UNIVERSITY OF NOVA GORICA
GRADUATE SCHOOL

INTEGRATION OF HETEROGENEOUS DATA SOURCES FOR PROTEIN CLASSIFICATION

DISSERTATION

Somdutta Dhir

Mentor: Dr. Sándor Pongor, Ph.D., D.Sc.

Nova Gorica, 2010

Dedicated to my family for their never-ending love and support

ABSTRACT

Bioinformatics is not only about processing more and more biological data but also about interpreting and classifying increasingly complex and heterogeneous data with computational tools. Most of current data interpretation ("data annotation") tasks are carried out by automated classifier algorithms, yet there are very few methods that enable one to compare the efficiency of classification algorithms in bioinformatics tasks. With this introduction in mind I have chosen the following areas:

i) How can we benchmark a data-interpretation method? I have approached this subject via the analysis of the protein classification problem and the development of a benchmark database of 6405 classification tasks, applicable to test structural and functional annotation of proteins. I illustrate the use of this collection by developing an algorithm based on a Committee of Classifiers.

ii) How can we integrate similarity data obtained from various data-sources? One of the most general schemes to define data-similarities is called a similarity space that can be represented as a network of similarities. I have developed Multi-Netclust, a straightforward algorithm that can combine similarity data from different sources and have showed that this approach can lead to better recognition as well as substantial data compression.

iii) How can we compare complete annotations, such as domain architectures predicted by various domain prediction algorithms? I have approached this problem by developing a general framework of comparison principles and numerical indices of similarity by which I could compare various protein domain annotation schemes. I show that similarity-based domain prediction performs as well, sometimes even better than generative models based on learning algorithms.

iv) Finally, how do we apply these principles to practical problems? I carried out the structural analysis and classification of the newly determined ^{1}H-NMR solution structure of an epidermal growth factor (EGF) domain encoded by exon 6 of the JAG1 protein. I have found that this domain has an atypical structure and is encoded by an atypical exon/intron arrangment which is conserved throughout evolution. I also carried out a systematic and

comprehensive analysis of mutations found in EGF domains and showed that specific residue requirements for folding, structural integrity and correct post-translational processing may provide a rationale for most of the disease-associated mutations.

Integracija heterogenih podatkovnih virov pri klasifikaciji proteinov

POVZETEK

Bioinformatika se ne ukvarja le s procesiranjem vedno številčnejših bioloških podatkov, pač pa vključuje tudi interpretacijo in klasificiranje vedno bolj kompleksnih in heterogenih podatkov z računskimi orodji. V zadnjem času večina interpretacije podatkov ("data annotation") poteka z avtomatskimi klasifikatorskimi algoritmi, obstaja pa tudi nekaj metod, s katerimi lahko primerjamo učinkovitost teh algoritmov v bioinformatiki. Na osnovi teh dejstev je bilo moje raziskovalno delo razdeljeno na naslednje sklope:

i) Kako oceniti uspešnost določene metode za interpretacijo podatkov? Te teme sem se lotila z analizo problema klasifikacije proteinov, pri čemer sem razvila ocenjevalno databazo 6405 klasifikatorskih opravil, s katerimi lahko testiramo strukturno in funkcijsko anotacijo proteinov. To zbirko sem uporabila za razvoj algoritma, ki temelji na "komisiji klasifikatorjev" (Committee of Classifiers).

ii) Kako lahko integriramo podatke o podobnostih, ki so bili pridobljeni iz različnih podatkovnih virov. Ena od splošnih shem za predstavljanje podatkovnih podobnosti se imenuje vesolje podobnosti , kar si lahko predstavljamo kot mrežo podobnih podatkov. Razvila sem enostaven algoritem z imenom Multi-Netclust, ki lahko primerja podobne podatke iz različnih virov. Poleg tega sem pokazala, da ta algoritem omogoča boljše prepoznavanje in tudi občutno kompresijo podatkov.

iii) Kako lahko primerjamo kompleksnejše anotacije, kot je na primer zgradba proteinskih domen, ki je predpostavljena na osnovi različnih napovednih algoritmov. Za rešitev tega problema sem razvila osnovni okvir primerjalnih principov in številčnih ocen podobnosti, s katerimi je mogoče primerjati različne sheme za anotacijo proteinov. Dokazala sem, da je strukturo proteinskih domen mogoče učinkovito napovedati tudi na osnovi podobnosti, kar je včasih celo bolj uspešno kot z uporabo generativnih modelov, ki temeljijo na logaritmih učenja.

iv) Kako lahko zgoraj navedene principe apliciramo na reševanje praktičnih problemov? Naredila sem strukturno analizo domene epidermalnega rastnega faktorja (epidermal growth factor; EGF), ki jo kodira exon 6 proteina JAG_1 in ki je bila pred kratkim določena v raztopini z metodo ^{1}H-NMR. Ugotovila sem, da ima ta domena neobičajno strukturo ter da jo kodira neobičajno prerazporejanje intronov in eksonov, ki je ohranjeno skozi evolucijo. Opravila sem tudi sistematično in poglobljeno analizo mutacij, ki jih najdemo v domenah EGF. Ugotovila sem, da so specifične aminiokisline zahteve za pravilno gubanje, strukturno integriteto in post-translacijske modifikacije verjetna podlaga za večino mutacij, ki so povezane z boleznimi.

Acknowledgments

Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.
Sir Winston Churchill

I would like to take this opportunity to thank all those people without whom I would not have successfully journeyed through this beginning.

First and foremost, I would like to express my sincere thanks my thesis supervisor, Dr. Sándor Pongor, for his inspiring and knowledgeable guidance throughout my research and for patiently guiding me during the writing of this thesis.Without his encouragement and magnanimous support, this thesis would never have been completed in its present form and I gratefully acknowledge his support.

I'd also like to sincerely thank Dr. Alessandro Pintar for the invaluable advice and guidance that he had extended throughout the course of my research especially for the JAG project.

During this work I have collaborated with many colleagues and I wish to extend my warmest thanks to all those who have helped me with my work. I have been fortunate enough to have wonderful lab mates and colleagues and I would like to thank all them for the excellent atmosphere, for the good company and interesting discussions over the years. I would like to express my heartfelt gratitude to Paolo, Mircea and Dino and all the members of the Protein Structure and Bioinformatics group. Special thanks to Attila, for the critical reading and commenting in detail on the thesis draft that prompted me to improve a lot. I owe my thanks to all my friends whom I have met in this my home far away from home. Their company is a constant reminder that there is much more to life than a thesis.

To conclude the acknowledgement, I would like to express my gratitude my parents. No words can ever describe the efforts they have put into giving me the life I have now. I would like to thank my brother for always being there for me and for believing in me. A special thanks to my in-laws for always having their faith in me. Thank you all for bringing me so much happiness over the years.

Finally, I wish to express special thanks and to dedicate this work to my husband, **Ashish**, for his endless care and relentless encouragement. I simply could not have taken the initial step of my study nor have gone this far without him.Without him by my side through the most difficult times of this journey, any accomplishment wouldn't have been half as sweet. I would like to share my happiness with him. You are the reason I did this; you are the reason I thrive to be better.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Foreword

Bioinformatics is a broad field of research that escapes easy definition. Various definitions agree that bioinformatics uses mathematical tools and computer technology in order to derive knowledge from biological data. However this broad definition tells experimental biologists little about how this goal can be achieved.

The starting point of my research at the Protein Structure and Bioinformatics Group of International Centre for Genetic Engineering and Biotechnology (ICGEB), Trieste, was the need of a unifying framework that would help biologists to understand how "biological knowledge" is formalized in bioinformatics and what the fundamental techniques are. The approach we adopted was to start from the simple facts of bioinformatics, namely from a broad overview of sequence and structural databases which are perhaps the most visible items that have been produced by bioinformatics. In the beginning, sequence databases contained only sequences provided by names and a literature reference (Doolittle, 1995). Gradually, the scope of added information started to expand and its form became more and more organized. For instance, earlier versions of the Swiss-Prot database (Boeckmann et al., 2003) included specific sections of protein function, cross references to genetic diseases as well as a feature table, containing local descriptors of the sequence in terms of structural and functional domains. These added sections of a protein sequence record was termed the annotation part, and producing the annotation part became a field of its own, highly used but rarely appreciated by end-users. As a subfield branching out of database annotation, secondary databases started to appear already in the early 90's, such as the Prosite (Bairoch, 1991), SBASE (Pongor et al., 1993) and Prodom (Corpet et al., 1998; Sonnhammer and Kahn, 1994) that concentrated on protein domain sequences, or SCOP (Murzin et al., 1995) and CATH (Orengo et al., 1997) on domain 3D structure. The common philosophy of the derived databases was to cluster the (sub-structural) data into similarity-based subgroups. This new, curated information was however soon mapped back to the original sequence collections, and databases such as UNIPROT (Apweiler et al., 2004) now also contains data on drug actions, genetic diseases etc. while others concentrate on protein architectures, exon and intron information(Schultz et al., 1998). These data are now integrated at higher levels, into genome-wide databases(Hubbard et al., 2002).This brief overview tells us that

the history of bioinformatics is not only about more and more data, but also about deeper and more integrated knowledge, which, in our view, is perhaps a bigger challenge. Simply put, the development of computer technology appears to cope with the data storage problems of the life sciences, and also, the primary data processing needs (sequence assembly, gene identification, and simple proteome annotations) are well covered by the emerging multiprocessor technologies. But the in-depth interpretation data has a high – and largely unexplored – complexity. The current status of bioinformatics is characterized by a very strong tendency of data-integration. Formats are unified, vocabularies are formalized into ontologies, etc. As more and more new data-collection technologies appear, there is a very strong need for automated computational methods that minimize human intervention.

Taken together, bioinformaticians have to be prepared to applying in-depth data interpretation methodologies to newer and newer kinds of data. Machine learning methods are becoming a standard in many areas, however they are like black boxes to practicing biologists, and even bioinformaticians not working in the very same field are often clueless when having to choose between alternative methodologies.

The subject of my thesis was established with this introduction in mind. We have chosen four areas:

i) How can we benchmark a data-interpretation method? I approach this subject via the analysis of the protein classification problem and the development of a benchmark database comprising of 6405 classification tasks, applicable to test structural and functional annotation of proteins. I illustrate the use of this collection by developing an algorithm based on a Committee of Classifiers.

ii) How can we integrate similarity data obtained from various data-sources? One of the most general schemes to represent data-similarities is called a similarity space that can be represented as a network of similarities. I developed Multi-Netclust, a straightforward algorithm that can combine similarity data from different sources and showed that this approach can lead to better recognition as well as substantial data compression.

ii) How can we compare complete annotations, such as domain architectures predicted by various prediction algorithms? I approached this problem by developing a general framework of comparison principles and numerical indices of similarity by which I could compare various protein domain annotation schemes. I show that similarity-based domain

prediction performs as well, sometimes even better than generative models based on learning algorithms.

iv) Finally, how do we apply these principles to practical problems? I carried out the structural analysis and classification of the newly determined ^{1}H-NMR solution structure of an epidermal growth factor (EGF) domain encoded by exon 6 of the JAG1 protein. I found that this domain has an atypical structure and is encoded by an atypical exon/intron arrangment which is conserved throughout evolution. I also carried out a systematic and comprehensive analysis of mutations found in EGF domains and showed that specific residue requirements for folding, structural integrity and correct post-translational processing may provide a rationale for most of the disease-associated mutations.

This thesis is structured as follows: There is a general introduction that describes the theoretical principles underlying the projects. This is followed by four sections, each of them provided by a brief introduction and a results and a summary section. The thesis finishes with a brief section of conclusions.

My thesis research is largely based on following publications:

Sonego, P, Pacurar, M., **Dhir, S.**, Kertész-Farkas, A, Kocsor, A., Gáspári, Z., Leunissen, J.A.M., and Pongor, S. (2007) A Protein Classification Benchmark collection for machine learning, Nucl. Acids. Res., 35, D232-236.

Kertész-Farkas, A., **Dhir, S.**, Sonego, P., Pacurar, M., Netoteia, S., Nijveen, H., Leunissen, J. A. M., Kocsor, A., Pongor S. (2008): A comparison of random and supervised cross-validation strategies and benchmark datasets for protein classification, Journal of Biochemical and Biophysical Methods, in, 35, 1215-1223 .

Pintar, A., Guarnaccia C., **Dhir S.**, Pongor S. (2009) Exon 6 of human JAG1 encodes a conserved structural unit, BMC Structural Biology, 9, 43.

Guarnaccia C., **Dhir S.**, Pintar A., Pongor S. (2009) The Tetralogy of Fallot-associated G274D mutation impairs folding of the second epidermal growth factor repeat in Jagged-1, FEBS Journal, 276, 6247-6257

Franklin D., **Dhir S.**, Pongor S. (2009) Analysis of Kernel Based Protein Classification Strategies Using Pairwise Sequence Alignment Measures, Computational Intelligence Methods for Bioinformatics and Biostatistics, Lecture Notes in Computer Science, 5488, 222-231.

Dhir S., Pacurar M., Franklin D., Gáspári Z., Kertész-Farkas A., Kocsor, A., Eisenhaber F., Pongor S.(2010) Detecting atypical examples of known domain types by sequence similarity searching: The SBASE domain library approach. *(in press)*

Gáspári Z., Ángyán A, **Dhir S.**, Franklin D., Perczel A., Pintar, A., Pongor S.(2010) Probing dynamic protein ensembles with atomic proximity measures. *(In press)*

Kuzniar A., **Dhir S.** et al, (2010) Multi-Netclust: A tool for finding connected clusters in multi-parametric data networks, Bioinformatics *(in press)*

GENERAL INTRODUCTION

1. Introduction

This thesis focuses on some of the basic problems pertinent to how heterogeneous data sources are used to create new knowledge in bioinformatics. This is a far too general definition of the topic, so in order to make it more coherent, we selected all the examples from one field, that of protein classification and the annotation of protein domains in protein sequences. The rationale is that protein similarity searching is actually a classification problem; we assign proteins to known classes. Moreover, since protein domain similarities are especially difficult to deal with, a large part of the pertinent bioinformatics techniques were developed in the framework of protein domain analysis. For this reason, this introductory chapter focuses on the general subjects of the bioinformatics of protein domains, machine learning, and protein classification. Due of space constraints, fundamental bioinformatics techniques, such as BLAST or CLUSTAL which are subjects of many good textbooks have not been included as a part of this introduction.

1.1. Machine Learning and Protein Classification

Machine learning can be defined as the study of computational methods and the construction of computer algorithms and programs capable of learning from their own previous experience, in order to improve their performance for a defined task(Mitchell, 1997). The major focus of machine learning research is to extract information from data automatically, by computational and statistical methods. This multi-disciplinary field is closely related not only to data mining and statistics, but also to theoretical computer science. Machine learning has a wide spectrum of applications that include natural language processing, pattern recognition, search engines, medical diagnosis, brain-machine interfaces and cheminformatics, speech and handwriting recognition and object recognition in computer vision.

Shavik and collegues (Shavik et al., 1995) described the field of molecular biology as tailor-made for machine learning approaches. Approaches of machine learning are suitable for application to bioinformatics because the subjects of investigation are highly complex biological systems. Generally, the basic concept of applying machine learning in bioinformatics research is to discover meaningful knowledge from the existing biological

databases and presented in a meaningful and understandable pattern. There are several biological domains where machine learning techniques are applied for knowledge extraction from data (**Figure 1.1**).

