figure 17.

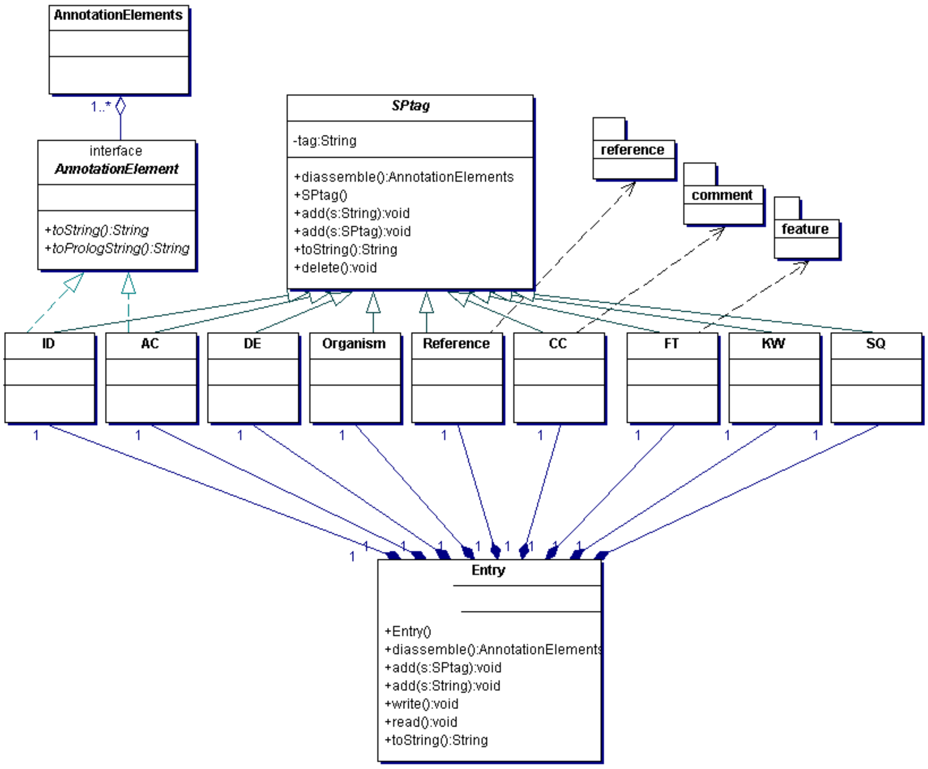


Figure 17: UML class diagram for the Java SWISS-KNIFE classes used within EDITtoTrEMBL

The entry class represents a single SWISS-PROT/TrEMBL entry and contains a collection of SWISS-PROT tags (SPtags). All these parts know how to disassemble themselves into annotation elements. These represent individual chunks of information that are considered to be treated independently by the conflict resolution engine. This and the following UML diagrams were created with the program TogetherJ (Togethersoft 2001).

Equivalent to Figure 7 and Figure 37, Figure 18 visualises the interdependencies of classes for the implementation of the agent system. The algorithm of Figure 9 is shown as an interaction diagram in Figure 19.

The environment in which the different tools for annotations are embedded is not supposed to run on a single computer only. Hence, to ensure consistency it was required that the same code is executed on all platforms.

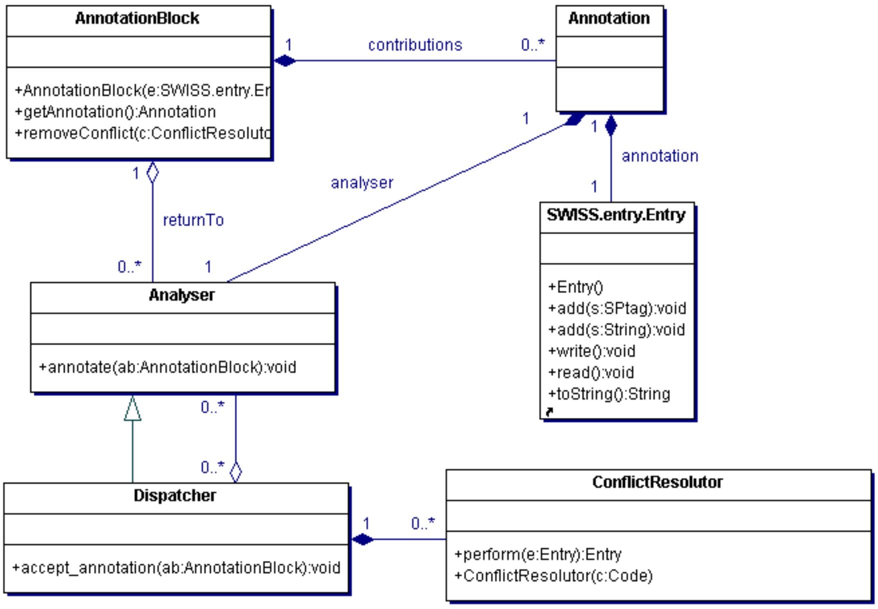


Figure 18: UML class diagram of EDITtoTrEMBL

The annotation process is controlled by the dispatcher. Semantics are controlled within the conflict resolution (see section E of this chapter), the annotation process of a single entry reflected in the annotation block and the actual annotation performed within the Analyser.

Platform Independence

Java provides platform independence. This means that code that was once compiled on one platform runs on any platform equipped with a Java virtual machine interpreting the byte-code.

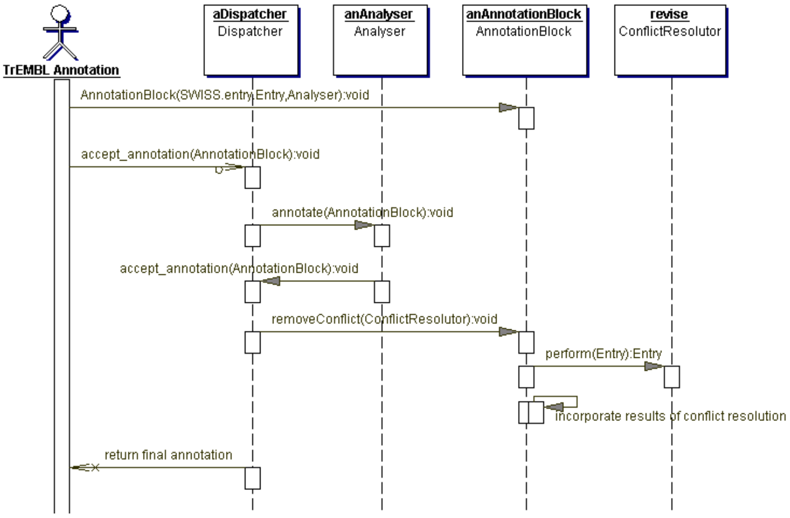

aDispatcher
Dispatcher
anAnalyser
Analyser
anAnnotationBlock
AnnotationBlock
revise
ConflictResolutor
TrEMBL Annotation
AnnotationBlock(SWISS.entry,Entry,Analyser):void
accept_annotation(AnnotationBlock):void
annotate(AnnotationBlock):void
accept_annotation(AnnotationBlock):void
removeConflict(ConflictResolutor):void
perform(Entry):Entry
incorporate results of conflict resolution
return final annotation

Figure 19: Class interaction diagram for the annotation process

This would be more difficult in standard programming languages like C or C++. However, this is not the most important difference of Java to Perl, Tcl/Tk or Python. These scripting languages are functioning without compilation and on different platforms as well. The primary reason for favouring Java was the ease of distributed computing via Remote Method Invocation.

The implementation raised the interest of a group of computer scientists who wanted to compare the principle of the implementation with the performance of a commercial agent system (Cogan, Gomoluch et al. 2001). It was shown that the performance of both environments does not differ very much, which also has the consequence that it would be feasible to increase flexibility even further with little additional coding effort.

PROLOG

The language PROLOG (Sterling 1994; Bratko 2000) differs from procedural languages in its principle of performing complete recursive searches through a supplied list of facts. For this application, the facts represent the original TrEMBL entry and annotation derived from incorporated tools.

A call of a function, which is a predicate in PROLOG, potentially succeeds more than once. Variables that are passed as arguments of the predicate become defined, they are instantiated or unified, such that the predicate succeeds.

Examples:

Teaching the PROLOG engine a fact on a specific protein, here a description:

```
?- assert(protein('TOL9','TOLL-LIKE PROTEIN 9')).
Yes
```

Seeking feasible instantiations of the variable X to match known facts:

```
?- protein('TOL9',X).
X = 'TOLL-LIKE PROTEIN 9' ;
No
```

Only one solution was found.

In EDITtoTrEMBL, all semantic analysis is done with assistance of rules stated and evaluated in PROLOG. The REVISE system is also implemented in this language. It is used as an engine, which is not only capable of presenting conclusions, but also to determine and to resolve conflicts (Damasio, Pereira et al. 1997).

Historically PROLOG has been the language of choice for many applications:

Language analysis

The analysis of context sensitive languages is probably the most eminent application of PROLOG. Natural Language Processing belongs to this group.

Knowledge processing

PROLOG can directly be used as in inference engine. However, in order to be able to trace the inference process an additional layer on top of the PROLOG engine is usually used.

Many different PROLOG engines are available. It was of importance for this thesis to choose an implementation, which is available on all the major platforms to allow an integration of EDITtoTrEMBL throughout the institute.

Java and PROLOG are very compatible with each other. They can even share variables. For EDITtoTrEMBL, a free implementation of PROLOG in Java, JavaLog (Amandi, Zunino et al. 1999), was chosen. It was chosen in favour over the alternative SWI PROLOG (Wielemaker 2001) to remain most platform-independent. This decision is due for revision, because of problems with the performance of the Java-based PROLOG system.

G. Future development of EDITtoTrEMBL

A few things have changed since EDITtoTrEMBL was created. Most importantly TrEMBL, but also SWISS-PROT, are available from within a relational database system. This section describes consequences for the functionality of the environment to adapt to these changes.

1. Integration of a relational database

The accessibility of SWISS-PROT/TrEMBL via a relational database helps to delay the occurrence of severe problems with an increasing number of analysers. The separation of entries by dispatchers can be substituted partially by relational queries that select all entries with certain properties. This can lead to bulk-submissions to analysers and these could write back their annotation into the database for a later retrieval without the program to be rerun. For the transmembrane proteins this was implemented, all tools store their results in a database, which can then be updated independently from any TrEMBL annotation run. The problem with this approach is that it does not scale well. With an increasing number of specialised applications and many interdependencies of applications this approach is no longer feasible.

2. Syntax versus content

Not to have used XML (W3C 1997) from the beginning was a wrong design decision. Much time during the development of EDITtoTrEMBL was spent on the adaption to the SWISS-PROT format which could have been saved since XML is used since late 2000 within the SWISS-PROT group. This move was initiated by the increased acceptance within the SWISS-PROT group of a relational database to store information, which led to a separation of content stored from the presentation in the entry as requested for the use of XML. In order to achieve the incorporation of XML, only the Java SWISS-KNIFE needs to be extended towards the XML representation. The remaining classes do not need to be adapted.

With XML the presentation of evidence for individual statements and other relationships between different parts of the annotation of an entry could be better represented than in the SWISS-PROT annotator's or programmer's section.

3. Workflows

The (indirect) specification of rules to determine the workflow is straightforward and a simple procedure. However, many issues towards the workflow problem have not been addressed, e.g. updates of SWISS-PROT/TrEMBL during annotation runs or the concurrent updates of analysers. There is no tool available to provide such a complete solution (Kreil 2001). EDITtoTrEMBL does not address most problems of workflow specifications in bioinformatics, e.g. the asynchronous update of databases. By design it relies on a stable system during the annotation process. More flexibility to allow partial annotation runs, reflecting variances in the availability of tools or other criteria, would be useful.

4. Creation and validation of rules

It is often problematic to generate of rules for protein annotation that depend on some computational analysis, such as the match of a sequence to an InterPro entry, and hold for all entries in SWISS-PROT and especially TrEMBL. The SWISS-PROT group at the EBI is maintaining an automation of a rule discovery process (Fleischmann, Möller et al. 1999; Kretschmann, Fleischmann et al. 2001) and caters for the manual verification of proposed rules.

The formulation of biologically sound rules and consistency checks to spot errors in the automated sequence annotation is of an increased difficulty. This is due to an extensive list of exceptions. The early acceptance of conflict-resolution is therefore seen in the integration of multiple databases rather than for finding conflicts within a single database like SWISS-PROT/TrEMBL. However, conflict-resolution may well be incorporated not as a tool to detect errors, but as part of knowledge representation, thereby simplifying rules and their maintenance, avoiding to make rules aware of too many exceptions and thereby extending their validity.

5. Intra-Entry Rules and Inter-Entry Relationships

All here described rules have a single entry as a basis although this is not a technical limitation. Links between entries are perfomed indirectly via sequence domains and rules for annotation that are triggered by them. Conversely one could select all entries with a certain InterPro domain and update the annotytion of these TrEMBL entries only. Since the annotation and conflict-resolution is performed context-free, i.e. without mutual dependencies or impact, there is also no feedback on the rules applied for annotation that would impose restrictions for the application of rules.

6. Summary

EDITtoTrEMBL enhanced the flexibility and scalability of automated protein annotation. The number of tools incorporated into the annotation process can be dynamically increased with no interference with the handling of other tools. The system needs little maintenance and the system scales with higher workloads.

V. Transmembrane Proteins

The field application of the designed system for automated protein sequence annotation are transmembrane proteins. For this, the tools incorporated into the system had to be evaluated (C) and for the evaluation a test set of transmembrane proteins had to be created (B). Finally, to implement the integration of membrane protein topology prediction methods, respective rules for conflict resolution needed to be established (D and E).

A. Introduction

This section is aimed at those readers with a computational sciences background. It presents some biological background information helpful to understand the motivation to focus on transmembrane proteins.

The following introduction will also contribute to a full understanding of

- facts and revisables
- rules for the integration of transmembrane topology predictors.

1. Function of transmembrane proteins

A protein is transmembrane if it extends to both sides of a membrane. This was first proven for glycophorin in the membranes of red blood cells (Bretscher 1971) in the MRC Laboratory of Molecular Biology here in Cambridge. Membranes are the boundaries of cells or their compartments. Thus information or molecules that are passed between cellular units, or subunits, must pass through membranes. About a fourth to a third (own findings, (Wallin and Heijne 1998; Stevens and Arkin 2000)) of all genes of an organism code for transmembrane proteins.

The membrane proteins are called integral, as they are not easily separable from the membrane. This is used almost synonymously with transmembrane, which states that a protein knowingly spans through the membrane. A protein is *not* transmembrane but merely membrane-associated if it is e.g. linked to the membrane via a post-translationally added lipid or the sequence only enters the membrane without fully spanning it.

For simpler unicellular organisms (Shapiro and Dworkin 1997) as for multicellular organisms, information has to be passed between cells to co-ordinate their differential development and their behaviour. Only by concerted actions of a multitude of functionally diverse cells, advanced activities may be carried out. This communication is performed by hormones or gasses via the blood, modulating ion concentrations in nervous tissue and by protein-protein contacts between neighbouring cells.

While some hormones can traverse the membrane (e.g. steroids), the vast majority of substances only has a controlled access to cells, at least while the substance is present in physiological concentrations. The membrane permeability of a substance depends upon both its size and polarity. The cell, serving energy management and communication, actively maintains different concentrations of substances, i.e. ions, which involves membrane proteins.

Many transmembrane protein families are involved in transport activities, either directly or indirectly. Signals can be passed through membranes also without movement of chemicals. Instead information can be transferred by an induced conformation change of a transmembrane protein, which in turn invokes a reaction on the other side of the membrane.

An example of such proteins are G protein-coupled receptors (GPCRs), a class of receptors described in detail in section F.

2. Transmembrane topology

For the understanding the function of of a transmembrane protein, as for any protein, it is helpful to determine the spatial organisation of the protein, and at best determine its full three-dimensional structure. If the sequence of a transmembrane protein is known, one can predict which residues are buried within the membrane and which parts of the protein form loops on either side of the membrane. This guides the analysis of proteins by experimental mutagenesis or by considering the significance of natural variations in its sequence. For a cell-surface receptor protein, a mutation on the outer side of the membrane would more likely affect the ligand binding, while mutations affecting the residues located on the inner side of the protein may lead to changed protein-protein interaction of the receptor with cytoplasmic proteins involved in intracellular signalling.

A protein sequence is represented as a linear chain of amino acids (Figure 20). When integrated into a membrane, proteins can be visualised as boxes spanning the membrane, connected by the protein chain loops. Transmembrane segments for cytoplasmic or mitochondrial inner membranes are assumed to be helical structures, and to have a length between 16 and 25 residues (Lemmon, MacKenzie et al. 1997). The majority of positively charged residues are found on the inner side of the membrane (Heijne 1986), a rule that holds especially for bacterial proteins.

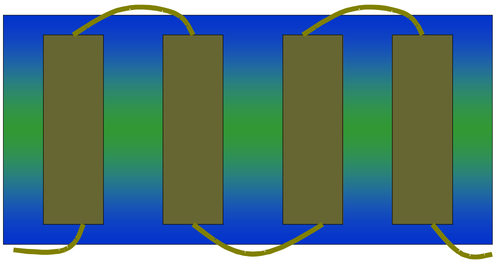


Figure 20: Schematic drawing of a transmembrane protein with four transmembrane regions

The dark boxes represent helices spanning the membrane. While in general the outer compartment is drawn to be on top, the converse holds for the community of structural biologists.

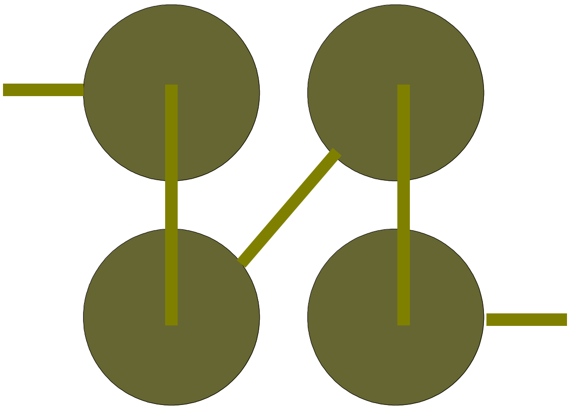


Figure 21: Schematic top-view of a transmembrane protein

The solid dark circles represent helices seen along their axes. The linking sequences connecting centers of helices are visible from the compartment of the observer.

While the definition of a transmembrane protein is very clear, classifying some proteins is difficult. There are cases for which it is not certain that the protein's N-terminus solely fully passes the membrane, or alternatively forms a turn within the membrane double layer, in spite this being considered as energetically unfavourable. In addition, a recently determined structure of a protein with six transmembrane regions (Fu, Libson et al. 2000) proved another surprise, as two helical moieties were shown to reach only half through the membrane. These two meet within the membrane and thereby form a transmembrane-like structure. From the viewpoint of a formal representation of topology, people regard these moieties as intramembrane, but not as transmembrane. However, when interested in the functional aspects of transmembrane regions, then it seems less certain whether this is the correct approach.

Gram-negative bacteria, such as *E. coli*, have a second layer of membrane that substitutes properties of the otherwise stronger sugar coating of the cells. In addition, mitochondria have a second, outer, membrane. Proteins with multiple transmembrane proteins in outer membranes are found to form a ring of beta-sheets, a barrel-fold, as their topology. The number of residues needed to span the outer membrane in this form is smaller than for their helical counterparts, requiring only about 12 residues instead of 17. Therefore, different predictors for the transmembrane topology of proteins have been developed. An overview on the development of signal prediction methods and determination of the subcellular location of proteins is given in a paper of Kenta Nakai (Nakai 2000).

Membrane proteins (Blobel 1980; Singer 1990) and transmembrane helices (Sakaguchi 1997) are grouped into classes in dependence

- of the number of membrane spanning regions (one or many),
- the direction of insertion and
- the prior cleavage of a signal.

Other classifications (Gennis 1989; Payne 2001) and extensions of the above additionally incorporate

- membrane-associated proteins and
- have an extra class for those transmembrane proteins that also have globular loops attached by lipid anchors.

The classification of transmembrane protein is usually performed by the assignment of a number to denote a particular class. However, due to the existence of different classification systems, the assignment of numbers to a protein without mentioning the classification continuously leads to considerable confusion. In the following the assignment of a number to denote the classification of a protein has been avoided in favour of the protein's description by its topological properties as summarised in Table 2 (Jennings 1989).

Monotopic	A membrane associated protein who's membrane region does not pass the bilayer.
Bitopic or singlespanning	A transmembrane protein with a single transmembrane region.
Polytopic or multispanning	A transmembrane protein with multiple transmembrane regions.

Table 2: Consensus of classifications for transmembrane proteins

The number of MSRs is the most eminent property of a transmembrane protein. A description that goes beyond the above offered detail, will most probably also name the protein family.

3. Determination of membrane spanning regions

Different techniques have been developed to study the topology of transmembrane proteins. An excellent review on the methods available for topology prediction, together with strengths and drawbacks is provided by (Geest and Lolkema 2000). These are summarised below.

C-terminal fusion with indicator protein

The most common procedure to determine the transmembrane regions of proteins is by the fusion of the C-terminal part of the sequence with an indicator protein, which is active only in a specific compartment of the cell. At different positions the indicator protein is fused to the C-terminus of a transmembrane protein under investigation. The original C-terminal sequence is substituted by the soluble indicator. These fused proteins are overexpressed and the activity of the indicator is measured.

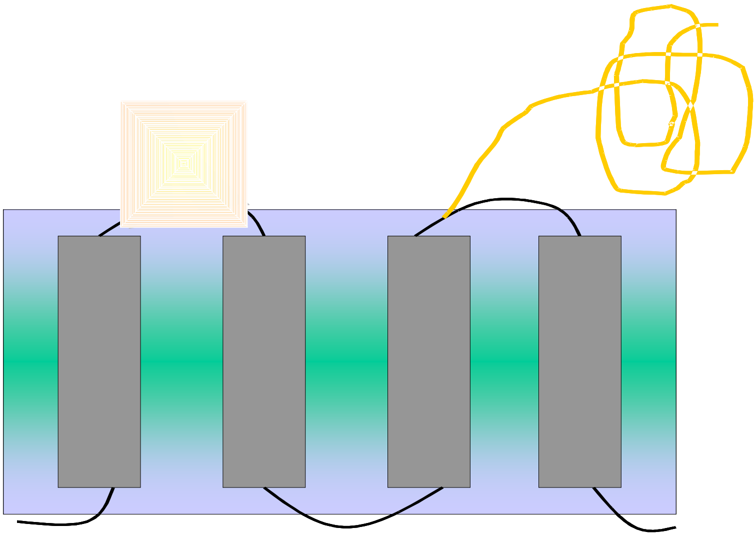


Figure 22: Visualisation of Experiments for the determination of transmembrane topology

The Y-like figure represents an antibody binding to a subsequence (epitope) or the transmembrane protein. The yellow line represents a C-terminal fusion with an indicator protein, who's activity depends on the fusion point with the transmembrane *protein as the indicator protein itself can't pass the membrane and is active only in a single compartment.*

To analyse the results, the values are sorted according to the position of the fusion point, usually from the N to the C-terminus. The change of the indicator protein's activity from high to low or from low to high between two fusion points would be interpreted as the location of a membrane spanning region. One should be aware of certain artefacts, though. The activity of the indicator protein should be normalised according to the protein concentration and to the maximum activity found in the series of experiments. Although the protein expressions are normalised, these don't take identical values for high and low activity. The normalised values may be regarded as a propensity of a protein to have a certain topology when cleaved at the respective position.

Most experiments, including a few examples of eukaryotic proteins, are performed by an expression of the protein in *E. coli* with one or two of the indicator proteins shown in Table 3. A few eukaryotic proteins have been studied in yeast, with the fusion of prolactin and a subsequent antibody binding or protease accessibility study.

Protein	Gene	Compartment where active
Alkaline phosphatase	phoA	Periplasm
β-lactamase	blaM	Periplasm
β-galactosidase	lacZ	Cytoplasm

Table 3: Commonly used indicator proteins for topological analysis

Many fusions to different positions, ideally with complementary indicator proteins, should be made for a complete analysis. Another variant are ‚sandwich fusions‘, in which the indicator protein is inserted rather than substituting the original C-terminus. This should preserve interactions of membrane spanning regions during the membrane-insertion process and thereby avoid the introduction of artefacts by the fusion.

Antibody binding experiments

The insertion of a known recognition site, called epitope, for a specific antibody into the protein sequence is used to direct the antibody to a specific part of the transmembrane protein. If the induced binding site is accessible to the antibody, then the protein's location can be visualised in the membrane by using a secondary fluorescent antibody and microscopy. The C-terminal fusion experiment with prolactin can be regarded as a special case of epitope insertion.

Proteolysis

Proteins are degraded by exposure to enzymes called proteases. Proteases are often specific for particular residues. As a result cleavage of a given protein sequence will generate fragments of characteristic lengths. The integration of a protein into a membrane prevents cleavage at residues in the membrane and on the cytoplasmic side of the membrane. Hence, analysing the peptide fractions that are produced following protease treatment by SDS-PAGE will allow insights into the protein's transmembrane topology.

Post-translational modification

A variety of chemical modifications is potentially performed upon a peptide chain, during and after its translation. Such modifications are performed in specific subcellular compartments. The finding that a certain residue becomes e.g. glycosylated, proofs that this is an extracellular residue.

4. The difficulty in experimentally determining the topology of transmembrane proteins

Biochemically, the topology of transmembrane proteins cannot be fully determined. Experiments testing the accessibility to proteases or antibodies still leave room for interpretation and so do fusion experiments with indicator proteins. An early overview on the field is given in (Jennings 1989). Transmembrane proteins are very hard to crystallise. The problems in growing crystals lie in the aliphatic nature of transmembrane proteins and in getting both large and pure quantities of protein. Hence, only very few crystal structures of membrane proteins (about 20) are available (Fyfe, McAuley et al. 2001). Even the few crystal structures available do not give experimental evidence on how exactly the protein is embedded in the membrane. They also do not answer, whether the membrane's thickness changes upon insertion of a protein.

NMR studies of membrane proteins have advantages over crystals in that these show proteins in solution, which is a more natural conditions for a protein. They can also give information about the flexibility of the protein. However, an NMR experiment only allows small proteins or a fraction of the protein to be analysed at a time. Again this leads to an investigation under unnatural conditions.

These unresolved issues make the evaluation of a computational prediction of membrane protein topology and structure difficult. On the other hand, this also means that the biologist does not expect a completely reliable annotation, as it is simply not possible with current models.

5. Dependence on programs for transmembrane segment prediction

Since biochemical experiments as described in the previous section 3 have only been carried out on a relatively small number of proteins, biochemical evidence to allow a reliable propagation of transmembrane annotation is only available for a few protein families. Methods for the prediction of membrane spanning regions in proteins focus on modelling the properties of individual helices on a very abstract level. A sequence that applies well to such a model then will be assumed to have the properties as assigned by the model, which is the prediction.

The ever-increasing number of hypothetical protein sequences being provided to the scientific community makes such predictions indispensable.

6. Topology predictions for heptahelical receptors

Heptahelical receptor proteins, as exemplified by the G protein-coupled receptors (GPCRs), are integral membrane proteins that play a central role in the cell-cell signalling mechanisms of eukaryotes. The actions of a large number of hormones, neurotransmitters, peptides, and odourants are initiated by their binding to such cell surface receptors. This triggers an intracellular signalling cascade, which eventually leads to the appropriate physiological response.

These receptors and their ligands have been the subject of intensive study, since they play key roles in many aspects of mammalian physiology, and provide numerous therapeutic targets for the pharmaceutical industry (Sautel and Milligan 2000). Mechanistic analysis of the functioning of GPCRs has been hampered by the lack of solved 3D structures for GPCRs. Instead, molecular models have had to been built based upon the structure of bacteriorhodopsin (a light-driven proton pump), and since the recent report of its crystal structure (Palczewski, Kumasaka et al. 2000) on that of bovine rhodopsin (a heptahelical light receptor).

In the absence of structural information for GPCRs, prediction of their membrane topology is critical, both for the building of homology-based structural models, and for making inferences about the likely functions of domains within the receptor sequence. Such hypotheses can then be tested by the design of suitable experiments e.g. mutagenesis studies.

Unfortunately, accurate prediction of the membrane spanning regions (MSRs) of GPCRs has proven difficult (section C), principally because of the amphipathic nature of some of their MSRs. This arises due to the presence of charged residues that are buried within the membrane helices, which (at least in some classes of GPCRs) are thought to participate in ligand binding. In section 2 an application of the topology prediction for a prediction of protein-protein interactions is described.

B. Test set of Transmembrane Proteins

This section describes the creation of a reference annotation set for membrane proteins. A collection of transmembrane proteins with annotated transmembrane regions, for which good experimental evidence exist, was created as a test- or training set for algorithms to predict transmembrane regions in proteins.

This collection unifies, updates and verifies existing test sets. These were created for the development of TMHMM (Sonnhammer, Heijne et al. 1998), HTP (Rost, Casadio et al. 1996), DAS (Cserzo, Wallin et al. 1997), CoPreTHi (Promponas, Palaios et al. 1998), SOSUI (Hirokawa, Boon-Chieng et al. 1998), TMPDB (Shimizu and Nakai 1994) and HMMTOP (Tusnády and Simon 1998). Additional references and information were extracted from SWISS-PROT (Bairoch and Apweiler 2000) and from the literature.

1. Motivation

A program for the prediction of membrane spanning regions in proteins needs reliable data both for its specification and for its verification. The created framework for automated annotation should be applied to transmembrane proteins and for the evaluation of this integration - again a good reference annotation is required. This analysis is presented in section C.

In order to benchmark the performance of transmembrane prediction programs it is necessary to use a test set of sequences with experimentally confirmed transmembrane regions. Previous test sets did not explicitly state their source of information, with one exception (Shimizu and Nakai 1994). Most programs for the prediction of transmembrane proteins just provide a list of SWISS-PROT entries, which where used for evaluation or training of the program. These entries where collected and re-annotated according to information from the literature.

The required interpretation of experimental data to derive the topological information makes the creation of test sets with concrete boundaries for transmembrane regions a difficult task. What was needed is a reference towards the original data, which is now provided by this work. Without this background information it could hardly be said, to what extend an aberration from the reference annotation could be tolerated.

The test set is available to the authors of topology prediction programs. The collection is used to improve their programs. This in return helps to further improve the annotation of

TrEMBL. A problem with this is that once the information is passed back to the programmers, a reliable evaluation of programs becomes again more difficult. The following section C describes the use of this test set for an evaluation of current prediction methods for MSRs.

The most important source for the test set is the available information in literature. Most publications retrieved where found in the literature database MEDLINE. Many others where referenced in those.

2. Concept

The test set is organised in two files, available for the scientific community from the EBI's FTP server (Möller 2000). The first file contains the sequences and the transmembrane annotation and the second file contains the experimental evidence for the respective annotation. Both files are stored in a syntax close to the one used by SWISS-PROT (Bairoch and Apweiler 1999). Four additional line types were introduced to ensure that all information relevant for this test set could be stored.

The TS-line (Test Set) lists references to papers and programs where an individual sequence was mentioned to have been used for training or evaluation. The TR line assigns a level of trust to the transmembrane annotation, based on the experimental data available. Table 4 presents the four basic categories of trust of a transmembrane topology.