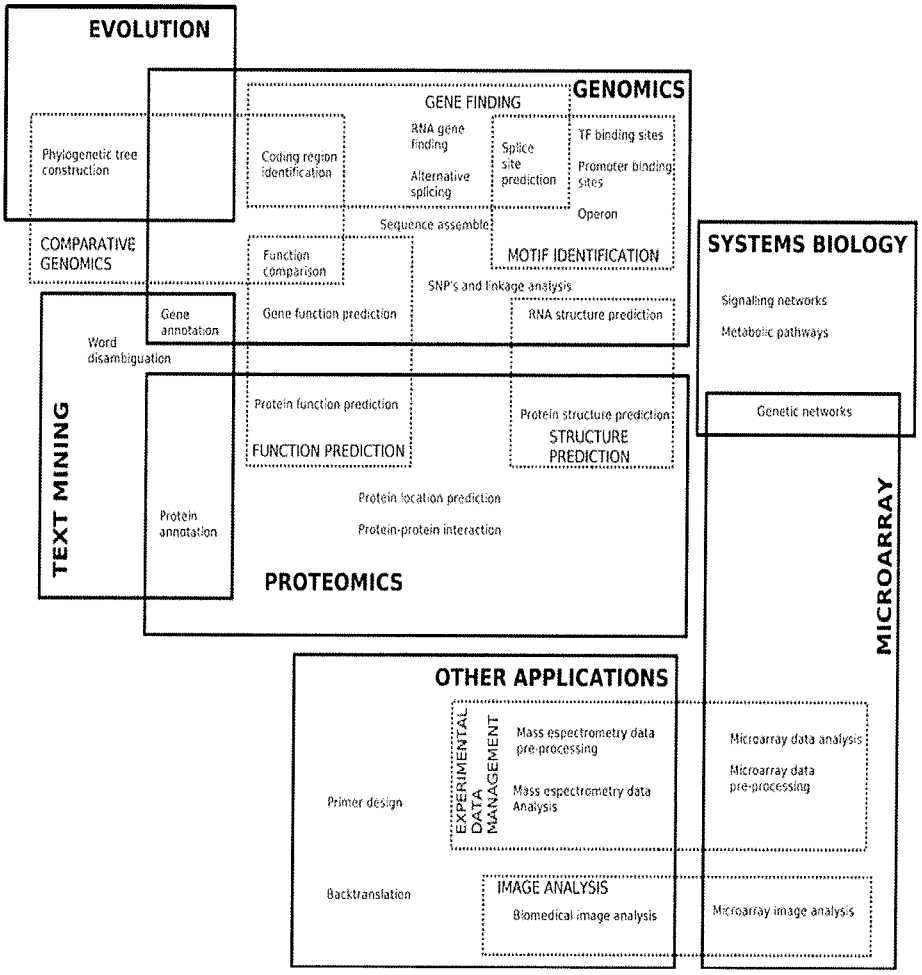


Figure 1.1: Machine learning in Bioinformatics.
Classification of the application where machine learning methods are applied in bioinformatics (Larranaga et al., 2006).

Pattern recognition, a subclass of machine learning, is the study of how machines can learn to distinguish patterns of interest based on either *a priori* knowledge or on statistical information extracted from the patterns, and make sound and reasonable decisions about the categories of the patterns. Watanabe (Watanabe, 1985) defines a pattern 'as opposite of

a chaos; it is an entity, vaguely defined, that could be given a name.' For example, a pattern could be the amino acid sequence of a protein domain.

The design of a pattern recognition system essentially involves a few general aspects. These are, (i) data acquisition and pre-processing which basically involves collecting data (variables and features) and performing feature selection (for instance removing irrelevant and redundant features), (ii) selecting the right learning algorithm depending on the task at hand (for instance evaluating several alternatives), (iii) training the classifier (or model), and finally evaluating the performance of the classifier (usually done on a separate test set).

A pattern recognition method can be carried out in several ways, the most common being, the supervised, unsupervised, semi-supervised and reinforcement learning. In the following paragraphs, I briefly describe the supervised and unsupervised learning methods, as they have been used extensively in this dissertation.

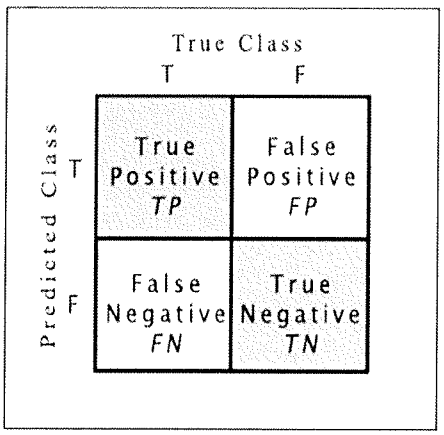
Supervised learning techniques attempt to learn associations from a manually curated data (Duda et al., 2000). This technique is carried out by partitioning the dataset into distinct classes. A class selected for an experiment is called a positive class while the complementer class within the database is called negative class. These classes are further divided into two distinct, non-empty classes called train and test set, thus resulting in a fourfold division of the dataset to yield, we get positive train, positive test, negative train, and negative test sets and such a division of the database is called a classification task. The train set consists of pairs of input objects (typically vectors) and desired output. This output could be a class label (in classification) or a continuous value (in regression). A supervised learning algorithm then adjusts its parameters according to the train set and its performance is determined on how well it predicts the output labels of the test set elements.

In an unsupervised learning approach, there is no outcome measure and the goal is to uncover trends, correlations, or patterns among a set of input measures(Hastie et al., 2003). Since all the data are unlabeled in unsupervised learning, the learning procedure consists of both defining the labels and associating objects with them. In other words, unsupervised learning tries to unveil natural groupings in the data. Clustering, principle component analysis and dimensionality reduction are some of the examples of unsupervised learning techniques. Clustering, an important process in pattern recognition and machine learning, is a process of identifying natural clusters of the data over some kind of similarity (eg.

Euclidean distance) measure. Central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered.

Assessing and evaluating classification algorithms

In many pattern recognition applications, it is not adequate to characterize the performance of a classifier by a single number, e.g. classification error rate. In a binary (two-class) classification problem, which maps an object (such as an un-annotated sequence of 3D structure) into one of two classes, the sequences with significant scores (above the set threshold value) are positive instances while the sequences with insignificant scores are negative instances, which we usually denote by '+' and '-' respectively. Based on the *priori* classification of samples, there are four possible outcomes: true positive, true negative, false positive, false negative (Duda et al., 2000). If an object is positive and it is classified as positive, it is counted as a true positive (TP); if it is classified as negative, it is counted as a false negative (FN). If the object is negative and it is classified as negative, it is counted as a true negative (TN); if it is classified as positive, it is counted as a false positive (FP). This can be summarized in a two-by-two contingency table also called a confusion matrix (**Figure 1.2**).



$$ACC = accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$SP = specificity = \frac{TN}{FP + TN}$$

$$TPR = sensitivity = \frac{TP}{TP + FN}$$

$$FPR = (1 - specificity) = \frac{FP}{FP + TN}$$

Figure 1.2: The confusion matrix and a few performance measures.
TP, TN, FP, FN are the number of true positives, true negatives, false positives and false negatives in a test set, respectively.

For binary predictions, the simplest measure of prediction accuracy from the confusion matrix is the proportion of cases that are classified correctly, termed as accuracy. However, it is possible to derive many other measures. Sensitivity and specificity are commonly used

to evaluate predictive accuracy. While sensitivity (also called true positive or recall rate) measures the proportion of actual positives that are correctly identified, specificity measures the proportion of negatives that are correctly identified. These quantities have values that lie between 0 and 1 and can be interpreted as probabilities. For instance, the false positive rate is the probability that a negative instance is incorrectly classified as being positive. Many similar indices are reviewed in (Baldi et al., 2000)

The goal behind developing classification models or classifiers is to use them to predict the class membership of new samples. With the expansion of machine learning methods in bioinformatics and other fields, researchers are frequently faced with the problem of evaluating the accuracy of a particular classifier. It is important to note that, accuracy as measured on the training set and accuracy as measured on unseen data (the test set) are often very different. It is the accuracy on the unseen data, when the true classification is unknown, that is of practical importance. In other words, it is important to have some idea about how well the classifier will perform with new data. This is known as generalization and it is linked to both classifier design and testing. It is important because the accuracy achieved with the original data is often much greater than that achieved with new data (Henery, 1994). Thus, in order to have a reliable estimate of the future classification performance, not only should the training set and the test set be sufficiently large, but the training samples and the test samples must be independent. There are no good guidelines available on how to divide the available samples into training and test sets; Fukunaga provides arguments in favour of using more samples for testing the classifier than for designing the classifier (Fukunaga, 1990).

Cross-validation procedures

Cross-validation is a commonly used and widely accepted technique in the fields of machine learning for estimating the generalization performance, model comparison and optimizing learning model parameters (Duda et al., 2000). This is one of the several approaches to estimate how well the model that just learned from some training data will perform on future as-yet-unseen data.

Hold-Out Validation: This method randomly split the available data into a training dataset and a test dataset. The model is trained on the training dataset, and predictive performance is assessed using the testing data. Hold-out validation avoids the overlap between training

data and test data, yielding a more accurate estimate for the generalization performance of the algorithm. The drawback is that this procedure does not use all the available data and the results are highly dependent on the choice for the training/test split.

K-fold cross validation: In this approach, all samples are partitioned randomly (independently and evenly) into k equal sized subset or folds with. Of the k subsets, a different single subset is held-out as the test data for testing the model, and the remaining $k-1$ subsets are used for learning. The cross-validation process is then repeated k times (the folds), with each of the k subsets used exactly once as the test data. The k results from the folds then can be averaged to produce a single estimation.

Leave-One-Out: Leave-One-Out cross-validation is a special case of k-fold cross-validation where k is the number of data points. The main drawback to the leave-one-out method is that it is expensive - the computation must be repeated as many times as there are training set data points.

ROC curve

Particularly useful for evaluating sequence and structure comparison algorithms, the Receiver Operating Characteristic (ROC) analysis (Egan, 1975; Zweig and Campbell, 1993) is the most widely used evaluation method in bioinformatics today (Sonego et al., 2008), as it is, both a visual and numeric method. It is a two dimensional measure of classification performance depicting the relationship between the true positive rate (sensitivity) and false positive rate (1 - specificity) at various thresholds T.

An ROC curve is formulated by plotting the sensitivity and specificity of the classifier against each other as a function of the threshold criterion, T. **Figure 1.3** shows an example of how to calculate a ROC curve. With the output of a classifier which is a ranked list as shown on the left hand side of figure, one can plot the ROC curve shown at the bottom left of the figure by varying the decision threshold between the minimum and maximum of the output values and plotting the FPR (1 - specificity) on the x-axis and the TPR (sensitivity) on the y-axis. One can tune the threshold of decision T, to change the number of true positives versus false positives. Increasing the number of true positives also increases the number of false alarms; decreasing the number of false alarms also decreases the number of hits. Depending on how good/costly these are for the particular application we

have, we decide on a point on this curve. For example, in **Figure 1.3**, when the threshold is set to 0.6, the TPR is 0.7, and the FPR is 0.1.

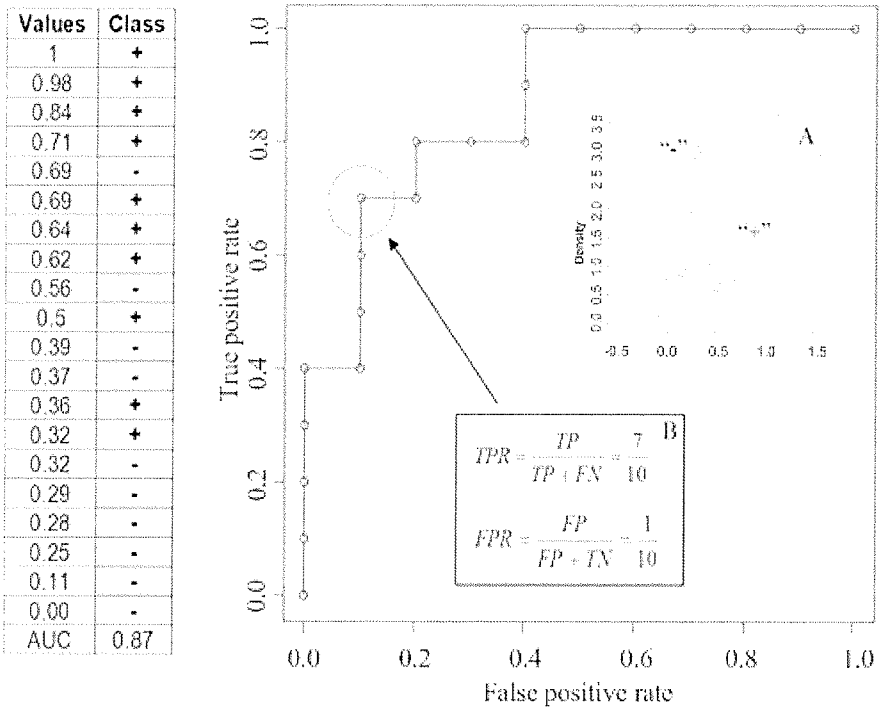


Figure 1.3: Constructing a ROC curve from ranked data.
The TP,TN, FP, FN values are determined by comparing them to a moving threshold, an example being shown by an arrow in the ranked list (left). Above the threshold + data items are TP, - data items are FP. Thus a threshold of 0.6 produces the point FPR=0.1, TPR=0.7 as shown in inset **B**. The plot is produced by moving the threshold through the entire range. The data items were randomly generated based on the distributions shown in inset **A**. (Sonego et al., 2008)

The ROC curve thus presents graphically the trade-off between false positives (FP) and false negatives (FN) in the classification process. The Area Under the ROC curve (AUC) provides a scalar value that reflects the overall quality of the classifier and is a value between 0 and 1. While a perfect classifier has a rectangular shape with an AUC equal to 1, a random classifier that returns random answers irrespective of the input is approximately a diagonal line and the integral of this curve is 0.5. A correct classifier has a ROC curve above the diagonal and an AUC > 0.5.

Protein Classification

The field of protein classification is a varied landscape and a large variety of methods have been used to approach this highly complex problem. When a new genome is sequenced perhaps, the first key question is the structure and function of the encoded proteins. The primary objective of protein classification is to automate the laborious task of functional annotation and functional prediction.

Proteins can be classified according to similarities in their sequences, in their structures and in their functions. The relationship between such levels of description makes them complementary and in practice, the best approach to protein classification utilizes several systems, in an attempt to leverage the advantages of each system.

General methods of protein classification fall into four broad categories:

- Nearest Neighbor based methods work by comparing an unknown object (protein sequence or structure) with members of an a *priori* classified database of protein objects. The results are ranked according to the similarities and the strongest similarities are evaluated in terms of biological or statistical significance, after which a query is assigned to the class of the most similar object.

- Generative models are based on consensus (or aggregate) descriptions of protein groups. Methods for preparing consensus descriptions include regular expressions, frequency matrices, profiles and Hidden Markov Models. The unknown query is then compared to a collection of generative models and the strongest similarities are evaluated and used to assign the protein to the given class.
- Discriminative models seek to determine a boundary between a class (positive group) and its immediate similarity neighbourhood (negative classes). Such boundaries are established with learning algorithms, among which kernel methods, in particular support vector machines are extremely popular.
- Network models use a graph-like representation in which proteins are the nodes and similarities are the (weighted) edges. Such a network can be evaluated by simple local statistics (Murvai et al., 2001) or by propagation algorithms such as the PageRank algorithm (Brin and Page, 1998) used in the Google search engine that was successfully applied later to protein similarity searching (Weston et al., 2004).

1.2. Protein Domains

It is long known that proteins exhibit a modular architecture comprising of several building blocks, known as domains. Generally speaking, domains are the structural and functional building blocks of proteins, but definitions vary according to the field of study. Structural biologists prefer 3D definitions based on compactness which provides a stable globular core, while the domain concept of molecular biologists is sometimes nothing more than that of a sequence segment that plays some functional role and/or is associated with a conserved exon/intron structure. According to PROSITE, protein domains or motifs are ideally defined by a specific combination of secondary structures that has a particular topology and is organized into a characteristic three dimensional (3D) structure (Sigrist et al., 2010). However, as this definition can obviously not be used in the absence of a known 3D structure, domains more commonly correspond to a region of sequence homology identified in otherwise apparently unrelated proteins. In this case, the conserved region is supposed to fold into a similar secondary and tertiary structure, independent of the context in which it is found. Domains not only share acommon structure but also often have similar function that contributes to the global activity of the protein that contains it.

All proteins, with exception to certain disordered proteins, consist of one or more domains. Several different domains, representing structural and functional units, can be found in the so-called modular or mosaic proteins (Bork, 1992; Doolittle and Bork, 1993). As proteins have variable multi-domain architectures, particularly in complex eukaryotes, one particular domain is not always found associated with the same surrounding domains, but can be found in various combinations in seemingly unrelated proteins. Moreover it is now known that domains differ in their propensity to form multi-domain proteins. While some domains are present only in specific combinations, others participate in diverse domain architectures. Such domains which occur with many different domains are termed 'promiscuous' or mobile domains, and are important in creating the observed diversity in protein domain architectures (Basu et al., 2008; Tordai et al., 2005). A well-known example of a domain that is found in modular proteins is the epidermal growth factor (EGF) module (Baron et al., 1992). The EGF protein itself is a small soluble peptide hormone that causes cell division in the skin and connective tissue. It is generated by proteolytic cleavage between repeated EGF domains in the EGF precursor protein that contains an additional membrane-spanning domain. The EGF domain is also found among others in association with

chymotryptic, immunoglobulin, fibronectin or kringle domains, in modular proteins involved in blood coagulation, fibrinolysis, neural development and cell adhesion (Campbell and Bork, 1993).

Accordingly, the presence of various combinations of domains can be used to classify proteins in a hierarchical way into superfamilies, families, subfamilies, etc. In addition, the domains themselves can be classified following their structural relationships (Bork and Koonin, 1996; Thornton et al., 1999). Evolutionarily, it is believed that protein domains could act as 'units of evolution' (Thornton et al., 1999). As domains are often flanked by introns, it is supposed that middle repetitious sequences in introns may create hotspots for recombination to shuffle exonic sequences. Hence, chimeric proteins with totally new combinations of pre-existing domains would arise. Because of the individual contribution of each domain, a protein with a potentially new function would be created. Interestingly, modular proteins are mainly, although not exclusively, found in multi-cellular animals. It has been proposed that the metazoan radiation was made possible by exon shuffling that led to the rapid construction of multi-domain extracellular and cell surface proteins, that are indispensable for multi-cellularity (Patthy, 1999).

Representation of Protein Domains

The concept underlying domain representation is the *similarity group*. Broadly speaking, a similarity group is a group of objects that share common properties that distinguish them from the rest of a database (Ágoston et al., 2005). There is a large variety of quantitative and qualitative similarity measures that can be applied to define groups of molecules based on structure, function, citations, ontological terms etc. Protein domains (as well as protein folds) refer to *structural similarity groups* whose members share commonalities defined either in terms of sequence or in terms of three-dimensional concepts such as secondary structures, backbone representations etc. A common description of the similarity group is called a *consensus description* that can take the form of regular expressions, consensus sequences, frequency matrices, Gribskov profiles (Gribskov et al., 1987), Hidden Markov Models (HMM) (Krogh et al., 1994) etc. (Attwood, 2000). The consensus description usually does not cover the entire structure of the molecules that constitute the similarity group; it is rather a partial representation, a pattern that includes only those features of the structure which are common to all or most of the members.

Domain Annotation

Finding domains in a protein refer to two types of problems: i) *De novo* or *ab initio* detection refers to finding domains in 3D structures irrespective of the already known domains or folds. ii) Domain annotation on the other hand refers to finding instances of known domain types in newly determined sequences of structures. This approach – the subject of this section – relies on a database of known domains, defined in terms of sequences and/or 3D structure. Sequence based domain annotation methods could be roughly categorized into four major types: Nearest-Neighbor comparison algorithms, generative models, discriminative classifiers and network based models. As the methods are usually associated with a database, below I describe a few examples of domain databases.

Nearest-Neighbor comparison based methods

These methods make use of pairwise similarities for detection of protein domains. The query sequence is compared against members of *priori* classified protein domain database. Among such algorithms, the Smith–Waterman dynamic programming algorithm (Smith and Waterman, 1981) is the most sensitive, whereas heuristic algorithms such as BLAST (Altschul et al., 1990) and FASTA (Pearson and Lipman, 1988) trade sensitivity for speed. Potential domains are evaluated by analyzing the distribution similarity scores provided by the selected pairwise algorithm and similar to the "Nearest-Neighbor" paradigm for supervised learning, the query sequence is then assigned with the domain nearest to it in the domain database.

The SBASE (Pongor et al., 1993; Simon et al., 1992; Vlahovicek et al., 2005) project was initiated in order to develop a prediction scheme that can automatically recognize instances of known protein domains in the newly determined sequences, using similarity search on a reference domain sequence database. The motivation behind the SBASE project was to use pairwise alignments directly for finding known domains, without the necessity to construct and curate MAs and /or generative models. It uses a curated collection of domain sequences – the SBASE domain library – and standard similarity search algorithms, followed by post-processing which is based on a simple statistics of the domain similarity network (http://hydra.icgeb.trieste.it/sbase/). Since SBASE uses a fairly simple strategy to construct domains by performing simple BLAST (Altschul et al., 1990) searches against a manually annotated database of subsequences, it heavily relies on good annotation of domains in primary databases. The SBASE approach is especially useful in detecting rare,

atypical examples of known domain types that are sometimes missed by more sophisticated methodologies. Today, SBASE is refreshed only once a year and contains curated subset of InterPro collection (Hunter et al., 2009) that is complemented with established domain types from other sources Pfam (Sonnhammer et al., 1997), SMART (Letunic et al., 2009), Swiss-Prot annotations (Boeckmann et al., 2003). The current size of the SBASE collection is approximately 736 thousand domain sequences.

ProDom (Bru et al., 2005; Corpet et al., 1998; Sonnhammer and Kahn, 1994), is a domain family database containing comprehensive set of protein domain families automatically generated by clustering homologous segments from the Swiss-Prot and TrEMBL (Boeckmann et al., 2003) sequence databases. ProDom is based on an algorithm originally developed by Sonnhammer and colleagues (Sonnhammer and Kahn, 1994) and extended to MKDOM2 (Gouzy et al., 1999) which exploits the features of the recursive PSI-BLAST homology search algorithm. One can query the ProDom by accession number (Display a ProDom entry),SWISS-PROT/TrEMBL identifier/accession number, keyword searchor selecting the display of all proteins belonging to one or several ProDom families. Moreover, it also allows for BLAST searches in ProDom, suggesting a possible domain arrangement for any query protein. The output is either information on a given domain family or cartoons displaying the domain arrangements of all proteins matching the query. Since, PrpDom also includes the use of three-dimensional (3D) information from the SCOP database (Murzin et al., 1995), the ProDom graphical interface also provides an option for the display of ProDom domains on 3D structures.

Generative Approaches

This methodology involves building a model or pattern from each group of protein domains and then evaluating each input candidate sequence to see how well it fits the model. The input is then classified according to the model it fits best.

Pfam (Sonnhammer et al., 1997) and SMART (Letunic et al., 2009), which use Hidden Markov Models (HMM) of protein families, domains and repeats, as well as PROSITE (Sigrist et al., 2010) which uses regular expressions and profile-based methods of domain classification are databases that fall under this category. These methods allow the computational biologist to infer nearly three times as many homologies as a simple pairwise alignment algorithm (Park et al., 1998).

PROSITE (Sigrist et al., 2010), initially termed as a 'signature' database was the first domain-related database, created by Amos Bairoch in 1988. PROSITE uses two kinds of signatures or descriptors to identify conserved regions, i.e. patterns and generalized profiles. The patterns are built from alignments of related sequences collected from well characterised protein families, from the literature, from the sequence searches against Swiss-Prot and TrEMBL. The alignments generated are then checked for conserved regions and a core pattern is created in the form of a regular expression. Since patterns have limitation across whole sequence, PROSITE also creates generalized profiles. The profile structure used in PROSITE is similar to but slightly more general than the one introduced by Gribskov and co-workers (Gribskov et al., 1987) .Each pattern and profile in PROSITE is linked to an annotation document where the user can find information on the protein family or domain detected by the signature, such as the origin of its name, taxonomic occurrence, domain architecture, function, 3D structure, main characteristics of the sequence, domain size and literature reference. They are also complemented by ProRule (Sigrist et al., 2005), a collection of rules which contain information for the automated annotation of domains in the UniProtKB/Swiss-Prot database that help to reliably identify to which known family of protein (if any) a new sequence belongs.

Pfam (Sonnhammer et al., 1997), a collection of multiple protein sequence alignments and HMMs, is a excellent repository of models for identifying domains, protein families and repeats. The starting point is manually curated multiple sequence alignments also known as the seed alignments, with each alignment containing a representative set of sequences that are relatively stable between releases of the database. The seed alignments are used to build profile hidden Markov models (HMMs) that can be used to search any sequence database for homologues in a sensitive and accurate fashion. Those homologues that score above the curated inclusion thresholds are aligned against the profile to make a *full* alignment. Pfam comes in two flavors, Pfam-A is a set of 11912 manually curated and annotated models and is found in approximately three quarters of known proteins. To be comprehensive and increase the coverage further, curated families in Pfam-A are augmented by Pfam-B, which is a set of automatically generated families built from homologous sequence clusters derived from the ADDA domain collection (Heger et al., 2005). ADDA has been used from Pfam release 23.0 onwards and is a method for automatically predicting protein sequence domains from protein sequence alignments alone.

Simple Modular Architecture Research Tool (SMART) (Letunic et al., 2009) contains manually curated HMMs for the annotation and identification of genetically mobile domains and analysis of domain architectures. Originally, it focused on eukaryotic signalling domains, as these were under-represented in other domain databases. Today, SMART contains a wider spectrum of protein domains from all kingdoms of life, with the current release containing manually curated models for 784 protein domains. The underlying protein database based on completely sequenced genomes was greatly expanded and now includes 630 species. The models rely on hand curated multiples sequence alignments of representative family members, based on tertiary structures (wherever available) otherwise found by PSI-BLAST. Users looking at genome wide domain counts often end up with wrong and highly inflated numbers due to the high redundancy in existing protein databases. SMART remedies this problem by making use of two search modes, namely, a 'genomic' analysis mode, which uses only those proteins that are from the completely sequenced genomes and a 'normal' analysis mode which uses the non-redundant protein database created by SMART. The main source of protein sequences is Uniprot (Apweiler et al., 2004), complemented with the full set of stable genomes from ENSEMBL.

A common difficulty of these approaches comes from the fact that a multiple alignment is necessary which requires human intervention.

Discriminative Approaches

In the case of discriminative approaches, protein sequences are seen as a set of labeled examples (for examples, positive if they are in the group of interest and negative otherwise). The learning algorithm then attempts to learn the distinction between the classes, creating a decision boundary between the positive and negative examples. Unlike the generative approaches, both positive and negative examples are used in the training process for a discriminative approach. One of the earliest examples in protein classification is the work of the Fisher kernel method (Jaakkola et al., 1999). Later applications include versions of SVM predictions. At present there are no domain collections based explicitly on discriminative approaches, but the methodology is used by several protein classification resources.

SVM-Fold (Melvin et al., 2007) is a resource that uses SVM search to complement PSI BLAST searches. The SVM predictors are trained on groups of the SCOP database. SVM-Prot (Cai et al., 2003) contains SVMs trained on 54 Pfam families. SVM predictors trained

on BLAST output parameters were also developed for SBASE. This is an approach that builds not on sequence inputs but on properties of protein alignments such as score coverage, length coverage, score/HSP length (Vlahovicek et al., 2005).

Neural networks, another classical example of discriminative models have been used in several experimental resources. Murvai and associates used this approach in the SBASE evaluation pipeline for several years but the method was abandoned due to the large updating overheads.

Integrated Databases for Protein Domains

While the databases described above have significant overlaps in the protein families and domains they predict, they arrive at these overlaps by different means. While, using just one of the databases to analyze a query sequence makes one vulnerable to any limitations the chosen database may have, trying to use all of them at the same time but from the separate sites may lead to confusion in trying to rationalize the different results obtained at each. Thus, Meta databases (database of databases) which integrates several databases into one coherent database have been compiled to catalog and categorize these databases. These integrated database resource include, InterPro (Hunter et al., 2009), Metafam (Silverstein et al., 2001), iProClass (Wu et al., 2004), CDD (Marchler-Bauer et al., 2009) and ProGMap (Kuzniar et al., 2009) .

The InterPro collection (Hunter et al., 2009) integrates together predictive models or 'signatures' representing protein domains, families and functional sites from multiple, diverse source databases: Gene3D (Lees et al., 2010), PANTHER (Mi et al., 2010), Pfam(Finn et al., 2008), PIRSF (Nikolskaya et al., 2006), PRINTS (Attwood et al., 2003), ProDom (Bru et al., 2005), PROSITE (Sigrist et al., 2010), SMART (Letunic et al., 2009), SUPERFAMILY (Wilson et al., 2007) and TIGRFAMs (Selengut et al., 2007). Signatures from these databases that describe the same domain, family, repeat, active site, binding site or post-translational modification, are grouped into single comprehensive format of InterPro entries with unique accession numbers. Two general types of relationships can exist between Interpro entries: the parent/child and contains/found-in relationships. While the contains/found-in relationship is used to indicate domain composition and generally refers to the presence of genetically mobile domains, the parent/child relationship is used to describe a common ancestry between entries. If one InterPro entry is described as the

child of another InterPro entry, this implies that the child entry is more specialized sequence than the parent, and that in all cases a protein sequence match to the child entry implies a match to the parent as well. Signatures for the parent and child entries must overlap. Integration is performed manually and approximately half of the total ~58,000 signatures available in the source databases belong to one InterPro entry. InterPro consists of nearly 5,000 entries. Each InterPro entry contains high-quality manual annotation providing useful information on the protein family, domain etc. in question. InterPro is implemented using Oracle relational database and is accessible using text or sequence searches.

MetaFam (Silverstein et al., 2001), is a comprehensive relational database of protein family information. This web-accessible resource creates supersets of overlapping families Pfam (Sonnhammer et al., 1997), PROSITE (Sigrist et al., 2010), SBASE (Vlahovicek et al., 2005), PRINTS (Attwood et al., 2003), DOMO, BLOCKS (Henikoff et al., 2000) and ProDom (Bru et al., 2005) databases. This is achieved using set theory to compare database with one another. Users can attempt to classify their own sequences from the MetaFam server (http://metafam.ahc.umn.edu/)

Conserved Domain Database (CDD) (Marchler-Bauer et al., 2009) from NCBI, was established to annotate protein sequences with footprints of ancient conserved domains. It is a collection of multiple sequence alignments and derived database search models, which represent protein domains conserved in molecular evolution. CDD is a database of domains from numerous information resources that provide computational annotation for protein sequences and protein domains. These include, Pfam (Sonnhammer et al., 1997), SMART (Letunic et al., 2009), COGs (Tatusov et al., 2003), Protein Clusters (Sayers et al., 2009). CDD uses a Reverse Position Specific BLAST (RPS-BLAST) for comparing a query sequence to a set of many Position specific scoring matrices. CDD's collection of models can be queried with novel protein sequences via the CD-Search (Marchler-Bauer and Bryant, 2004) service at http://www.ncbi.nlm.nih.gov/Structure/cdd/wrpsb.cgi.

ProGMap (Protein Group Mappings), is a single-entry web-tool which unifies the classification information present in the current protein databases that can be queried via a single interface (Kuzniar et al., 2009). It is designed to help researchers and database annotators to assess the coherence of protein groups defined in various databases thereby facilitating the annotation of newly sequenced proteins. ProGMap is based on a non-

redundant dataset of over 6.6 million protein sequences which is mapped to 240,000 protein group descriptions collected from UniProt (UniProt, 2010), RefSeq (Pruitt et al., 2009), Ensembl (Kersey et al., 2010), COG and KOG (Tatusov et al., 2003), OrthoMCL-DB (Chen et al., 2006), HomoloGene (Sayers et al., 2009), TRIBES (Enright et al., 2003) and PIRSF (Nikolskaya et al., 2006). Instead of creating a new classification scheme, ProGMap combines the underlying classification schemes via a network of links constructed by a fast and fully automated mapping approach originally developed for document classification. The web interface (http://www.bioinformatics.nl/progmap) enables queries to be made using sequence identifiers, gene symbols, protein functions or amino acid and nucleotide sequences.