A	Structure available
B	Very good biochemical characterisation with at least two complementary methods
C	Basic biochemical characterisation done. Some type C annotation got the additional flag "PARTIAL" to emphasise that the annotation is reliable only for a part of a sequence.
D	No biochemical characterisation available. This includes papers that present only a hydrophobicity analysis or an alignment as a basis for their annotation.

Table 4: Categories of trust for the transmembrane proteins in the test set

It should be emphasised, that those experiments flagged with C or C.PARTIAL still give essential information when one looks in more detail at the provided experimental data.

Finally the PL line type was introduced to store the plain experimental data as explained in the next section.

```
ID   ATP6_ECOLI     STANDARD;      PRT;   271 AA.
AC   P00855; Q47708;
TR   C.
TS   TOPOLOGY.
FT   DOMAIN       1    38  PERIPLASMIC (PROBABLE).(MEDLINE; 98298136)
FT   TRANSMEM    39    60  PROBABLE.              (MEDLINE; 98298136)
FT   DOMAIN      61   105  CYTOPLASMIC (PROBABLE).(MEDLINE; 98298136)
//
```

Figure 23: A typical entry in the annotation file

The ID an AC line are taken from the respective SWISS-PROT entry, the FT line follows the SWISS-PROT syntax, except for the reference of the evidence which is the MEDLINE UI.

Figure 23 shows an example for a typical entry in the annotation file. The FT (feature table) lines in the annotation file contain the annotation and the source of the experimental evidence for the annotation. The cited reference links to the corresponding experimental data stored in the second file. The evidence flags in SWISS-PROT have a different format since this solution was in place before these were introduced. Unfortunately, rather than the MEDLINE UI (from which information on the publication date could be derived) the PUBMED ID should have been chosen since the UI is no longer provided. Since the collection's publication new important structures have been derived, hence a new edition is due and should then also adapt the evidence scheme of SWISS-PROT.

The qualifier "POTENTIAL" for an FT line means that this feature was predicted. This qualifier is only found in entries of trust (TR) level D and in some FT lines of entries with partial biochemical characterisation of membrane spanning regions. In entries with a trust level B or C you will find the qualifier "PROBABLE" in FT lines. No additional flag is given for transmembrane proteins with a solved structure.

Formalism to store experimental data

Three kinds of experiments were accepted as a source for transmembrane annotation which were explained before (page 94):

- *C-terminal fusion* with indicator proteins
- *Antibody binding*: it is tested if an antibody binds to known epitopes
- *X-ray diffraction* to determine a 3D structure

The `PL` (predicate language) lines represent experimental data. These are predicates with two arguments:

```
experiment(description, measurements).
```

The description explains modifications made to the original peptide. The vast majority of papers utilised *(93%)* describe at least one C-terminal fusion. Such a fusion is expressed with the predicate stated as

```
fusion(indicator, position).
```

Measurements are expressed as a combination of the measured property and a value. The description of experiments to determine a protein's topology may look like the one below:

```
experiment(fusion(lacZ,150-151),activity(2781)        ) ).
experiment(fusion(blaM,188-189),resistance(positive) ) ).
```

Figure 24: Format of stored experimental data for transmembrane topology

Every experiment is stored in a single line. There are two arguments, the first describes the experiment, the second the observation. LacZ and blaM are indictor proteins of fusion experiments as explained in Table 3.

Figure 25 shows a complete entry for the experimental evidence of a 1991 paper leading to a topology for the *E. coli* mannose transporter.

```
ID   MEDLINE@96370834
DE   Membrane topology of the mannose transporter of Escherichia coli
DE   K12.
PA   [1] IIC
DR   SWISS-PROT; P08187; PTNC_ECOLI.
TR   B.
PL   num_tm_regions(6).
FT   DOMAIN        1      4       CYTOPLASMIC.
FT   TRANSMEM      5     24
FT   DOMAIN       25     49       PERIPLASMIC.
FT   TRANSMEM     50     69
FT   DOMAIN       70     83       CYTOPLASMIC.
FT   TRANSMEM     84    102
FT   DOMAIN      103    105       PERIPLASMIC.
FT   TRANSMEM    106    124
FT   DOMAIN      125    150       CYTOPLASMIC.
FT   TRANSMEM    151    170
FT   DOMAIN      171    181       PERIPLASMIC.
FT   TRANSMEM    182    201
FT   DOMAIN      201    266       CYTOPLASMIC.
PL   experiment(fusion(phoA, 21), activity( 455)).
PL   experiment(fusion(phoA, 22), activity( 513)).
PL   experiment(fusion(lacZ, 22), activity(1024)).
PL   experiment(fusion(phoA, 24), activity( 346)).
PL   experiment(fusion(phoA, 28), activity( 394)).
PL   experiment(fusion(phoA, 33), activity( 291)).
PL   experiment(fusion(phoA, 43), activity(  71)).
PL   experiment(fusion(phoA, 47), activity( 223)).
PL   experiment(fusion(phoA, 52), activity( 192)).
PL   experiment(fusion(phoA, 54), activity( 169)).
PL   experiment(fusion(phoA, 55), activity( 177)).
PL   experiment(fusion(phoA, 59), activity(  91)).
PL   experiment(fusion(lacZ, 75), activity(1756)).
PL   experiment(fusion(phoA, 79), activity(   9)).
PL   experiment(fusion(lacZ, 81), activity(2642)).
PL   experiment(fusion(phoA, 82), activity(  20)).
PL   experiment(fusion(phoA, 87), activity(  17)).
PL   experiment(fusion(phoA,101), activity(  17))
PL   experiment(fusion(phoA,102), activity(  73)).
PL   experiment(fusion(phoA,106), activity( 252)).
PL   experiment(fusion(phoA,110), activity(  78)).
PL   experiment(fusion(lacZ,120), activity(1385)).
PL   experiment(fusion(lacZ,137), activity( 496)).
PL   experiment(fusion(lacZ,138), activity(2783)).
PL   experiment(fusion(lacZ,152), activity(1349)).
PL   experiment(fusion(phoA,153), activity(  41)).
PL   experiment(fusion(phoA,164), activity( 222)).
PL   experiment(fusion(phoA,171), activity( 386)).
PL   experiment(fusion(phoA,173), activity( 140)).
PL   experiment(fusion(phoA,175), activity( 425)).
PL   experiment(fusion(phoA,179), activity( 241)).
PL   experiment(fusion(phoA,184), activity( 520)).
PL   experiment(fusion(phoA,185), activity( 668)).
PL   experiment(fusion(phoA,226), activity(  61)).
PL   experiment(fusion(phoA,237), activity(  54)).
PL   experiment(fusion(phoA,241), activity(  36)).
PL   experiment(fusion(lacZ,245), activity(2708)).
PL   experiment(fusion(lacZ,252), activity( 611)).
PL   experiment(fusion(lacZ,266), activity(2702)).
[...second subunit omitted...]
//
```

Figure 25: Example entry describing experimental evidence for a transmembrane protein's topology

For the dataset an interface to SRS 5 (Etzold, Ulyanov et al. 1996) has been created, equivalently grammars for other languages can be created for the retrieval of data.

3. Results

The current release covers cytoplasmic membrane and proteins of the mitochondrial inner membrane and includes 320 sequences of which 69 have not previously been used for the training or analysis of transmembrane region prediction methods. Information from 214 papers was used to find 33 membrane proteins with known structures, 24 with an in-depth biochemical characterisation and 142 with at least partial biochemical evidence. A dozen new membrane proteins annotations will appear in the next release.

It was tried to find a confirmation for all proteins referred to by other test sets and to add new characterisations (see Table 5).

Test set	A	B	C	D	Total
CoPreTHi	9	9	64	70	152
DAS	4	6	31	3	44
HMMTOP	11	9	65	71	156
HTP	11	3	15	51	86
TMHMM 83	7	3	21	52	83
TMHMM 160	13	11	60	76	160
TMPDB	12	5	32	5	54
TrEMBL	1	1	5	1	8
SWISS-PROT	32	23	137	120	312
Non redundant	28	24	97	0	149
New (% of prev. known)	7 (27%)	10 (71%)	52 (58%)	0 (0%)	69 (27%)
Total	33	24	142	121	320

Table 5: Incorporated test sets

The table gives an overview how sequences of other test sets were assigned to the four categories of quality in this work.

Annotations assigned the trust level D should not be used for training or testing purposes. This test set has 199 entries assigned to trust levels A-C. This is more than twice the number of reliable entries that any of the other test sets can offer. The test set gives also information on the reliability, a feature not available in previous test sets. Besides the evaluation of current test sets, this test set contributed a significant number of new reliable annotations.

“Non-redundant” in Table 5 stands for a subset of entries of trust levels `A` to `C`, derived from a protein sequence clustering. Entries with the highest trust level and different topologies were selected from each cluster. The clustering procedure is based on pairwise sequence similarities of ‘all against all’, using the Smith-Waterman algorithm (Smith and Waterman 1981). To measure the statistical significance of each Smith-Waterman score, additional searches with 1000 shuffled copies of the query sequence were performed to determine the Z-score (Comet, Aude et al. 1999; Pearson 2000). Clusters were built using a single linkage algorithm for a Z score of 10 or higher.

Summary of the test set work

The borders of membrane spanning regions cannot be clearly determined and a final ambiguity always remains. Due to that a reference annotation alone did not seem sufficient for an evaluation of predictions.

The augmentation of this test set with the underlying experimental data provides necessary background information on remaining ambiguities in the transmembrane annotation and therewith facilitates the evaluation of an algorithm’s performance. The numeric values of the indicator protein’s activities may also be used to calibrate the prediction methods.

The interpretation of individual experiments with indicator proteins (Prinz and Beckwith 1994) (Traxler, Boyd et al. 1993) or post-translational modifications is sometimes difficult. Even data that was derived from crystal structures does leave room for interpretation since the membrane’s thickness may be different where the protein is inserted and transmembrane helices often extend beyond their intramembranous residues. No definite rules can hence be given how experimental data should be mapped to transmembrane annotation.

The full test set is redundant in terms of sequence similarity and thus a non-redundant subset of the test set is also provided.

The test set will be further maintained. A number of additional proteins have been described that will find their way to the FTP server after the thesis has been submitted. Additionally, outer membrane proteins are currently being added. The work will have a positive impact on current and future algorithms (and potentially biological methods) for the determination of transmembrane regions. It was used in new versions of both HMMTOP (Tusnády and Simon 2001) and TMHMM.

C. Evaluation of Predictors of Membrane Spanning Regions

This section summarises the evaluation of the performance of the current most widely used and best known methods for the prediction of transmembrane regions in proteins.

Such predictions are possible because of distinctive patterns of hydrophobic (intra-membranous) and polar (loops) regions within the sequence. The topology of the vast majority of membrane proteins remains biochemically undetermined. A collection of proteins with known biochemical characterisations of membrane topology (Möller, Kriventseva et al. 2000) was described in the previous section. However, this collection contains only ~200 well-characterised sequences. Hence, the characterisation of the remaining membrane proteins requires an accurate method for the automated prediction of MSRs.

Reliable computational methods for topology predictions are very valuable as they provide the basis for further experimental analysis. A variety of tools have been implemented, with the first being about 20 years old. For an evaluation of predictions, it is important not only to look at individual MSRs but at the whole protein. To make a prediction for proteins with seven MSRs 95% reliable, individual segments would need to be 99,96% reliable, and additionally, the method must never over-predict. Current tools are far away from achieving this. The present study provides an evaluation of their actual performance.

1. Evaluation

The following methods for prediction of MSRs have been evaluated: TMHMM 1.0, 2.0, and a retrained version of 2.0 (Sonnhammer, Heijne et al. 1998), MEMSAT 1.5 (Jones, Taylor et al.

1994), Eisenberg (Eisenberg, Weiss et al. 1982), Kyte/Doolittle (Kyte and Doolittle 1982), TMAP (Persson and Argos 1997), DAS (Cserzo, Wallin et al. 1997), HMMTOP (Tusnády and Simon 1998), SOSUI (Hirokawa, Boon-Chieng et al. 1998), PHD (Rost, Casadio et al. 1996), TMpred (Hofmann and Stoffel 1993), KKD (Klein, Kanehisa et al. 1985), ALOM 2 (Nakai and Kanehisa 1992), and Toppred 2 (Claros and Heijne 1994).

The previously mentioned collection of well-characterised membrane proteins was used as the reference annotation to evaluate the predictions of the various methods. This test set contains 188 proteins with 883 MSRs that have been determined from either their elucidated structures or by fusion experiments. As described in section B the interpretation of experiments does not allow one to set unambiguous borders for transmembrane regions. Therefore, some deviation of the prediction from the reference annotation must be tolerated. In accord with the authors of TMHMM (Sonnhammer, Heijne et al. 1998), for an MSR to be evaluated as correct, it is requested to share at least 9 residues with the reference annotation's MSR. This threshold is a little less than half that the ~20 residues expected for an MSR.

Every program was rated by three values.

1. The percentage of predicted transmembrane regions that could be assigned to a reference MSR (true positive predictions),
2. the percentage of reference MSRs that were not predicted (false negatives) and
3. by the percentage of predicted MSRs which are not existent as MSRs in the reference protein test set (false positives).

Also, but not applicable to all methods, the reliability of a prediction was investigated to determine the sidedness of the protein's membrane integration.

The reference annotation describes proteins of both the mitochondrial inner membrane and plasma membrane. Proteins in these membranes are generally believed to span the lipid bilayer in a helical secondary structure. For this reason a minimum length of 15 residues would be expected in order to fully span the membrane. The default parameter sets were used for the evaluation of all methods. For the evaluation of TMpred, parameter values of a minimum length of 15 and a maximum length of 25 residues for MSRs were utilised. It is likely that in order to achieve optimal results these values should be varied, depending on the organism (varying thickness of the membrane) or organelle (hypothetical influence of the length of MSRs on protein sorting (Munro 1995)). This optimisation of parameters was not performed in the present study, in order to keep the evaluation straightforward, and subsequently easily reproducible.

2. Results

Performance on transmembrane regions of all biochemically characterised proteins

Method	***TP***	***FN***	***FP***	***(FN+FP)***
TMHMM 2.0 (Sonnhammer, Heijne et al. 1998)	812	65	38	103
TMHMM 1.0 (Sonnhammer, Heijne et al. 1998)	818	63	45	108
TMHMM – Retrain*	811	70	38	108
HMMTOP (Tusnády and Simon 1998)	841	40	97	137
MEMSAT 1.5 (Jones, Taylor et al. 1994)	772	110	78	188
Eisenberg (Eisenberg, Weiss et al. 1982)	809	72	163	235
KKD (Klein, Kanehisa et al. 1985)	719	164	72	236
KD5 (Kyte and Doolittle 1982)	773	139	125	259
TMAP (Persson and Argos 1997)	675	191	82	273
DAS (Cserzo, Wallin et al. 1997)	829	38	243	281
SOSUI (Hirokawa, Boon-Chieng et al. 1998)	686	192	137	329
KD9 (Kyte and Doolittle 1982)	494	391	25	416
TMpred (Hofmann and Stoffel 1993)	525	357	80	437
ALOM 2 (Nakai and Kanehisa 1992)	429	545	17	471
PHD (Rost, Casadio et al. 1996)	564	319	207	526
Toppred 2 (Claros and Heijne 1994)	468	417	123	540
Total number of MSRs	883			

Table 6: Performance on Known Transmembrane Regions

TP stands for the number of correctly predicted MSRs, FN for MSRs that where not predicted and FP for predictions that where not confirmed by the reference annotation. The methods are sorted by the sum of false negative and false positive predictions. False negatives and true positives should sum up to the same number (883) for all the methods. This is not the case when a predicted MSR spans two reference regions. Also two predicted MSRs overlapping a single reference MSR would not be noticed in this table.

Table 6 shows the performance of the evaluated methods on individual MSRs. The methods are ranked according to the number of errors detected (FN+FP). The method TMHMM in all its three versions is by far the best in this comparison. MEMSAT is the second best method, although it produces twice as many errors as TMHMM. The only additional interesting result here is the low number of false positives assigned by ALOM. Its FP/TP ratio is even slightly lower than the one of TMHMM.

Performance on all MSRs within a protein

Method	*All MSRs Found*	*Additionally Correct Sidedness*
TMHMM – Retrain[*]	129 (69%)	102 (79% of 129)
TMHMM 2.0	128 (68%)	89 (70%)
TMHMM 1.0	126 (67%)	91 (72%)
HMMTOP	104 (55%)	84 (81%)
MEMSAT 1.5	100 (53%)	77 (77%)
KKD	85 (45%)	n/a
HMMTOP	83 (44%)	68 (82%)
TMAP	80 (43%)	21 (26%)
Eisenberg	72 (38%)	n/a
DAS	70 (37%)	n/a
TMpred	70 (37%)	12 (17%)
SOSUI	68 (36%)	n/a
KD5	61 (32%)	n/a
KD9	49 (26%)	n/a
PHD	49 (26%)	34 (69%)
Toppred 2	48 (26%)	23 (48%)
ALOM 2	14 (7%)	n/a
Total number of Proteins	188 (100%)	

Table 7: Performance on Proteins with characterised MSRs

This table presents an analysis of the program's performance in predicting all MSRs within a transmembrane protein. It displays in the second column the number of predictions that had all MSRs correctly assigned. This was defined as being the case when a sequence had no false positives, no false negatives and also the correct number of MSRs predicted. The third column shows how often the sidedness of the integration was predicted correctly.

Table 7 shows the performance of the evaluated method on all MSRs within a protein and basically confirms the results of Table 6. The TMHMM versions predicted in approximately two thirds of the reference proteins all MSRs correctly. In about 70 to 80% of these correctly predicted proteins, the sidedness was correctly predicted, too. The

[*] This version of TMHMM was developed for this evaluation only and is not available to the public.

retrained TMHMM performs better in the determination of the sidedness. MEMSAT was able to predict all MSRs correctly in 53% of the cases. While HMMTOP is the best method to predict the sidedness of a transmembrane protein, Toppred 2, TMAP and TMpred decide the sidedness less reliably than by random choice.

Performance on transmembrane regions of proteins unknown to the method

Method	*TP+FN*	*TP*	*FN*	*FP*	*FN+FP*	*%correct*
TMHMM–Retrain*	322	294	28	20	48	85.1
TMHMM 2.0	469	415	54	27	81	82.7
TMHMM 1.0	471	413	58	36	94	80
HMMTOP	452	421	31	63	94	79.2
MEMSAT 1.5	722	620	102	69	171	76.3
Eisenberg	881	809	72	163	235	73.3
KKD	883	719	164	72	236	73.3
KD5	907	773	134	125	259	71.4
TMAP	696	538	158	68	226	67.5
DAS	626	598	28	210	238	62
SOSUI	829	638	191	137	328	60.4
KD9	885	494	391	25	416	53
TMpred	882	525	357	80	437	50.5
HMMTOP	453	251	202	33	235	48.1
ALOM 2	883	429	454	17	471	46.7
PHD	883	564	319	207	526	40.4
Toppred 2	885	468	417	123	540	39

Table 8: Performance on known MSRs not used in the training sets of the method

TP stands for the number of correctly predicted MSRs, FN for MSRs that where not predicted and FP for predictions that where not confirmed by the reference annotation. The methods are sorted by the percentage of correct predictions. Please be aware that the number of MSR differs for different methods since the training/evaluation set of the methods differ. The set is smallest for the newer versions of TMHMM and HMMTOP.

Table 8 presents a variation of the analysis shown in Table 6, by being based on only those MSRs that were not presented to the respective program for its training or analysis. Again, the TMHMM versions performed best (80-85% correct predictions), slightly ahead of MEMSAT, which confirmed 76% of the MSRs correctly. The Eisenberg and KKD methods are very close runner-ups with 73.3% each. The low number of false negatives of the Eisenberg method (8.1%) and especially of DAS (4.5%) should be mentioned. The false negative rates of the best performing TMHMM version and of MEMSAT are 8.6% and 14%, respectively.

Performance on all MSRs within the proteins that where not used for training.

Table 9 and Table 10 present, like Table 7, a view on whole proteins rather than on individual MSRs. The intersection of proteins that were not used for training or analysis by any of the programs contains only *87* proteins. A larger data set optimises the reliability of this analysis for all individual methods. Hence, Table 9 presents the analysis of Table 7 based on different protein sets for each method. This allows to present each method with the maximal number of proteins unknown to it.

Method	*#proteins*	*All MSRs Found*	*Additionally Correct Sidedness*
TMHMM 2.0	108	64 (59%)	40 (63%)
TMHMM 1.0	108	57 (53%)	21 (53%)
TMHMM-Retrain	69	35 (51%)	22 (62%)
MEMSAT 1.5	159	80 (50%)	58 (73%)
KKD	188	85 (45%)	n/a
TMAP	156	69 (44%)	18 (26%)
HMMTOP	106	54 (51%)	42 (78%)
Eisenberg	188	72 (38%)	n/a
TMpred	188	70 (37%)	12 (17%)
KD5	188	61 (32%)	N/a
SOSUI	147	53 (36%)	N/a
DAS	148	50 (33%)	n/a
PHD	151	49 (33%)	34 (70%)
Toppred 2	188	48 (26%)	23 (48%)
KD9	188	48 (26%)	n/a
ALOM 2	188	14 (7%)	n/a
Total number of proteins	188		

Table 9: Performance on Proteins with characterised MSRs not known to the method

This table presents an analysis of the programs's performance on the whole transmembrane protein. Methods are sorted by the percentage of correctly predicted proteins. The second column shows the number of proteins that could be used for the evaluation since they were not presented to the respective program for its training or analysis. The third column shows the number of proteins whose MSRs where all correctly predicted. This was defined as being the case when a sequence had no false positives, no false negatives and also the correct number of MSRs predicted. The fourth column shows how often the sidedness of the integration was predicted correctly.

The drawback of this approach is that the methods are not constrained to the identical weaknesses and difficulties present in the evaluation set. Table 10 shows therefore the same analysis on the set of 87 proteins that were not involved in the training of any of these methods.

Method	*All MSRs Found*	*Additionally Correct Sidedness*
TMHMM-Retrain	52 (60%)	43 (83% of 52)
TMHMM 2.0	48 (55%)	36 (75% of 48)
TMHMM 1.0	45 (52%)	33 (73% of 45)
MEMSAT 1.5	41 (47%)	33 (80% of 41)
KKD	39 (45%)	n/a
HMMTOP	38 (43%)	30 (79% of 38)
TMAP	35 (40%)	12 (34% of 35)
TMpred	29 (33%)	9 (31% of 29)
Eisenberg	27 (31%)	n/a
SOSUI	27 (31%)	n/a
KD5	26 (30%)	n/a
KD9	25 (29%)	19 (83% of 23)
DAS	24 (28%)	n/a
KD6	21 (24%)	n/a
PHD	18 (21%)	17 (94% of 18)
Toppred 2	16 (18%)	6 (38% of 16)
ALOM 2	9 (10%)	n/a

Table 10: Comparison of Performance on an Identical Set of Proteins Unknown to Methods

This table presents an analysis of the program's performance on the whole transmembrane protein. The set of 87 proteins not involved in the training of any of the prediction methods was used as the basis for this analysis. Methods are sorted by the percentage of correctly predicted proteins. The second column shows the number of proteins whose MSRs where all correctly predicted. This was defined as being the case when a sequence had no false positives, no false negatives and also the correct number of MSRs predicted. The third column shows how often the sidedness of the integration was predicted correctly.

Both Table 9 and Table 10 confirm the dominance of TMHMM. The three versions of this method predict all MSRs within proteins that were not used for training in 51-60% of the cases correctly. MEMSAT correctly predicted 47% of all MSRs within proteins that are not used for training of the program.

Influence of signal peptides and transit peptides

Transmembrane prediction programs have the tendency to interpret the hydrophobic parts of signal sequences and transit peptides as MSRs. The transmembrane test set contains 34 proteins with a cleavable signal and 8 proteins with transit peptides. Table 11 shows that only ALOM 2 correctly predicted not a single signal sequence as transmembrane. ALOM 2 is followed by PHD with one error and Toppred 2 with three errors. The 7 errors of

TMHMM 2.0 account for 16% of the total TMHMM false positives from Table 6. Only the Kyte/Doolittle hydropathy analysis methods (KD5-KD9) predicted the 8 mitochondrial transit peptides as MSRs.

Method	*# Signal sequences predicted as MSRs*	*# Transit peptides predicted as MSRs s*
ALOM 2	0	0
PHD	1	0
Toppred 2	3	0
TMHMM 1.0	7	0
TMHMM 2.0	7	0
TMHMM-Retrain	9	0
MEMSAT 1.5	12	0
SOSUI	14	0
TMAP	20	0
Eisenberg	26	0
KKD	26	0
HMMTOP	29	0
TMpred	31	0
DAS	33	0
KD5	34	8
KD9	34	8
Maximum	34 of 34	8

Table 11: Discriminative performance on Signal and Transit Peptides

The second column displays the number of proteins in which a signal sequence was predicted to be a MSR. The third column shows the number of proteins in which a transit peptide was predicted to be a MSR.

Summary of evaluation based on reference TM annotation

Of the reference test set's 188 proteins, 162 proteins (85%) have their MSRs correctly predicted by at least one program. When the sidedness is included in this analysis, this reduces the number of correct predictions to 131 (70%). Table 7 shows TMHMM to be best performing. Its versions were able to predict at least 89 (48% of all proteins) entries completely correct, including their sidedness. In its retrained variant, it was even predicting 54% of the entries completely correct, although this improvement of the retrained version was due to the better performance on the determination of the sidedness. Table 12 shows the entries from the collection for which the MSRs could not be correctly assigned by any method.

Trust level	*Number of problematic proteins*	Number of Test Set Proteins	*Test Set Entries: SWISS-PROT ID [SWISS-PROT AC]*[1]
A	1 (3%)	34	PGH1_SHEEP[P05979]
B	5 (22%)	23	ARSB_ECOLI[P37310], DTPT_LACLA[P36574], HLYB_ECOLI [P08716], PTNC_ECOLI[P08187], PTND_ECOLI[P08188]
C	17 (15%)	108	ADT2_YEAST[P18239], ALKB_PSEOL[P12691], B3AT_HUMAN[P02730], CYB_RHOSH[Q02761], CYDA_ECOLI [P11026], CYOE_ECOLI[P18404], FLO1_HUMAN[P41440], PMA1_NEUCR[P07038], RBSC_ECOLI[P04984], S61A_YEAST [P32915], SCAA_RAT[P37089], STE6_YEAST[P12866], [LEP00030], [LEP00130], [LEP00330], [LEP03300], [LEP03303]
C*	3 (13%)	23	GAA4_BOVIN[P20237], GRA1_HUMAN[P23415], GRA3_RAT [P24524]
Sum	26 (14%)	188	

Table 12: Membrane proteins whose MSRs were not correctly predicted by any program

Column one shows the category of trust as set in the collection of transmembrane proteins for individual entries. Trust level A *stands for an available crystal structure,* B *for strong biochemical evidence and* C *for less reliable biochemical evidence.* C* *denotes entries with MSR annotation labelled in SWISS-PROT as highly reliable. The fourth column lists the entries of the test set with their entry name and the accession number in brackets.*

All proteins in the test set, except the LEP0xxxx proteins, are SWISS-PROT entries. The LEP0xxxx proteins are artificial proteins, resulting from fusions of the *E. coli* leader peptidase with itself. Polar residues were introduced in the loops, which led to topologically “frustrated” membrane regions (Gafvelin and Heijne 1994). None of the current methods seems sensitive enough for these subtle changes.

Trust level	*Number of problematic proteins*	*Number of Test Set Proteins*	Test Set Entries SWISS-PROT ID [SWISS-PROT AC]
A	3 (9%)	34	*ATPL_ECOLI[P00844], CB22_PEA[P07371], COX3_PARDE [P06030]*
C	12 (11%)	108	*CITN_KLEPN[P31602], CLC1_HUMAN[P35523], CYOA_ECOLI [P18400], CYOC_ECOLI[P18402], GAB1_HUMAN[P18505], IM23_YEAST[P32897], MDFA_ECOLI[Q46966], ROM1_BOVIN [P52205], [LEP00000], [LEP00003], [LEP00300], [LEP00303]*

[1] The constructed LEP0xxx proteins are not in SWISS-PROT/TrEMBL

C*	16 (70%)	23	*GAA1_CHICK[P19150], GAA2_HUMAN[P47869], GAA3_HUMAN [P34903], GAA5_HUMAN[P31644], GAA6_MOUSE[P16305], GAB2_HUMAN[P47870], GAB3_HUMAN[P28472], GAB4_CHICK [P24045], GAC1_RAT[P23574], GAC3_MOUSE[P27681], GAC4_CHICK[P34904], GAD_MOUSE[P22933], GAR1_HUMAN [P24046], GAR2_HUMAN[P28476], GRB_RAT[P20781], SSRG_RAT[Q08013]*
Sum	31 (16%)	188	

Table 13: Membrane proteins for which only sidedness was not correctly predicted

Column one shows the category of trust as set in the collection of transmembrane proteins for individual entries. Trust level `A` stands for an available crystal structure, `B` for strong biochemical evidence and `C` for less reliable biochemical evidence. `C*` denotes entries with MSR annotation labelled in SWISS-PROT as highly reliable. The fourth column lists the entries of the test set with their entry name and the accession number in brackets.

The *E. coli* leader peptidase in its native form is among the proteins of the test set and is correctly predicted. The LEP-LEP fusions though irritate the prediction methods, especially for the determination of their sidedness.

Other proteins involved in the integration of membrane proteins into the membrane, e.g. SecY and SecE seem to be reliably predicted. Exceptions are the yeast Sec61A (Table 12) and the mitochondrial IM23 (Table 13).

It is not too surprising that proteins within larger membrane complexes are harder to predict since their properties are less constrained by the membrane than by their interaction with other proteins within their complex. In addition, it is not clear if they are integrated into the membrane by the same mechanism. The problems with COX and CYO proteins can possibly be explained this way.

The remaining problematic proteins of Table 12 and Table 13 have in common that they have at least four transmembrane regions. Most of them are ion transporters, which have polar residues within their MSRs. This may have contributed to the difficulty of an automated prediction of MSRs and the sidedness.

Correlation of tools with respect to the number of predicted transmembrane helices

The pairwise comparison of tools yielded the matrix shown in Table 1. It presents the correlation on the number of transmembrane helices. This measure was preferred over the correlation on individual residues since the percentage of membrane-buries helices on proteins veries strongly between proteins (from 0 for soluble proteins, about 17 for transmembrane proteins to alsmost 100% for very short membrane proteins like mature melittin) and little shifts in the prediction have an inpact on the correlation that does not relect the semantics of a transmembrane topology. Hence, to compare the number membrane spanning regions reflects more the similarity of the tools than an analysis that is based on the residues.