The iProClass (Wu et al., 2004) database is an integrated resource that provides comprehensive family relationships, structural and functional classifications and features of proteins. It provides rich links to over 50 databases of protein sequences, families, functions and pathways, post-translational modifications, protein–protein interactions, protein expressions, structures and structural classifications, genes and genomes, ontologies, taxonomy and literature. The current version consists of about 830,000 non-redundant PIR-PSD, SWISS-PROT, and TrEMBL proteins organized with more than 36 000 PIR superfamilies, 145,000 families, 4000 domains, 1300 motifs and 550,000 FASTA similarity clusters integrates PIR superfamilies and PROSITE motifs. Implemented in the Oracle object-relational database system, iProClass employs an open and modular architecture for interoperability and scalability. The integrative data warehouse approach like iProClass allows systematic detection of genome annotation errors, comparative studies of protein function and evolution, and provides sensible propagation and standardization of protein annotations. The database is freely accessible from the web site at http://pir.georgetown.edu/iproclass/. Protein entries can be retrieved using a single protein ID or one of many other sequence database identifiers. It provides two types of summary report for the information retrieved: Protein summary report, which contains information about protein ID and name, source organism taxonomy, sequence annotations, data cross-references, family classification, and graphical display of domains and motifs on the amino acid sequence. The second type of information, known as the Family summary report, is available for PIRSF families and contains information about PIRSF number and general statistics, family and function/structure relationships, database cross-references, and graphical display of domain and motif architecture of seed members or all members.

The SCOP and CATH Classification Schemes

SCOP (Murzin et al., 1995) and CATH (Orengo et al., 1997) are the two well documented protein 3D classification schemes representing the most significant efforts to classify structural information available from the Protein Data Bank (PDB) (Berman et al., 2002). CATH and SCOP are based on hierarchical classification of protein domains into structural groups. They are widely used as gold standards to benchmark novel protein structure comparison methods as well as to train machine learning approaches for protein structure classification and prediction. Both SCOP and CATH partition the protein structure universe hierarchically (nested groups), proceeding from coarse-grained to fine-grained partitions. The top levels of the hierarchy are defined by the three-dimensional structure, whereas lower levels are identified on the basis of sequence similarity and functional considerations.

SCOP was among the earliest efforts to classify protein structures into folds. It aims to provide a detailed and comprehensive description of the structural and evolutionary relationships between all protein structures by manually labelling them (Murzin et al., 1995). The SCOP database uses four-level taxonomy: class, fold, superfamily, and family (**Figure 1.4).** Each *domain* in a protein structure is assigned to one category in each of these four levels. There are three additional levels, namely, protein domain, species, and entry domain. The topmost level of SCOP, *class*, defines 11 different classes. Four of these are not true classes; these are short peptides, low resolution structures, and engineered proteins. The four major classes, ones where the majority of structures reside, are 'all α', 'all β', 'α/β' and 'α+β' roughly describing the content of secondary structure elements in the domain. The three remaining actual classes are multi-domain, membrane, and small proteins. Proteins in a common fold have the same major secondary structures in the same arrangement with the same topological connections. The *superfamily* level groups together structures with a *probable common evolutionary origin.* Proteins with low (insignificant) sequence similarity, but whose structural and , in many cases, functional features suggest a common evolutionary origin, are grouped in the same superfamily. Domains clustered in the same *family* are likely to have a common evolutionary origin based on sequence similarity or functional evidence. Generally the sequence identities (between the sequences for structures belonging to the same family) are above 30%. However, in some cases structural and functional features can provide the evidence alone, in spite of lower sequence similarity.

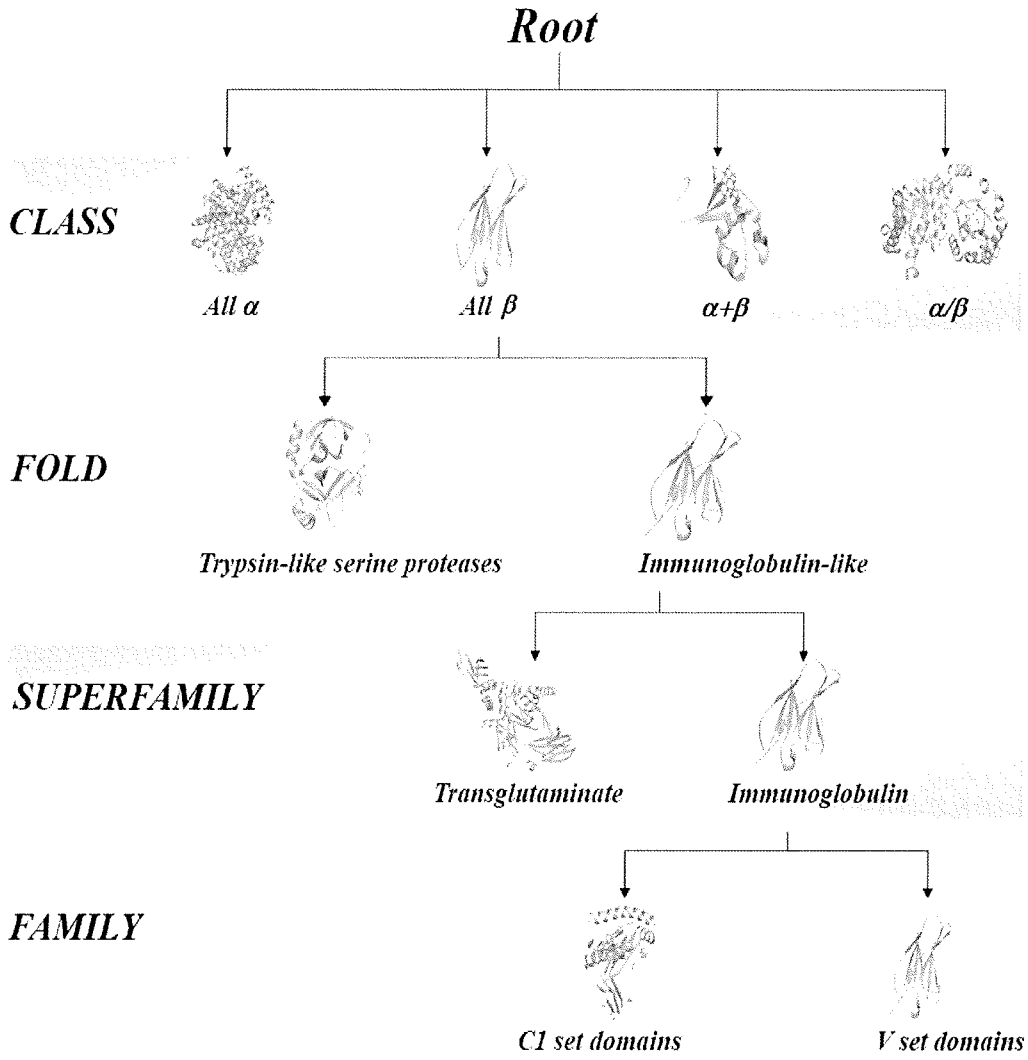


Figure 1.4: The SCOP hierarchy

Unique identifier of a domain in SCOP contains concatenation of 4 subparts, a letter 'd', the PDB ID of the protein that the domain is a part of, the chain letter, and a digit indicating its domain number within the chain. For instance, the second domain from chain C of the protein with PDB ID 1FJG, is assigned the SCOP identifier 'd1fjgc2'. The latest release (1.73) contains 92,927 domains organized into 3464 families, 1777 superfamilies and 1086 folds. The SCOP domains correspond to 34,495 entries in the PDB. The actual part of a PDB file corresponding to a SCOP domain can be retrieved from the ASTRAL database.

As compared to the SCOP, the building process of CATH contains more automatic steps and less human intervention. Recognized fold groups and families are stored in the CATH database, so-called because the organization of the database reflects the hierarchy of protein (C)lass, (A)rchitecture, (T)opology or fold group, and (H)omologous superfamily. Analogous to SCOP, CATH starts at the class level defining three major classes of secondary structure content ('all α', 'all β' and 'α–β' and proteins with few secondary structures (FSS)). Domains within each class are then assigned the next level of classification, called 'Architecture' based on the similarities in their architecture, i.e. the shape created by the relative orientation of the secondary structure units in 3D space, the connectivity is not taken into account though. These shape families are chosen according to a commonly used structure classification (e.g., barrel, sandwich, roll, etc.). The first two levels of the hierarchy are phenetic, and do not say anything about the evolutionary relationship between domains in the same group The 'Topology' level is analogous to the SCOP 'fold' level and groups structures that have a similar number and arrangement of secondary structure elements with the same connectivity. The last (major) level, 'Homologous superfamily', clusters proteins with highly similar structures, sequences and/or functions, which suggest that they may have evolved from a common ancestor. Both the topology and homologous superfamily levels are assigned by thresholding a calculated structural similarity measure (SSAP) at two different levels. In addition to these four levels of classification, the CATH classification system also includes five 'SOLID' sequence levels. While, S, O, L, I further divides domains within the H-level using multi-linkage clustering with successively higher sequence identity cut-offs (35, 60, 95 and 100%), the leaves of the hierarchy 'D' are individual domains and is a simple counter appended to the 'I' level to ensure that every domain in CATH has a unique CATH solid identification code.

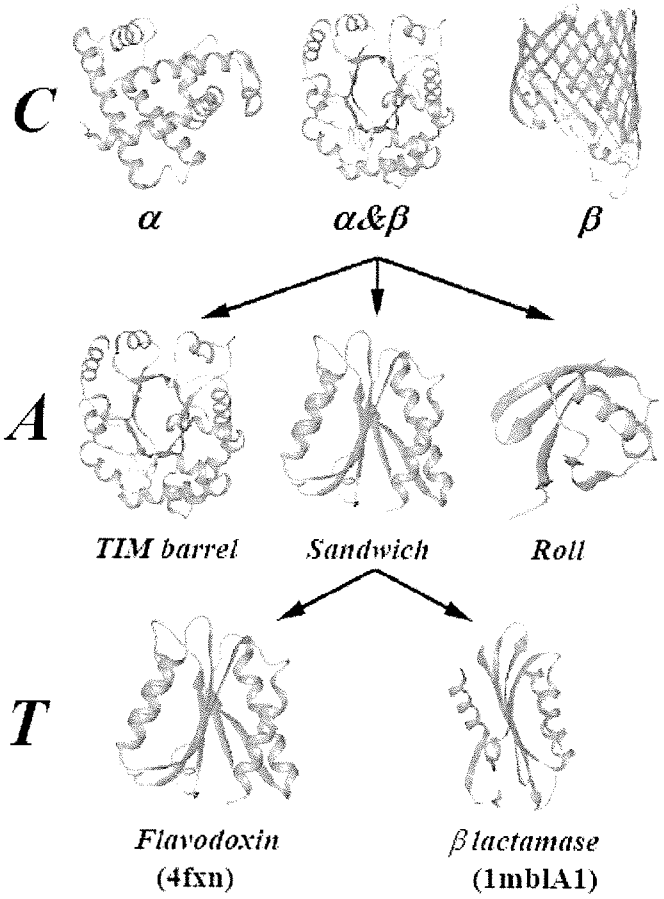


Figure 1.5: The CATH hierarchy

A particular node in the CATH hierarchy is referenced using a number for each level above and including that node. For example there is a superfamily of serine proteases domains denoted 3.40.50.200. Here the class is 3 (α/β), the architecture is 40 (3-layer αβα sandwich), the topology is 50 (Rossman fold) and the superfamily is 200 (serine protease). The latest release of CATH (v.3.2) comprises of 1233 fold groups and 2178 homologous superfamilies.

Dali Fold Classification

Dali Fold Classification (http://www.ebi.ac.uk/dali,(Holm and Sander, 1994)) provides domain classification using structure-structure alignment of proteins. The classification is based on an exhaustive all-against-all structure comparison using Dali (Holm and Sander, 1998) structure comparison. Each domain is regarded as a point in a high-dimensional fold

space, and a multivariate scaling method is used to find the groups of proteins sharing common features. Dali Database is updated twice a year and contains precomputed structural alignments of PDB90 against the full PDB(PDB90 is a representative subset of the PDB. This is a non-redundant subset in which no two chains share more than 90% sequence identity). The query structure is mapped to the closest representative in PDB90 and the structure comparison scores are recomputed using the transitive alignment via the representative.

Sequence Databases

Until now this chapter this chapter provided a description of the existing systems for protein classification based on the underlying methodology: domain/motif-based and structure-based which has been used extensively in this dissertation. In the following paragraph I would like to provide a brief description of the frequently used public protein sequence database, Swiss-Prot and TrEMBL (Boeckmann et al., 2003). Recent years have seen explosive growth in publicly available biological data and protein sequence databases play a vital role as a central resource for storing this data, and making them available to the scientific community. Depending on the type of data they contain, these databases can be categorized into two classes: universal and specialized. While universal databases store sequences from all species, specialized databases focus on specific families of proteins, or proteins from a specific organism. Major universal protein database archives include Swiss-Prot (Boeckmann et al., 2003), TrEMBL (Boeckmann et al., 2003), PIRSF (Nikolskaya et al., 2006), NCBI's Entrez Protein and RefSeq (Pruitt et al., 2009).Here we focus on the Swiss-Prot and TrEMBL used extensively in this dissertation.

The most commonly cited protein database is Swiss-Prot (Boeckmann et al., 2003), which is an annotated protein sequence database established back in 1986 and maintained collaboratively by the Swiss Institute of Bioinformatics and the European Bioinformatics Institute (EBI). The Swiss-Prot database strives to provide a high level of annotation through a process of literature-based manual curation and this allows the addition of as much accurate and up-to-date information as possible about each protein. The database is non-redundant, merging all reports for a given protein into a single entry, thus summarizing many pages of scientific literature into a concise yet comprehensive report. It also provides a high level of integration with other databases in the form of cross-references to other sequence databases as well as to specialized data collections.

Since maintaining the high quality annotation of Swiss-Prot limited its growth, a supplement database called TrEMBL (Translation of EMBL nucleotide sequence database) was introduced in 1996 (Boeckmann et al., 2003). TreMBL consists of computer-annotated entries automatically derived from the translation of all coding sequences in the EMBL/GenBank/ DDBJ nucleotide sequence databases that are not yet included in Swiss-Prot. To ensure completeness, it also contains a number of protein sequences extracted from the literature or submitted directly by the user community.

Both TrEMBL and SWISS-PROT are internally maintained in a relational database. Both the databases store data in a highly structured and uniform manner, which simplifies data access for users and data retrieval by computer programs. The databases are distributed in flat-files, which is a textual representation of the database in a format. They consist of a large number of structurally homogeneous entries, each representing one protein sequence together with its annotation. 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.

1.3. Kernel methods in bioinformatics

Learning from Diverse Types of Data

The large volume and complexity of biological data being generated represents both a challenge and an opportunity for bioinformatics research and development. As biology is a knowledge-driven discipline, access to information is of utmost importance and the process of successfully gaining insight into complex biological mechanisms increasingly depends on a complementary use of a variety of resources. Availability of new high-throughput data acquisition methods and advances in computing, communications and digital storage technologies has paved the way for growth in the generation and storage of large and diverse biological data sets. These large data sets are usually stored in different, autonomously structured, and relational data repositories. Mining information from these dispersed database entries of hundreds of genes/proteins is notably inefficient and shows the need for higher-level integrated views that can be captured more easily by an expert's mind.

Conceptual data integration is concerned with combining data from different databases, in different formats, into a global (conceptual) scheme. However, the growing influx of biological databases on the internet has made manual integration of relevant biological information a seemingly impossible task.

In the practice of protein bioinformatics, combinations of various methods such as sequence comparison, structure comparison, phylogenetic information is routinely used, but mainly on an intuitive basis. In the practice of machine learning, there are well established methods to combine heterogeneous sources. In many practical cases, no single method is able to provide the acceptable reliability of classification and intuitively, it makes sense that combination classifiers might be able to harness the complementary information provided by different methods and to improve the generalization performance of the resulting classifier. The goal of combining methods is the extension of the information contained in the data used as the training set. In bioinformatics we may use as classifiers sequence comparison, structure comparison, etc. Combining can occur at different levels of the analysis, namely, Sensor or the data level (early integration), Classifier score level (intermediate) and Decision level (late integration) (Mottl et al., 2007) as shown in **Figure 1.6**. The different stages of integration correspond to the different stages of knowledge

acquisition into the pipeline of extracting meaningful information from experimental data to infer new biological knowledge.

a) Sensor level combination is the fusion of the signals or features obtained directly from the data objects, like for instance, using properties such as size, composition, as dimensions of a vector.

b) Classifier score level, which presupposes classifiers for the various features producing independent scores that are then combined.

c) Decision level implies fusing final decisions made separately by single classifiers.

Thus these methods provide a potential framework for integrating various sources of data in various ways. Still, at present, there is little work done on systematically applying them to bioinformatics problems.

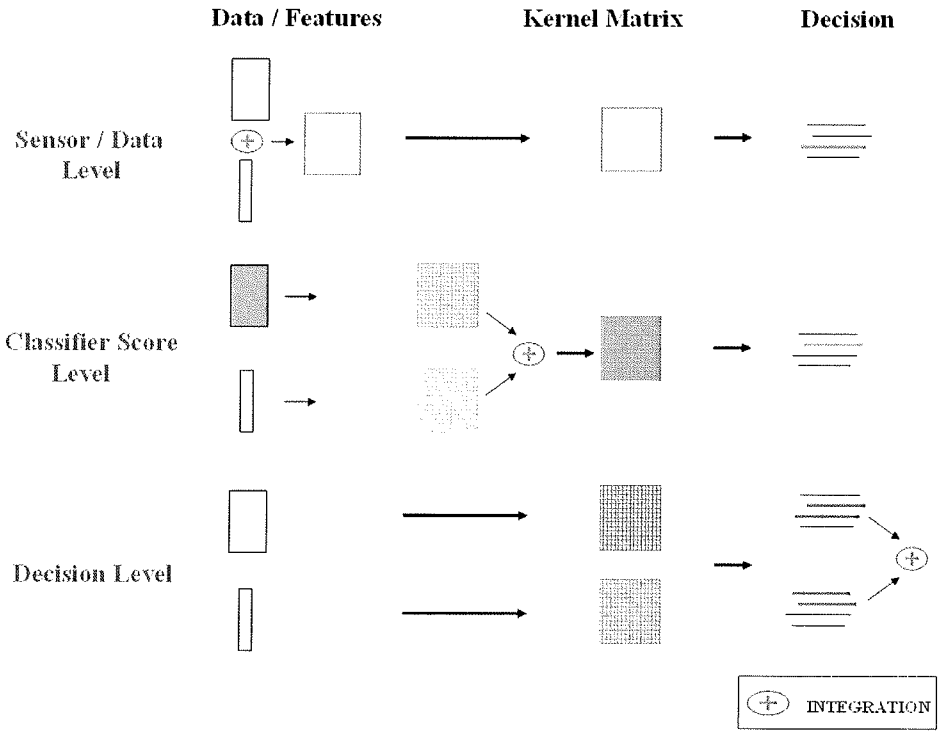
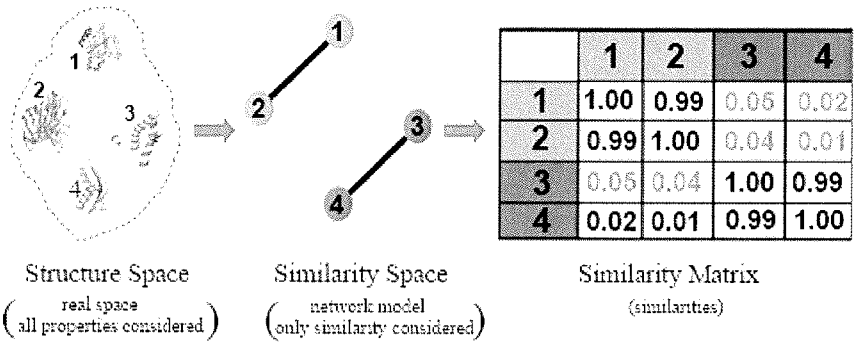


Figure 1.6: The different stages of data integration

Kernel methods

Consider an object in the real life which has many properties, entities in a database can have long lists of functions, attributes represented in a vectorial form, structural coordinates, etc. The attributes assigned to database objects and there ensemble is known as the *feature space.* However, many algorithms do not use the feature representation directly; they exploit just the relation between the objects, for instance similarity relations that numerically express how similar the objects are to each other. Using a similarity measure, we simplify this picture, as if we transferred the data into a simpler world where there are no properties, just similarities between the objects. This is the *similarity space* and it can also be depicted as a network of similarities (**Figure 1.7**). Such a network is a graph which by definition has a matrix form that stores the similarity measures resulting from all pairwise comparisons between the data objects. One of the basic problems in Bioinformatics is the comparison of DNA or protein sequences and structures and there are many algorithms specially designed for the purpose, such as BLAST (Altschul et al., 1990), PSI-BLAST (Altschul et al., 1997), FASTA (Pearson, 1990), and the Smith-Waterman algorithm (Smith and Waterman, 1981), DALI (Holm and Park, 2000), PRIDE (Gaspari et al., 2005) etc. These algorithms return a numeric similarity score expressing how similar or different the two sequences or structures are.

Many computational methods (like classification, prediction, noise filter methods, etc.) require additional properties from this similarity matrix, that is, it has to be symmetric and positive semi-definite. Support vector machines (SVMs) (Vapnik, 1998) are the most popular in this category of methods. Other methods operating on positive semi-definite matrices include Gaussian processes, Fisher's Linear Discriminant Analysis (LDA), Principal Components Analysis (PCA), Canonical Correlation Analysis (CCA), ridge regression, spectral clustering, linear adaptive filters and many others. The most common tool to obtain a positive, sem-idefinite matrix is the inner product of vectors.



	1	2	3	4
1	1.00	0.99	0.05	0.02
2	0.99	1.00	0.04	0.01
3	0.05	0.04	1.00	0.99
4	0.02	0.01	0.99	1.00

Figure 1.7: Representation of objects in the real space and in the similarity-space. The similarity matrix will be kernel matrix if similarity function is a kernel function.

Kernel functions are symmetric, semi-definite functions that provide another alternative to produce positive, semi-definite matrices. Kernel functions have two important advantages over the simple inner product: 1) they provide a convenient way to extend linear methods to non-linear ones (for example linear classifiers to non-linear ones) without raising the complexity. 2) Kernel functions can be applied directly to non-vectorial data, like collections of trees, graphs, images, DNA and protein sequences, microarray gene expression chips, etc. These have made kernel methods very popular in bioinformatics over the last few years.

The Fisher kernel (Jaakkola et al., 1999; Jaakkola et al., 2000)derived from a Hidden Markov Model (HMM) was one of the first application of kernels to protein sequence comparison. Since then there have been improvements on the performance of the Fisher kernel. The Local Alignement kernel (Saigo et al., 2004) is a convolution of all the possible gapped local alignment and it can be considered as the "kernelized" version of the Smith-Waterman. The Spectrum kernel (Leslie et al., 2002) which compares all possible k-mers with a sequence and the Mismatch kernel (Leslie et al., 2002) which compares k-mers and considers them identical if they have at most m mismatches, both gave state of the art performance when used within a support vector machine.

While classical kernel-based algorithms are based on only one kernel, recent applications (Lanckriet et al., 2004b) have shown that multiple kernels can enhance interpretability of the decision function and improve classifier performance This can be fulfilled due to the fact that the class of kernel functions are closed under the positively weighted addition. The

idea behind these multi-kernel methods is to represent a set of heterogeneous features via different types of kernels and to use the resulting combined kernel as an input to machine learning algorithms (**Figure 1.8**).

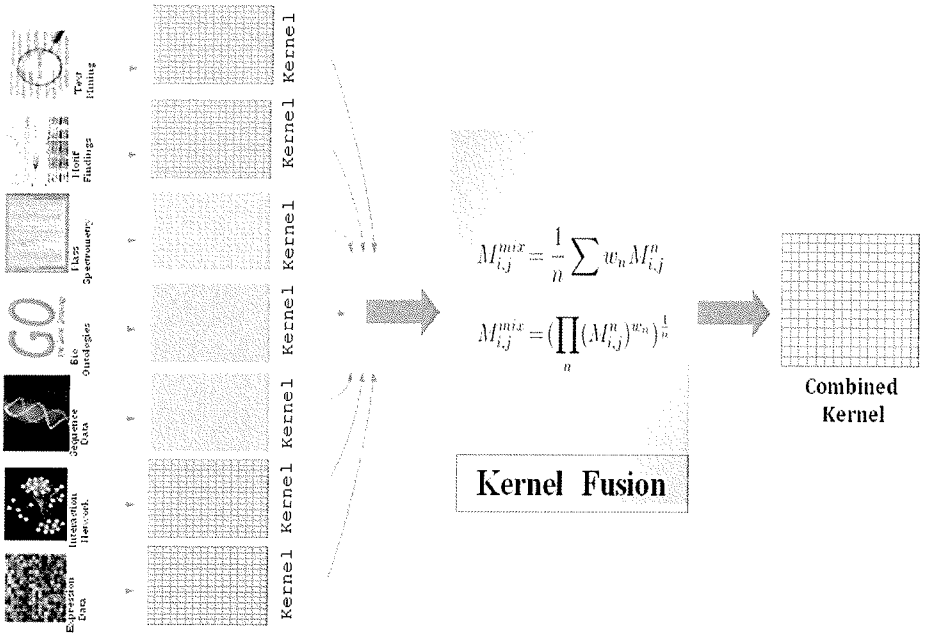


Figure 1.8: Applying kernel fusion to combine molecular biology data.
The crucial point is to convert data into relations (such as similarities, concept-distances in ontologies, interactions determined by experiment, etc.) that can in turn be represented as matrices. These matrices are normalized to the kernel form and then combined using the kernel fusion principle.

Recently, several methods have been described for handling heterogeneous data sets by combining kernels in the context of SVM learning. Pavlidis and associates used un-weighted sum of kernels to combine kernel matrices generated from microarray gene expression data as well as phylogenetic profiles, and trained SVMs to recognize functional categories of yeast genes (Pavlidis et al., 2002). Vert and co-workers proposed a "metric learning pairwise kernel" for biological network inference (Vert et al., 2007), whereas Ben-Hur and Noble predicted PPIs using pairwise kernel and simple linear combination with sequence kernels (Ben-Hur and Noble, 2005). All the above mentioned approaches used simple, unweighted linear combination of kernels with an equal weight given to all the data sources. Efforts were also made in combining multiple kernels using optimal weights, Lanckriet and

associates formulated a multiple kernel learning (MKL) problem which optimizes kernel weights by training a SVM classifier using semi-definite programming problem(Lanckriet et al., 2004a). This approach was further improved by Sonnenburg (Sonnenburg et al., 2006) who proposed multiple kernel learning based on linear semi-infinite programming, as well as by Bach and associates who suggested an algorithm based on sequential minimal optimization (SMO) (Bach et al., 2004).

RESULTS

2. Protein Benchmark Collection

2.1. Background

One of the fundamental tasks in bioinformatics is the structural and functional annotation of proteins. In a typical application, proteins of a newly sequenced genome are to be classified into one of the several thousand a priori known structural or functional categories and in view of the large number of new genomes sequenced this task is carried out to a large extent, by automated machine learning methods.

Application of machine learning techniques to proteins is a delicate task and is usually hampered by the fact that the clusters in the protein universe are highly variable in most of their characteristics (e.g. average sequence length, number of known members, within-group similarity, etc.). Despite the fact that application of machine learning algorithms to protein classification is a frequent topic in the literature, comparing the performance of a new classification method with the figures published on other methods often becomes quite difficult. In our opinion this is mainly because (i) the published results are often based on different and sometimes obsolete databases and program versions, (ii) the fine-tuning of the program parameters is sometimes not described in sufficient detail and finally, (iii) the classification performance is characterized by various, often ad hoc chosen performance measures and validation protocols.

In the practice of machine learning, cross-validation techniques are used to assess the accuracy of classification methods which involves making the algorithm learn from one randomly picked part of the data and then testing its classification ability on another part. However this approach is not very suitable in the case of protein classification because of the biological nature of the problem itself. Namely, most genomes contain novel variants of the known proteins, i.e., the similarity distribution of a known protein family in a newly sequenced genome is in fact expected to be different from rather than similar to that of its known variants. So, the foremost question then becomes, how well a given algorithm generalizes to novel subtypes and how one can assess the generalization capability of a classifier algorithm making use of some additional knowledge on the protein class.

Haussler and co-workers suggested the use of benchmarking datasets (Jaakkola et al., 2000) that were *a)* difficult enough to show differences between various methods and *b)* well enough populated so as to provide robust statistics. For example, taking into consideration the hierarchical schemes in which the protein universe is organised, if it is known that a given group consists of subgroups, one can then use one subgroup as the positive test set and pool the others as the positive training set. One can repeat this procedure for each of the subgroups. With this method each of the subgroups is considered one-by-one as a "newly discovered" subtype, so the method will estimate the classifier's average ability to discover new variants. Thus this "knowledge based cross-validation" as opposed to the random cross-validation techniques gives a more realistic estimate of the generalization capability of a classification algorithm.

In view of the above difficulties and the number of new genomes sequenced, it is critically important to define benchmark datasets for assessing the accuracy of classification algorithms. Knowledge-based or supervised cross-validation, i.e., selection of test and train sets according to the known subtypes within a database has been successfully used earlier in conjunction with the superfamilies and families of the SCOP database (Dong et al., 2006; Jaakkola et al., 1999; Jaakkola et al., 2000; Liao and Noble, 2003; Lindahl and Elofsson, 2000). The goal of the Protein Classification Benchmark collection, described in this chapter, was to extend this principle to other databases and devise standardized sets of protein data and procedures that make it easier to compare new methods with the established ones. Primarily meant for those interested in developing sequence or structure comparison algorithms and/or machine learning methods for protein classification, the collection is based on two general ideas:

(i) since, protein groups are highly variable, the performance of an algorithm has to be tested on a wide range of classification tasks, such as the recognition of all the protein families in a given database;

(ii) the utility of a classifier is determined by its ability to recognize novel subtypes of the existing proteins.

2.2. Overview of methods and data used to create the Benchmark Protein Collection

This section provides a description of the methods involved in designing and creation of the Protein Benchmark Collection:

Selection of the source data.

The collection contains datasets of protein sequences, 3D structures and in a few cases, reading frame DNA sequences of the same molecules. The sequences are deposited in concatenated FASTA format (http://www.ncbi.nlm.nih.gov/ blast/fasta.shtml), the structures are in PDB format (http:// www.rcsb.org/static.do?p=file_formats/pdb/index.html or http://www.pdb.org/).

Protein Sequence Data

SCOP95. The sequences were taken from the SCOP database 1.69 (Andreeva et al., 2004) and were downloaded from the ASTRAL Compendium (Chandonia et al., 2004) http: //astral.berkeley.edu. ASTRAL has available sequence files filtered to different levels of residue identity. For this dataset, sequences with less than 95% identity to each other were selected. In ASTRAL, domains that are non-contiguous in sequence, i.e. parts of the domain separated by the insertion of another domain, are marked with separators between the fragments representing regions belonging to other domains. Of the 12065 domain sequences downloaded, 121 non-contiguous domains were discarded, resulting in a total of 11944 entries in the dataset. The domain sequences included in this dataset are variable in terms of length and often there is relatively little sequence similarity between the protein families.