	TEST				**HMM-**					**TMHMM**	**TMHMM**
	SET	**ALOM**	**DAS**	**ESKM**	**TOP**	**KKD**	**MEMSAT**	**SOSUI**	**TMAP**	**1.0**	**20**
TEST SET	1,00	0,85	0,83	0,89	0,92	0,90	0,82	0,83	0,78	0,93	0,93
ALOM	0,85	1,00	0,81	0,86	0,89	0,92	0,79	0,83	0,77	0,89	0,90
DAS	0,83	0,81	1,00	0,93	0,88	0,87	0,74	0,79	0,81	0,87	0,87
ESKM	0,89	0,86	0,93	1,00	0,93	0,92	0,82	0,85	0,85	0,94	0,93
HMMTOP	0,92	0,89	0,88	0,93	1,00	0,93	0,83	0,87	0,84	0,96	0,96
KKD	0,90	0,92	0,87	0,92	0,93	1,00	0,82	0,87	0,82	0,96	0,96
MEMSAT	0,82	0,79	0,74	0,82	0,83	0,82	1,00	0,90	0,74	0,85	0,86
SOSUI	0,83	0,83	0,79	0,85	0,87	0,87	0,90	1,00	0,77	0,90	0,89
TMAP	0,78	0,77	0,81	0,85	0,84	0,82	0,74	0,77	1,00	0,85	0,85
TMHMM	0,93	0,89	0,87	0,94	0,96	0,96	0,85	0,90	0,85	1,00	0,99
TMHMM2	0,93	0,90	0,87	0,93	0,96	0,96	0,86	0,89	0,85	0,99	1,00

Table 14: Correlation between transmembrane prediction methods

The table shows the correlation coefficients of tools, calculated on the number of transmembrane helices these predict. The matrix is symmetrical, values higher than 0.95 are marked yellow, the best correlations (>0,9) are marked in green. The leftmost column and the top row list the methods in alphabetical order. This table reflects all sequences from the collection, inclding those of the previously omitted cathegory D.

It can be seen that the hidden marcov models but also the method of Klein et al. are very similar to each other, which is not too surprising since these are also closest to the preseumed correct annotation. The other tools do not repeat each others mistakes, otherwise these would be more similar to each other. This observation explains the success of consensus predictions for transmembrane topology (Nilsson, Persson et al. 2000). It is surprising to see that HMMTOP is closer to TMHMM than to the test set, it has not been analysed to what degree this table changes if only sequences from trust cathegory A and B would have been selected.

Evaluation on seven-transmembrane proteins

In the following analysis, a subset of the available methods to predict the MSRs is applied on a set of 833 G Protein-Coupled Receptors (GPCRs). They are determined by the database reference of SWISS-PROT to the GPCR DB (Horn, Weare et al. 1998). Table 15 shows the prediction results of ALOM 2, DAS, HMMTOP, MEMSAT, TMHMM 1.0 and TMHMM 2.0. MSRs predicted N-terminal of a potential signal-peptide cleavage-point (as annotated in SWISS-PROT) were ignored.

Program	***Number of Proteins with Specific Number of Predicted Membrane Spanning Regions*** ***(Percentage of all GPCRs)*** ***Number without correction of overlap with signal sequence***									
	0 ***MSRs*** ***predicted***	***1***	***2***	***3***	***4***	***5***	***6***	***7*** ***correct***	***8***	***>8***
HMMTOP	0 (0) 0	0 (0) 0	0 (0) 0	0 (0) 0	1 (0) 1	1 (0) 1	27 (3) 25	712 (85) 644	88 (11) 154	4 (0) 8
TMHMM 2.0	0 (0) 0	0 (0) 0	0 (0) 0	1 (0) 1	3 (0) 3	12 (1) 12	98 (12) 96	711 (85) 698	8 (1) 23	0 (0) 0
TMHMM 1.0	0 (0) 0	0 (0) 0	1 (0) 1	0 (0) 0	2 (0) 2	12 (1) 12	98 (12) 96	707 (85) 696	13 (2) 26	0 (0) 0
MEMSAT 1.5	0 (0) 0	23 (3) 23	21 (3) 21	22 (3) 0	14 (2) 16	40 (5) 33	100 (12) 106	551 (66) 531	56 (7) 73	6 (1) 10

ALOM 2	0 (0) 0	6 (1) 6	20 (2) 16	57 (7) 57	176 (21) 170	291 (35) 271	248 (30) 269	29 (3) 35	6 (1) 9	0 (0) 0
DAS	2 (0) 2	0 (0) 0	5 (1) 5	42 (5) 42	212 (25) 194	369 (44) 357	173 (21) 156	24 (3) 62	3 (0) 9	1 (0) 4

Table 15: Performance on G-Protein Coupled Receptors

The columns reflect the number of MSRs that were predicted for single sequence, rows represent the different methods. All GPCR proteins should have seven MSR. The column representing this expected number is coloured in green. For each combination of method and number of MSRs predicted three values are presented. The first value shows the number of proteins predicted to have the respective number of MSRs, for which a predicted MSRs overlapping a signal sequence as annotated in SWISS-PROT is not counted. The second number gives the percentage of the first value of all 833 GPCRs. The third number shows the number of proteins with the respective number of MSRs when no information on signal peptides is taken into account.

One should note that the numbers of MSRs possessed by these proteins have not been biochemically determined. However, GPCRs are generally accepted to have seven transmembrane regions with an extracellular N-terminus.

The Hidden-Markov-Model based methods TMHMM and HMMTOP performed well in this evaluation, reaching 85% correct MSR assignments. MEMSAT's performance was less satisfying with 66%. ALOM 2 and DAS failed completely with only 3% of the 7TM proteins showing the expected number of 7 MSRs. An explanation may be that the membrane topology of GPCRs is rather hard to predict, possibly reflecting a high proportion of polar residues within their transmembrane helices (Ji, Grossmann et al. 1998).

Negative set of soluble proteins

It was mentioned before that MSR prediction methods often predict hydrophobic parts of the N-terminal signal as transmembrane. This error can easily be corrected by an additional run of a tool for signal prediction. What should not happen, though, is that hydrophobic regions within soluble proteins or globular loops of transmembrane proteins are predicted as transmembrane. To evaluate the ability of transmembrane prediction programs in discriminating transmembrane proteins from soluble proteins, all the programs were run on a set of 634 known cytoplasmic or periplasmic soluble proteins derived from SWISS-PROT release 38. Accordingly, not a single MSR should have been assigned to any one of these proteins.

Method	*# FP Proteins*	*# FP MSR (- signals)*	*#entries/100s*
TMHMM 1.0	8 (1.26%)	8 (-1)	37
TMHMM 2.0	8 (1.26%)	8 (-2)	37
SOSUI	19 (2.99%)	27 (-3)	10
ALOM 2	61 (9.6%)	65 (-0)	2438
HMMTOP	70 (11.0%)	84 (-9)	72
Eisenberg	84 (13.0%)	290 (-2)	3993
PHD	120 (18.9%)	212 (-1)	18
KKD	136 (21.5%)	166 (-7)	5835
Tmap	203 (32.0%)	276 (-6)	352
TMpred	350 (55.2%)	434 (-3)	n/a
MEMSAT 1.5	431 (68.0%)	784 (-8)	84
Toppred 2	472 (76.0%)	1198 (-8)	40
DAS	524 (82.6%)	1257 (-9)	5

Table 16: Performance on a Set of Soluble Proteins

The first column presents the method's name, the second the number of proteins that are false positive and the third presents the number of false positive MSRs. The number of signal sequences predicted as transmembrane is stated as a negative number in parentheses behind the total number of false positive MSRs. The fourth column compares the CPU time.

The performance of the majority of these tools in Table 16 seems disappointing. Except for TMHMM (8=1% false annotations) and SOSUI (19=3%) they all have the tendency to strongly over-predict. Even ALOM 2 (61=10%), while predicting only few false positive MSRs on real transmembrane proteins, does not perform so well in this evaluation against soluble proteins.

Summary of the evaluation

In order to compare the performance of current methods for the prediction of MSRs and their sidedness, the tools were run on a set of well-characterised transmembrane proteins.

These served as positive control, while and a set of soluble proteins served as a negative control. This work has shown the performance of some of these tools to be good, while not perfect, in determining the location of transmembrane regions. However, it seems that the determination of the sidedness of transmembrane proteins is not well modelled by most of the tools.

Overall, TMHMM performs best, closely followed by HMMTOP and with, a stronger tendency to overpredict, by MEMSAT. TMHMM especially convinces with its capability to distinguish most reliably between soluble and transmembrane proteins. In addition, for proteins known to be transmembrane it performs best, seconded by MEMSAT. ALOM 2 performed well in confirming transmembrane regions with a very low number of false positives.

In a similar comparison performed by the authors of HMMTOP (Tusnády and Simon 2001) the dominance of TMHMM was confirmed, while HMMTOP is raised to a similar level. This may be due to a difference in the evaluation. It is most interesting to see the number of correctly predicted topologies with disregard of the sidedness to be much higher (best method HMMTOP with 90%, TMHMM 89%) than in this study (HMMTOP 55%, TMHMM 69%).

The here presented evaluation is from the viewpoint of sequence annotation. Another question that could be asked is, what tool best models the real world, and that may just perform less well in this evaluation because of artefacts or different sizes of the respective training- oder evaluation data set. While different accessments have been proposed (Baldi, Brunak et al. 2000), a considerable difference in the here presented ranking of tools is not expected, especially since sequences applied in the training were not used for the evaluation the tools.

It was surprising to see that the simple hydrophobicity analysis or the analysis of the hydrophobic moment are relatively reliable predictors for the MSRs of membrane proteins.

Their main weakness, which they share with other window-based methods, is their lack of specificity for membrane proteins.

No method was able to predict more than 52% of the proteins correctly. However, 86% of the proteins had all their MSRs correctly predicted by at least one method and for 70% a correct prediction that includes the sidedness could be achieved by at least one method.

The results from the prediction of MSRs within GPCRs reveal how varying the performance of the prediction methods can be. It also demonstrates that Hidden-Markov-Models have superiority over sliding-window-approaches in such difficult cases.

Although TMHMM proves very robust against signal sequences, the topology prediction should not be performed without the consultation of signal-peptide prediction methods like SignalP (Nielsen and Krogh 1998; Nielsen 1999; Nielsen, Brunak et al. 1999; Emmanuelsson, Nielsen et al. 2000). TMHMM is the first choice to decide if a protein is transmembraneous or not. HMMTOP is best in determining the sidedness of the protein. When there is doubt in the correctness of the TMHMM prediction, additional evidence like determined protein domains or post-translational modifications should be considered and additional tools should be consulted to derive at a conflict-free consensus. The strongly underpredicting tool ALOM 2 might serve to increase the degree of confidence in individual MSRs, while more sensitive tools can be used to increase the number of candidates for a MSR.

However, all the tools should be only considered as help to biologists to make an educated guess about membrane spanning regions in a protein.

D. Linking sequence motifs with transmembrane annotation

The evaluation of membrane predictors has shown that the interpretation of the results remains error prone. Also it was shown that the number of transmembrane proteins whose topology could potentially be correctly predicted in an automated way could be considerably increased by using multiple methods, if there was an additional evidence for a single method to be the most promising.

Some programs have addressed this issue. Toppred2 asside hyrophobicity analysis applies additional information from the positive-inside rule, PHD and TMAP utilise sequence similarity. Similarly one could investigate if the utilisation of secondary structure prediction, the prediction of coiled-coils or of glycosylation sites could potentially yield a better picture of the transmembrane topology. This information could help to select the overall most-plausible topology of a membrane protein from the predictions made by different tools.

The additional sources of sequence information addressed in this work are protein domains described by the member databases of InterPro (Apweiler, Attwood et al. 2001; Mulder and Apweiler 2001; Kanapin, Apweiler et al. 2002). These member databases store sequence motifs (patterns) to which a biological property is assigned. The content of the here described database 'TransMotif' is generated by an automated merging of the transmembrane information from SWISS-PROT with protein motif information from InterPro. The latter can be understood as a unification of protein domain databases. TransMotif stores the extend to which the protein domains are specific to regions of a protein that reside in a certain subcellular compartment, i.e. that are transmembraneous, extracellular or intracellular.

When applied to an otherwise uncharacterised sequence of a transmembrane protein, the regions of such a protein domain described in TransMotif should constrain the protein's topology. Hence it could be utilised as an additional selection criteria for topology predictions. Means to implement this selection are presented in section E. Per se, the information in TransMotif represents partial knowledge on transmembrane topology.

1. Introduction

A first approach towards linking protein domains and sequence annotation was described in a paper of Wolfgang Fleischmann in the SWISS-PROT group (Fleischmann, Möller et al. 1999). For that work an analysis of the correlation of SWISS-PROT (Bairoch and Apweiler 2000) keywords, description lines and comment lines with PROSITE (Hofmann, Bucher et al. 1999) entries was determined. The effort led towards a manually verified set of rules, that *position-independently* characterise novel protein sequences and are utilised in the automated annotation of proteins in TrEMBL (Bairoch and Apweiler 2000).

About two years later, TransMotif presents an extension to include *position-dependent* annotation as found in SWISS-PROT feature lines that previously have not been covered. Another approach towards a position-independent automated generation of rules for protein annotation was presented in mid-2001 (Kretschmann, Fleischmann et al. 2001).

While TransMotif was created in a totally automated fashion, a precursor to this work (unpublished) was created manually. Back in January 1999 the description of PROSITE patterns that is given in PROSITEDOC (Figure 26) was manually formalised, without any further verification on SWISS-PROT sequences.

```
ID    PROSITEDOC@PDOC00251
DE    Membrane attack complex components / perforin signature
DR    PROSITE; PS00279; MAC_PERFORIN.
PL    is_transmembrane.
PL    probable(transmembrane(X,Y,-1):-
PL                        matches('PROSITE'@'PS00279',X,Y)).
 //
ID    PROSITEDOC@PDOC00878
DE    Bacterial type II secretion system protein C signature
DR    PROSITE; PS01141; T2SP_C.
PL    is_transmembrane.
PL    num_tm_regions(1).
PL    probable(topology('CYTOPLASMIC','PERIPLASMIC').
```

Figure 26: Excerpt from a manual formalisation of PROSITEDOC

The figure gives an example for two out of 144 such formalisations performed on the basis of the statements made in the PROSITEDOC file. The :- token reads as „holds if“. The function „matches“ describes a match to the respective pattern. The first description states that if a protein sequence was matched by the PROSITE pattern PS00279 from position X to Y, then the region X to Y is known to be transmembrane.

The description of the second entry tells that from a match of the patter PS01141 an in-out sidedness can be deduced.

This manual set of rules was not very expressive in the actual location of transmembrane regions in dependence of the sequence match to PROSITE. However, it gave an impression on parts of the protein topology. The most imminent problem with the set of rules were:

- The manual effort cannot be computationally updated.
- No information is given on the number of sequences from which the description was induced to serve as a measure of impact and reliability.
- The information in PROSITEDOC seems sparse with respect to information on transmembrane topology.
- No information on other protein domain databases was presented.

These problems with the manual approach motivated to automate this work and to implemented it in collaboration with Özgün Babur, a summer student at the EBI.

Automated approach by induction from InterPro matches to SWISS-PROT

Since the earlier paper appeared, two things have happened that made this work possible. Firstly, SWISS-PROT and TrEMBL are now accessible in a relational database and secondly, the integration of protein domain databases in the InterPro project (Apweiler, Attwood et al. 2001) allowed an extension to PFAM (Bateman, Birney et al. 2000), PRODOM (Corpet, Servant et al. 2000) and PRINTS (Attwood, Croning et al. 2000) with almost no extra costs within the same database system.

This section describes the following parts of the work on TransMotif:

1. For every protein domain list all membrane spanning regions (MSRs) in SWISS-PROT that are closest to the domain.
2. Investigate for which protein domains the distances to transmembrane regions seem conserved throughout SWISS-PROT. This may be regarded as an annotation of the

domains, since it could be converted to text and stored as an addendum to PROSITEDOC.

3. For all domains show what impact these rules have when applied to the automated transmembrane annotation of TrEMBL.

2. Methods

InterPro

The InterPro database is an integration of protein domain databases. For this work the provided list of matches of protein domain descriptions to SWISS-PROT and TrEMBL was utilised. InterPro member databases that are used in this study are Pfam, PRODOM, PROSITE and PRINTS. A match of an InterPro member database is described with the attributes "from", "to" (positions on the sequence) together with the accession numbers of the SWISS-PROT entry and of the InterPro member database.

The vast majority of matches of entries in Pfam, PRODOM and PROSITE to SWISS-PROT can be described by such a quadruple. Multiple matches on the same protein sequences would then indicate a repeat of a functional unit within the protein. However, this does not hold for PRINTS. The PRINTS database characterises a protein family by a set of conserved motifs, for which only a subset is required to match a sequence for its classification. This matching subset is referred to as a fingerprint.

For this work, PRINTS motifs are distinguished by enumerating them as they appear from the N- to the C-terminus of a sequence matched. Problems arise for approximately 17% of the PRINTS entries, which feature partial matches and the enumeration of motifs was therefore not leading to an unambiguous identification. PRINTS entries with a varying number of motifs were not considered for this study. It should be emphasised, that PRINTS motifs are distinguished within the PRINTS database, it is only during the transfer to InterPro, that the distinction is lost.

To simplify the terminology, it is in the following not distinguished between matches to protein domains, family assignments or the motif of a fingerprint. These are all referred to as *matches*.

Determination of relative positions of transmembrane regions

InterPro gives access to the start and the end of a domain match within the protein sequence. Except for PRINTS, with its concept of multiple motifs to be contributing to a family assignment, there is no further information on internal states available within InterPro. The focus is on the relative positions of transmembrane regions to these motif matches. These are stored relative to the beginning and the end of the motif (Figure 27).

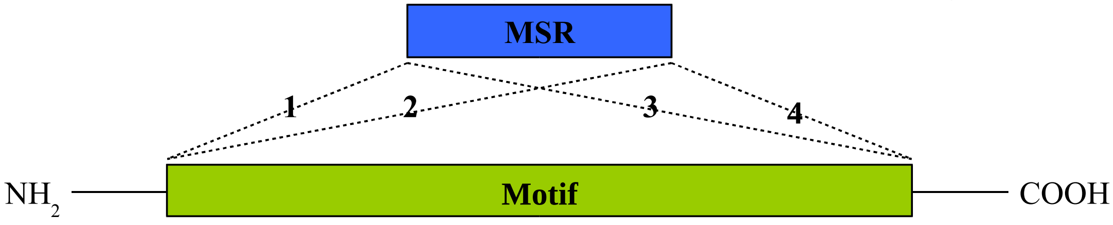


Figure 27: Relative distances between motif and transmembrane segment

The figure shows a sequence for which both a membrane spanning region (MSR) and a protein domain (motif) was determined. Within this analysis, the location of transmembrane segments is described in dependence of the begin and end of a motif. This dependency varies as the same motif may vary in length in different sequences. For greater accuracy both the dependency of the MSR positions relative to the N-terminal and the C-terminal end of the motif are stored. Hence, the following four values are needed for a complete description:

1 - msts : from start of match to start of transmembrane region

2 - mste : from start of match to end of transmembrane region

3 - mets : from end of match to start of transmembrane region

4 - mete : from end of match to end of transmembrane region

The basis for the analysis is a textual representation of Figure 27, implemented as a table in the database system (Table 17). The table joins the information from InterPro and SWISS-PROT entries of transmembrane proteins. It shows the number and positions of residues between begin or end of a motif relative to any transmembrane regions of a protein, the four distances described in Figure 27.

ID	SP ACC#	IPRO ACC#	SUB	MSTS	MSTE	METS	METE	POS_FROM	SEG_START
80768	Q47825	PS50286	0	4	24	-359	-339	9	13
80769	Q47825	PS50286	0	25	45	-338	-318	9	34
80770	Q47825	PS50286	0	77	97	-286	-266	9	86
80771	Q47825	PS50286	0	118	138	-245	-225	9	127
80772	Q47825	PS50286	0	144	164	-219	-199	9	153
80773	Q47825	PS50286	0	183	203	-180	-160	9	192

Table 17: Transmembrane regions of SWISS-PROT entry Q47825 relative to motif PS50286

The first column enumerates all rows, the second and third columns store the SWISS-PROT accession number of the protein and the accession number of the motif in InterPro, the column sub differentiates multiple occurrences of a the same motif. POS_FROM is the start point of the motif and SEG_START is the starting point of the transmembrane segment within the sequence. The column headers are four-letter abbreviations as visualised in Figure 27, MSTS stands for the number of residues (from the N- to the C-terminus) from the start of the motif (MS) to the start of the transmembrane region (TS), MSTS equivalently stores the distance in residues to the end of the transmembrane region. METS and METE are determined equivalently for the end of the motif.

Unification of MSR positions and the creation of reference MSRs

For each InterPro entry, the smallest distinctive region were determined that embrace at least one transmembrane segment on the motif. It is relevant to investigate both dependencies from both ends of the motif. Some protein domains may vary in length and as a consequence one end may correlate better than the other with transmembrane regions.

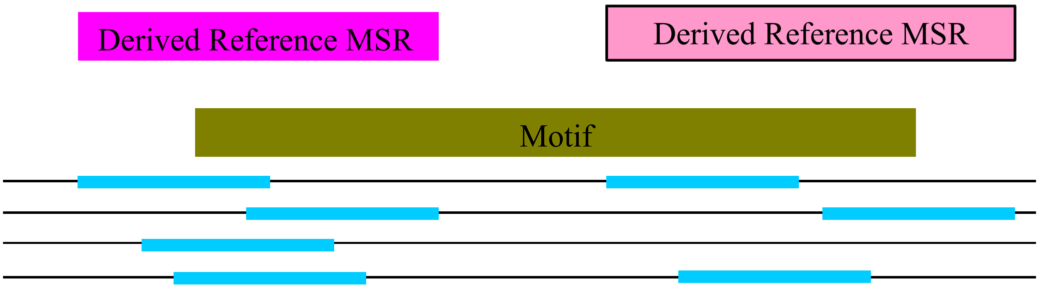


Figure 28: MSRs from multiple sequences relative to a common motif

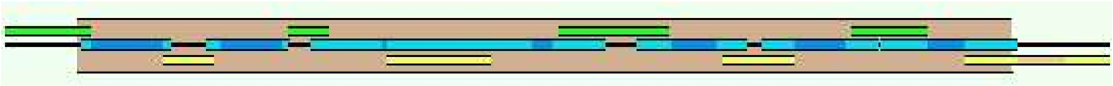
The figure displays the determination of reference transmembrane regions relative to a common motif on the proteins. All sequences feature a MSR on the N-terminal (left) and *all but one also have a MSR on the C-terminal end of the motif. From these positions two reference MSRs have been defined that include the outermost positions of overlapping MSRs. The N-terminal MSR (dark purple) is perfect (subsequently called reliable) as all sequences have a MSR (light purple) in that region, while the C-terminal domain is not reliable since the third sequence does not show a MSR in that designated region.*

Those regions that describe the location of membrane spanning regions in dependency of a match to a specific protein domain, in whatever protein sequence the match occurs, are referred to as *reference regions*. Figure 28 schematically demonstrates how these regions are determined. All reference regions are enumerated and stored a separate table. The average and standard deviation of the four distances (msts, mste, mets and mete) were calculated. The regions with the smallest standard variation would then be suggested to be applied as constraints for the automated annotation.

Determination of the status of each residue within and around a motif

Besides the description of regions in dependence of the position of sequence motifs, the location of individual residues were investigated. It was found, that matches to protein domain databases can reliably describe the extend to which if individual residues lie within the outer cellular compartment (extracellular, periplasmic, lumenal, mitochondrial intermembrane), are transmembrane or lie within the inner compartment (cytoplasmic, stromal, matrix). The information was again derived from the SWISS-PROT FT lines. All residues within a motif and 20 residues around the borders of a motif were analysed. Again, any position relative to the start or and of the motif that is always assigned to a specific domain will serve as a constraint for the automated sequence annotation of transmembrane proteins.

The information from the residue-based approach is much more detailed than the region-based approach. This is visualised in Figure 29.



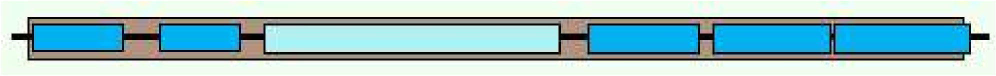


Figure 29: Representation of constraints

The figures visualises the location of individual residues within a protein motif, here described by the PFAM entry PF00003 (G Protein-Coupled Receptors family 3). The figure first shows the residue-based information, then the minimal regions known to contain MSRs. The intensity of a colour represents if the assignment as perfect for all sequences (dark) or if there were exceptions (light). The area described by the motif is represented by a larger light violet box encapsulating all the visualised constraints. In the first graph, the top line represents the outer domain, the middle line is transmembrane and the bottom line represents the inner domain. The second graph shows the regions that were found to embrace transmembrane segments. The web presentation offers more detailed information for each graph.

Figure 29 visualises both a region-based (bottom) and residue-based (top) analysis. It can be seen how a varying degree of certainty is displayed in either analysis and how much more information the residue-based information contains. From the latter, all expected seven MSRs for the here displayed GPCR could be derived, while the focus on regions only shows five reliable reference regions that may act as constraints for sequence annotation.

Automated Annotation of Transmembrane Proteins

Finally, the expected gain in incorporating reference regions and residues as constraints in sequence annotation is evaluated. For this purpose the annotation of transmembrane predictors was compared with the constraints determined as previously explained. The basis of this analysis was, created in a separate effort, a database with results from a variety of prediction methods for all SWISS-PROT and TrEMBL entries. These predictions are compared with the information available from the region-based constraints. Those constraints that could not be satisfied by the predictor TMHMM, because of its inherent tendency to underpredict, were satisfied by using the annotation from other less careful predictors. The preference for TMHMM results from the evaluation of methods in section C, in which TMHMM was found to be the best general prediction method.

This evaluation of constraints focuses on region-based constraints for transmembrane regions. For the automated sequence annotation this has the consequence, that only false negative MSRs are pointed out by this approach, while a general problem of transmembrane annotation is overprediction. However, TMHMM was found to underpredict membrane-spanning regions. The constraints point to MSRs missed in the annotation that can be completed by the other methods. No effort was yet put into creating constraints that also determine extracellular and intracellular moieties of proteins, which would also help to overcome overprediction.

3. Results

In September 2000, SWISS-PROT stored 14133 transmembrane proteins of about 95000 in SWISS-PROT annotated with the keyword 'Transmembrane'. 11503 entries were not-fragmented.and form the basis for this analysis.

This set is matched by 1098 different InterPro entries, 574 of which are found only in transmembrane proteins. For these 574 InterPro entries constraints were determined as previously described. The total number of transmembrane proteins in SWISS-PROT with domains described by these InterPro entries is 8544 (Table 18).

Number of TM proteins (all)	14133
Number of TM proteins (+location, non-fragment)	11503
Number of InterPro entries	3210
Number of InterPro entries matching only transmembrane proteins	1098
Number of InterPro entries matching only transmembrane proteins	574
Number of TM proteins (+location, non-fragment) that have transmembrane-specific InterPro entries matching ⬋ refered to as *useful proteins*	8544

Table 18: Description of data serving as input for the analysis

The left column describes a data source and the right column lists the respective number of entries.

Comparison of InterPro member databases

In order to determine which InterPro member databases were most descriptive for transmembrane proteins, this section presents numbers of InterPro motifs and SWISS-PROT entries that are described with derived transmembrane constraints in the TransMotif database (Figure 30 and Figure 32).

One important issue is reliability. One must ask how many SWISS-PROT entries are needed to confirm an individual constraint. For instance, some constraints are created on the basis of no more than a single protein in SWISS-PROT, with only TrEMBL entries being additionally assigned to the InterPro entry. Since neither InterPro nor the member databases themselves provide any information on the reliability of a match of a sequence to a protein domain, it was not tried in this study to develop any statistical means to derive sensible suitable thresholds. Hence, the user should decide upon the minimum number of confirmations necessary before accepting a constraint. The figures below draw the numbers of matches or the number of motifs in dependence of the number of confirmations requested for a reference region in SWISS-PROT to be used as a constraint. The member databases are distinguished by the coulouring.

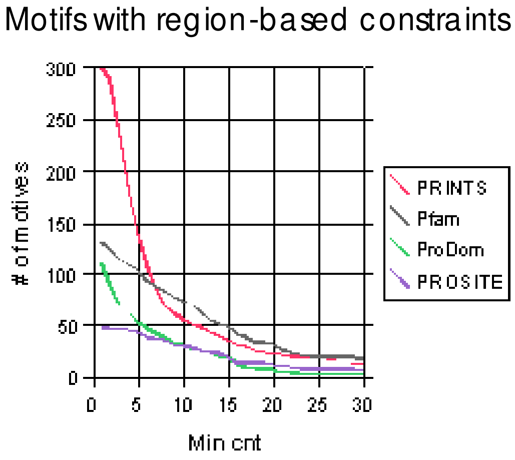


Figure 30: Motifs with region-based constraints

The figure visualises the dependency on the number of confirmations that are requested for the application of a reference region as a constraint. The X-axis represents the number of confirmations (Min cnt) minimally required for a constraint, the Y-axis represents the number of motifs of a member database from which constraints could be derived. The member databases are distinguished by colours, the top line is PRINTS, then taken over by Pfam for min cnt larger than 7.

Figure 31 and Figure 33 equivalently show the number of rules that can be created based on the data supplied by member databases. This is not linearly depending on the number of motifs since a single domain or family assignment may be related to multiple MSRs. The increase of min cnt also increases the reliability; but the number of constraints and annotated motifs decrease.

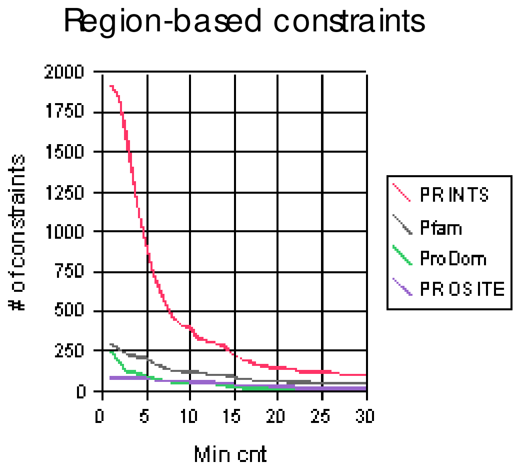


Figure 31: Region-based constraints

The X-axis again represents the number of confirmations of constraints, the Y-axis respresents the number of region-based constraints. Colours differentiate the member databases, PRINTS (red) is the top line.

When individual residues rather than stretches of membrane spanning regions are investigated, the information becomes more explicit. The single-linkage clustering on membrane spanning regions to determine the reference regions may artificially extend the reference beyond an optimal range for the utilisation as a constraint. On a residue level this is not a problem and the number of InterPro entries (motifs) for which constraints can be derived is increased.

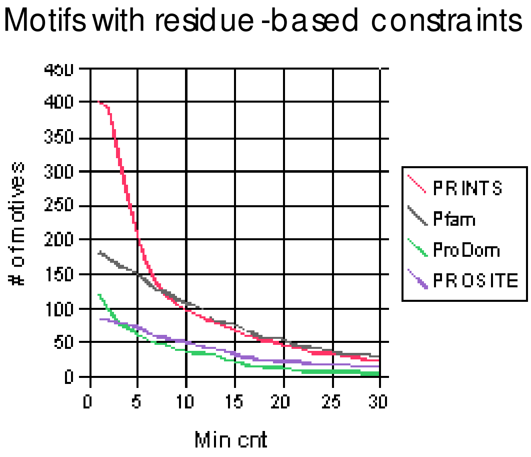


Figure 32: Motifs with residue-based constraints

This graph is equivalent to Figure 30 on the basis of residue-based constraints rather than region-based constraints. Again PRINTS and PFAM are practically equivalent for constraints with at least 7 confirmations in SWISS-PROT.

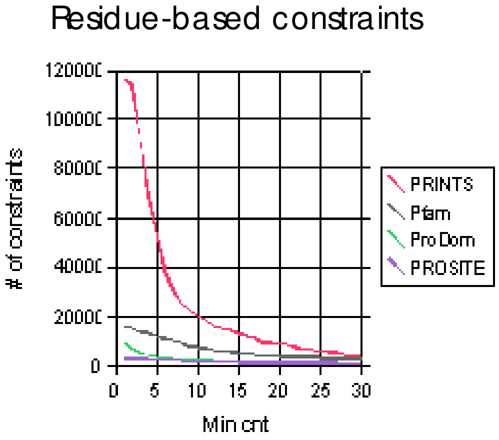


Figure 33: Residue-based constraints

The graph is equivalent to Figure 31 except for the referral to individual residues rather than whole transmembrane regions. Again, PRINTS is the best method.

These figures demonstrate the superiority of PRINTS to describe transmembrane regions. This is presumably (supported by Table 19 and Table 21) mainly due to the direct access of fingerprints via InterPro where more complicated means would be necessary for the other member databases. Due to the proximity of some PRINTS motifs within sequences, the reported number of constraints is expected to be partially contributed to by redundant constraints, i.e. a description of the same MSR location by multiple motifs.

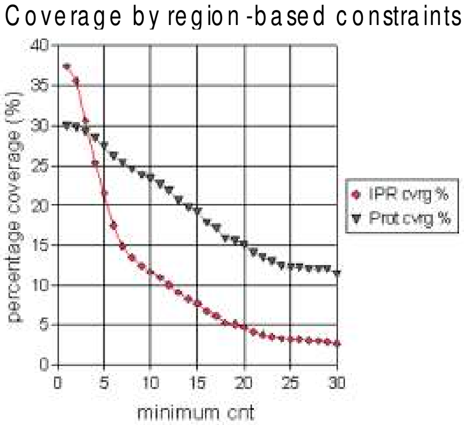


Figure 34: Coverage of by region-based constraints

The figure visualises the percentage of motifs under an InterPro accession number (IPR cvrg) from which region-based constraints with requested number of confirmations on SWISS-PROT (minimum cnt) could be derived. Also the percentage of transmembrane proteins in SWISS-PROT (Prot cvrg) for which constraints are available is shown. The X-axis represents the minimal number of confirmations of SWISS-PROT for constraints, the Y-axis represents the percentage. It can be seen that the percentage of sequences drops slower than the number of InterPro entries.

Of interest was the coverage of the constraints on the transmembrane SWISS-PROT sequences and on the InterPro entries and is visualised in Figure 34 and Figure 35.

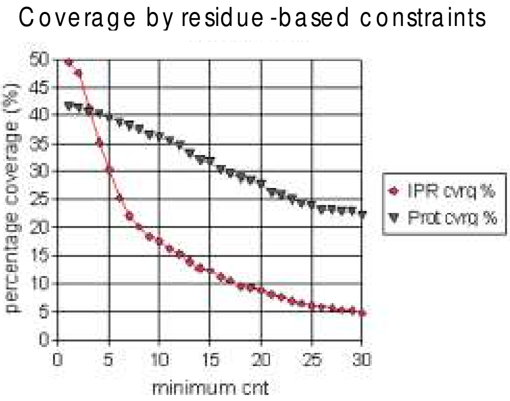


Figure 35: Coverage by residue-based constraints

This figure is the equivalent to Figure 34 for residue-based constraints.

From these graphs, it can be seen that a few protein domain database entries have a big overall impact on the coverage because they match so many proteins. Therefore, even with a high level of requested reliability one can still get many transmembrane annotations confirmed. Also the residue-based constraints seem at least 25% more effective on sequence level than the region-based constraints.

Impact on automated annotation

With five as a minimum number of confirmations required, it is investigated how many transmembrane proteins miss transmembrane regions when these are annotated with TMHMM. At this threshold, Table 19 shows that PRINTS offers the largest number of constraints for MSRs while Pfam covers the largest number of proteins. For PRODOM the smallest number of constraints per protein was made and PROSITE performs surprisingly well with a coverage of proteins a little below PRINTS and a number of constraints twice as large as PRODOM's.

Database	#Constraints	#Proteins
PFAM	34445	10591
PRINTS	52052	4085
PROSITE	14187	3712
PRODOM	7934	3222

Table 19: Performance of individual member databases to describe the positions of MSRs

The first column lists the InterPro member databases, the second column (#constraints) shows the number of both N- and C-terminal constraints formulated on the basis of the respective InterPro member database. The third column shows the number of proteins to which these could be applied. This table reflects the influence of individual databases on the automated annotation. PRINTS is most specialised and Pfam has the highest impact.

Database	Average length of constraint	Standard deviation of length
PRINTS	25.13	6.24
PRODOM	30.53	18.03
Pfam	36.77	21.18
PROSITE	40.71	16.87

Table 20: Average length of constraints in sequences of SWISS-PROT and TrEMBL

The first column lists the InterPro member databases, the second column shows the average lengths of region-based constraints, The third column shows the standard deviation of these lengths. Smaller values are better, PRINTS is the most constraining (smallest lengths) and the most consistent (smallest deviation).

The relevance of these constraints varies. As seen in Table 20, PRINTS is the most specific in the positions and from Table 21 can be derived that it also contributes the highest number of confirmed constraints.

Database	Confirmed by TMHMM 2.0		Confirmed by other methods		Unconfirmed	
	Constraints Tolerance 3 or 9 AA	Proteins 3 / 9	Constraints 9	Proteins 9	Constraints 3 / 9	Proteins 3 / 9
Pfam	15985 / 19116	5846 / 6844	2633	1465	18460 / 12696	6404 / 4720
PRINTS	39926 / 44203	3598 / 3623	2285	648	12126 / 5564	1633 / 466
PRODOM	2908 / 3724	1171 / 1552	459	327	5026 / 3751	2269 / 1793
PROSITE	10446 / 11206	3032 / 3053	362	180	3597 / 2475	1090 / 710

Table 21: Effect of the constraints on the automated prediction of membrane helices

The table has four columns, all except the first are split in a left half with a number for constraints and a right with a number for proteins. The first column lists the InterPro member databases from which information was derived, the remaining columns describe to what extend the information was confirmed. Constraint here reads as a match a of a sequence to a reference region that is used as a constraint. A single protein can have multiple matches, therefore the number of proteins to which constraints from the respective domain database apply is also listed. A tolerance (allowed extend beyond constraint) of 3 or 9 residues was allowed for the acceptance of a MSR prediction. The second column displays the number of constraints that could be confirmed by TMHMM; the second allowed any method to contribute MSRs, the last lists the number for constraints that could not be satisfied.

Only those constraints that are left fulfilled that are not satisfied by any of the programs ALOM 2 (Klein, Kanehisa et al. 1985)*, DAS* (Cserzo, Wallin et al. 1997)*, HMMTOP* (Tusnády and Simon 1998)*, MEMSAT* (Jones, Taylor et al. 1994) *or TMHMM 1.0* (Sonnhammer, Heijne et al. 1998)*. TMHMM 2.0 contributed 102879 membrane spanning regions as the default method, the others in sum contributed an additional 5739 (5.3%).*

The current annotation in SWISS-PROT is strongly influenced by the MEMSAT prediction method (Jones, Taylor et al. 1994) while in this analysis the constraints are compared with the the prediction of TMHMM. To ensure the large number of unconfirmed constraints is not simply due to a slight shift of the transmembrane region, a deviation of no more than nine residues to either side of the constraint was accepted as a match. This is about half the expected length of a transmembrane helix.

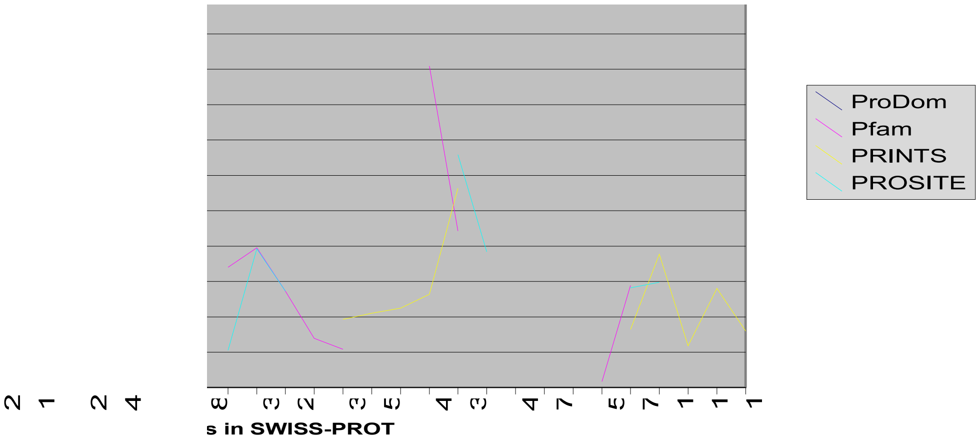


Figure 36: The percentage of constraints not be fulfilled by any of the transmembrane prediction tools

The graph visualises the percentage of constraints that could not be fulfilled by predictions dependence of the confirmations in SWISS-PROT. Methods are drawn in different colours, the lower the percentage the better a method is thought to be.

It still remains to be investigated, if the high percentage of unconfirmed constraints of Pfam and especially PRODOM is rather due to a higher coverage of sequences or due to problems with the accuracy the localisation of the MSR. From Figure 36 it can be seen that Pfam and PRODOM show the highest percentage for constraints derived from domains with a low number of confirmations of SWISS-PROT, while for PRINTS and PROSITE the source of unconfirmed predictions seems independent from the number of confirmations a constraint has. This figure suggests that one should not accept any constraint from these databases that has not been confirmed at least 10 times in SWISS-PROT. PRINTS does perform very well, only about 10 percent of the constraints could not be confirmed, and most importantly this seems independent from the number of confirmations PRINTS-based constraints have.

4. Summary and Conclusion

It was demonstrated that the use of protein domain information for automated annotation is feasible beyond position-independent information. The links can be seen both as constraints to an automated annotation of protein sequences, and as an automated improvement of the annotation of the domain databases. This technique can be applied when evidence leads to the exclusion of certain properites within a parts of the peptide was confermed or conversely the existence of a property is known but the actual location cannot be explicitly stated. Constraints represent partial knowledge and implement the validation of predictions from independent sources. SWISS-PROT already uses constraints. Greated than ('>') and less than ('<') symbols are used in its annotation in cases when the localisation of a feature can be only vaguely described.

An interesting aspect to the analysis is the annotation of protein fragments for which the respective transmembrane region lies on the border of the available sequence, but a link to an InterPro member database could be found. With current standard technology, this information would not be available since alignments do not visualise sequence annotation. The method to propagate sequence features (Velds 1999) would not be applicable to non-conserved regions.

The transmembrane regions documented in SWISS-PROT are the result of a manually verified consensus of different transmembrane prediction methods, sequence similarity, and also biochemical evidence. Of the 1323 N-terminal references with a minimal confirmation in SWISS-PROT of 5, a subset of 276 references could be verified as they match sequences of the test set of well characterised membrane proteins that was presented in the previous section B. The remaining links must still be regarded as predictions. Though this problem is inherent to transmembrane predictions for which only limited experimental evidence exists, other FT lines do not have this problem. The transmembrane annotation was chosen as a proof of principle. The approach can be easily adapted to other FT lines and hence to other protein properties..

The high number of unconfirmed constraints is surprising but would still be in accordance with results from the program's evaluation in from section C. The best programs annotated less than 80% of proteins correctly and constraints from PRINTS left 10% unconfirmed. The low rate of unconfirmed constraints in PROSITE and PRINTS is presumably due to the comparatively short and conserved nature of the motifs that allowed a high accuracy for the specification of a linked MSR's position.

The use of Pfam from within InterPro was shown to be problematic in that for matches that partially fall under a threshold a gap is introduced as the sequence falls out of and back into the profile HMM. In InterPro this is not machine-distinguishable from multiple matches to the pattern.

E. Integration of Prediction methods for the automated annotation of transmembrane proteins

The previous section explained a concept for the automated generation of constraints for automated sequence annotation. The higher sensitivity of residue-based versus region-based constraints and different properties of predictors of topology predictors leads to potentially conflicting annotation for a single entry. This the adaptation of a novel method for the integration of these sources of information.

1. Representation of the Biological Knowledge and the Analyser Beliefs

This section presents biological background for a selection of conflicts and their formal representation in REVISE (section E). A prediction of transmembrane topology is presented as a set of facts, which syntactically very much resembles the original representation as FT lines in SWISS-PROT.

The numbers denote the respective start and end of a specific region of sequence described as a feature. A fact's first argument is the source of information, if available the evidence tags would be used. The second argument is the entry's accession number in order to identify the entry the statement was derived from.

```
ft(swissprot,p17353,transmem,31,50).
ft(swissprot,carbohyd,20,20).
ft(swissprot,domain,1,22,extracellular).
```

Besides the localisation of transmembrane regions it is important in what direction the protein is integrated into the membrane, referred to as the membrane protein's sidedness. This is denoted by the predicate

```
    topology(Source,Accession,Domain1,Domain2).
```

It describes the direction of the first transmembrane helix.

Post-translational modifications are subject to individual residues of a peptide sequence only. The two positions will hence be identical. The only exceptions to this are disulphide bridges, which connect two residues.

```
ft(mod_res,5,5,phosphatation).
ft(carbohyd,10,10).
ft(disulfid,66,99).
```

For the conflict resolution, rules have been created. Some only serve to interpret the SWISS-PROT annotation, e.g. the rule *in_or_out* derives if a specific residue is in the inner or outer domain in according to SWISS-PROT annotation. Other rules code for potential conflicts, these constrain the automated annotation further to what is biologically meaningful.

A disulphide bridge links to residues within the same compartment only:

```
← ft(Agent,Acc,disulfid,Pos1,Pos2),
  in_or_out(Agent,Acc,Pos1,D1),
  in_or_out(Agent,Acc,Pos2,D2),
  D1 ≠ D2.
```

A residue's glycosylation is established in the outer domain only:

```
←  ft(Agent,Acc,carbohyd,Pos,Pos),
   in_or_out(Agent,Acc,Pos,D),
   D ≠ outer.
```

It must be checked certain other modifications are made to residues of the inner compartment:

```
← ft(Agent,Acc,Modification,Pos,Pos,_),
  member(Modification,[lipid,mod_res]),
  in_or_out(Agent,Acc,Pos,D),
  D ≠ inner.
```

Matches with domain databases and derived knowledge

The statement that a protein sequence contains a domain specified in a protein domain database is made with the predicate *matches*:

```
matches(ProtAccession, DomainAccession, From, To).
```

A sequence's match to a protein domain can be dynamically determined for a novel sequence by an analyser or it is already stated in the TrEMBL entry. To improve the efficiency of the revision process, this information is used to present only those rules to REVISE that have a chance to fire.

The documentation to the PROSITE entry PS00538 "Chemotaxis Transducer" describes well how the protein domain fits with the protein topology. Below the information from the PROSITEDOC entry with respect to transmembrane topology is first verbosely repeated and then formally stated as available for all PROSITEDOC entries:

1. The protein matched by the PROSITE pattern PS00538 has two transmembrane regions:

   ```
   num_tm_regions(ProteinAcc,2):-
               matches(ProteinAcc, prosite, ps00538, X, Y).
   ```

   More closely to the formalism it reads "it is true that the protein has two transmembrane regions if it is matched".

2. The protein has an N-in sidedness:

   ```
   topology(ProteinAcc,cytoplasmic, periplasmic)
                         matches(ProteinAcc, prosite, ps00538, X, Y).
   ```

3. If the pattern matches then it does so in the second loop which spans the inner domain.

   ```
   loop(ProteinAcc, cytoplasmic,X,Y,2) :-
                        matches(ProteinAcc, prosite, ps00538, X, Y).
   ```

   The predicate ft could not be used to express this, since only a fraction of the loop is described, no concrete boundaries for this intracellular loop could be derived.

Also the statement ‘is_transmembrane’ is available to express that the protein spans the bilayer and the predicate ‘transmembrane’ expresses that a range of residues lies within the bilayer.

Constraints representing Knowledge specific for transmembrane proteins

The rules presented before only looked at individual revisables and their consistence with knowledge independent from the transmembrane prediction process. The following rules compare transmembrane predictions with each other.

All methods must agree on a protein being transmembraneous:

```
← ft(Agent,Acc,transmem,_From,_To),
   not is_transmembrane(Agent2,Acc).
```

If two transmembrane regions are predicted to overlap then neither border should differ more than four residues from the border of the other predictions:

```
←  ft(Agent1,Acc,transmem,From1,To1),
  ft(Agent2,Acc,transmem,From2,To2),
  (From1>From2,From1<To2 ; To1>From2,To1<To2),
  (abs(From1-From2)>4;abs(To1-To2)>4).
```

The length of a transmembrane region is limited:

```
← ft(_Origin,_AccessionNumber,transmem,From,To),
   X is To-From,X≤15.

← ft(_Origin,_AccessionNumber,transmem,From,To),
   X is To-From,X>25.
```

Further heuristics, like the positive-inside rule (Heijne 1986) have been implemented and can be used to support or to refuse a prediction.

2. Analysers for transmembrane topology prediction

A variety of sources has been integrated at this stage. Based on the database ENZYME (Bairoch 1996) there is AddEnzyme serving as an annotation tool. With the aid of the InterPro project (Apweiler, Attwood et al. 2001), the databases PROSITE (Hofmann, Bucher et al. 1999), PFAM (Bateman, Birney et al. 2000), PRINTS (Attwood, Croning et al. 2000) could be added. Finally, applications such as TMHMM (Sonnhammer, Heijne et al. 1998) (prediction of transmembrane proteins) or NNPSL (Reinhardt and Hubbard 1998) (prediction of the sub-cellular location) are potential sources of protein annotation. Figure 37 visualises the data flow of entries between the collaborating analysers and dispatchers. The persistent and temporal knowledge differs from analyser to analyser for several reasons. Most importantly, this is to keep the system as small as possible in order to avoid unforeseen conflicts. Secondly, this has the technical aspect that with a lower number of rules in the system the computational effort is decreased.

In addition, the types of constraints differ between applications. In the here presented integration of membrane prediction methods many different checks on mutual compatibility are performed. This process involves constraints for protein domain information that are dynamically loaded to the system. In contrast hereto, the application of the later described predictor of receptor-G protein coupling (section 3) would rely on the prior derived protein domain information and topology prediction. While at this stage the constraint is only on the topological information, an additional masking could be applied to domain matches that are known not to be functioning as potential binding sites.

For this thesis, only proteins in the cytoplasmic or mitochondrial inner membranes are covered, because the incorporated signal prediction SignalP (Nielsen and Krogh 1998; Nielsen 1999) does not allow a prediction of the final subcellular location. This prediction is performed by PSORT (Nakai and Horton 1999). If PSORT was integrated, together with information on the respective organism and sequence similarities, this could be addressed. EDITtoTrEMBL was designed for such extensions, and a predictor for the topology of outer membrane proteins, such as (Diederichs, Freigang et al. 1998), could be incorporated to allow the topology of these proteins to be automatically annotated.

If available, the information presented by protein domain databases is most valuable for a first characterisation of proteins and should therefore be requested first. Dependencies between participating agents can be dynamically derived (Möller, Leser et al. 1999) or otherwise declared (Gaasterland, Maltsev et al. 1994).

Three dispatchers are involved. The first controls the whole process, the second integrates the domain databases and the third integrates the prediction methods.

We focus on the annotation of transmembrane proteins by interpreting the results of a set of different programs for the prediction of membrane spanning regions. The programs make different strength and weaknesses and differ in their quality for different protein families.

A match of a domain's pattern in a protein sequence is associated with a probability by which a random sequence might contain it. Similarly, other tools provide reliability factors, which the analyser uses as a basis to determine the probability for the correctness of the information.

The basic assumption underlying the approach of integration is that if a protein's features are equally determined by different methods, then these should most likely be correct. Consequently, the number of proteins to be annotated and the reliability of the derived annotation should be possible to increase since more information is available.

The transmembrane dispatcher cannot revise information provided by protein domain databases. When constraints have been derived from protein domains for the automated annotation as described in section D then these are fully trusted. They serve as a referee to resolve ambiguities and to avoid wrong annotation. This should be changed as soon as probabilities reflecting reliability are available for the assignments of protein domains to protein sequences.

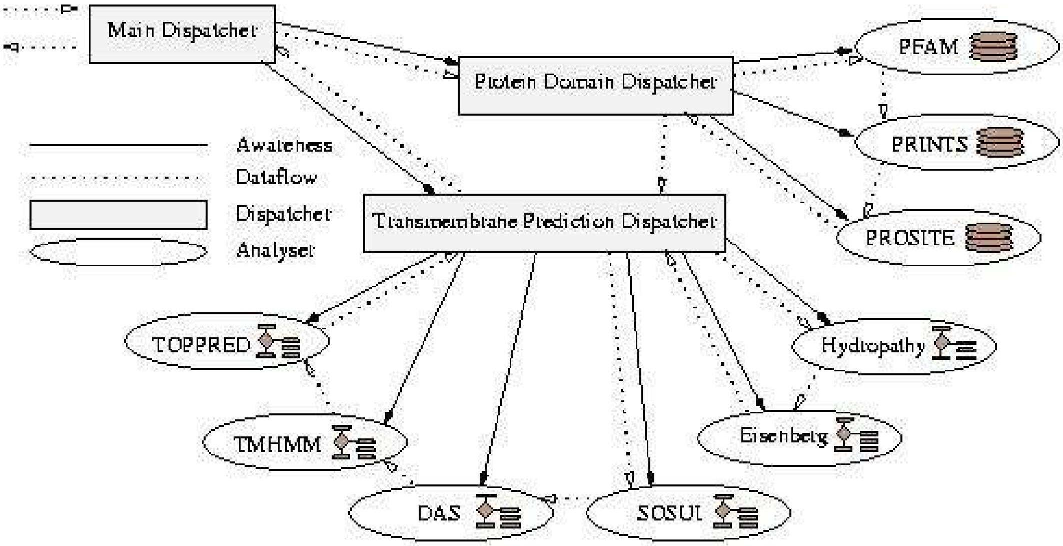


Figure 37: Visualisation of analysers and the dataflow of entries

Dotted arrows represent the data flow of entries through the analysers. This shows that dispatchers can decide, not to expect the annotated entry before all analysers of a certain selection have performed on a specific entry. This further removes strain on network traffic. It was previously demonstrated that the rules for integration of tools must be carefully curated. It is necessary to distinguish between domain knowledge that is independent from any application, and temporary knowledge that is acquired selectively for individual sequences.

3. Application of the conflict resolution

This section gives an example of how REVISE works, by using the set of revisables shown in Figure 38, and the rules as described above.

```
revisable(ft(das,p04633, transmem,19,  29 ),true).
revisable(ft(das,p04633, transmem,120, 128),true).
revisable(ft(das,p04633, transmem,214, 229),true).
revisable(ft(das,p04633, transmem,216, 227),true).
revisable(ft(das,p04633, transmem,280, 285),true).

revisable(topology(phd,p04633,inner,outer),true).
revisable(ft(phd,p04633, transmem,18, 35), true).
revisable(ft(phd,p04633, transmem,117,133),true).
revisable(ft(phd,p04633, transmem,214,231),true).
revisable(ft(phd,p04633, transmem,271,288),true).

revisable(transmembrane(tmhmm,p04633),false).

revisable(ft(toppred,p04633, transmem, 13, 33),true).
revisable(ft(toppred,p04633, transmem,113,133),true).
revisable(ft(toppred,p04633, transmem,212,232),true).
revisable(ft(toppred,p04633, transmem,239,259),true).
revisable(ft(toppred,p04633, transmem,269,289),true).
```

Figure 38: Revisables presented to REVISE.

The figure presents the predicate-representation of the prediction results on the sequence with the accession number P04633. The predicate-reformulation is performed within EDITtoTrEMBL. Outstanding is the (wrong) classification of the sequence as soluble by TMHMM. It is represented as a negative statement that must be revised to become true.

The predicate *solution* returns a list of minimal revisions, consisting of a set of revisables changed from false to true, and a list of those revisables changed from true to false.

Local conflict checks only

If neither the domain information is present, nor the domain database has any rules available, *solution*(*X*) will return a single solution:

```
X = [[], [ft(das, p04633, transmem, 19, 29),
      ft(das, p04633, transmem, 120, 128),
      ft(das, p04633, transmem, 216, 227),
      ft(das, p04633, transmem, 280, 285)]
     ] ;
```

This represents the transmembrane regions that are too short.

Balance with other predictions

With the additional constraint, that all predictions must agree on a protein to be either integrated into the membrane or soluble a revision of TMHMM's prediction from true to false is introduced. The third solution trusts TMHMM and assumes all transmembrane regions to be false positive:

```
[[is_transmembrane(tmhmm, p04633)],
 [ft(das, p04633, transmem, 19,..., 285),
  ft(phd, p04633, transmem, 18, 35)]
] ;
[[$is_transmembrane$($tmhmm$, p04633)],
 [ft(das, p04633, transmem, 19,..., 285),
  ft(toppred, p04633, transmem, 13, 33)]
] ;
[[],
 [ft(das, p04633, transmem, 19,..., 285),
  ft(phd, p04633, transmem, 18, 35),
  ft(phd, p04633, transmem, 117, 133),
  ft(phd, p04633, transmem, 214, 231),
  ft(phd, p04633, transmem, 271, 288),
  ft(toppred, p04633, transmem, 13, 33),
  ft(toppred, p04633, transmem, 113, 133),
  ft(toppred, p04633, transmem, 212, 232),
  ft(toppred, p04633, transmem, 239, 259),
  ft(toppred, p04633, transmem, 269, 289)]
];
```

Use of pattern database

The following matches to the PROSITE database have been derived:

```
matches(p04633,ps00215,prosite,32,41).
matches(p04633,ps00215,prosite,132,141).
matches(p04633,ps00215,prosite,231,240).
```

Addition information can be retrieved from the protein match data found in PROSITE:

```
is_transmembrane(prositedoc,Acc)←
    matches(Acc, prosite, ps00215, _,_).
num_tm_regions(prositedoc,Acc,6)←
    matches(Acc, prosite, ps00215, _,_).
loop(prositedoc,Acc,'',X,T)←
    matches(Acc, prosite, ps00215, F,T),\;X\;is\;F+3).
transmembrane(prositedoc,Acc,X,X,)←
    matches(Acc, prosite, ps00215, F,T),\;X\;is\;F-3).
```

The knowledge that this protein sequence indeed belongs to a transmembrane protein led to the exclusion of the third option in the previous output. The then second solution needed to be removed because PHD's prediction of a transmembrane region from residues 18 to 35 is in conflict with a loop region between residues 35 and 41.

```
X = [[], [ft(das, p04633, transmem, 19, ..., 285),
          ft(phd,   p04633, transmem, 18, 35),
          ft(toppred, p04633, transmem, 239, 259)]
    ] ;
```

Interpretation of REVISE's output

The constraints supplied to the system guarantee that the final solution will not be in conflict with the information known for specific protein domains. When multiple tools predict the same transmembrane region then they tend to vary only slightly in their description of the positions of MSRs.

REVISE presents all possible interpretations of the prediction methods consistent with itself and the extra knowledge from protein domain databases. This can be visualised as follows:

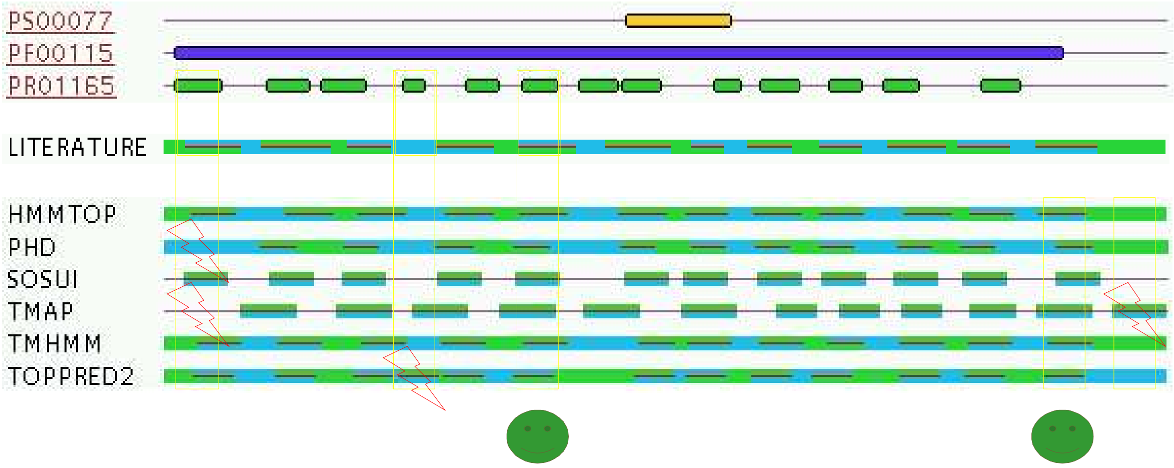


Figure 39:Graphical interpretation of the revision process

From the literature or a consensus within SWISS-PROT links were created between matches of protein domain databases and transmembrane annotation. These are then used to constrain the prediction of transmembrane topology predictions. Where no domain information is available, this is calculated as a majority vote, unless probabilities could be assigned to reflect the reliability of a TM region's assignment.

Horizontal lines represent the protein sequences, boxes a protein domain match, and barred boxes are predicted transmembrane regions. Flashed transmembrane regions are found to be in conflict with the remaining annotation.

For the annotation that is produced in order to become transferred to TrEMBL, the medians of the transmembrane regions's borders are chosen. The case of remaining ambiguities after conflict resolution, i.e. multiple solutions, has not been catered for in the current implementation. In this case only the common annotation should be transferred into TrEMBL being appropriately marked.

Mutual independence of revisables

The revision of a fact may give rise to new conflicts. This inherent non-monotonicity is the major problem in conflict resolution. It is not possible to give prior advise to the system as to which revisions should be attempted, in cases when the specific revision is suggested by occurrence of a particular conflict.

Also one cannot define semantic dependencies among revisables. If these exist, those must be defined in REVISE as conflicts. There are two problems with this approach. The obvious one is that these rules may be rather hard to maintain.

To allow individual revisions of facts also for the transmembrane annotation, the redundant transmembrane annotation style of SWISS-PROT, in which moieties of either loops are explicitly stated, was reduced. The transmembrane annotation was represented by the transmembrane regions as predicate each, plus a single predicate to represent the protein's sidedness. The alternative would be to provide an additional description of the individual loops, leading to an increased efficiency for rules, though thereby loosing the revisables's mutual independence. A revision of the first loop should also change semantics for the sidedness, which is the only dependency between revisables remaining.

4. Summary

Constraint technology is now becoming increasingly accepted. A very recent example is the latest version of HMMTOP (Tusnády and Simon 2001) that facilitates constraints on the protein topology to be imposed by the user. It seems feasible to integrate a protein domain detection with HMMTOP and to apply constraints from this work in an automated manner.

F. Specialisation and data-mining on 7TM proteins

1. Motivation and history

At this stage of my studies, I had the best infrastructure to annotate the topology of transmembrane proteins. However, the work had not yet been employed to solve a significant biological problem.

With Dr. Michael Croning from the PRINTS group at the EBI the idea was developed, to first use the TM prediction in his GPCR (G Protein-Coupled Receptor) discovery environment and second to hunt for signals that determine the coupling specificity of GPCRs. It was expected that if such a signal existed, it would be located on the then determined cytoplasmic side of the receptor. Because of this, any approach utilising the whole receptor sequence would be likely to fail, as other functional properties of the receptor (such as ligand-interaction determinants) would confound detection.