SCOP40 mini database. This small dataset comprised of sequences taken from the SCOP database 1.69 (Andreeva et al., 2004). The entries of the SCOP40, with less than 40% identity to each other were downloaded from ASTRAL. Removal of 53 non-contiguous domains resulted in 7237 entries.

Only those protein families were selected for this set that had least 5 members within the family and at least 10 members outside the family but within the same superfamily in SCOP95 thus resulting in 1375 sequences. The SCOP40mini dataset is even more difficult

since here sequences more similar to each other than 40% are represented by a single prototype sequence.

CATH95. The sequences were taken from the CATH database v.3.0.0 (Greene et al., 2007). The entries of the CATH95 (>95% identity) selection were downloaded from the ftp://ftp.biochem.ucl.ac.uk/pub/cathdata/v3.0.0/ site. 1648 non-contiguous domains were discarded and 11373 were retained for this dataset.

3PGK. The dataset was constructed from evolutionarily related sequences of a ubiquitous glycolytic enzyme, 3-phosphoglycerate kinase (3PGK, 358 to 505 residues in length) found in Archaea, Bacteria, and Eukaryota (Pollack et al., 2005). 131 3PGK sequences were selected which represent various species of the archaean, bacterial and eukaryotic kingdom (Pollack et al., 2005). The Archea consist of Euryarchaeota(11of species) and Crenarchaeota(4) phylums, the Bacteria consist of 4 phylums, namely Proteobacteria(30), Firmicutes (35), Chlamydia(3), Actinobacteridae(5) and finally the Eucaryota sequences were obtained from 7 phylums, namely Metazoa(12), Euglenozoa(5), Fungi(10), Alveolata(4), Mycetozoa(1), Viridaeplantae(8) and Stramenopiles(3).

COG. This dataset is a subset of the COG database of functionally annotated orthologous sequence clusters (Tatusov et al., 2003). In the COG database, each COG cluster contains functionally related orthologous sequences belonging to unicellular organisms, including archaea, bacteria, and unicellular eukaryotes. For a given COG group, the positive test set included the yeast sequences, while the positive training set was the rest of the sequences. Of the over 5665 COGs we selected 117 that contained at least 8 eukaryotic sequences and 16 additional prokaryotic sequences. This dataset contains 17973 sequences.

Protein Structure Data

SCOP95. 3D structures were taken from the SCOP database 1.69 (Andreeva et al., 2004). Domain structures with less than 95% identity to each other were downloaded from the ASTRAL Compendium (Chandonia et al., 2004) site http://astral.berkeley.edu/pdbstyle-1.69.html. Of the 12065 domain sequences downloaded, 121 non-contiguous domains were discarded, resulting in a total of 11944 entries in the dataset. Table 2.1 provides a distribution of the sequences/structures included in this dataset.

SCOP40 mini database. This small dataset comprised of 3D structures were taken from the SCOP database 1.69 (Andreeva et al., 2004). The entries of the SCOP40 (<40% identity) were downloaded from the ASTRAL (Chandonia et al., 2004) site http://astral.berkeley.edu/pdbstyle-1.69.html. Removal of 53 non-contiguous domains resulted in a total of 7237 structures.

Only those protein families were selected for this set that had least 5 members within the family and at least 10 members outside the family but within the same superfamily in SCOP95. This resulted in a total of 1375 structures in this dataset. Table 2.2 provides a distribution of the sequences/structures included in this dataset.

Table 2.1: *Classification of SCOP95 sequences/structures*

SCOP95 Classes	#Sequences	#Families	#Superfamilies	#Folds
α	2141	607	375	218
β	3077	559	289	143
α/β	2801	629	222	136
α+β	2612	711	407	278
Mulitdomain	204	60	45	45
Membrane and cell surface	222	98	87	47
Small	887	170	107	74
Total	11944	2834	1532	941

Table 2.2: *Classification of SCOP40mini sequences/structures*

SCOP40mini Classes	#Sequences	#Families	#Superfamilies	#Folds
α	258	102	5	4
β	377	65	6	5
α/β	679	113	11	11
α+β	23	5	1	1
Mulitdomain	20	6	1	1
Membrane and cell surface	0	0	0	0
Small	0	0	0	0
Total	1375	291	24	22

CATH95. 3D structures of CATH95 (>95% identity) were taken from the CATH database v.3.0.0 (Greene et al., 2007). The entries of the selection were downloaded from the http://cathwww.biochem.ucl.ac.uk/staticdata/v3_0_0/dompdb/ site. The 1648 non-contiguous domains were discarded thus retaining 11373 domain structures. Table 2.3 provides a distribution of the sequences/structures included in this dataset.

Table 2.3: *Classification of CATH sequences/structures*

CATH	#Sequences	#H Groups	#T Groups	#A Groups
α	2672	628	279	5
β	3334	393	176	19
α–β	5107	839	445	14
Few SS	260	100	89	1
Total	11373	1960	989	39

b) Protein comparison data. Dataset vs. dataset comparison were performed using several sequence and structure comparison methods. The methods include sequence comparisons such as BLAST (Altschul et al., 1990), Smith–Waterman (Smith and Waterman, 1981), Needleman–Wunsch (Needleman and Wunsch, 1970), compression-based distances (Kocsor et al., 2006) and the local alignment kernel (Saigo et al., 2004). The structure comparison algorithms included are PRIDE2 (Gaspari et al., 2005) and DALI (Holm and Park, 2000). Data were deposited in the form of distance matrices stored as tab-delimited ASCII files. These data can then be used directly in nearest neighbor classification schemes as well as for the training of kernel methods.

Protein Sequence Comparison Methods

Basic Local Alignment Search Tool (BLAST). An all against all comparison was carried out using BLAST (Altschul et al., 1990) version 2.2.13 downloaded from http://www.ncbi.nlm.nih.gov/BLAST/download.shtml The BLOSUM62 matrix was used with a gap opening penalty of 11 and a gap extension penalty of 1 (default parameters). The results were then stored in a compressed, tab-delimited ASCII file.

Smith-Waterman (SW). All against all comparison was carried out using the Smith-Waterman algorithm (Smith and Waterman, 1981) as implemented in the water program of EMBOSS (Rice et al., 2000). The program was downloaded from ftp://ftp.bioinformatics.org/pub/biobrew/. The BLOSUM62 matrix was used with a gap opening penalty of 10 and a gap extension penalty of 0.5 (default parameters). Results were stored in a compressed, tab-delimited ASCII file.

Needleman-Wunsch (NW). An all against all comparison was carried out using the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) as implemented in the needle program of EMBOSS (Rice, et al., 2000). The program was downloaded from ftp://ftp.bioinformatics.org/pub/biobrew/. The BLOSUM62 matrix was used with a gap-opening penalty of 10 and a gap extension penalty of 0.5 (default). The results were stored in a compressed, tab-delimited ASCII file.

Local Alignment Kernel (LAK). The Local Alignment Kernel program version 0.3 of Saigo and associates (Jean-Philippe Vert, 2004) was downloaded from http://cg.ensmp.fr/~vert/. The following run parameters were used: Default comparison matrix found in the parameters.h file. Gap opening penalty = 11 (default), Gap extension penalty = 1 (default), Scaling parameter = 0.5.

Protein Structure Comparison Methods

PRobability of IDEntity (PRIDE). Designed to compare the fold (backbone conformation)of protein structures, PRIDE is based on representing protein structures in terms of alpha-carbon distance distributions, and comparing two sets of distributions (representing two protein structures, respectively) via contingency table analysis. The program was provided by Zoltán Gaspari.

Distance-matrix ALIgnement (DALI). A protein structure comparison algorithm proposed by Holm and Sander, DALI is based on the alignment of 2-dimensional distance matrices, representing all intra-molecular alpha-carbon distances of a protein structure. In order to evaluate the DALI method the program a standalone package of the Dali algorithm known as DALI-lite version 2.4.2 (Holm and Park, 2000) was downloaded from http://ekhidna.biocenter.helsinki.fi/dali_lite/downloads.

c) Classifier algorithms. Results of various machine learning algorithms are also a part of this collection. Results are deposited for Nearest Neighbor (1NN), Support Vector Machines (SVM) (Vapnik, 1998), Artificial Neural Networks (ANN) (Bishop, 1995), Random Forest (RF) (Breiman, 2001) and Logistic Regression (LogReg) (Rice, 1994) learning algorithms. In general, the input of these algorithms is a feature vector whose parameters are comparison scores calculated between a protein of interest and the members of the training set.

d) Evaluation of classifier performance. The primary evaluation protocol used here is standard receiver operator characteristic (ROC) analysis (Egan, 1975). This method is especially useful for protein classification as it includes both sensitivity and specificity, and it is based on a ranking of the objects to be classified (Gribskov and Robinson, 1996). The ranking variable is a numerical value, such as a BLAST score, or an output variable produced by a machine-learning algorithm. For nearest neighbor classification, the ranking variable is the similarity/distance between a test example and the nearest member of the positive training set, which corresponds to one-class classification with outlier detection. As a benchmark test contains several ROC experiments, one can draw a cumulative distribution curve of the AUC values. The integral of this cumulative curve, divided by the number of the classification experiments is in [0,1], the higher values represent the better classifier performances (Jaakkola et al., 1999). Alternatively, the average AUC can be used as summary characteristics for a database, and this value is given for each benchmark test within the database.

2.3. Results

A system capable of testing and comparing machine learning algorithms should include (i) datasets and classification tasks; (ii) sequence/structure comparison methods;(iii) classification algorithms; and (iv) a validation protocol.

The most important and critical part of this research is centred on the design and creation of standardized benchmark datasets and classification tasks based on arbitrary–hierarchical schemes using which one can train a classifier and further evaluate its performance. In this process we had to create new definitions and concepts described in the following section.

Supervised Cross-Validation

A *classification task* is the subdivision of a dataset into positive train, positive test, negative train and – test groups. Training a classifier algorithm involves subdividing the database into positive and negative groups. These two groups are then further subdivided into test and train sets resulting in a subdivision of the dataset into positive train, positive test, – train and – test groups that will be used for training and testing a classifier algorithm. We will term this fourfold subdivision a "classification task". In order to get a reliable estimate of the performance of a machine-learning method on an entire database, the algorithm needs to be tested on not only one but many protein groups selected from within the database. In other words, one can choose to conduct a test at different levels of a classification hierarchy, and within each of these levels one can define many different classification tasks. We term the ensemble of the classification tasks as *benchmark test*, which may be defined as a collection of several classification tasks defined on a given database.

Since the aim was to design benchmark tests for data arranged in a tree structure, let us begin with a few words on classification hierarchies. Hierarchical classification trees of protein categories provide a simple and general framework for designing supervised cross-validation strategies for protein classification. Making use of simple graph-theoretic distance, benchmark datasets can be designed at various levels of the concept hierarchy. The resulting datasets provide lower and in our opinion more realistic estimate of the classifier performance than do random cross-validation schemes.

Let us assume that a database consist of objects that are defined according to terms arranged into a hierarchical classification tree. The dataset can then be represented as a rooted tree with the root being the database itself and the leaves are protein entries of the database. Each of the other nodes defines a subgroup of protein entries that are the leaves connected to the given node. Let $D(e, f)$ be the distance between node e and node f, which is defined here as the number of edges on the shortest path between e and f. The distance of a node from the root is called the depth of the node. We call a tree a balanced tree, if all the depth of the leaves are the same, and this distance is called the height of the tree, denoted by H.

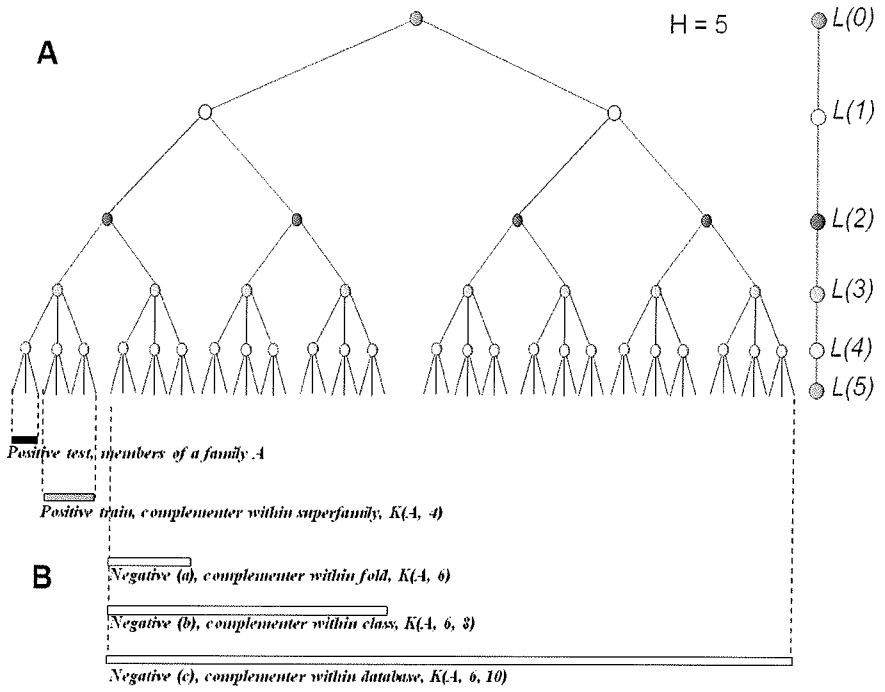


Figure 2.1: Application of supervised cross-validation scheme to an arbitrary classification hierarchy.
(**A**) Definition possibilities for positive and negative sets within a classification hierarchy. The hierarchy is a schematic and partial representation of that of the SCOP database. The positive set is defined at the superfamily level, the +test and +train sets are defined at the underlying family level. (**B**) The boundaries of the negative set can be fixed in terms of the number of steps within the tree hierarchy, calculated with respect to the positive set A. For instance, $K(A, 4)$ defines a neighborhood (a) whose members are 4 steps apart from the members of group A.

Figure 2.1 shows a typical example of a balanced tree. Any set of nodes at depth i or level i is denoted by $L(i)$ and these nodes in $L(i)$ represent a partition of the database into disjoint

groups labelled by the categories at level *i*. The tree hierarchy thus provides a supervised way to partition the database as one can define group neighborhoods by applying the *D(e,f)* distance as a proximity measure over proteins (i.e., the leaves).Constructing supervised classification tasks for a given database needs the subdivision of the data at least two adjacent hierarchical levels (e.g. superfamily/family) with the positive and negative groups defined at the higher lever *L(i)*, while the training/test subdivision is defined at the lower level *L(i* + 1). This is depicted in **Figure 2.1**, where various subgroups are defined based on the number of steps between two proteins within the hierarchy. As shown in the figure, members of the same family form the positive sets, which are two steps away from each other. On the other hand, members of the negative set are at least 4 steps away from any member of the positive set. Because of its generality, this principle can be applied to other levels of this hierarchy and to any other tree hierarchies.

Representation of classification tasks

From the implementation point of view, a classification task is described as a 'cast-vector' that assigns a membership code (positive test, positive train, negative test, negative train) to each entry in a given database. Thus, a benchmark test would be an ensemble of such cast-vectors which is represented in the form of a 'cast-matrix' or membership table. In a cast-matrix each column vector represents a classification task. In the Protein benchmark collection, for each benchmark test a cast-matrix is deposited as a tab-delimited ASCII files, with headers. The header line contains the names of the classification experiments that are represented by a column of the cast matrix. The classification experiments are named according to the group used as positive set and the subgroup used as positive test set using the general form "group_subgroup". For example, a.1.1_a.1.1.1 denotes a classification experiment where the positive set is the a.1.1. group of the database, and the positive test set is a.1.1.1 group. Similarly, Archaea_Euryarchaeota denotes a classification experiment wherein the positive set are the Archaea sequences and the positive test set are those of Euryarchaeota. The first name in a header line is "ID".