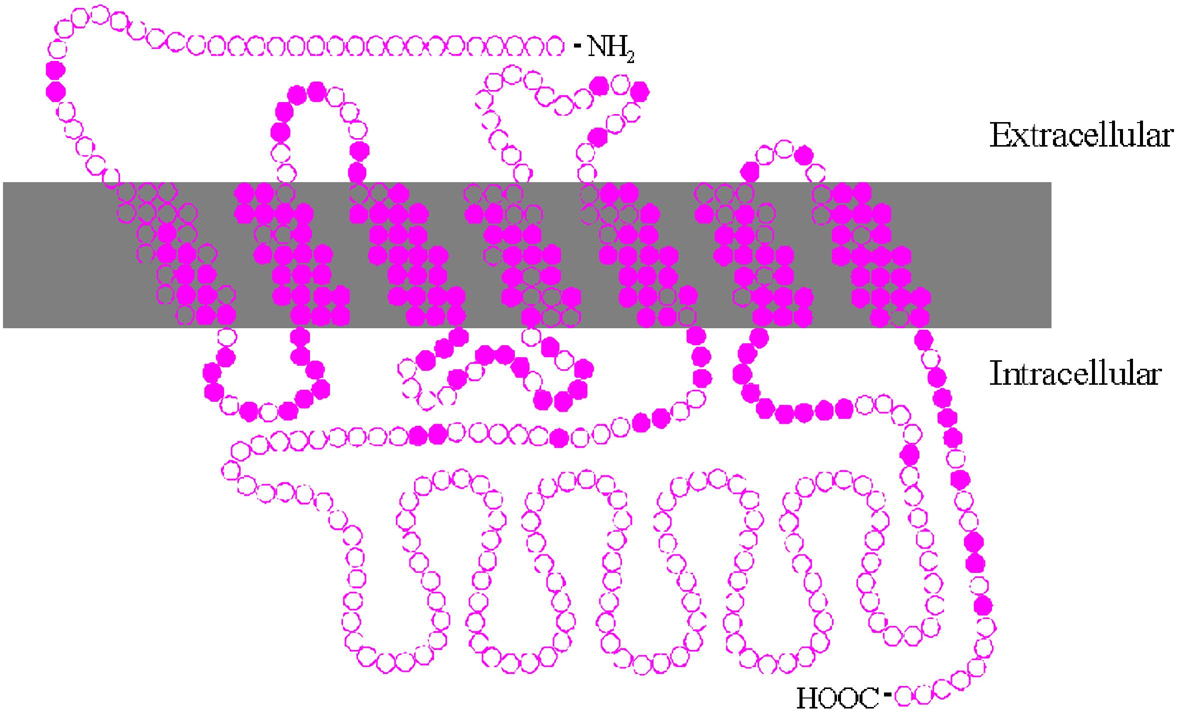


Figure 40: Conserved topology and residues of GPCRs in family one

All GPCRs are assumed to have a 7TM N-out topology. Conserved residues are drawn in as full circles, adapted from (Lynch 1998)

From the evaluation of membrane topology prediction methods (section C) it was known that the topology prediction for GPCRs is difficult. The GPCRs are generally assumed to have exactly seven membrane-spanning regions with an extracellular N-terminus and cytoplasmic C-terminus. Unexpectedly polar transmembrane helices would tend to prevent a transmembrane helix from being accurately predicted as such. Therefore, the data mining effort on poorly predicted cytoplasmic residues would be likely fail. To address this problem, a prediction tool specific for the GPCR topology had to be created. In addition, if this is achievable, this should also be applied to protein sequences of other families with a conserved topology.

To determine characteristic patterns it was first thought to apply a standard sequence similarity approach or the generation of PRINTS-like patterns to specify signals. Then it was found that this problem seems similar or even equivalent to the location of DNA binding sites.

2. Prediction of Membrane-Spanning Regions in Heptahelical Receptors

This section explains the development of a new algorithm dedicated to the prediction of the topology of 7TM proteins which is based on work of Henrik Nielsen and Anders Krogh from the Danish Technical University in Lynbgy, Denmark.

Introduction

If a number of topological constraints which are peculiar to GPCRs could be integrated into a prediction algorithm, then one might be able to improve the quality of MSR prediction for these difficult proteins. These constraints are:

- an extracellular N-terminus,
- seven MSRs,
- and a cytoplasmic C-terminus.

Another issue is the differentiation of signal sequences from precursors of transmembrane proteins. Some GPCRs are special (Nielsen and Krogh 1998) in featuring a long N-terminal translocated extracellular region with no prior signal sequence. In SWISS-PROT about 20% of all GPCRs are annotated as precursors. Although TMHMM is most robust in this differentiation, for this method the previously described development of an incorporated signal detection on the basis of SignalP and TMHMM is applied.

In an evaluation of current prediction methods (Möller, Croning et al. 2001) on GPCRs, TMHMM (Sonnhammer, Heijne et al. 1998; Krogh, Larsson et al. 2001) and HMMTOP (Tusnády and Simon 1998; Tusnády and Simon 2001), which where the only two HMM-based methods, performed best. Especially TMHMM seems to under-predict which may be due to its intrinsic difficulty to detect those periodic occurrences of polar residues in membrane helices that contribute to the helix's aliphaticity.

This work provides the means to modify the HMM of TMHMM to match the conditions stated above and thereby create a new customised model, for GPCRs as well as for other transmembrane families such as integrins, connexins or various well-characterised channels. This forces the topological predictions to obey such constraints. Such customised models can be generated for any number of MSRs, sidedness of integration and signal peptides.

Methods

The Hidden Markov Model (Durbin, Eddy et al. 1998) underlying TMHMM features separate units for outer, inner and membrane moieties in either direction. The circular orientation of these moieties allows recognising any number of transmembrane helices. HMMs are not capable of counting the number times a particular state was visited. This is a consequence of the Markov property. Hence, in order to make the HMM underlying TMHMM sensitive to the number of helices detected, the circularity had to be unrolled in order to match the number of helices expected. The development of the model from its atomic units, i.e. signal detection, outer and inner loops and transmembrane regions, is visualised in Figure 41 and the model specific for GPCRs again in Figure 44.

To allow only a specific number of MSRs to be predicted, the circular model of TMHMM needed to be unrolled.

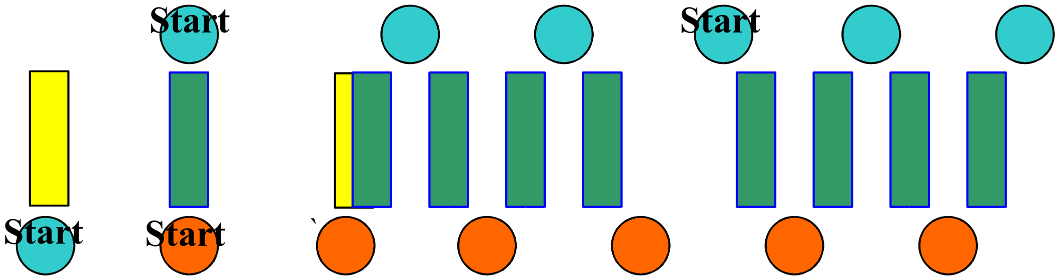


Figure 41: Graphical visualisation of Hidden Markov Models towards the development of 7TMHMM

The yellow box represents a detected signal sequence, the green boxes represent a detected transmembrane region. Blue circles represent extracellular domains, red regions intracellular domains. Hence, the third and forth schema shown in the figure represent a topology model for 5TM proteins with either direction of insertion. The cleavable signal is shown to compete with the first outwards directed TM region.

A combination of both programs is difficult to evaluate since no complete test set is available that features both experimentally verified signals and biochemically characterised transmembrane regions. The initial approach avoided any retraining for proteins of different topologies and used the respective original transition probabilities for all loops. The integration of SignalP and TMHMM still has some problems. As pointed out by Henrik Nielsen, the Hidden Markov Model of SignalP is excellent for detecting signals due to their amino acid distribution, but is inferior to the earlier Neural Network based solution for the actual determination of the cleavage site (Nielsen 1999; Nakai 2000). That is not surprising as the actual cleavage of residues is dependent on a fairly definite co-occurrence of residues. This is more difficult for an HMM to model, while – also depending on its architecture – a Neural Network may more easily classify such gapped correlations (Baldi and Brunak 1998; Durbin, Eddy et al. 1998), e.g. by the introduction of an additional layer. I am investigating at the moment to what extend the context of amino acids, e.g. the predicted secondary structure, can be used to extend the residue alphabet for the HMM. Such an approach brings the context in via the alphabet rather than by changing the HMM's Markov property or the model of membrane insertion itself.

The script performing the unrolling of the original HMM was written in JavaScript. This can be executed in all standard web browsers, the user is requested to enter the number of MSRs expected, whether signal peptides should be detected and finally whether the N-terminus is inside or outside. It generates an HMM for the TMHMM engine that recognises any specific specified number of MSRs in a protein. This uses the original values for transitions of TMHMM 1.0 but instead of reiterating through the same model for all the helices the generated model no longer features loops.

Summary

The generation of HMMs seems to be a promising way to incorporate global constraints for protein families in large-scale sequence analysis. The TMHMM engine states the likelihood for a model to represent a certain topology. For many protein families a constant transmembrane topology is assumed, e.g. integrins and connexins have previously been mentioned.

When topology of the protein is known due to strong sequence similarity then in theory this could directly be used as an input for a feature propagation (Velds 1999) of the transmembrane annotation. However, the overall sequence similarity between GPCRs is only about 30%. A specific algorithm like 7TMHMM is supposedly simpler to invoke and provides better consistency since all annotation is created independently from an otherwise varying external input. It was suggested to annotate the HMMs of Pfam with respect of protein topology, in order to serve as a family-specific sequence annotator. The prior section D on finding constraints for topology by InterPro addresses this issue.

A program like TMHMM could also be integrated dynamic environments for the annotation of peptide sequences to combine information from protein domain databases with a topology prediction as presented in the previous section. In these, information from domain databases can be accumulated to determine the most constrained HMM for a prediction of the actual membrane spanning regions.

Since only very limited reliable information on the topology of GPCRs is available in sequence databases it is not clear how good the performance of 7TMHMM really is. For now one relies on the idea that it represents the most probable model for a protein to feature seven transmembrane regions, since values from a training of the best-performing underlying circular model were transferred.

Some problems have been observed with the analysis of some olfactory receptors, more work remains necessary for a complete understanding. The current hypothesis is that remaining problems are due to the competition of the signal prediction with transmembrane regions. A first transmembrane region strongly resembling a cleavable signal, while the HMM has problems with the recognition of signal cleavag sites, may wrongly be accepted as a signal peptide and 7 MSRs would be predicted where there are only 6 left. With respect to the evaluation of prediction methods (section C) which only requested seven MSRs to be predicted, with all its weaknesses this algorithm would achieve 100% versus 85% of the native TMHMM. A second version of 7TMHMM without the integration of the signal prediction was created, in order to have a pointer to remaining ambiguities by a differential topology prediction.

3. Prediction of the coupling specificity of GPCRs to their G proteins

G protein coupled receptors (GPCRs) are found in great numbers in most eukaryotic genomes. They are responsible for sensing a staggering variety of structurally diverse ligands, with their activation resulting in the initiation of a variety of cellular signalling cascades. The physiological response, which is observed following receptor activation, is governed by the guanine nucleotide-binding proteins (G proteins), to which a particular receptor chooses to couple. Previous investigations have demonstrated that the intracellular domains of the receptor govern the specificity of the receptor-G protein interaction. Despite many studies, it has proven very difficult to predict *de novo*, from the receptor sequence alone, the G proteins to which a GPCR is most likely to couple. In order to find patterns of amino acid residues in the intracellular domains of GPCR sequences that are specific for coupling to a particular functional class of G proteins, a data-mining approach was combined with a pattern discovery and with membrane topology prediction. A prediction system was then built, being based on these discovered patterns. This approach was successful in the prediction of G protein coupling specificity of unknown sequences. Such predictions should be of great use in providing *in silico* characterisation of newly cloned receptor sequences and for improving the annotation of GPCRs stored in protein sequence databases.

Introduction

GPCRs are the biggest single class of receptors in biology, playing essential roles in a remarkably wide range of physiological and patho-physiological conditions. The actions of a large and structurally diverse range of hormones, neurotransmitters, tastants, odourants, photons, and peptidases, are initiated by their binding to GPCRs located on the cell surface (Bockaert and Pin 1999). Such binding activates the receptor, which cause helical rearrangements within the receptor. These (by way of unmasking binding sites) transmit the activation signal to a guanine nucleotide-binding protein (G protein) located on the cytoplasmic surface of the membrane, closely apposed to the receptor (Schoneberg, Schultz et al. 1999; Gether 2000).

Activation of the heterotrimeric G protein (consisting of α, β, and γ subunits) promotes exchange of the guanosine diphosphate (GDP), bound to the α subunit, for guanosine triphosphate (GTP) (Figure 42). This allows the dissociation of the α subunit (with GTP bound) from both the receptor and βγ complex. The separate moieties can then modulate several cell-signalling pathways, and the activities of certain ion channels. Termination of the response occurs as a result of the intrinsic catalytic activity of the α subunit, which hydrolyses the bound GTP to GDP. Subsequently the α -GDP then re-associates with the βγ complex to form the inactive heterotrimer.

Amongst the biochemical responses that have been observed following receptor activation (III 1999) are both stimulation and inhibition of adenylate cyclase activity. The G_s class, and the $G_{i/o}$ class of G proteins, respectively, mediate these opposing effects. The $G_{q/11}$ family activate phospholipase C enzymes, resulting in phosphatidylinositol hydrolysis. Together these three families constitute the major functional classes of G proteins, and studies have revealed this specificity is determined by the particular subtype of the α subunit, making up the G protein (Simon, Strathmann et al. 1991; Bourne 1997). Characteristically each GPCR subtype appears to only couple to a subset of the G proteins that may be found in a particular cell. Elucidation of the mechanism(s) underlying this coupling specificity has been a central theme in GPCR research over the last 15 years. Biochemical studies, especially those that involve the creation of chimeric receptors, have been used in order to locate domains within receptor sequences that may define their specificity of G protein coupling. Other strategies that have been employed are the use of synthetic peptides, which are designed to mimic or inhibit the normal interactions of receptor-G protein, and the neutralisation of specific G proteins with antibodies. Together this large number of studies has revealed that the selectivity of G protein recognition (and hence coupling) is determined by multiple intracellular receptor regions. The most important regions appear to be the second intracellular loop, and the start and end of the third intracellular loop, which are close to the cytoplasmic surface of the membrane (Wess 1998).

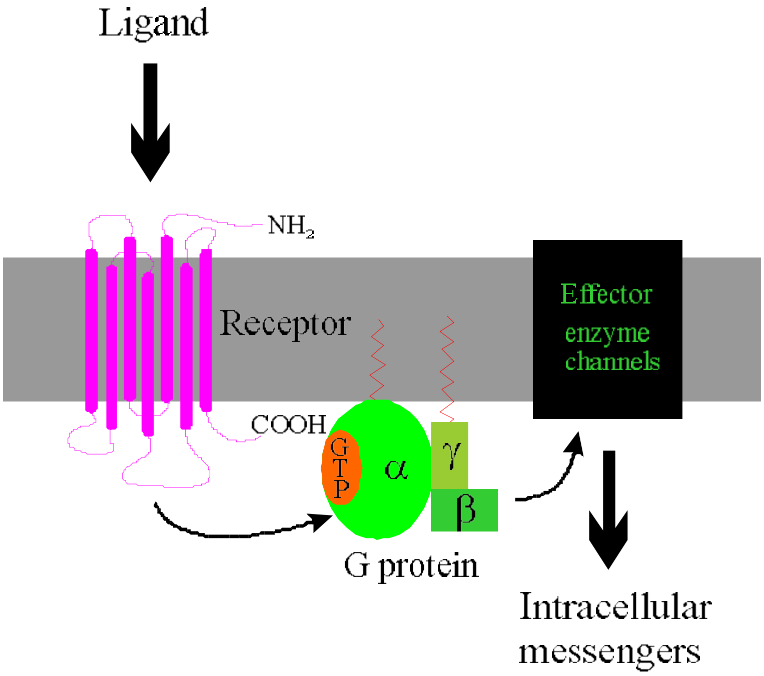


Figure 42: An activated G protein approaching its effector enzyme

With a ligand binding to a GPCR, a coupled G protein changes from being GDP bound to GTP and leaves the GPCR. The α subunit eventually splits from the βγ subunits, either may be catalytically active (courtesy of M.Croning).

However, the coupling specificity has yet to be experimentally determined for many hundreds of mammalian GPCRs, including many peptide receptors (Liu and Wess 1996). This knowledge is important for two main reasons, firstly, to understand the physiological mechanisms underlying the response mediated by activation of a given GPCR, and secondly, in order to choose appropriate cell lines for the heterologous expression of newly-cloned GPCRs. This is crucial for the study of the increasing catalogue of GPCRs that have been cloned but for which the endogenous agonist is unknown, the so-called orphan receptors (Wilson, Bergsma et al. 1998). The ligand-identification strategy that is applied to these orphans depends upon the functional coupling of the receptor to a G protein. A downstream change (such as a change in second messenger concentration), can be observed with a suitable assay. One then passes appropriate tissue extracts (or libraries of chemical compounds) over the cells, hoping to observe a response. Of course, for such a method for an identification to succeed G proteins must be present in the chosen cell, to which the receptor is willing to couple. In an effort to improve this likelihood, a number of transgenic systems have been developed, by introducing native or engineered G protein α subunits that are promiscuous in their coupling to receptors (Wess 1998). An overview on the role G proteins play in the development of disease is given in (Milligan and Wakelam 1992) and a more recent overview is given in (Roche 1996).

Clearly, the development of an accurate method for the prediction of the coupling specificity of a receptor to G protein(s) would be of great utility in guiding experimental investigations for the characterisation of GPCRs. No receptor sequence motifs that represent preferences for G protein coupling have previously been reported, which unambiguously determine coupling specificity across GPCR families and subtypes. It was thought unlikely with respect to the simple human inspection of GPCR sequences that a successful prediction system would be within reach. Instead, it was decided to use the prior knowledge that the sequences motifs for the coupling would likely be located in the intracellular domains of the receptor. To find those, a protein pattern discovery algorithm would be applied to hunt for commonly occurring patterns in these intracellular loops and C-termini. Following the identification of a large number of patterns, their usefulness in prediction was subsequently evaluated. This was based upon a set of ≈ 100 paralogous receptor sequences for which the G protein coupling specificity had previously been experimentally determined. Using such a bioinformatics approach (combining membrane topology prediction with pattern discovery), one can indeed discover combinations of patterns that are characteristic for the coupling of receptors to G proteins.

Methods

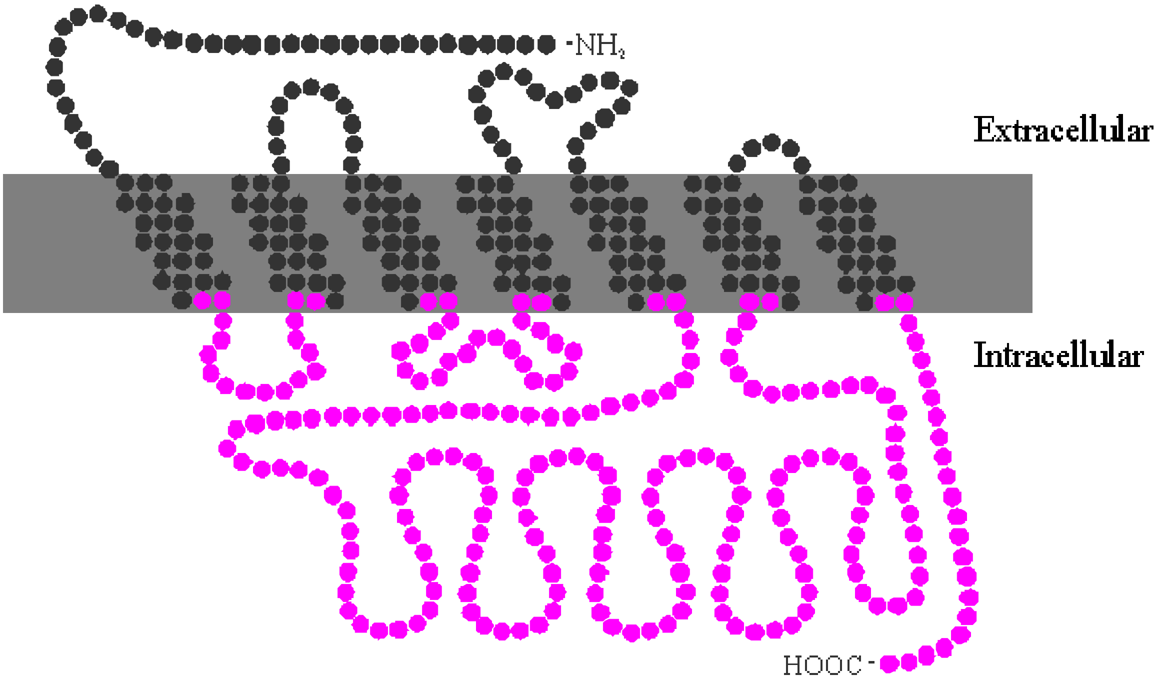


Figure 43: Visualisation of the search space for coupling-specific patterns of amino acids

Some patterns, as the DRY pattern are found directly ‚underneath' the membrane layer. To allow a slight variation of the topology prediction, certain ‚fuzziness' was offered. The intracellular regions reach three residues into the membrane (courtesy M. Croning).

The strategy to predict the coupling specificity of GPCRs for their G proteins was to attempt to find patterns of amino acid residues in their sequences that appeared to be specific for a particular class of G protein. In order to do this a set of GPCR sequences is required for which the coupling specificity has been reported, and a pattern discovery algorithm. From SWISS-PROT and TrEMBL 103 diverse receptor sequences were selected, for which an apparently non-promiscuous coupling had been determined and was summarised in the TIPS Nomenclature Supplement (TiPS 2000). These were grouped into the three functional classes $G_{i/o}$, G_s and $G_{q/11}$.

To constrain the search for patterns to the putative intracellular domains of the sequences and an incorporation of the signal peptide prediction, an accurate method was required as it was presented in the previous section 2. The model employed assumes exactly 7 MSRs, with extracellular N-terminus and intracellular C-termini, as shown in Figure 44.

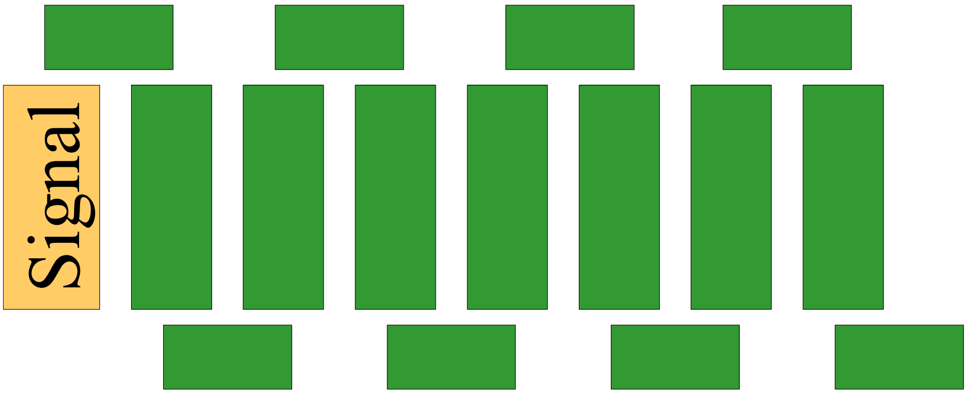


Figure 44: 7TMHMM for the prediction of GPCR topology

The model assumes the topology 7TM with N-terminus outside and allows an N-terminal signal sequence.

Generation and evaluation of patterns

The pattern discovery was carried out using the program SPEXS (Vilo 1998). This performs an exhaustive search within the input sequences with regular expressions as the predefined pattern language. Amino acids were grouped by property as described in (Livingstone and Barton 1993). This produced a large number of patterns (>4000), which was then evaluated for their usefulness. The most discriminative patterns are those that occur in large number of sequences of one receptor-G protein-coupling group and infrequently in the others. The specificity of all the patterns occurring in each group of receptor sequences was determined.

For every pattern, the likelihood of its appearance in its respective receptor-G protein-coupling group was calculated and normalised by its occurrence in all the sequences contained in the three functional classes of G protein coupling. The pattern score was calculated as the inverse of the probability, adapting an earlier method used to estimate the significance of patterns found in DNA sequences (Brazma, Jonassen et al. 1998; Vilo, Brazma et al. 2000). Thus, the smaller the probability of a random match was determined, the higher the pattern scores.

Hypothetically, it might improve the classification of the receptor-G protein coupling groups (and thus subsequent prediction of the coupling for a novel sequence) if considering the specificity of combinations of patterns, rather than just single patterns. This is similar to the concept of collections of motifs (called fingerprints) that are found in the secondary protein database PRINTS (Attwood, Croning et al. 2000), or the analysis of the regulatory regions in DNA (Scherf, Klingenhoff et al. 2000). Derived pairs and triplets of patterns that are specific for the binding to G proteins are used in conjunction to act as a classifier.

If all the patterns making up a particular combination where found in a sequence, the combination was said to match as a whole. For each sequence presented to the classifier, the total number of combinations found is reported. If 30% or more of the matches belonged to a specific receptor-G protein-coupling group, then this coupling was assumed a putative prediction. This potentially allows one to predict promiscuous receptor-G protein coupling.

In order to test the resultant classifier, the G protein coupling specificity of 10 human GPCR subtypes was predicted, for which the G protein coupling had either been experimentally determined or could be inferred from the biochemical responses following the activation of the receptor. It was ensured that the sequences of this test set are paralogue to those of the training set (compare Table 23).

Results

Pattern	Gio	Gq/11	Gs	Sensitivity	Specificity	Best class
Total	55	33	25			
[ILV]...SG.{0,10}R	15	0	0	0.273	1	Gi/o
N..R.{1,4}R	15	0	0	0.273	1	Gi/o
Y.A.{1,8}A[ILV]	15	0	0	0.273	1	Gi/o
A[ILV].{2,5}RT	15	0	0	0.273	1	Gi/o
N..[RK]..R	17	1	0	0.309	0.9444	Gi/o
K.[RK].{0,10}K.[ILV]	17	1	0	0.309	0.9444	Gi/o
V...[RK]....R	17	1	0	0.309	0.9444	Gi/o
[RK]...[CM][RK]	23	1	2	0.418	0.8846	Gi/o
V[RK].{1,10}SG	16	1	0	0.291	0.9412	Gi/o
K.[RK].{1,4}L[RK]	16	1	0	0.291	0.9412	Gi/o
[FWY][ILV]..V.{2,10}R	15	1	0	0.273	0.9375	Gi/o
Y.[RK].[RK].{0,9}T	15	1	0	0.273	0.9375	Gi/o
[ILV].A[AGS].{1,4}R	15	1	0	0.273	0.9375	Gi/o
FR....[RK].{0,3}L	15	1	0	0.273	0.9375	Gi/o
DRY.[AGS].{3,6}A	15	1	0	0.273	0.9375	Gi/o
F[RK]....K.{1,7}C	15	0	1	0.273	0.9375	Gi/o
A....[ILV].{1,8}RT	15	1	0	0.273	0.9375	Gi/o
[RK]....R.{0,9}EK	15	0	1	0.273	0.9375	Gi/o
[RK]R.{0,3}TR	15	1	0	0.273	0.9375	Gi/o
KA.{3,6}T	15	1	0	0.273	0.9375	Gi/o
DR.{4,11}H...[AGS]	15	1	0	0.273	0.9375	Gi/o
R....K.{0,8}T[AGS]	15	1	0	0.273	0.9375	Gi/o
[RK][FWY][ILV].{2,5}V	18	1	1	0.327	0.9000	Gi/o
N.{2,5}R.[FWY]	18	1	1	0.327	0.9000	Gi/o
Y.[AGS].{1,8}A[ILV]	18	2	0	0.327	0.9000	Gi/o
N..[RK].{1,4}R	23	3	1	0.418	0.8519	Gi/o
[ED].{0,3}N..[RK]	23	2	2	0.418	0.8519	Gi/o
Y.{2,5}I..[AGS]	23	0	4	0.418	0.8519	Gi/o
N..[RK].{1,11}R	30	6	2	0.545	0.7895	Gi/o
[RK].R.{2,12}K[RK]	20	4	0	0.364	0.8333	Gi/o
[ILV]...SG	20	1	2	0.364	0.8696	Gi/o
[AGS][RK]..[ED].{0,10}R	17	1	1	0.309	0.8947	Gi/o
[FWY].A.{1,9}A[ILV]	17	2	0	0.309	0.8947	Gi/o
R[FWY].[AGS][ILV].{0,7}A[ILV]	17	2	0	0.309	0.8947	Gi/o
[ILV].R....V	17	0	2	0.309	0.8947	Gi/o
[RK]Y.[AGS].{3,5}A	17	0	2	0.309	0.8947	Gi/o
[ILV]...SG.{0,8}E	17	0	2	0.309	0.8947	Gi/o
[FWY].[AGS][ILV]..A	17	1	1	0.309	0.8947	Gi/o
[RK]..[RK].{0,3}R[ILV]	32	8	2	0.582	0.7619	Gi/o
[ED]A.{0,3}E	19	3	0	0.345	0.8636	Gi/o

T..[RK].{0,10}S..T	0	11	0	0.333	1	Gq/11
A.{3,6}V[ILV][RK]	0	11	0	0.333	1	Gq/11
P..[AGS]T.{0,10}S	0	10	0	0.303	1	Gq/11
[AGS][ILV][ILV][RK].{2,10}S	0	10	0	0.303	1	Gq/11
S[FWY].{1,11}Q[ILV]	0	10	0	0.303	1	Gq/11
[AGS].{0,3}S..T[ILV]	0	10	0	0.303	1	Gq/11
S...L.{2,9}TL	0	10	0	0.303	1	Gq/11
[RK]F....K	0	10	0	0.303	1	Gq/11
[AGS].[ILV].{0,10}K.F	0	10	0	0.303	1	Gq/11
[AGS].S.[RK].{0,10}F	1	13	0	0.394	0.9286	Gq/11
S...L.{1,10}T[ILV]	1	12	0	0.364	0.9231	Gq/11
[RK].T.{0,10}Q[AGS]	0	12	1	0.364	0.9231	Gq/11
[AGS]...L.{1,10}TL	1	12	0	0.364	0.9231	Gq/11
[AGS][ILV][ILV][RK]	0	12	1	0.364	0.9231	Gq/11
A.{0,10}V[ILV][RK]	1	14	1	0.424	0.8750	Gq/11
[AGS].{0,3}V[ILV][RK]	1	14	1	0.424	0.8750	Gq/11
F.{0,10}Y...[RK]	0	14	2	0.424	0.8750	Gq/11
[CM].[FWY].{3,12}P	1	11	0	0.333	0.9167	Gq/11
S.[AGS].{3,13}TL	1	11	0	0.333	0.9167	Gq/11
V[AGS].{0,10}S.[AGS].[ILV]	1	11	0	0.333	0.9167	Gq/11
Y....[RK]P.{2,10}A	0	11	0	0.333	1	Gq/11
[ILV]......A.T	1	11	0	0.333	0.9167	Gq/11
S...L.{1,11}Y	1	11	0	0.333	0.9167	Gq/11
A.{3,12}V[ILV][RK]	0	11	1	0.333	0.9167	Gq/11
[AGS].{2,5}V[ILV][RK]	1	11	0	0.333	0.9167	Gq/11
[FWY].{4,7}KP	1	11	0	0.333	0.9167	Gq/11
R.[RK].{0,10}K[AGS][AGS]	1	11	0	0.333	0.9167	Gq/11
[ILV]A.{2,4}S.[ILV]	1	11	0	0.333	0.9167	Gq/11
[AGS].[ILV].{2,10}L.[FWY]	0	11	1	0.333	0.9167	Gq/11
[AGS][FWY]..[FWY]	1	11	0	0.333	0.9167	Gq/11
S.S.{1,11}L.S	0	11	1	0.333	0.9167	Gq/11
[ILV].L.{6,11}A.T	1	11	0	0.333	0.9167	Gq/11
K.{0,3}N.P	1	11	0	0.333	0.9167	Gq/11
[ILV].L.{6,10}A.T	0	11	0	0.333	1	Gq/11
[RK][FWY]....K	2	13	0	0.394	0.8667	Gq/11
[AGS].S.[RK].{2,10}F	1	13	0	0.394	0.9286	Gq/11
[ILV].{3,6}S.Q	3	18	3	0.545	0.7500	Gq/11
C.[FWY].{2,11}K	0	10	1	0.303	0.9091	Gq/11
C.[FWY].{2,12}K	0	10	1	0.303	0.9091	Gq/11
S....[RK]A.{3,10}S	1	10	0	0.303	0.9091	Gq/11
A[ILV].{1,5}Y..[ILV].T	0	0	10	0.400	1	Gs
A.{1,5}RY....T	0	0	10	0.400	1	Gs
I....RY.{1,10}R	0	0	9	0.360	1	Gs
I....RY.{4,6}T	0	0	9	0.360	1	Gs
LR.{1,9}T...[ILV]	0	0	9	0.360	1	Gs
RS.{3,13}C[AGS]	0	0	9	0.360	1	Gs
[ILV].[FWY]H.{1,3}I	0	0	9	0.360	1	Gs
F.{1,4}Y....T	0	0	9	0.360	1	Gs
I....RY.{4,4}T	0	0	9	0.360	1	Gs
I....R[FWY]	0	0	9	0.360	1	Gs
I....RY....T	0	0	9	0.360	1	Gs

I....RY	0	0	9	0.360	1	Gs
[FWY].A.{2,6}Y..[ILV]	0	0	9	0.360	1	Gs
I.[AGS].{1,10}S...R	0	0	8	0.320	1	Gs
[ILV].[FWY]H.{3,12}T	0	0	8	0.320	1	Gs
L..H.[ILV]	0	0	8	0.320	1	Gs
[ILV].[FWY]H.[ILV]	0	0	8	0.320	1	Gs
[ILV].[FWY]H.I	0	0	8	0.320	1	Gs
A....[RK][RK]I	0	0	8	0.320	1	Gs
[AGS].{0,10}L..H.[ILV]	0	0	8	0.320	1	Gs
[ILV].[FWY]H.{3,10}T	0	0	8	0.320	1	Gs
[FWY]H.I.{0,3}T	0	0	8	0.320	1	Gs
S.{5,12}S.L.[RK]	0	0	8	0.320	1	Gs
S.{5,9}S.L.[RK]	0	0	8	0.320	1	Gs
Q.{0,9}S.L.[RK]	0	0	8	0.320	1	Gs
A.{1,5}RY..[ILV].T	0	0	8	0.320	1	Gs
F.{1,10}A...H	0	0	8	0.320	1	Gs
[ILV]..H.[ILV].{1,3}T	0	0	8	0.320	1	Gs
[FWY]H.I.{0,10}V	0	0	8	0.320	1	Gs
A..[FWY].{0,3}H	0	1	10	0.400	0.9091	Gs
I....[RK]Y.{4,6}T	0	0	10	0.400	1	Gs
A.{1,5}R[FWY]....T	1	0	10	0.400	0.9091	Gs
A.{2,6}Y..[ILV].T	0	1	10	0.400	0.9091	Gs
A..[FWY].{0,8}H	1	1	11	0.440	0.8462	Gs
[AGS].{1,5}RY....T	0	2	11	0.440	0.8462	Gs
I....[RK]Y.{1,10}R	0	1	9	0.360	0.9000	Gs
R[FWY]H.{5,14}R	0	1	9	0.360	0.9000	Gs
[RK]S.{3,13}C[AGS]	1	0	9	0.360	0.9000	Gs
[RK].[ILV].C.R	1	0	9	0.360	0.9000	Gs
[RK].[ILV].C.[RK]	1	0	9	0.360	0.9000	Gs

Table 22: List of generated patterns found to best represent a specific coupling mechanism

Column 1 shows the pattern as a regular expression, columns 2-4 show the number of matches to the different receptor-G protein coupling groups, columns 5 and 6 show sensitivity and specificity, respectively.

Lists of patterns

Table 22 shows the 40 best patterns found for each receptor-G protein coupling group, together with the number of times they match in each of the three training set groups, and their calculated sensitivity and specificity.

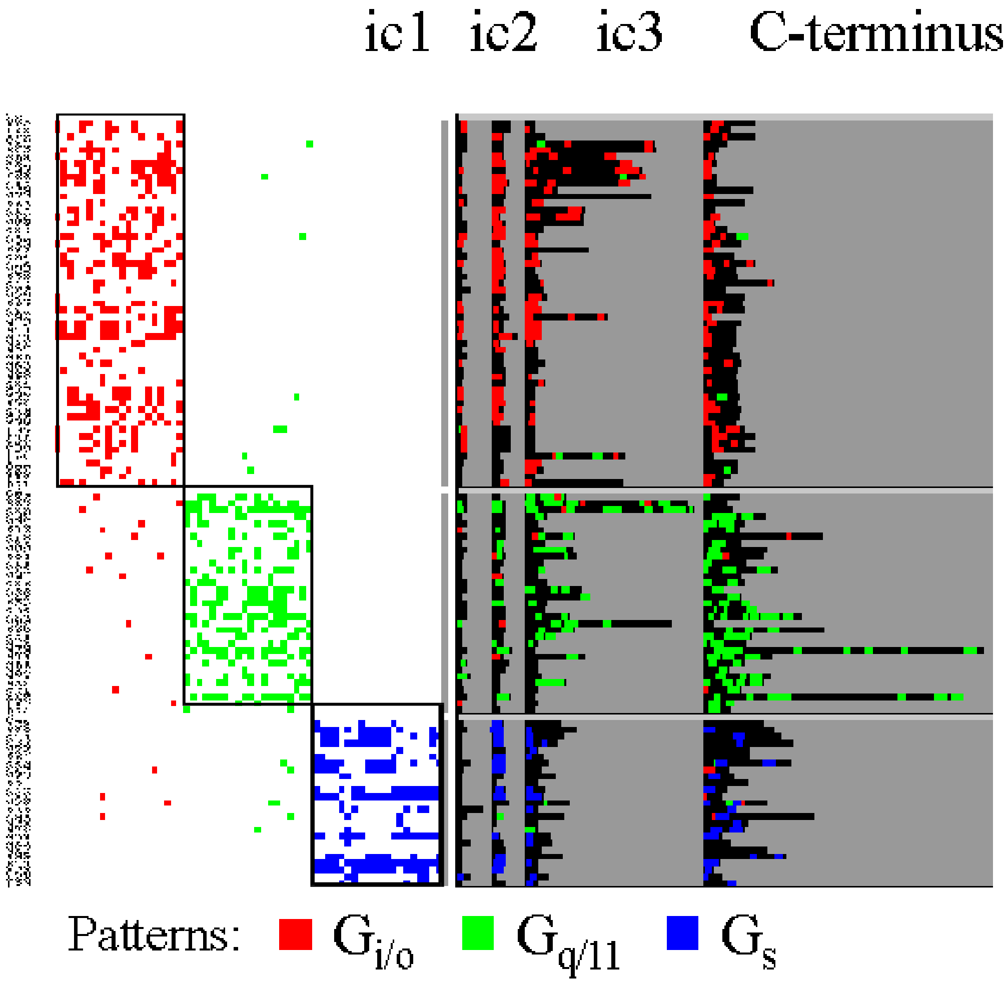


Figure 45: Visualisation of the intracellular positions of pattern matches

On the right side of the figure the sequences of all intracellular loop regions are displayed as black horizontal bars, proportional the sequence length, ordered by their coupling specificity. The left side shows all patterns that match a specific sequence, again ordered and coloured according to their coupling specificity. Ic1, ic2, ic3 and C-terminus stand for the intracellular loops. The formation of blocks on the left side is evidence for the quality of the patterns.

A visual inspection of the patterns confirmed the numerical analysis (as tabulated). The tool PATMATCH, part of the Expression Profiler package (Vilo et al., in preparation), was used for this purpose. It facilitates visualising pattern matches upon the sequences, having grouped the latter by their G protein coupling specificity (see Figure 44). Most of the patterns were seen to match in just one of the three groups of receptor sequences, with few matches to the other two groups, demonstrating their usefulness and specificity.

Additionally all of the GPCR sequences were matched by at least a few patterns. PATMATCH also allows determining where the patterns matched onto the intracellular domains of the receptor sequences, and from this to deduce whether match positions are conserved both within a particular receptor-G protein-coupling group, and between the three groups.

Figure 46 shows that low numbers of matches are found for about 20% of the sequences in the training set. Particularly, match totals in the range 31-100 are of concern. This may have resulted from a rather limited number of patterns being found in these sequences. No attempt was made to reduce redundancy in the selected patterns (Table 22), athough this might have helped to reduce bias in the number of patterns available per sequence. Another possible confounding factor is an inaccuracy in the membrane topology prediction for particular receptor sequences, which would presumably constrain the pattern discovery to portions of the molecule that are unlikely to govern G protein coupling specificity. The predictions are dependent upon the accuracy of the receptor-G protein coupling specificity information summarised in (TiPS 2000). It is possible that some useful patterns might have been lost due to an incorrect or promiscuous coupling assignment.

SWISS-PROT Accession	Class	Hits (class/total)	Protein description
Q9Y5N1	$G_{i/o}$	125 / 130	HISTAMINE H3 RECEPTOR
P49190	G_s	123 / 124	PARATHYROID HORMONE RECEPTOR
Q03431	G_s	123 / 132	PARATHYROID HORMONE/PARATHYROID HORMONE-RELATED PEPTIDE RECEPTOR
Q02643	G_s	123 / 124	GROWTH HORMONE-RELEASING HORMONE RECEPTOR
O95838	G_s	181 / 192	GLUCAGON-LIKE PEPTIDE 2 RECEPTOR
P41180	$G_{q/11}$	11 / 14	EXTRACELLULAR CALCIUM-SENSING RECEPTOR
P47872	G_s	123 / 124	SECRETIN RECEPTOR
P43220	G_s	125 / 142	GLUCAGON-LIKE PEPTIDE 1 RECEPTOR
P48546	G_s	124 / 140	GASTRIC INHIBITORY POLYPEPTIDE RECEPTOR
P25105	$G_{q/11}$	4 / 7	PLATELET ACTIVATING FACTOR RECEPTOR
O43613	$G_{q/11}$	52 / 56	OREXIN RECEPTOR TYPE 1

Table 23: Predictions for 10 sequences that are unrelated to the training set

Column 1 lists the SWISS-PROT or TrEMBL accession numbers, column 2 the predicted receptor-G protein coupling. Column 3 shows two numbers, the total number of matches and the number of matches contributed by the pairs and triplets to the predicted class. Column 4 shows the description of the protein.

Verification of classification on novel GPCR sequences.

The classifier was applied to the sequences of 10 receptor subtypes not present in the training set, as shown in Table 2. Pairwise alignments revealed that these test sequences were in general 30-40% identical to their most similar paralogue in the training set. All 10 predictions appeared to be correct after consulting the primary literature. With reference to Figure 46, no trust is in predictions that are based on less than 50 matches. It seemed surprising that the predictions for both P41180 and P25105 were correct given that they resulted from a rather low number of matches.

In order to determine whether pattern discovery was strictly necessary for correct prediction, in addition the simpler approach was taken of building a dendrogram from a multiple sequence alignment of the inner domains of the training set sequences. There does seem to be propensity for receptors with the same coupling preference to have a high sequence similarity, reflected by low distances in the tree. However, the delineation between the three groups of receptors was far from distinct.

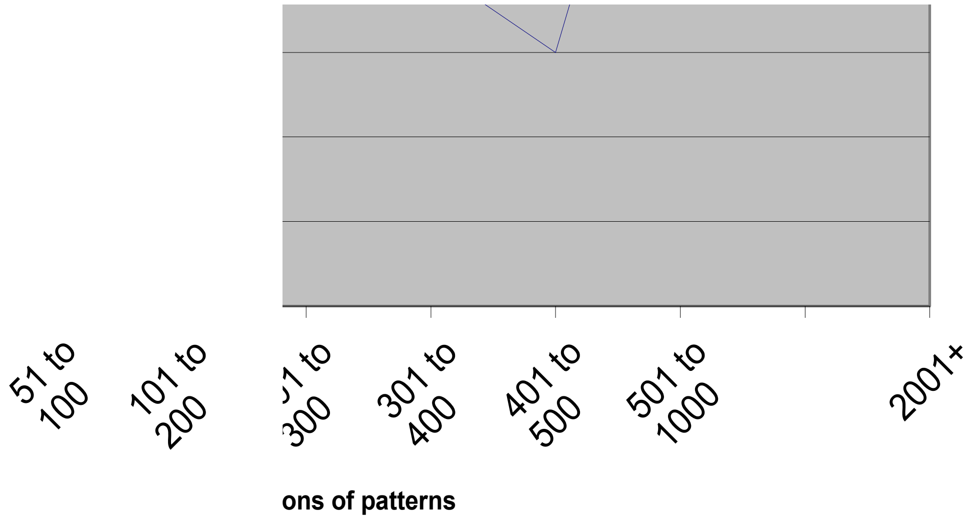


Figure 46: Distribution of the number of combinations

The graph summarises, how often in the training set a certain number of combinations is found to match a single sequence.

Positions of patterns on the sequence

To go further with the study of patterns determined, a concrete analysis of their positions on the sequence is required. The following sections investigate the redundancy of individual patterns that is expected to be reflected by a similarity in the occurrences of individual patterns within the GPCR sequences. Of special interest are the constraints that are imposed on individual residues within a matched region that are visualised in Figure 47.

I….RY	ICVPLRY	I….RY….T	ICVPLRYKSLVT	I....R[FWY]	ICVPLRY
	IFHALRY		IFHALRYHSIVT		IFHALRY
	IFYALRY		IFYALRYHHIMT		IFYALRY
	IFYALRY		IFYALRYHSIMT		IFYALRY
	IFYALRY		IFYALRYHSIVT		IFYALRY
	ILSPLRY		ILSPLRYKLRMT		ILSPLRY
	IRIPLRY		IRIPLRYNGLVT		IRIPLRY
	ISRPFRY		ISRPFRYKRKMT		ISRPFRY
	ISSPFRY		ISSPFRYERKMT		ISSPFRY
	ITSPFRY		ITSPFRYQSLLT		ITSPFRY
	IYVILRY		IYVILRYAKMKT		IYVILRY

Table 24: Patterns and the subsequences these match

The table helps to understand the redundancy between the patterns. Columns 1,3 and 5 show a pattern and next to it in columns 2,4 and 6 the matches for these patterns are listed in different sequences. The list of matches is complete for all the three patterns presented.

The position of a pattern on receptor sequences also determines the residues matched. Table 24 shows such matched sequences for a small selection of very similar patterns. A similarity score between patterns can be defined on this information about matched residues rather than the pattern’s lexical representation as a regular expression. The similarity of two patterns is calculated as the sum of residues that patterns share when applied on all sequences.

Table 24 demonstrates the redundancy between some of the patterns as identical regions are matched. The weakening of the first pattern by the substitution of the final tyrosine for a selection of aromatic residues that include tyrosine in order to yield the third pattern does not seem justified on the basis of this training set as neither more sequences are matched nor the length of the match was extended. Under the same considerations, the addition of threonine seems very reasonable.

SWISS-PROT/TrEMBL Acc.-Nr.: Q01726	
ID	MSHR_HUMAN
Sequence	AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW
G_{io}	AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW
G_s	AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW AKNRNLHSPMY@ISIFYALRYHSIVTLPRARRA@LARACQHAQGIARLHKRQRPV@FHSQELRRTLKEVLTCSW
$G_{q/11}$	<no matches>
Assigned G protein	G_s
Frequencies	G_s : 6 G_{io} : 1

Figure 47: Visualisation of matches on the protein sequence

Shown are the raw amino acid sequence of the cytoplasmic loops, separated by the @ sign. The loops include one residue to the N and C terminus that is predicted as transmembrane. Fragments of the protein sequence that are matched by a certain pattern are highlighted. Residues that were matched by a wildcard in the pattern were *left grey, while a residue determined by a selection of residues, e.g. [ILV], are shown in yellow. Residues that are explicitly requested by the pattern are depicted in red.*

Figure 47 visualises that different patterns may be very similar in the regions they match on the protein sequence and that these impose constraints on the same set of residues within these regions.

Patterns that span multiple loops were not considered at all for the training since the prediction was not assumed to vary in the assignment. The patterns are very much limited in maximum length and hence have problems to ignore too many irrelevant intramembranous characters. A representation of patterns as HMMs rather than regular expressions would allow a greater flexibility since gaps can be modelled to vary in length. Hence, it seems surprising that some patterns have a tendency to stretch across multiple inner loops, i.e. over the membrane boundary. For best results, the pattern finding and variables in the loop-determination need to be synchronised.

	7TMHMM annotation	1 residue of TM accompanies inner loops	2 residues	3 residues	4 residues
Pattern matches	1838	1958	2046	2151	2192
Spanning domains	232	224	237	226	172
Percentage	12.6	11.4	11.6	10.5	7.86

Table 25: The number of matches in dependence on the number of transmembrane residues

The columns reflect the number of transmembrane residues that are presented to the classifier, reaching from 0 to 4 on either side of a cytoplasmic loop. The first row presents the total number of matches of patterns, not their combinations, to the sequences. The second the number of those matches that span across the separating symbol for the membrane (represented by the character @ in Figure 47), i.e. represent patterns that span two intracellular domains. The last row shows the percentage of multi-loop-spanning matches of the total number of matches.

Table 25 shows that the number of matching patterns raises most sharply with addition of a single transmembrane residue and again with the addition of another two. This is not too surprising. Especially the DRY motif sits directly at the end of a transmembrane helix and the definition of transmembrane borders is not sharp, also other motifs have a tendency to seek either the N- or C-terminal portion of a loop. It can be found that the number of multi-loop-spanning patterns decreases with an increasing number of transmembrane residues being added to the inner loops.

Still, the concept of a pattern that spans multiple loops and that depends on the size of the gap between the helices are of interest. These would support the idea that a rearrangement of the helices, and thereby a change in the distance between the end of helices, is directly influencing the coupling. A secondary structure prediction is currently performed as it may explain the high frequency and clustered positioning of wild cards in the regular expressions, e.g. by emphasising a specific direction in a putative helical arrangements of the coupling sites. This is still investigated.

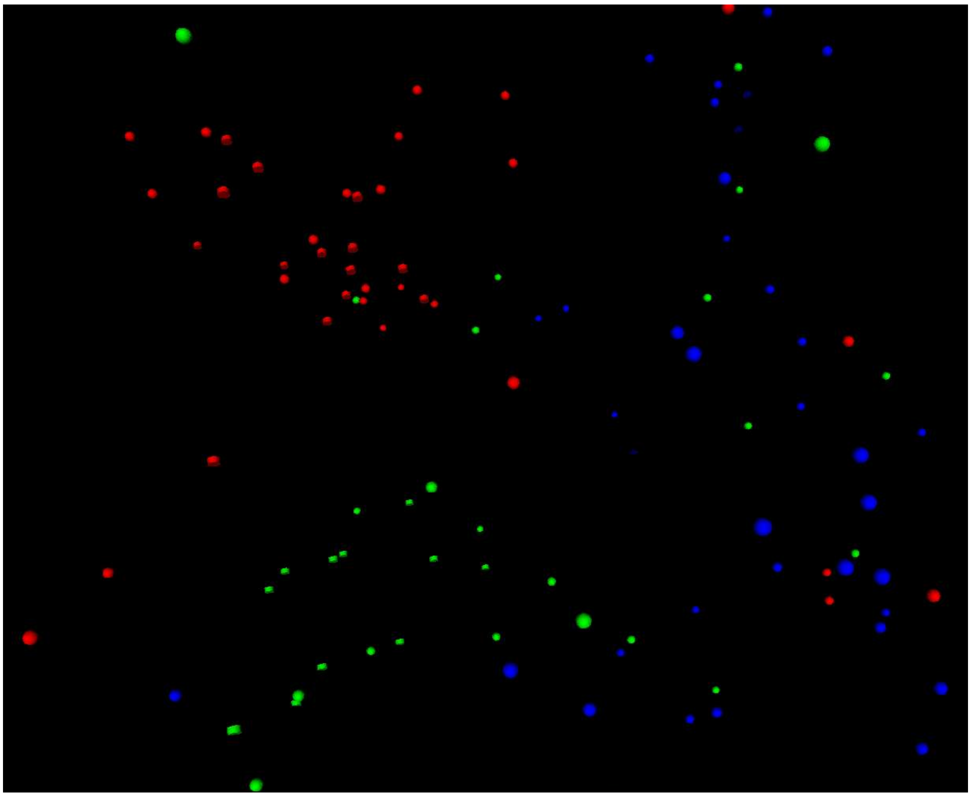


Figure 48: 3D Visualisation of pattern similarities

The similarity of patterns was determined pairwise on the basis of residues shared in their matches to sequences in the test set. As a distance matrix these were passed to the *program SpaceExplorer (Gilbert, Schroeder et al. 2000). Patterns indicating a coupling to G_s are shown in red, $G_{i/o}$ in green and $G_{q/11}$ in blue.*

Figure 48 visualises the similarity of patterns. Obviously, patterns describing the same coupling mechanisms cluster together. The figure also shows, especially for Gi/o coloured in red, that there are subgroups of patterns. It is yet unknown if these result from patterns describing matches in specific intracellular loops or if these code for specifc G-protein subunits.

Table 26 answers which intracellular loop is most determinant for the coupling preference, according to the distribution of matches on the sequences. The C-terminus, to which most residues have been assigned, has the least number of matches (5%). The first (ic1) and third loop (ic3) differ in only 0.3% of the matches with 33.5% and 33.2%, respectively. The second intracellular loop (ic2) has only slightly less with 28.2%. However, in relation with the number of residues of the intracellular loops, the matches of patterns occur 2 times more often in ic1 than in ic2, 4.5 times more often than in ic3 and 37 times more often than in the C-terminus. This speaks against a random distribution of matches.

Domain	ic1	ic2	ic3	C-Terminus
Number of residues	1330	2273	5742	7458
Number of matches	616	518	610	94
#residues/#matches	2.16	4.39	9.41	79.3
Percent of matches	33.5	28.2	33.2	5.1

Table 26: Distribution of patterns over intracellular domains

The first row shows the number of amino acids representing the individual intracellular moieties in the test set's GPCRs. The second row displays the number of matches that occurred in the respective loop, which are also given in percent just underneath.

Discussion

Patterns have been found within GPCR sequences, which seem to be involved in determining their selectivity to different functional classes of G proteins. The pattern discovery process focuses on the intracellular domains of GPCRs (previously reported to be involved in the receptor-G protein interaction) by using a novel membrane topology prediction algorithm designed specifically for GPCRs. The resulting patterns were employed in a combinatorial manner to build a classifier, allowing one to predict the G protein coupling specificity of "unknown" receptor sequences that were not present in the training set.

It could be assumed that the pairwise sequence similarity of intracellular domains by itself with no further pattern determination could be used to derive a grouping of different coupling mechanisms. Such a dendrogram is shown in Figure 49. It does not allow an assignment of larger subtrees to a specific coupling mechanism..

The dependency of the classification on a prior determination of the analysed protein sequence as a GPCR implies a context-dependence for the usage of the patterns. This a-priori knowledge can be derived from protein domain databases like PRINTS or PFAM, from the results of similarity searches, or can be read directly from the manual annotations present in databases such as SWISS-PROT. Additionally, the patterns should be employed in the context of the prediction of receptor sequence's membrane topology. With the increasing modularity of large-scale annotation efforts (Fleischmann, Möller et al. 1999; Möller, Leser et al. 1999; Ensembl 2001) such contextual information can now be technically incorporated into genome annotation. This work represents a context-dependent annotation of protein domains.

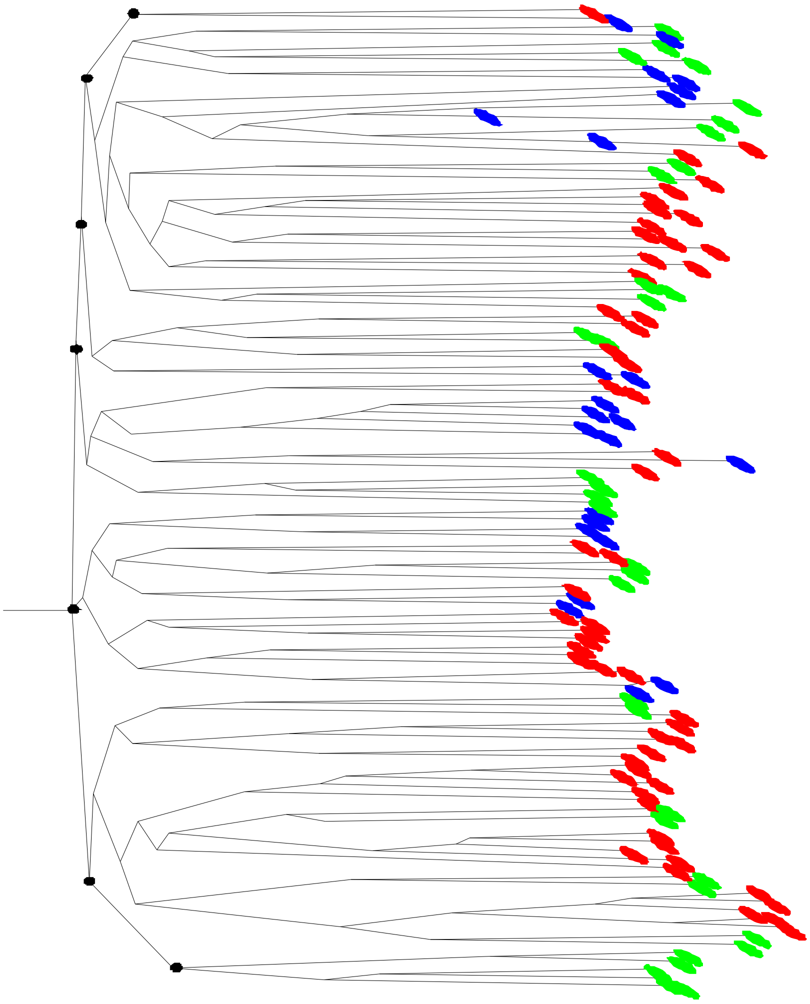


Figure 49: Dendrogram of cytoplasmic residues of GPCRs

The Gi/o class is depicted in red, Gq/11 in green, Gs in blue. A clustering can be found, but is inferior to the descriptiveness of the patterns as shown in Figure 45 (Courtesy M. D. R. Croning).

Clearly many aspects of the interaction between a receptor and its G protein(s) remain to be investigated. The extension of the method towards a modelling of promiscuous G protein coupling may be possible, while these should be carefully avoided during the training process, expecting patterns representing different coupling mechanisms to match a sequence. Additionally, it would be worthwhile determining whether unique interaction motifs exist for promiscuous coupling in receptors that have been demonstrated to lack selectivity in their G protein interactions. It was not attempted to construct patterns for the exclusion of certain G protein couplings, i.e. a pattern to represent an exception to a rule.

The approach could eventually be improved by ignoring any pattern combination that does not span at least two inner loops. The reason behind this is, that from prior biochemical investigations it is unlikely that a match to a single loop would be sufficient to provide an effective and selective G protein interaction (Wess 1998). Similarly, whether additional predictive power can be gained from analysis of where the patterns match in the context of a particular intracellular domain and their distance from the membrane remains to be investigated. Receptor-G protein recognition is known to be regulated by both post-transcriptional and post-translational modifications, likely of both the GPCR and the G-protein heterotrimer (Wess 1998). Analysing just the translated receptor coding sequence does not allow us to model such events. However, the discovered patterns are sensitive and selective enough to construct a useful predictor.

In conclusion, this approach uses protein sequence data to find patterns that from which GPCR signalling can be inferred. The applications of such patterns range from predicting the G protein coupling preferences of newly cloned receptors to designing biochemical experiments to increase the understanding of the molecular basis of the coupling of receptors with G proteins. Developing and improving functional predictions between interacting proteins, and the subsequent reconstruction of cellular pathways, constitutes one of the key challenges to bioinformatics as the post-genomic era is entered.

The G_z family of G Proteins, relevant in oncogenesis, is now regarded as playing a separate role from the G_i family to which these were previously assigned (Ho and Wong 2001). The coupling prediction will respectively be extended for this fourth coupling mechanism.

This expert-predictor on properties of GPCRs also closes the link to EDITtoTrEMBL. A static dependency that requires all proteins of TrEMBL to pass the tool only would be unappropriate.

VI.Summary and Discussion

A. Overview

In this thesis, work on automated sequence annotation was presented. Most of the work is published or will be published soon:

- the creation of an automated sequence annotation framework,
- the integration of logic programming for the conflict resolution,
- the creation of a test set of transmembrane proteins with the storage of biochemical data to allow subsequent automated reasoning,
- the evaluation of current transmembrane prediction methods,
- the automated creation of rules for the construction of constraints for transmembrane annotation from protein domain databases,
- the automated creation of Hidden-Markov-Models for protein families with know topology
- and finally the use of transmembrane topology prediction for the mining for patterns to predict protein-protein interactions.

Milestones of this work were the creation of an agent system for protein annotation with the combination of the automated workflow determination with a conflict resolution mechanism.

The test set of transmembrane proteins is of special importance since it allowed the evaluation of TM prediction methods, a prerequisite for the remainder of this thesis.

B. This work and the future of Automated Sequence Annotation

1. Distributed computing becomes routine

Flood of data and an increasing specialisation of tools

Nobody predicts the number of sequences submitted to nucleotide sequence databases to decrease over the next decade, in the contrary. The same holds for structure databases to be derived and almost any other molecular database. Many new and very different proteins should be expected that make the automation of annotation more complex and difficult. Regarding the process of automated sequence annotation this will be reflected in an increase of number and degree of specialisation of tools for annotation.

Global collaboration

The biochemical research community shares clones and antibodies for decades. In addition, the bioinformatics community is strong in sharing infrastructure, especially manifested in the Open Source community.

It should be expected that this trend of collaboration is extended towards a further distribution of processes for annotation. Today research groups promote their work on web pages, offering submission pages to their software or make them available for local installation. .Additionally, principles as presented in this thesis for distributed computing are expected to be implemented to achieve a symbiois of respective local expertise and resources.

Change in computer infrastructure

For large-scale automated sequence annotation the individual units can be treated independently. While multi-processor machines do have the advantage of a very fast communication between agents, the cost of communication is little when compared to the cost of computation. Therefore, clusters of low cost PCs substitute big mainframes very successfully.