A screenshot of a cast-matrix has been provided in ***Appendix-A*, Figure A1**. Each line of the cast matrix corresponds to a sequence or structure specified by the row-name (first column). The row-names are those used in the corresponding sequence (*.fasta) or structure (*.pdb) file, and the serial order of the rows is identical with that used in those files. The values stored in the cells of each column (classification experiment, specified by

the column header) are integers that denote a role that a sequence plays in the given experiment. "0"= no role in the classification experiment; "1"= positive train; "2"= negative train; "3"= positive test; "4"= negative test;

The Protein classification Benchmark Collection

Based on this principle we have designed a simple program that can divide a hierarchically organized dataset into classification tasks. The input to the program is the classification hierarchy supplied in the form of a tab-delimited file, containing information about the each protein domain and its position in the hierarchy; the choice of the hierarchical level that is used in the classification and the minimal size of the positive training set desired. The later is necessary so as to avoid statistical bias caused by too small groups. As an example, we may want to create classification tasks of the SCOP database, at the superfamily level, so that there should be a minimum of 5 proteins in the positive training set. We will use the non-redundant SCOP95 dataset (version 1.69) in which sequences more than 95% identical are represented by one member of the group. The program will identify 246 classification tasks from this dataset, each corresponding to a family within a given superfamily.

We have applied this principle to databases of structures, protein sequences, DNA sequences, where a hierarchical classification scheme was available or could be designed.

Currently, the benchmark collection contains classification tasks for the following type of data:

- Protein 3D: SCOP (Andreeva et al., 2008) and CATH (Greene et al., 2007) (grouped according to structural hierarchy).
- Protein sequence: SCOP (Andreeva et al., 2008) and CATH (Greene et al., 2007) (grouped according to structural hierarchy).
- Protein and DNA sequence: 3PGK (Pollack et al., 2005) (grouped according to phylogenetic hierarchy).
- Protein sequence and function COG (Tatusov et al., 2003) (grouped according to functional hierarchy).

Altogether, the collection now contains a total 34 benchmark tests spread over 6405 classification tasks, 3297 on protein sequences, 3095 on protein structures and 10 on

protein coding regions in DNA. These tests were designed so as to represent various degrees of difficulty and complexity.

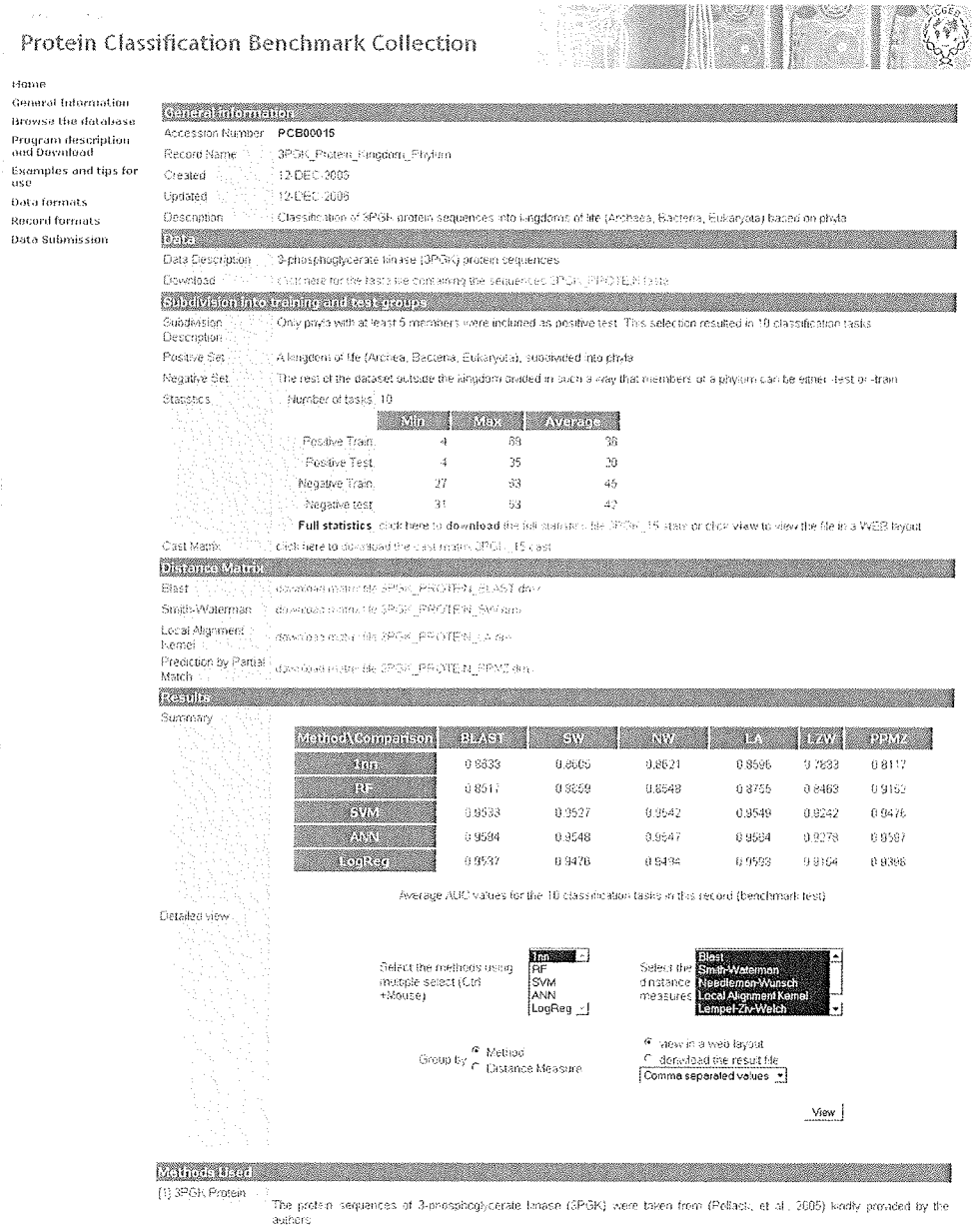


Figure 2.2: A screenshot of the benchmark database.

For the SCOP95 database, there are 6 benchmark tests defined at the *family*, *superfamily*, *fold*, *class* levels of the hierarchy, comprising of a total of 3258 classification tasks. **Table 2.4** summarizes the distribution of proteins in benchmark tests defined on the SCOP95 dataset.

Table 2.4: *The distribution of proteins in benchmark tests defined on SCOP95 dataset*

SCOP95	Superfamilies into Families		Folds into superfamilies		Classes into folds	
	A	**B**	**A**	**B**	**A**	**B**
α	614	43	522	145	1899	453
β	1507	55	1727	178	2921	466
α/β	1392	86	734	150	2675	557
α+β	503	41	583	134	2300	505
Multidomain	33	4	0	0	148	24
Membrane & cell surface	0	0	27	5	167	62
Small	274	17	308	38	817	120
Total	4323	246	3901	650	10927	2187

A = No. of positive test sequences; **B**= no. of positive test families

Similarly, one could define 8 benchmark tests on the CATH hierarchy (defined at the *homologous superfamily, topology, architecture* and *class* Levels, respectively) with 2828 classification tasks on the CATH95 dataset. **Table 2.5** summarizes the distribution of proteins in benchmark tests defined on the CATH95.

Table 2.5: *The distribution of proteins in benchmark tests defined on CATH95 dataset.*

CATH95	(1) Homology into sequence similarity groups		(2) Topology into homology groups		(3)Architecture into topology		(4) Classes into architecture	
	A	**B**	**A**	**B**	**A**	**B**	**A**	**B**
α	503	198	1277	403	2329	594	2590	617
β	498	134	1508	262	2896	360	3253	390
α–β	773	282	2370	578	4478	800	5009	833
Few SS	58	35	133	67	235	92	251	99
Total	1832	649	5288	1310	9938	1846	11103	1939

A = No. of positive test sequences; **B**= no. of positive H-groups

The collection also contains a small dataset meant for those interested in benchmarking of a new machine learning method. As the calculations are repeated many times during program development, the SCO40mini database was created. It is a small subset of SCOP comprising of 55 classification tasks (corresponding to 8 all-α, 15 all-β, 30 α/β and 2 other classes).

Designing classification tasks on the COG database (Tatusov et al., 2003) of protein functions represents another level of granularity. COG contains mostly well-characterized protein sets classified by orthology and presents a case where there is a strong sequence similarity between the members of a group but very weak similarity within groups. The recognition tasks were designed to answer the following question: can we annotate genomes of unicellular eukaryotes based on prokaryotic genomes? The collection contains a total of 189 classification tasks spread over two types of benchmark tests designed on the COG database, namely, the Taxonomic classification (Classification of Archaean protein sequences of the COG database) and Functional classification (functional annotation of unicellular eukaryotic proteins based on prokaryotic sequences in the COG database).

The 3PGK (Pollack et al., 2005) presents a case where both the within-group and the between-group sequence similarities are very high. Sequences in the 3PGK dataset are uniform in length and are closely related to each other. Despite its small size, this set is quite difficult to handle because the groups greatly differ in the number of members, and the average similarity within and between groups with any particular sequence similarity method. Two benchmark tests with 10 classification tasks each have been designed on this dataset. The benchmark tests for 3PGK fall under the taxonomic classification, involving tests such as classification of 3PGK DNA sequences (reading frames) into kingdoms of life (Archaea, Bacteria, Eukaryota) based on phyla and classification of 3PGK protein sequences into kingdoms of life (Archaea, Bacteria, Eukaryota) based on phyla.

Database Structure

The database consists of records. Each record contains a benchmark test, which consists of several (10–490) classification tasks defined on a given database. Each record contains at least one distance matrix (an all versus all comparison of the dataset) as well as performance measures (typically ROC analysis results) for all the classification tasks for at least one classification algorithm. The details are included in **Table 2.6.**

Table 2.6 : *Examples of records (benchmark tests) included in the collection*

Benchmark tests	Data	Classification tasks	Comparison methods
Classification of protein domains in SCOP [PCB0001, PCB00003, PDB0005]	11 944 Protein sequences/or protein structures from SCOP95	Superfamilies subdivided into families.....**246** Folds subdivided into superfamilies.....**191** Classes subdivided into folds.....**377**	BLAST, Smith–Waterman, Needleman–Wunsch, LA–kernel, PRIDE2
Classification of protein domains in CATH [PCB00007, PCB00009, PCB00011, PCB00013]	11 373 Protein sequences/or protein structures from CATH	H groups subdivided into S groups.....**165** T groups subdivided into H groups.....**199** A groups subdivided into T groups.....**297** Classes subdivided into A groups.....**33**	BLAST, Smith–Waterman, Needleman–Wunsch, LA–kernel, PRIDE2
CLassification of phyla based on 3 phospho-glycerate kinase (3PGK) sequences. [PCB00031, PCB00032]	131 3PGK Protein and DNA sequences	Groups of kingdoms (Archaea, Bacteria, Eucarya) subdivided into phyla.....**10**	BLAST, Smith–Waterman, Needleman–Wunsch, LA–kernel, LZW, PPMZ
Functional annotation of unicellular eukaryotic sequences based on prokaryotic orthologs. [PCB00031]	17 973 Sequences of prokaryotes and unicellular eukaryotes from the COG databases	Orthologous groups subdivided into prokaryotes and eukaryotes.....**119**	BLAST, Smith–Waterman, Needleman–Wunsch, LA–kernel, LZW, PPMZ

The accession numbers of the records are given in square brackets. The number of classification tasks for each benchmark test is shown in bold.

Random vs. supervised cross-validation

This section is meant to illustrate the difference between random and knowledge based or supervised cross-validation of data, which is the primary reason behind creating the benchmark collection. Nearest neighbour classification on the Smith-Waterman comparison was used for evaluating a particular task, the Lipocalin superfamily from the SCOP95 dataset. The superfamily consists of 58 sequences which is considered as positive set, this includes 27 sequences which belong to the fatty acid binding family, 28 sequences fall under the retinol-binding family and 3 further sequences that, in the supervised case, were not used as members of the positive test group.

We first apply two traditional machine learning approaches of random subdivision of the dataset, namely, the leave-one-out method and the 5-fold cross-validation strategy. Using the leave-one-out method, each member was used as positivetest and the rest of the superfamily was used as positivetrain. For the 5-fold cross-validation strategy, a randomly

chosen one fifth of the group was used as positive test with the remaining four fifth being used as positive train. The negative group was the rest of the SCOP95 database which was randomly subdivided either into equal test and train groups.

Table 2.7 summarises the performance of the strategies mentioned above along with that of the knowledge based cross-validation. As it can be seen, the two random cross-validation tests give rather high results (AUC values are close to 1.00), whereas dividing the samples in a supervised manner, i.e., according to the known subgroups (lines 3–4 of the table), we get substantially lower AUC values. This can be better understood if we focus our attention on the number of the subgroup members included in the positive test and positive train groups (columns D and E, respectively).

Table 2.7: *Comparison of random and supervised cross-validation strategies on the example of the Lipocalin superfamily of the SCOP database.*

	Family	No. of family members in +test	No. of family members in +train	Area under curve	Average area under curve
A	**B**	**C**	**D**	**E**	**F**
(1) Leave one out	Retinol-binding	1	27	0.9908	0.9908
	Fatty acid binding	1	26		
(2) 5-Fold cross-validation	Retinol-binding	3-7	19-20	0.9906, 0.9982,1, 0.9979	0.999127 (0.99-1.0)
	Fatty acid binding	4.7	19-23	1.000	
	Retinol-binding	28	0	0.8635	
(3) Supervised cross-validation	Fatty acid binding	0	27		0.8243 (0.79-0.86)
	Retinol-binding	0	28	0.7851	
	Fatty acid binding	27	0		

In the two random subdivisions, members of the retinol binding and the fatty-acid-binding subgroups are included in both the positive test and the positive train sets, so the comparison scores will be high. On the other hand, in the case of supervised subdivisions, members of the subgroups make part either of the positive test or of the positive train group, so the comparison scores will be lower, resulting in lower AUC values. Thus the two supervised calculations refer to situations where we attempt to predict one subgroup based

on the other one, i.e., they estimate the generalization capability of a method on this particular classification task.

Figure 2.3 compares supervised and the random cross-validation methods at various levels of the SCOP and CATH hierarchies. It can be seen that the AUC values of random cross-validation tests are substantially higher than the supervised values, i.e., the qualitative picture obtained on the example shown in **Table 2.7** is in fact general to all the classification levels of SCOP and CATH. At all levels of the hierarchies there is a clear tendency in the ranking: leave-one-out test ~5-fold cross-validation scores higher than knowledge based cross-validation.

This tendency indirectly explains why an excellent performance obtained with random cross-validation techniques does not necessarily guarantee good performance on sequences from new genomes. In other words, random cross-validation techniques may grossly overestimate the predictive power of a method on new genomes. The tendencies shown in **Figure 2.3** confirm the well-known fact that the prediction at the lower levels of the hierarchy is more efficient than at the higher levels. It is also apparent that the difference between the random and the supervised subdivision is larger at the higher levels of the hierarchy, in spite of the fact that the domain definitions and the hierarchies of SCOP and CATH are different. On the other hand, the differences of the two databases are reflected by the different shapes of the corresponding curves.