More biological knowledge will be computer accessible and understandable

With the advent of the GeneOntology, a very important first step towards a formal representation of biological knowledge is made. It is still far away from being fully usable as a general exchange of biological information, especially since no grammar for the use of the ontology was suggested. This is due to the paradox that the driving force behind a more formal representation of biological knowledge is less the desire for a deeper integration of computational tools, but a more consistent representation of knowledge in sequence databases.

The developers of GO seem very open towards extending its current scope of application and therefore GO should be regarded as a very promising path towards a nice integration of tools and databases in molecular biology.

Summary

Above comments should support the prospect of a logic-supported distributed annotation environment. The Environment of Distributed Information Transfer to TrEMBL (EDITtoTrEMBL) with its symbiosis of the revision system (Revise) may be perceived as a first implementation of such a system.

2. Importance of constraint-based techniques

Constraints and conflict resolution applied to the protein annotation process represents the central innovation of this thesis. It is my perception that these might almost be taken for granted in near future when more formal ways to express biological knowledge have been both developed and established.

Until then, the constraints will, as they did in this work or in HMMTOP (Tusnády and Simon 2001), be used on specialised topics like the membrane protein topology. In HMMTOP constraints become a central means of communication of the experimentalist with the program performing the actual annotation, to share knowledge on the protein.

3. Future sources to be integrated

In this work, only sequence-derived information was addressed. The EBI together with an international group of researchers works on the project named *integr8,* which suggest to integrate the most eminent molecular databases, i.e. the protein-protein interaction data, structural information and the achievements transcriptome and proteome efforts. Interestingly, at this stage there is no formalism available to perform queries on a higher level than the actual stored data. Such semantic queries, as addressed in a poster shown in the appendix (Figure 51), are not supported.

During the time at the EBI, I was given the opportunity to address three main areas of bioinformatics:

- integration of data and tools (*EDITtoTrEMBL*)
- provision of data by formalisation of information published (*transmembrane proteins*)
- biological inference from gathered data (topology predictor and GPCR-G protein coupling prediction)

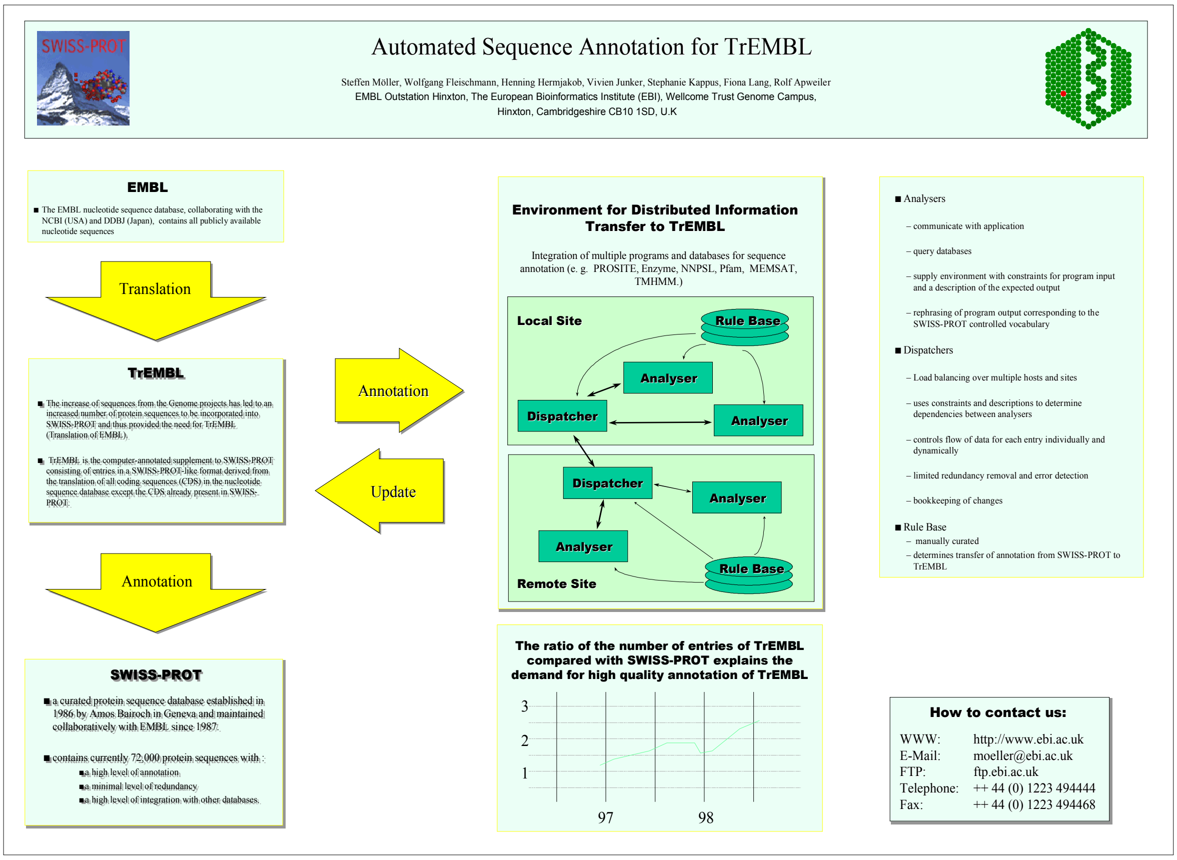
It is my perception that this work has been of help to become aware of the prospects of an integrated world of computational molecular biology**.**

VII.Appendix

A. Posters

1. EDITtoTrEMBL

The poster was shown on the 1998 ISMB in Montreal. It explains the interaction of analysers and the annotation process of TrEMBL in general.



Automated Sequence Annotation for TrEMBL

Steffen Möller, Wolfgang Fleischmann, Henning Hermjakob, Vivien Junker, Stephanie Kappus, Fiona Lang, Rolf Apweiler
EMBL Outstation Hinxton, The European Bioinformatics Institute (EBI), Wellcome Trust Genome Campus,
Hinxton, Cambridgeshire CB10 1SD, U.K

EMBL

- The EMBL nucleotide sequence database, collaborating with the NCBI (USA) and DDBJ (Japan), contains all publicly available nucleotide sequences

TrEMBL

- The increase of sequences from the Genome projects has led to an increased number of protein sequences to be incorporated into SWISS-PROT and thus provided the need for TrEMBL (Translation of EMBL).
- TrEMBL is the computer-annotated supplement to SWISS-PROT consisting of entries in a SWISS-PROT-like format derived from the translation of all coding sequences (CDS) in the nucleotide sequence database except the CDS already present in SWISS-PROT.

SWISS-PROT

- a curated protein sequence database established in 1986 by Amos Bairoch in Geneva and maintained collaboratively with EMBL since 1987.
- contains currently 72,000 protein sequences with :
  - a high level of annotation
  - a minimal level of redundancy
  - a high level of integration with other databases.

Environment for Distributed Information Transfer to TrEMBL

Integration of multiple programs and databases for sequence annotation (e. g. PROSITE, Enzyme, NNPSL, Pfam, MEMSAT, TMHMM.)

The ratio of the number of entries of TrEMBL compared with SWISS-PROT explains the demand for high quality annotation of TrEMBL

- Analysers
  - communicate with application
  - query databases
  - supply environment with constraints for program input and a description of the expected output
  - rephrasing of program output corresponding to the SWISS-PROT controlled vocabulary
- Dispatchers
  - Load balancing over multiple hosts and sites
  - uses constraints and descriptions to determine dependencies between analysers
  - controls flow of data for each entry individually and dynamically
  - limited redundancy removal and error detection
  - bookkeeping of changes
- Rule Base
  - manually curated
  - determines transfer of annotation from SWISS-PROT to TrEMBL

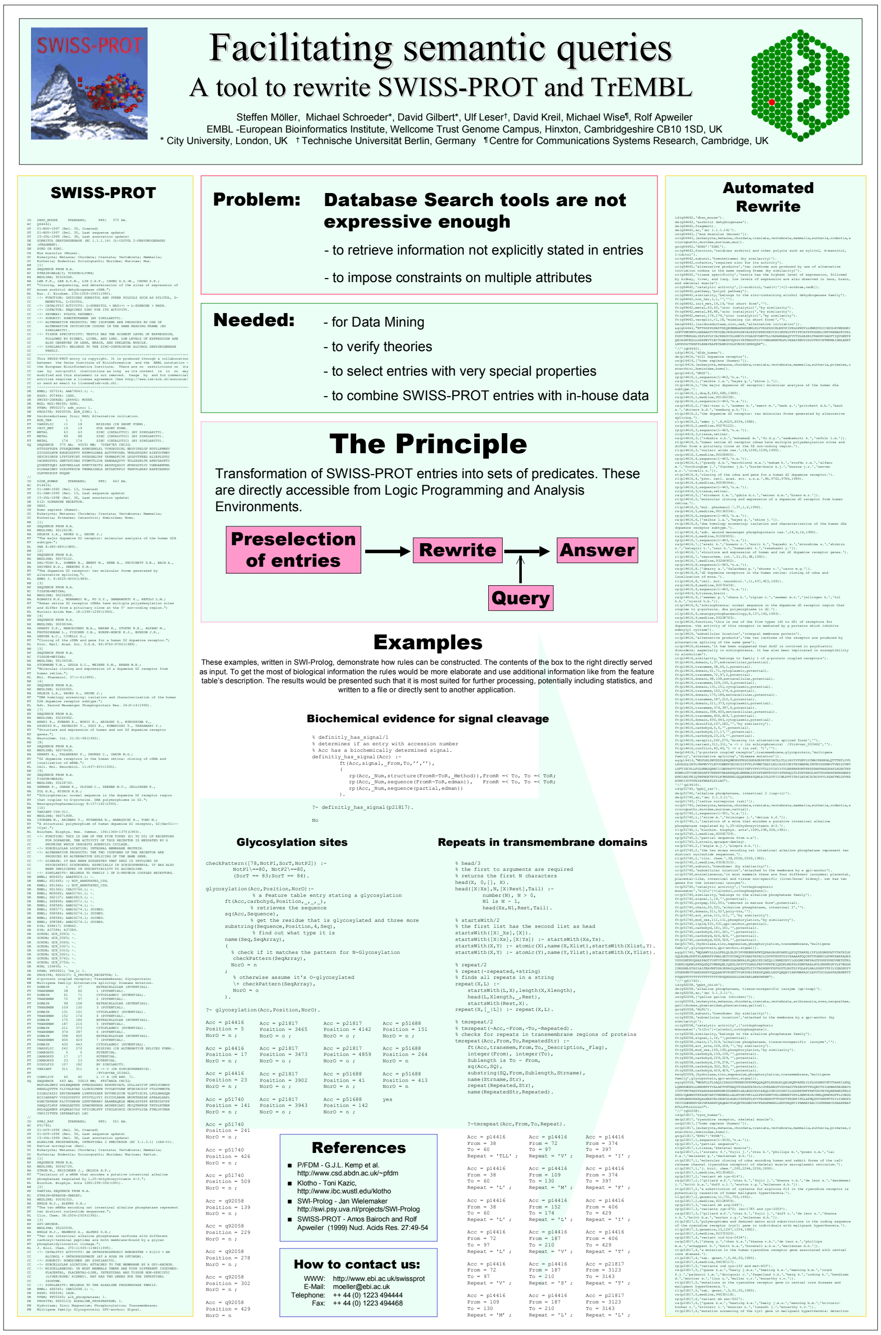
How to contact us:

WWW:	http://www.ebi.ac.uk
E-Mail:	moeller@ebi.ac.uk
FTP:	ftp.ebi.ac.uk
Telephone:	++ 44 (0) 1223 494444
Fax:	++ 44 (0) 1223 494468

Figure 50: Poster for the presentation of EDITtoTrEMBL

2. Facilitating Semantic Queries

Shown on the German Conference in Bioinformatics 1999 this poster explained the use of Logic Programming to search in flat file representations of bioinformatics databases.



Facilitating semantic queries

A tool to rewrite SWISS-PROT and TrEMBL

Steffen Möller, Michael Schroeder*, David Gilbert*, Ulf Leser†, David Kreil, Michael Wise¶, Rolf Apweiler
EMBL -European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridgeshire CB10 1SD, UK
* City University, London, UK † Technische Universität Berlin, Germany ¶Centre for Communications Systems Research, Cambridge, UK

SWISS-PROT

Problem: Database Search tools are not expressive enough

- to retrieve information not explicitly stated in entries
- to impose constraints on multiple attributes

Needed:

- for Data Mining
- to verify theories
- to select entries with very special properties
- to combine SWISS-PROT entries with in-house data

The Principle

Transformation of SWISS-PROT entries into sets of predicates. These are directly accessible from Logic Programming and Analysis Environments.

Automated Rewrite

Examples

These examples, written in SWI-Prolog, demonstrate how rules can be constructed. The contents of the box to the right directly served as input. To get the most of biological information the rules would be more elaborate and use additional information like from the feature table's description. The results would be presented such that it is most suited for further processing, potentially including statistics, and written to a file or directly sent to another application.

Biochemical evidence for signal cleavage

```
% definitly_has_signal/1
% determines if an entry with accession number
% Acc has a biochemically determined signal.
definitly_has_signal(Acc) :-
        ft(Acc,signal,_From,To,'',''),
        (
           rp(Acc,_Num,structure(FromR-ToR,_Method)),FromR =< To, To =< ToR;
           rp(Acc,_Num,sequence(FromR-ToR,edman)),   FromR =< To, To =< ToR;
           rp(Acc,_Num,sequence(partial,edman))
        ).

?- definitly_has_signal(p21817).

No
```

Glycosylation sites

```
checkPattern([78,NotP1,SorT,NotP2]) :-
        NotP1\==80, NotP2\==80,
        (SorT == 83;SorT == 84).

glycosylation(Acc,Position,NorO):-
              % a Feature table entry stating a glycosylation
      ft(Acc,carbohyd,Position,_,_,_),
              % retrieves the sequence
      sq(Acc,Sequence),
              % get the residue that is glycosylated and three more
      substring(Sequence,Position,4,Seq),
              % find out what type it is
      name(Seq,SeqArray),
      (
        % check if it matches the pattern for N-Glycosylation
         checkPattern(SeqArray),
        NorO = n
      ;
         % otherwise assume it's O-glycosylated
         \+ checkPattern(SeqArray),
         NorO = o
      ).

?- glycosylation(Acc,Position,NorO).

Acc = p14416     Acc = p21817     Acc = p21817     Acc = p51688
Position = 5     Position = 3465  Position = 4142  Position = 151
NorO = n ;       NorO = o ;       NorO = o ;       NorO = n ;

Acc = p14416     Acc = p21817     Acc = p21817     Acc = p51688
Position = 17    Position = 3473  Position = 4859  Position = 264
NorO = n ;       NorO = o ;       NorO = o ;       NorO = n

Acc = p14416     Acc = p21817     Acc = p51688     Acc = p51688
Position = 23    Position = 3902  Position = 41    Position = 413
NorO = n ;       NorO = o ;       NorO = n ;       NorO = n

Acc = p51740     Acc = p21817     Acc = p51688     yes
Position = 141   Position = 3943  Position = 142
NorO = n ;       NorO = o ;       NorO = n ;

Acc = p51740
Position = 241
NorO = n ;

Acc = p51740
Position = 426
NorO = n ;

Acc = p51740
Position = 509
NorO = n ;

Acc = q92058
Position = 139
NorO = n ;

Acc = q92058
Position = 229
NorO = n ;

Acc = q92058
Position = 278
NorO = n ;

Acc = q92058
Position = 302
NorO = n ;

Acc = q92058
Position = 429
NorO = n
```

Repeats in transmembrane domains

```
% head/3
% the first to arguments are required
% returns the first N characters
head(X, 0,[], X).
head([X|Xs],N,[X|Rest],Tail) :-
        number(N), N > 0,
        N1 is N - 1,
        head(Xs,N1,Rest,Tail).

% startsWith/2
% the first list has the second list as head
startsWith([X|_Xs],[X]).
startsWith([X|Xs],[X|Ys]) :- startsWith(Xs,Ys).
startsWith(X,Y) :- atomic(X),name(X,Xlist),startsWith(Xlist,Y).
startsWith(X,Y) :- atomic(Y),name(Y,Ylist),startsWith(X,Ylist).

% repeat/2
% repeat(-repeated,+string)
% finds all repeats in a string
repeat(X,L) :-
    startsWith(L,X),length(X,Xlength),
    head(L,Xlength,_,Rest),
    startsWith(Rest,X).
repeat(X,[_|L]) :- repeat(X,L).

% tmrepeat/2
% tmrepeat(-Acc,-From,-To,-Repeated).
% checks for repeats in transmembrane regions of proteins
tmrepeat(Acc,From,To,RepeatedStr) :-
    ft(Acc,transmem,From,To,_Description,_Flag),
    integer(From), integer(To),
    Sublength is To - From,
    sq(Acc,SQ),
    substring(SQ,From,Sublength,Strname),
    name(Strname,Str),
    repeat(Repeated,Str),
    name(RepeatedStr,Repeated).



    ?-tmrepeat(Acc,From,To,Repeat).

    Acc = p14416      Acc = p14416     Acc = p14416
    From = 38         From = 72        From = 374
    To = 60           To = 97          To = 397
    Repeat = 'TLL' ;  Repeat = 'V' ;   Repeat = 'I' ;

    Acc = p14416      Acc = p14416     Acc = p14416
    From = 38         From = 109       From = 374
    To = 60           To = 130         To = 397
    Repeat = 'L' ;    Repeat = 'M' ;   Repeat = 'F' ;

    Acc = p14416      Acc = p14416     Acc = p14416
    From = 38         From = 152       From = 406
    To = 60           To = 174         To = 429
    Repeat = 'L' ;    Repeat = 'L' ;   Repeat = 'I' ;

    Acc = p14416      Acc = p14416     Acc = p14416
    From = 72         From = 187       From = 406
    To = 97           To = 210         To = 429
    Repeat = 'L' ;    Repeat = 'V' ;   Repeat = 'T' ;

    Acc = p14416      Acc = p14416     Acc = p21817
    From = 72         From = 187       From = 3123
    To = 97           To = 210         To = 3143
    Repeat = 'V' ;    Repeat = 'S' ;   Repeat = 'T' ;

    Acc = p14416      Acc = p14416     Acc = p21817
    From = 109        From = 187       From = 3123
    To = 130          To = 210         To = 3143
    Repeat = 'M' ;    Repeat = 'L' ;   Repeat = 'L' ;
```

References

- P/FDM - G.J.L. Kemp et al. http://www.csd.abdn.ac.uk/~pfdm
- Klotho - Toni Kazic, http://www.ibc.wustl.edu/klotho
- SWI-Prolog - Jan Wielemaker http://swi.psy.uva.nl/projects/SWI-Prolog
- SWISS-PROT - Amos Bairoch and Rolf Apweiler (1999) Nucl. Acids Res. 27:49-54

How to contact us:

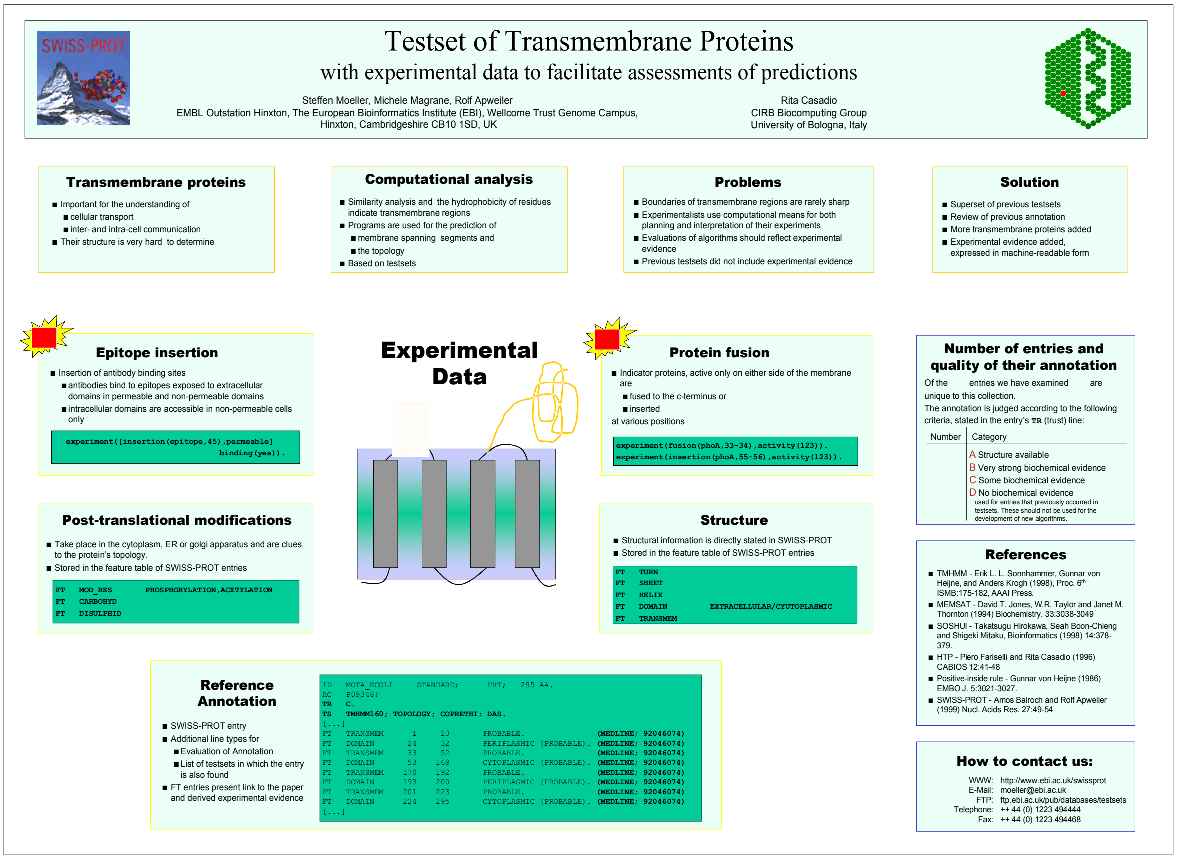
WWW: http://www.ebi.ac.uk/swissprot
E-Mail: moeller@ebi.ac.uk
Telephone: ++ 44 (0) 1223 494444
Fax: ++ 44 (0) 1223 494468

Figure 51: Presentation of idea to facilitate semantic queries with PROLOG

The poster shows how from a range of SWISS-PROT entries (left side) a list of predicates could be derived that are then used as an input for PROLOG. This work was a antecedent for the conflict resolution presented in this thesis.

3. Test set of Transmembrane Proteins

The poster was first shown on the ISMB 1999. A year later it was shown on the conference on membrane proteins in Amsterdam. The motivation was to bring the work closer to biologists working on membrane proteins.



Testset of Transmembrane Proteins

with experimental data to facilitate assessments of predictions

Steffen Moeller, Michele Magrane, Rolf Apweiler
EMBL Outstation Hinxton, The European Bioinformatics Institute (EBI), Wellcome Trust Genome Campus,
Hinxton, Cambridgeshire CB10 1SD, UK

Rita Casadio
CIRB Biocomputing Group
University of Bologna, Italy

Transmembrane proteins

- Important for the understanding of
  - cellular transport
  - inter- and intra-cell communication
- Their structure is very hard to determine

Computational analysis

- Similarity analysis and the hydrophobicity of residues indicate transmembrane regions
- Programs are used for the prediction of
  - membrane spanning segments and
  - the topology
- Based on testsets

Problems

- Boundaries of transmembrane regions are rarely sharp
- Experimentalists use computational means for both planning and interpretation of their experiments
- Evaluations of algorithms should reflect experimental evidence
- Previous testsets did not include experimental evidence

Solution

- Superset of previous testsets
- Review of previous annotation
- More transmembrane proteins added
- Experimental evidence added, expressed in machine-readable form

Epitope insertion

- Insertion of antibody binding sites
  - antibodies bind to epitopes exposed to extracellular domains in permeable and non-permeable domains
  - intracellular domains are accessible in non-permeable cells only

```
experiment([insertion(epitope,45),permeable]
                                binding(yes)).
```

Post-translational modifications

- Take place in the cytoplasm, ER or golgi apparatus and are clues to the protein's topology.
- Stored in the feature table of SWISS-PROT entries

```
FT   MOD_RES       PHOSPHORYLATION,ACETYLATION
FT   CARBOHYD
FT   DISULPHID
```

Experimental Data

Protein fusion

- Indicator proteins, active only on either side of the membrane are
  - fused to the c-terminus or
  - inserted

at various positions

```
experiment(fusion(phoA,33-34),activity(123)).
experiment(insertion(phoA,55-56),activity(123)).
```

Structure

- Structural information is directly stated in SWISS-PROT
- Stored in the feature table of SWISS-PROT entries

```
FT   TURN
FT   SHEET
FT   HELIX
FT   DOMAIN          EXTRACELLULAR/CYUTOPLASMIC
FT   TRANSMEM
```

Number of entries and quality of their annotation

Of the entries we have examined are unique to this collection.

The annotation is judged according to the following criteria, stated in the entry's **TR** (trust) line:

Number	Category
	A Structure available
	B Very strong biochemical evidence
	C Some biochemical evidence
	D No biochemical evidence used for entries that previously occurred in testsets. These should not be used for the development of new algorithms.

References

- TMHMM - Erik L. L. Sonnhammer, Gunnar von Heijne, and Anders Krogh (1998), Proc. 6th ISMB:175-182, AAAI Press.
- MEMSAT - David T. Jones, W.R. Taylor and Janet M. Thornton (1994) Biochemistry. 33:3038-3049
- SOSHUI - Takatsugu Hirokawa, Seah Boon-Chieng and Shigeki Mitaku, Bioinformatics (1998) 14:378-379.
- HTP - Piero Fariselli and Rita Casadio (1996) CABIOS 12:41-48
- Positive-inside rule - Gunnar von Heijne (1986) EMBO J. 5:3021-3027.
- SWISS-PROT - Amos Bairoch and Rolf Apweiler (1999) Nucl. Acids Res. 27:49-54

How to contact us:

WWW:	http://www.ebi.ac.uk/swissprot
E-Mail:	moeller@ebi.ac.uk
FTP:	ftp.ebi.ac.uk/pub/databases/testsets
Telephone:	++ 44 (0) 1223 494444
Fax:	++ 44 (0) 1223 494468

Reference Annotation

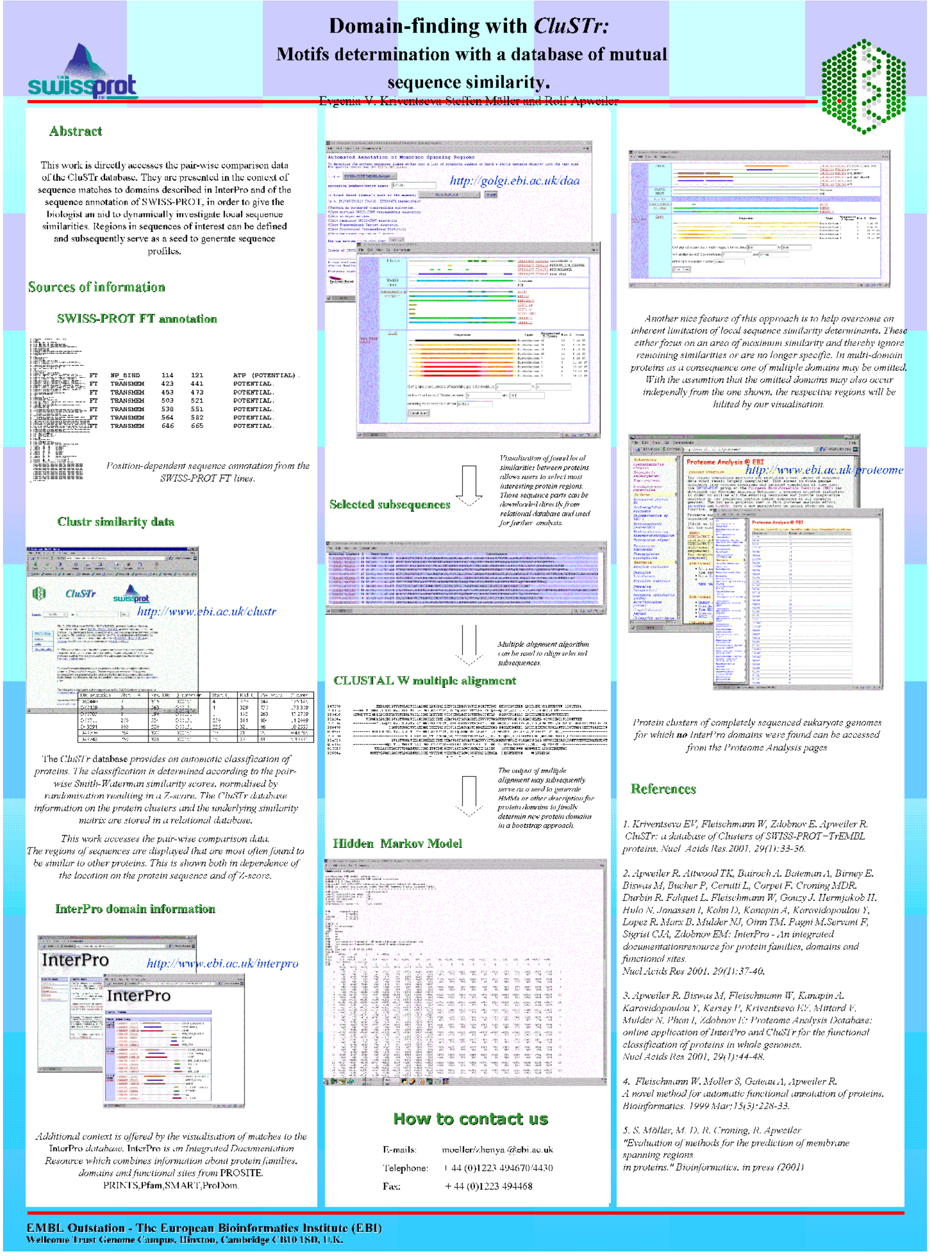
- SWISS-PROT entry
- Additional line types for
  - Evaluation of Annotation
  - List of testsets in which the entry is also found
- FT entries present link to the paper and derived experimental evidence

```
ID   MOTA_ECOLI     STANDARD;      PRT;   295 AA.
AC   P09348;
TR   C.
TS   TMHMM160; TOPOLOGY; COPRETHI; DAS.
[...]
FT   TRANSMEM      1     23       PROBABLE.                  (MEDLINE; 92046074)
FT   DOMAIN       24     32       PERIPLASMIC (PROBABLE).    (MEDLINE; 92046074)
FT   TRANSMEM     33     52       PROBABLE.                  (MEDLINE; 92046074)
FT   DOMAIN       53    169       CYTOPLASMIC (PROBABLE).    (MEDLINE; 92046074)
FT   TRANSMEM    170    192       PROBABLE.                  (MEDLINE; 92046074)
FT   DOMAIN      193    200       PERIPLASMIC (PROBABLE).    (MEDLINE; 92046074)
FT   TRANSMEM    201    223       PROBABLE.                  (MEDLINE; 92046074)
FT   DOMAIN      224    295       CYTOPLASMIC (PROBABLE).    (MEDLINE; 92046074)
[...]
```

Figure 52: Presentation of the collection of well-annotated transmembrane proteins.

4. Domain-finding with CluSTr: Re-occurring motifs determined with a database of mutual sequence similarity

Presented on the ISMB 2001 in Copenhagen, the poster explains the application of the CluSTr project for the hunt for new protein domains. Areas of local similarity are presented together with the most important information from SWISS-PROT/TrEMBL and InterPro.



swissprot
http://golgi.ebi.ac.uk/daa
http://www.ebi.ac.uk/clustr
CluSTr
http://www.ebi.ac.uk/interpro
InterPro
http://www.ebi.ac.uk/proteome
Proteome Analysis @ EBI

Domain-finding with *CluSTr*:
Motifs determination with a database of mutual sequence similarity.

Evgenia V. Kriventseva Steffen Möller and Rolf Apweiler

Abstract

This work is directly accesses the pair-wise comparison data of the CluSTr database. They are presented in the context of sequence matches to domains described in InterPro and of the sequence annotation of SWISS-PROT, in order to give the biologist an aid to dynamically investigate local sequence similarities. Regions in sequences of interest can be defined and subsequently serve as a seed to generate sequence profiles.

Sources of information

SWISS-PROT FT annotation

```
FT   NP_BIND     114    121       ATP (POTENTIAL).
FT   TRANSMEM    423    441       POTENTIAL.
FT   TRANSMEM    453    473       POTENTIAL.
FT   TRANSMEM    503    521       POTENTIAL.
FT   TRANSMEM    530    551       POTENTIAL.
FT   TRANSMEM    564    582       POTENTIAL.
FT   TRANSMEM    646    665       POTENTIAL.
```

Position-dependent sequence annotation from the SWISS-PROT FT lines.

Clustr similarity data

The *CluSTr* database *provides an automatic classification of proteins. The classification is determined according to the pair-wise Smith-Waterman similarity scores, normalised by randomisation resulting in a Z-score. The CluSTr database information on the protein clusters and the underlying similarity matrix are stored in a relational database.*

This work accesses the pair-wise comparison data. The regions of sequences are displayed that are most often found to be similar to other proteins. This is shown both in dependence of the location on the protein sequence and of Z-score.

InterPro domain information

Additional context is offered by the visualisation of matches to the InterPro *database.* InterPro *is an Integrated Documentation Resource which combines information about protein families, domains and functional sites from* PROSITE, PRINTS,Pfam,SMART,ProDom.

Visualisation of found local similarities between proteins allows users to select most interesting protein regions. Those sequence parts can be downloaded directly from relational database and used for further analysis.

Selected subsequences

Multiple alignment algorithm can be used to align selected subsequences.

CLUSTAL W multiple alignment

The output of multiple alignment may subsequently serve as a seed to generate HMMs or other description for protein domains to finally determin new protein domains in a bootstrap approach.

Hidden Markov Model

How to contact us

E-mails: moeller/zhenya @ebi.ac.uk
Telephone: + 44 (0)1223 494670/4430
Fax: + 44 (0)1223 494468

Another nice feature of this approach is to help overcome an inherent limitation of local sequence similarity determinants. These either focus on an area of maximum similarity and thereby ignore remaining similarities or are no longer specific. In multi-domain proteins as a consequence one of multiple domains may be omitted. With the assumtion that the omitted domains may also occur independly from the one shown, the respective regions will be hilited by our visualisation.

Protein clusters of completely sequenced eukaryote genomes for which ***no*** *InterPro domains were found can be accessed from the Proteome Analysis pages*

References

1. Kriventseva EV, Fleischmann W, Zdobnov E. Apweiler R. CluSTr: a database of Clusters of SWISS-PROT−TrEMBL proteins. Nucl Acids Res.2001, 29(1):33-36.

2. Apweiler R. Attwood TK, Bairoch A. Bateman A, Birney E. Biswas M, Bucher P, Cerutti L, Corpet F. Croning MDR. Durbin R. Falquet L. Fleischmann W, Gouzy J. Hermjakob H. Hulo N. Jonassen I, Kahn D, Kanapin A, Karavidopoulou Y, Lopez R. Marx B. Mulder NJ, Oinn TM. Pagni M.Servant F, Sigrist CJA, Zdobnov EM: InterPro - An integrated documentationresource for protein families, domains and functional sites.
Nucl Acids Res 2001. 29(1):37-40.

3. Apweiler R. Biswas M, Fleischmann W, Kanapin A. Karavidopoulou Y, Kersey P, Kriventseva EV, Mittard V. Mulder N. Phan I, Zdobnov E: Proteome Analysis Database: online application of InterPro and CluSTr for the functional classification of proteins in whole genomes.
Nucl Acids Res 2001, 29(1):44-48.

4. Fleischmann W. Moller S, Gateau A, Apweiler R. A novel method for automatic functional annotation of proteins. Bioinformatics. 1999 Mar;15(3):228-33.

5. S. Möller, M. D. R. Croning, R. Apweiler
"Evaluation of methods for the prediction of membrane spanning regions
in proteins." Bioinformatics, in press (2001)

EMBL Outstation - The European Bioinformatics Institute (EBI)
Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, U.K.

Figure 53: Domain finding with CluSTr

VIII.References

Alberts, B., D. Bray, et al. (1994). Molecular Biology of the Cell, Garland Publishing, Inc.

Alferes, J. J. and L. M. Pereira (1996). Reasoning with Logic Programming, Springer-Verlag.

Amandi, A., A. Zunino, et al. (1999). Multi-paradigm languages supporting multi-agent development. Multi-Agent System Engineering, 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'99. F. J. Garijo and M. Bornan, Springer-Verlag.

Apweiler, R., T. K. Attwood, et al. (2001). “The InterPro database, an integrated documentation resource for protein families, domains and functional sites.” Nucleic Acids Res. **29**(1): 37-40.

Apweiler, R., T. K. Attwood, et al. (2001). “The InterPro database, an integrated documentation resource for protein families, domains and functional sites.” Nucl. Acid Res. **29**(1): 37-40.

Apweiler, R., C. O'Donovan, et al. (1998). SWISS-PROT and its Computer-Annotated Supplement TrEMBL: How to Produce High-Quality Automatic Annotation. Systems, Cybernetics and Informatics, Orlando, Florida, IIIS.

Ashburner, M. (2001). Gene Ontology.

Ashburner, M., C. A. Ball, et al. (2000). “Gene ontology: tool for the unification of biology.” Nat. Genet. **25** (1): 25-29.

Attwood, T. K., M. D. Croning, et al. (2000). “PRINTS-S: the database formerly known as PRINTS.” Nucl. Acid Res. **28**(1): 225-227.

Babur, Ö., S. Möller, et al. (2001). “TransMotif.” submitted.

Bairoch, A. (1996). “The ENZYME data bank in 1995.” Nucl. Acids Res. **24** (1): 221-222.

Bairoch, A. (2000). “Serendipity in bioinformatics, the tribulations of a Swiss bioinformatician through exciting times!” Bioinformatics **16**(1): 48-64.

Bairoch, A. and R. Apweiler (1999). The SWISS-PROT Protein Sequence Database User Manual.

Bairoch, A. and R. Apweiler (2000). “The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.” Nucleic Acids Res. **28**: 45-48.

Bairoch, A. and R. Apweiler (2000). “The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.” Nucl. Acid. Res. **28**(1): 45-48.

Baldi, P. and S. Brunak (1998). Bioinformatics, The MIT Press.

Baldi, P., S. Brunak, et al. (2000). “Assessing the accuracy of prediction algorithms for classification: an overview.” Bioinformatics **16**(5): 412-424.

Bass, B. L. (2001). RNA Editing, Oxford University Press.

Bateman, A., E. Birney, et al. (2000). “The Pfam protein families database.” Nucl. Acid Res. **28**(1): 263-266.

Bentley, D. R. (2000). “Decoding the human genome sequence.” Hum. Mol. Genet. **9**(16): 2353-2358.

Birney, E. (2001). Ensembl. Nucl. Acid Res. **2001**.

Blobel, G. (1980). “Intracellular protein topogenesis.” Proc. Natl. Acad. Sci. **77**: 1496-1500.

Bock, A., K. Forchhammer, et al. (1991). “Selenoprotein synthesis: an expansion of the genetic code.” Trends Biochem Sci. **16**(12): 463-467.

Bockaert, J. and J. P. Pin (1999). “Molecular tinkering of G protein-coupled receptors: an evolutionary success.” EMBO J. **18**(7): 1723-1729.

Bourne, H. R. (1997). “How receptors talk to trimeric G proteins.” Curr. Opin. Cell Biol. **9**(2): 134-142.

Bratko, I. (2000). PROLOG Programming for Artificial Intelligence, Addison-Wesley.

Brazma, A., I. Jonassen, et al. (1998). “Approaches to Automatic Discovery of Patterns in Biosequences.” Journal of Computational Biology **5**(2): 277-304.

Bretscher, M. S. (1971). “Major human erythrocyte glycoprotein spans the cell membrane.” Nature New Biol. **231**: 229-232.

Bryson, K., M. Joy, et al. (1999). Using Software Agents to Investigate Genomes, BBSRC CCP11 Newsletter.

Bryson, K., M. Luck, et al. (2000). Applying Agents to Bioinformatics in GeneWeaver. International Workshop on Collaborative Information Agents.

Butler, D. (2000). "Ensembl gets a Wellcome boost." Nature **406**(6794): 333.

Casari, G., M. A. Andrade, et al. (1995). "Challenging times for bioinformatics." Nature **376**(6542): 647-648.

Casari, G., A. D. Daruvar, et al. (1996). "Bioinformatics and the discovery of gene function." Trends Genet. **12**(7): 244-245.

Casari, G., C. Ouzounis, et al. (1996). GeneQuiz II: automatic function assignment for genome sequence analysis. Pacific Symposium on Biocomputing, World Scientific.

Casati, F., P. Grefen, et al. (1996). WIDE Workflow model and architecture, University of Twente.

Claros, M. G. and G. v. Heijne (1994). "TopPred II: an improved software for membrane protein structure predictions." Comput Appl Biosci. **10** (6): 685-686.

Cogan, P., J. Gomoluch, et al. (2001). "A Quantitative and Qualitative Comparison Of Distributed Information Processing Using Mobile Agents Realised in RMI and Voyager." submitted.

Comet, J. P., J. C. Aude, et al. (1999). "Significance of Z-value statistics of Smith-Waterman

scores for protein alignments." Comput Chem. **23**(3-4): 317-331.

Consortium, T. G. I. S. (2001). "Initial sequencing and analysis of the human genome." Nature **409**: 860-921.

Corpet, F., F. Servant, et al. (2000). "ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons." Nucl. Acid Res. **28**(1): 267-269.

Cserzo, M., E. Wallin, et al. (1997). "Prediction of transmembrane alpha-helices in prokaryotic membrane proteins: the dense alignment surface method." Protein Eng. **10**(6): 673-676.

Cuff, J. A., M. E. Clamp, et al. (1998). "JPred: a consensus secondary structure prediction server." Bioinformatics **14**(1o): 892-893.

Dahm, R. (2001). Fish facts, MPI für Entwicklungsbiologie, http://www.eb.tuebingen.mpg.de/home/news/fish_facts.html.

Damasio, C. V., L. M. Pereira, et al. (1997). REVISE: Logic Programming and Diagnosis. Logic Programming and Non-monotonic Reasoning, Springer--Verlag.

Decker, K., S. Khan, et al. (2001). "Extending a Multi-agent System for Genomic Annotation." Lecture Notes in Computer Science **2182**: 106.

Decker, K., X. Zheng, et al. (2001). A multi-agent system for automated genomic annotation. Fifth International Conference on Autonomous Agents, Montreal, Canada, ACM Press.

Diederichs, K., J. Freigang, et al. (1998). "Prediction by a neural network of outer membrane beta-strand protein topology." Protein Sci. **7**(11): 2413-2420.

Dowell, D. R., R. M. Jokerst, et al. (2001). "The Distributed Annotation System." BMC Bioinformatics **2**(1): 7.

Durbin, R., S. R. Eddy, et al. (1998). Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge, UK, Cambridge University Press.

Eisenberg, D., R. M. Weiss, et al. (1982). "The helical hydrophobic moment: a measure of the amphiphilicity of a helix." Nature **299**(5881): 371-374.

Emmanuelsson, O., H. Nielsen, et al. (2000). "Predicting subcellular localization of proteins based on their N-terminal amino acid sequence." J. Mol. Biol. **300**: 1005-1016.

Ensembl (2001). Ensembl, http://www.ensembl.org.

Etzold, T., A. Ulyanov, et al. (1996). "SRS: information retrieval system for molecular biology data banks." Methods Enzymol. **266**: 114-128.

Fleischmann, W., S. Möller, et al. (1999). "A novel method for automatic functional annotation of proteins." Bioinformatics **15**(3): 228-233.

Flores-Mendez, R. A. (1999). "Towards the Standardization of Multi-Agent System Architectures: An Overview."

ACM Crossroads Special Issue on Intelligent Agents(5.4): 18-24.

Frishman, D. and H.-W. Mewes (1997). “PEDANTic genome analysis.” Trends in Genetics **13**: 415-416.

Frishman, D. and H.-W. Mewes (1997). “Protein structural classes in five complete genomes.” Nature Structural Biology **4**(8): 626-628.

Fu, D., A. Libson, et al. (2000). “Structure of a Glycerol-Conducting Channel and the Basis fo Its Selectivity.” Science **290**: 481-486.

Fyfe, P. K., K. E. McAuley, et al. (2001). “Probing the interface between membrane proteins and membrane lipids by X-ray crystallography.” Trends Biochem Sci. **26**(2): 106-111.

Gaasterland, T. and J. Lobo (1997). “Qualifying answers according to user needs and preferences.” Fundamenta informatica **32**: 121-137.

Gaasterland, T., N. Maltsev, et al. (1994). Assigning function to CDS through qualified query answering: beyond alignment and motifs. Int. Conf. Intell. Syst. Mol. Biol., AAAI Press.

Gaasterland, T., A. Sczyrba, et al. (2000). “MAGPIE/EGRET Annotation of the 2.9-Mb Drosophila melanogaster Adh Region.” Genome Res. **10**(4).

Gaasterland, T. and C. Sensen (1996). “MAGPIE - Automated Genome Interpretation.” Trends in Genetics **12** (2): 76-78.

Gaasterland, T. and C. W. Sensen (1996). “Fully automated genome analysis that reflects user needs and preferences. A detailed introduction to the MAGPIE system architecture.” Biochimie **78**(5): 302-310.

Gafvelin, G. and G. v. Heijne (1994). “Topological 'Frustration' in Multispanning E. coli Inner Membrane Proteins.” Cell **77**: 401-412.

Gasteiger, E. (2001). NiceProt, http://www.expasy.org/sprot/.

Geest, M. v. and J. S. Lolkema (2000). “Membrane topology and insertion of membrane proteins: search for topogenic signals.” Microbiol Mol Biol Rev. **64**(1): 13-33.

Gennis, R. B. (1989). Biomembranes : Molecular Structure and Function, Springer Verlag.

Georgskopoulos, D., M. Hornick, et al. (1995). “An overview of workflow management: From process modeling to workflow automation infrastructure.” Journal of Distributed and Parallel Databases **3**(22): 119-154.

Gether, U. (2000). “Uncovering molecular mechanisms involved in activation of G protein-coupled receptors.” Endocrine Reviews **21**(1): 90-113.

Gilbert, D. R., M. Schroeder, et al. (2000). “Interactive visualization and exploration of relationships between biological objects.” Trends Biotechnol. **18**(12): 487-494.

Graham, J. and K. Decker (2000). Towards Distributed, Environment Centered Agent Framework. Intelligent Agents IV, Agent Theories, Architectures, and Languages. Y. L. Nicholas Jennings, Springer-Verlag. **4**.

Greiner, R., B. A. Smith, et al. (1989). “A correction of the Algorithm in Reiter's Theory of Diagnosis.” Artificial Intelligence **41**(1): 79-88.

Grey, A. D. N. J. d. (2000). “Mitochondrial gene therapy: an arena for the biomedical use of inteins.” Trends in Biotechnology **18**(9): 394-399.

Hao, B., W. Gong, et al. (2002). “A new UAG-encoded residue in the structure of a methanogen methyltransferase.” Science **296**(5572): 1462-1466.

Heijne, G. v. (1986). “The distribution of positively charged residues in bacterial inner membrane proteins correlates with the trans-membrane topology.” EMBO J. **5**(11): 3021-3027.

Henikoff, J., S. Henikoff, et al. (1999). “New features of the blocks database servers.” Nucleic Acids Res. **27**(226-228).

Hermjakob, H., W. Fleischmann, et al. (1999). “Swissknife - 'lazy parsing' of SWISS-PROT entries.” Bioinformatics **15**(9): 771-772.

Hirokawa, T., S. Boon-Chieng, et al. (1998). “SOSUI: classification and secondary structure prediction system for membrane proteins.” Bioinformatics **14**(4): 378-379.

Ho, M. K. and Y. H. Wong (2001). “G(z) signaling: emerging divergence from G(i) signaling.” Oncogene **20**(13): 1615-1625.

Hofmann, K., P. Bucher, et al. (1999). “The PROSITE database, its status in

1999." Nucl. Acid Res. **21**(1): 215-219.

Hofmann, K. and W. Stoffel (1993). "TMBASE - A database of membrane spanning protein segments." Biol. Chem. Hoppe-Seyler **374**: 166.

Horn, F., J. Weare, et al. (1998). "GPCRDB: an information system for G protein-coupled receptors." Nucleic Acids Res. **26**(1): 275-279.

Hubbard, T., D. Barker, et al. (2002). "The Ensembl genome database project." Nucleic Acids Res. **30**(1): 38-41.

III, H. L. (1999). "Structural features of heterotrimeric G-protein-coupled receptors and their modulatory proteins." Mol. Neurobiol. **19**(2): 111-149.

Jeanmougin, F., J. D. Thompson, et al. (1998). "Multiple sequence alignment with Clustal X." Trends Biochem Sci. **23**(10): 403-405.

Jennings, M. L. (1989). Topography of membrane proteins. Annu. Rev. Biochem., Annual Reviews Inc. **58:** 999-1027.

Ji, T. H., M. Grossmann, et al. (1998). "G protein-coupled receptors. I. Diversity of receptor-ligand interactions." J. Biol. Chem. **273**(28): 17299-17302.

Jones, D. T., W. R. Taylor, et al. (1994). "A model recognition approach to the prediction of all-helical membrane protein structure and topology." Biochemistry **33**(10): 3038-3049.

Kanapin, A., R. Apweiler, et al. (2002). "Interactive InterPro-based comparisons of proteins in whole genomes." Bioinformatics **18**: 374-375.

Klein, P., M. Kanehisa, et al. (1985). "The detection and classification of membrane-spanning proteins." Biochim. Biophys. Acta **815**: 468-476.

Klein, P., M. Kanehisa, et al. (1985). "The detection and classification of membrane-spanning proteins." Biochim. Biophys. Acta **815**(3): 468-476.

Klusch, M. (1998). Kooperative Informationsagenten im Internet. Computer Science, University of Kiel.

Kreil, D. P. (2001). From General Scientific Workflows to Specific Sequence Analysis Applications:The study of compositionally biased proteins. Biology. Cambridge, University of Cambridge.

Kreil, D. P. and T. Etzold (1999). "DATABANKS - a catalogue database of molecular biology databases." Trends Biochem Sci. **24** (4): 155-157.

Kretschmann, E., W. Fleischmann, et al. (2001). "Automatic rule generation for protein annotation with the C4.5 data mining algorith applied on SWISS-PROT." Bioinformatics **17**: 920-926.

Kriventseva, E., W. Fleischmann, et al. (2001). "CluSTr: a database of clusters of SWISS-PROT+TrEMBL proteins." Nucleic Acids Res. **29**(1): 33-36.

Krogh, A., B. Larsson, et al. (2001). "Predicting transmembrane protein topology with a hidden Markov model: Application to compete genomes." J. Mol. Biol. **305**(3): 567-580.

Kyte, J. and R. F. Doolittle (1982). "A simple method for displaying the hydropathic character of a protein." J. Mol. Biol. **157**(1): 105/132.

L. Charles Bailey, J., S. Fisher, et al. (1998). "GAIA: Framework Annotation of Genomic Sequence." Genome Research **8**(3): 234-250.

Lappe, M., J. Park, et al. (2001). Generating Protein Interaction Maps from Incomplete Data: Application to Fold Assignment. Int. Sys. Mol. Biol., Copenhagen, DK.

Lemmon, M. A., K. R. MacKenzie, et al. (1997). 1997. Membrane Protein Assembly. G. v. Heijne, R. G. Landes Company.

Liu, J. and J. Wess (1996). "Different single receptor domains determine the distinct G protein coupling profiles of members of the vasopression receptor family." J. Biol. Chem. **271**: 8772-8778.

Livingstone, C. D. and G. J. Barton (1993). "Protein sequence allignments: a strategy for the hierachical analysis of residue conservation." Comp. Appl. Biosci. **9** (6): 745-756.

Lynch, K. R. (1998). G Protein-Coupled Receptor informatics and the orphan problem. Identification and Expression of G Protein-Coupled Receptors. K. R. Lynch, John Wiley and Sons**:** 54-72.

Milligan, G. and M. Wakelam (1992). G Proteins Signal Transduction & Disease. London, Academic Press Ltd.

Möller, S. (2000). Files describing well-analysed transmembrane proteins, ftp://ftp.ebi.ac.uk/pub/databases/testsets/transmembrane.

Möller, S., M. D. R. Croning, et al. (2001). "Evaluation of Methods for the prediction of membrane spanning regions in transmembrane proteins." in press.

Möller, S., E. Kriventseva, et al. (2000). "A collection of well characterised integral membrane proteins." Bioinformatics **16**(12): 1159-1160.

Möller, S., U. Leser, et al. (1999). "EDITtoTrEMBL: a distributed approach to high-quality automated protein sequence annotation." Bioinformatics **15**(3): 219-227.

Möller, S., J. Vilo, et al. (2001). Prediction of coupling of GPCRs to G -proteins. Conf. Int. Sys. Mol. Biol., Copenhagen, AAAI Press.

Mulder, N. J. and R. Apweiler (2001). "Tools and resources for identifying protein families, domains and motifs." Genome Biology **3**(1): 8.

Munro, S. (1995). "An investigation of the role of transmembrane domains in Golgi protein retention." EMBO J. **14** (19): 4695-4704.

Nakai, K. (2000). Protein sorting signals and prediction of subcellular localization. Advances in Protein Chemistry, Academic Press. **54:** 277-344.

Nakai, K. and P. Horton (1999). "PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization." Trends Biochem Sci. **24**(1): 34-36.

Nakai, K. and M. Kanehisa (1992). "A knowledge base for predicting protein localization sites in eukaryotic cells." Genomics **14**(4): 897-911.

NCBI (2001). Genome resources - All Organisms, http://www.ncbi.nlm.nih.gov/PMGifs/Genomes/allorg.html.

NHGRI (2001). Positional Cloning Fact Sheet, http://www.nhgri.nih.gov/Policy_and_public_affairs/Communications/Fact_sheets/positional_cloning.html.

Nielsen, H. (1999). From sequence to sorting. Prediction of signal peptides. Center for Biological Sequence Analysis. Lyngby, Danish Technical University.

Nielsen, H., S. Brunak, et al. (1999). "Machine learning approaches to the prediction of signal peptides and other protein sorting signal." Protein Eng. **12**(1): 3-9.

Nielsen, H. and A. Krogh (1998). Prediction of signal peptides and signal anchors by a hidden Markov model. Int. Conf. Intell. Syst. Mol. Biol., Montreal, CA, AAAI Press.

Nilsson, J., B. Persson, et al. (2000). "Consensus prediction of membrane protein topology." FEBS Lett. **486**: 267-269.

ObjectSpace (2001). Voyager, http://www.objectspace.com/products/voyager/.

O'Donovan, C., M. J. Martin, et al. (1999). "Removing redundancy in SWISS-PROT and TrEMBL." Bioinformatics **Bioinfomatics**(15): 258-259.

Ohlsson, R., B. Tycko, et al. (1998). "Monoallelic expression: 'there can be only one'." Trends Genet. **14**(11): 435-438.

Ouzounis, C., P. Bork, et al. (1995). "New protein functions in yeast chromosome VIII." Protein Sci. **4**(11): 24242428.

Ouzounis, C., G. Casari, et al. (1996). "Computational comparisons of model genomes." Trends Biotechnol. **14**(8): 280-285.

Palczewski, K., T. Kumasaka, et al. (2000). "Crystal structure of rhodopsin: A G protein-coupled receptor." Science **289**(5480): 739-745.

Payne, M. A. (2001). Biochemistry 491 Lecture Notes, http://www.lasierra.edu/~mpayne/.

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

Persson, B. and P. Argos (1997). "Prediction of membrane protein topology utilizing multiple sequence alignments." J. Protein Chem. **16**(5): 453-457.

Prinz, W. A. and J. Beckwith (1994). "Gene fusion analysis of membrane protein topology: a direct comparison of alkaline phosphatase and beta-lactamase fusions." J Bacteriol **176** (20): 6410-6413.

Promponas, V. J., G. A. Palaios, et al. (1998). "CoPreTHi: A Web tool which combines transmembrane protein segment prediction methods." In Silicio Biology **1**.

Raymond Reiter (1987). "A Theory of Diagnosis from First Principles." Artificial Intelligence **32**(1): 57-96.

Reinhardt, A. and T. Hubbard (1998). "Using neural networks for prediction of the subcellular location of proteins." Nucl. Acids Research **26** (9).

Ridley, M. (1999). Genome: the autobiography of a species in 23 chapters. New York, HarperCollins.

Roche, P. C. (1996). G Proteins. San Diego, CA, Academic Press, Inc.

Rost, B., R. Casadio, et al. (1996). Refining neural network predictions for helical transmembrane proteins by dynamic programming. Int. Conf. Intell. Syst. Mol. Biol., St. Louis, U.S.A., AAAI Press.

Roy, P. V. and S. Haridi (1999). Mozart: A Programming System for Agent Applications. International Conference on Logic Programming, Workshop on Distributed and Internet Programming with Logic and Constraint Languages, http://www.mozart-oz.org.

Sakaguchi, M. (1997). Mutational nalysis of Signal-Anchor and Stop-Transfer Sequences in Membrane Proteins. Membrane Protein Assembly. G. v. Heijne, Springer-Verlag**:** 135-150.

Sanger, F., S. Nicklen, et al. (1977). DNA sequencing with chain-terminating inhibitors. Proc. Natl. Acad. Sci. USA. **74:** 5463-5467.

Sanger, F., S. Nicklen, et al. (1992). "DNA sequencing with chain-terminating inhibitors. 1977." Biotechnology **24**: 104-108.

Sautel, M. and G. Milligan (2000). "Molecular manipulation of G-protein-coupled receptors: a new avenue into drug discovery." Current Medicinal Chemistry **7**(9): 889-896.

Scharf, M., R. Schneider, et al. (1994). GeneQuiz: a workbench for sequence analysis. Proc. Int. Conf. Intell. Syst. Mol. Biol.

Scherf, M., A. Klingenhoff, et al. (2000). "Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach." J Mol. Biol. **297**(3): 599-606.

Schoneberg, T., G. Schultz, et al. (1999). "Structural basis of G protein-coupled receptor function." Mol. Cell. Endocrinol. **151**(1-2): 181-193.

Shapiro, J. A. and M. Dworkin (1997). Bacteria As Multicellular Organisms, Oxford University Press.

Shimizu, T. and K. Nakai (1994). Construction of a Membrane Protein Database and an Evaluation of Several Prediction Methods of Transmembrane Segments. Genome Informatics Workshop, Universal Academic Press**:** 148-149.

Shriver, C. R., A. L. Beaudet, et al. (1995). The metabolic and molecular bases of inherited disease., McGraw-Hill.

Simon, M. I., M. P. Strathmann, et al. (1991). "Diversity of G proteins in signal transduction." Science **252** (5007): 802-808.

Singer, S. J. (1990). "The Structure and Insertion of Integral Proteins in Membranes." Annu. Rev. Cell Biol. **6**: 247-296.

Smith, T. F. and M. S. Waterman (1981). "Identification of common molecular subsequences." J Mol Biol. **147**(1): 195-197.

Sonnhammer, E. L. L., G. v. Heijne, et al. (1998). A hidden Markov model for predicting transmembrane helices in protein sequences. Int. Conf. Intell. Syst. Mol. Biol., Montreal, CA, AAAI Press.

Sterk, P. (2001). EBI Genome MOT, http://www.ebi.ac.uk/genomes/mot/index.html.

Sterling, L. (1994). The Art of Prolog: Advanced Programming Techniques, MIT Press.

Stevens, T. J. and I. T. Arkin (2000). "Do more complex organisms have a greater proportion of membrane proteins in their genomes?" Proteins **39**(4): 417-420.

Stoesser, G., W. Baker, et al. (2001). "The EMBL Nucleotide Sequence Database." Nucleic Acids Res. **29**(1): 17-21.

TiPS (2000). Receptor & ion channel nomenclature supplement.

Togethersoft (2001). TogetherJ, http://www.togetherJ.com.

Traxler, B., D. Boyd, et al. (1993). “The topological analysis of integral cytoplasmic membrane proteins.” J Membr Biol **132**(1): 1-11.

Tusnády, G. E. and I. Simon (1998). “Principles Governing Amino Acid Composition of Integral Membrane Proteins: Application to Topology Prediction.” Journal of Molecular Biology **283**: 489-506.

Tusnády, G. E. and I. Simon (2001). “The HMMTOP transmembrane topology prediction server.” Bioinformatics **17** (9): 849-850.

Tusnády, G. E. and I. Simon (2001). “Topology of Membrane Proteins.” J. Chem. Info. Comput. Sci. **41**: 364-368.

Ullman, J. D. (1988). Priciples of database and knowledgebase systems. Rockville, Maryland, Computer Science Press.

Velds, A. (1999). Feature Propagation. German Conference in Bioinformatics, Hannover, Germany.

Venter, J. C. (2001). “The Sequence of the Human Genome.” Science **291**(5507): 1304-1351.

Vilo, J. (1998). Discovering Frequent Patterns from Strings. Helsinki, Department of Computer Science, University of Helsinki.

Vilo, J., A. Brazma, et al. (2000). Mining for putative regulatory elements in the yeast genome using gene expression data. Int. Conf. Intell. Syst. Mol. Biol., San Diego,CA.

W3C (1997). Extensible Markup Language (XML), http://www.w3.org/XML.

Wallin, E. and G. v. Heijne (1998). “Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms.” Protein Sci. **7**(4): 1029-1038.

Wess, J. (1998). “Molecular basis of receptor/G-protein-coupling selectivity.” Pharmacol. Ther. **80**(3): 231-264.

Wiederhold, G. and M. Genesereth (1996). The Basis for Mediation, Standford University.

Wiederhold, G., M. R. Genesereth, et al. (1997). “The Conceptual Basis for Mediation Services.” IEEE Expert **12** (5).

Wielemaker, J. (2001). SWI Prolog. Amsterdam, http://www.swi.psy.uva.nl/projects/SWI-Prolog.

Wilson, S., D. J. Bergsma, et al. (1998). “Orphan G-protein-coupled receptors: the next generation of drug targets?” Br. J. Pharmacol. **125**(7): 1387-1393.

Zdobnov, E., R. Lopez, et al. (2000). The EBI SRS server- recent developments. German Conference on Bioinformatics, Hannover, Logos Verlag, Berlin, Germany.

Zdobnov, E. M., R. Lopez, et al. (2002). “The EBI SRS server—recent developments.” Bioinformatics **18**: 368-373.