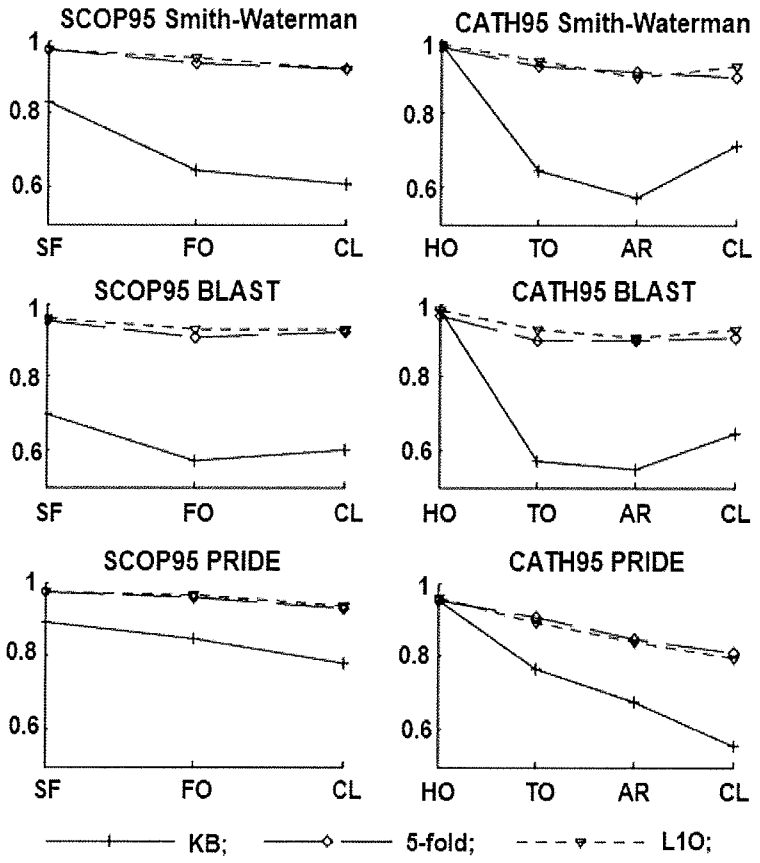


Figure 2.3: A comparison of supervised and random cross-validation schemes.
A comparison of supervised and random cross-validation schemes on the SCOP and CATH databases benchmark tests at various levels of the classification hierarchy, using Smith-Waterman (top), BLAST (middle) for sequence comparison and PRIDE (bottom) for structure comparison. Categories on the X axis are the levels of the classification (In SCOP: **SF**: superfamilies; **FO**: folds divided; **CL**; classes; In CATH: **HO**: homology groups; **TO**: topology groups; **AR**; architecture groups; **CL**: classes), the Y axis shows the average ROC scores in a benchmark test. KB=supervised (knowledge-based); **L1O**=leave-one-out; 5-fold=5-fold cross-validation. Note that the random subdivisions give higher values than the supervised (knowledge-based) ones.

Similarly one can also vary the way how the negative test is subdivided (this is not in the scope of this thesis). For instance, subdivision by superfamily means that members of a superfamily can be either –train or –test. This subdivision would then correspond to a hypothetical situation where a newly sequenced genome contains only novel superfamilies that have not been used for training. This is a stringent test, since subdivision at the fold level would mean that the new genome contains only novel folds, which is not a likely event. Subdivision by sequence, on the other hand corresponds to the random subdivision strategy employed in the general practice of machine learning.

Application example: Optimizing BLAST-based predictions

The Benchmark Database allows one to study subtle differences between predictors in a statistically well-determined environment. Sequence similarity search is by far the most frequently used methodology for predicting any property (function, domain-type etc.) from sequence. The question we are asking is whether we can leverage the efficiency of the prediction by including more the one parameter from the BLAST search, i.e. in addition to the score/e-value. A BLAST search yields the parameters shown in the following sketch:

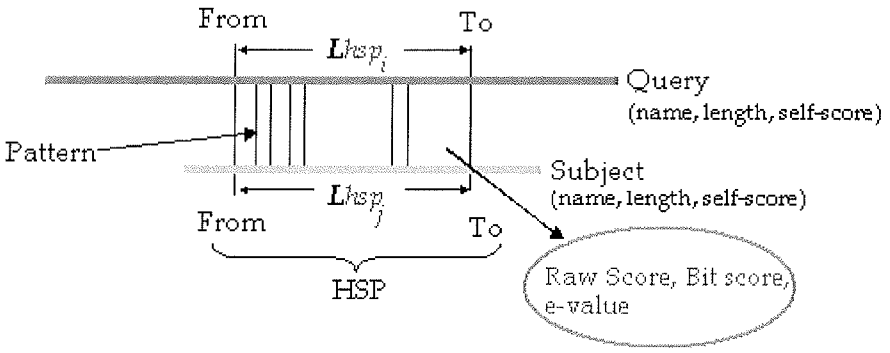


Figure 2.4: Parameters of a pairwise alignment as used by BLAST.
It is to be noted that the apparent length of the HSP may be different on the query and on the subject, as the number of gaps introduced is different in each case.

Previous studies have showed that various other derived parameters can prove to be useful predictors (Vlahovicek et al., 2005):

Length coverage = hsp/query length or hsp/subject length.

Score coverage = raw score/query self score or raw score/subject self score.

In other earlier studies we found that *NSD*, the number of significant hits found between the query and a group of proteins (in this case, the positive train group) is also an efficient prediction parameter (Murvai et al., 2001). These calculations were performed on two datasets from the Protein Benchmark Collection, namely, SCOP40mini with 55 classification tasks and the larger dataset of SCOP, SCOP95 with 246 classification tasks.

We carried out the calculations with the simple nearest neighbour algorithm, in two ways, i) using the maximal score to the positive train as the measure of similarity ("*max*") and ii) using the average score to the positive train as the measure of similarity ("*avg*").

Table 2.8: *Classification efficiency (AAUC) calculated using 1NN derived from BLAST output parameters*

Combined BLAST output	SCOP40mini (55)	SCOP95 (246)
Avg(rs/rs(query))	0.8187	0.685
Avg(rs/len(query))	0.8135	0.685
Avg(rs/len(subject))	0.7942	0.6949
Max(hsp/len(subject))	0.7828	0.6976
Avg(hsp/len(subject))	0.7802	0.6946
NSD *	0.7774	0.6975
Max(rs/len(subject))	0.7744	0.6976
Max(rs/rs(query))	0.7738	0.6976
Max(hsp/len(query))	0.7086	0.6934
Avg(hsp/len(query))	0.69	0.6892
Max(rs/rs(query))	0.5935	0.6873
Max(rs/len(query))	0.5909	0.6872

* *NSD* means number of neighbors. *Max* and *Avg* mean maximal and average aggregation. *len ()* is for the sequence length of subject or query.

From these data, it is apparent that there is variation between the various parameters. We note that averaging or maximum selections are aggregation operation, and in terms of data-integration they correspond to "sensor-level integration" (**Section 1.3 of the Introduction**) since the input of the predictor is an aggregate of several values.

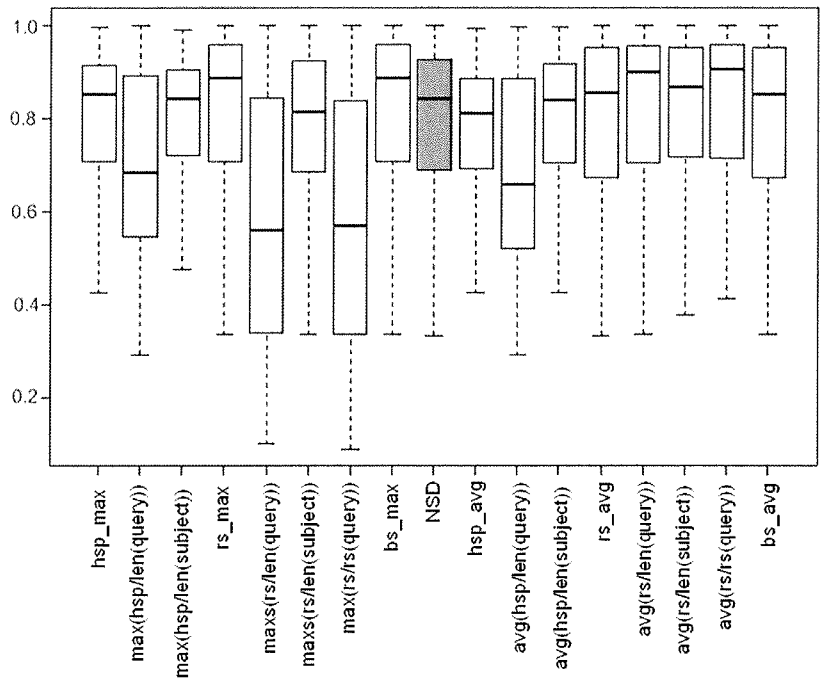


Figure 2.5: Boxplot of AAUCs (1NN) for some single and combined BLAST output parameters for the SCOP40mini.
The boxes for the combined output parameters are shown in light gray.

In earlier studies we used some of the parameters as input to train Support Vector Machine classifiers, which were used in the SBASE WWW-server. Now, as a control, we constructed a simple committee of nearest neighbour classifiers, based on the following categories:

a) protein (in the training set) which has higher raw-score *(rs);*

b) protein which has higher high-scoring segment pairs *(hsp);*

c) proteins which have the five highest bit-score *(bs).*

The predictor uses the categories, i.e. functions of (a) and (b) as votes. A third vote comes from an aggregate feature that is given by the majority function from the five proteins in (c). The third vote is a variation of the *NSD* approach, adapted in such a way that categories with few instances are not underweighted. The basic idea is that the majority vote suggests the query function. If no majority is obtained, the Committee Classifier follows the NSD vote, because it has yielded the best results during tests in SBASE (Murvai et al., 2001). In terms of data-integration, this strategy corresponds to decision-level integration, as discussed in the general introduction (**Section 1.3 of the Introduction**). The

computational efficiency of this Committee Classifier is very good as it needs no training – in contrast to SVM and other machine learning algorithms. On the other hand, it is as accurate as the SVM classifier (**Table 2.9**)

Since the Committee Classifier can not be assessed by ROC Curve, the classifiers performances are compared in terms of True Positive Rate (TPR) and True Negative Rate (TNR). It can be seen that the Committee Classifier gives a slightly worse TPR but better TNR values, in other words the results indicate that it is possible to design a simple and efficient protein classifier using a combination of separate classifiers into a voting system.

Table 2.9: *Average Classification in terms of True Positive and Negative Rate for a SVM classifier (Vlahovicek et al., 2005) with 6 BLAST output parameters as input vs. the simple Committee Classifier for the SCOP40mini dataset.*

Classifier	TPR	TNR
SVM	0.5532	0.9446
1 NN-Committee Machine	0.5067	0.9777

2.4. Summary

In this chapter, we have described the method of supervised cross-validation, which is a strategy that allows one to estimate the capability of an algorithm to recognize novel subtypes of known categories. As novel protein types abound in newly sequenced genomes, generalization capability of an algorithm is crucial for genome annotation. One can design classification tasks in a supervised way if there are known subclasses within the classes to be studied. If the categories are hierarchically organized, one can define classification tasks at various levels of the hierarchy.

Here we have presented a method that can be used to construct classification tasks (positive train, positive test, negative train, negative test groups) on any database that has a category hierarchy (such as protein domain databases, protein family databases, phylogenetic hierarchies etc.). The distinctive feature of this method is the explicit subdivision termed "supervised cross-validation" which is based on two successive classification levels. The

positive and negative groups are defined at the higher level (i), while the learning/test subdivision is at the lower level (i+1). We created a collection of protein and DNA sequences and protein DNA structures classified according to structural, functional, and phylogenetic similarity. Altogether, the collection has 6405 classification tasks spread over 42778 protein sequences and 24674 structures (**Table 2.10**). The Protein Classification Benchmark and a collection of documents and help files can be accessed at http://hydra.icgeb.trieste.it/benchmark/.

Table 2.10: *Summary of the Protein Benchmark Collection.*

Database	Classification Tasks	Number of Sequences	Number of structures
SCOP95	3258	11944	11944
SCOP40mini	110	1357	1357
CATH95	2828	11373	11373
COG	189	17973	-
3PGK	20	131	-
Total	6405	42778	24674

The collection has been used to test and compare the performance of the main types of machine learning algorithms in protein classification (P. Sonego, PhD thesis in preparation). Here we also show that it is possible to design a simple and efficient protein classifier using a combination of BLAST-based classifiers into a voting system. Since its publication in 2007, the collection was cited in nine publications.

3. Integration of heterogeneous data sources using a multi-parametric network model

3.1. Background

Biomedical research uses heterogeneous data sources –a researcher often needs to handle, say microarray, protein interaction, and DNA sequence data in one experiment. The datasets are not only varied but also large and often noisy: high throughput methods typically provide large numbers of inaccurate data that cover an entire genome or proteome, and a researcher first has to pick "coherent groups" of genes or proteins on which to work further. Finding such groups usually includes human intervention and background knowledge to which new experimental data can be mapped. The problem can appear untrivial at the first sight since, for instance, one may have expression data on some genes, interaction data on others, and very few genes on which we have both kinds of data. The general task can thus be formulated: how to find coherent groups in large heterogeneous datasets?

In homogeneous datasets, such problems can be approached by one of the many clustering techniques. Classical clustering algorithms have difficulties in handling large data sets used in bioinformatics owing to high demands on computer resources. Fast heuristic algorithms have been developed for specific tasks, for example BLASTClust from the NCBI-BLAST package (Altschul et al., 1990), Tribe-MCL (Enright et al., 2003) or the CD-HIT (Li and Godzik, 2006) that can delineate protein or gene families in a large network of sequence similarities (e.g. BLAST e-values). However, there are no apparent tools that could efficiently handle large multiple datasets, such as those necessary to group proteins using more than one similarity criterion (e.g. based on sequence, structure and/or function).

I propose to approach this problem using two simple ideas borrowed from machine learning and kernel methods.

i) Representing data as a network of relations. As mentioned in the introduction, many machine learning methods and kernel methods in particular, do not use the actual pro-

-perties ("features") of the objects, but the relationships between objects. These relations can be depicted as a network, for instance a database of protein sequences (or structures) can be transformed into a network of sequence (or structural) similarities. When we convert heterogeneous data into networks, we create a multi-parametric network that can be best pictured as a graph where several edges of different colour can exist between nodes.

ii) Fusing relation-matrices. A network can also be represented as a matrix (containing, for instance, as many rows and columns as there are proteins in the database) in which the individual cells represent a numerical index between two proteins. In kernel computations, it is required that this matrix should have specific mathematical properties in order to be called a kernel matrix, here we are not directly concerned with these limitations, so we simply speak about similarity matrices or relation matrices. In graph-theoretical terms, these are the adjacency matrices of a weighted network.

Transforming data into relational (or kernel) matrices have a simple advantage: heterogeneous data on the same object can be transformed into matrices of the same size. Such matrices can be then added, averaged, aggregated in various ways. In kernel computations, this is called kernel combination or kernel fusion.

A kernel combination process is usually comprised of two phases, first being, constructing the individual matrices whereas the second phase comprises of combining these individual kernel matrices into one. In the first phase, various models can be adopted to construct different kernel matrices, or they can be constructed on different features or from different sample datasets. In the second phase, the kernels are combined by fixed or trained rules. The simplest way to combine kernels is by averaging them. But not each kernel matrix should receive the same weight in the decision process, and therefore the main force of the kernel combination study is to determine the optimal weight for each kernel.

These mathematical formalisms provide us with a straightforward way to combine heterogeneous data. Given a set of proteins represented as sequences or structures, one can easily compute a kernel matrix based on various pairwise similarity measures available. The sum or product of various such kernels would be a new kernel, which is an extension of the incorporating kernels. Two cases emerge when we try to combine kernels, namely, the kernels are either built on the same feature space or they work on different feature space. The operations of linear combination are valid for both the cases. We did not mention

the computational load associated with large multiple networks. We have already mentioned that conventional clustering algorithms have difficulties in handling large bioinformatics dataset. This problem is further aggravated if we use multiple networks.

The aim of this chapter is to design a simple algorithm and tool that overcomes these difficulties and allows one to handle heterogeneous data (large biological networks). In particular, I present an efficient pre-processing tool that can aid exploratory analyses of large biological networks using an ordinary computer.

3.2. Informal Description of the Algorithm

We developed a heuristic algorithm, Multi-Netclust that takes the users' empirical knowledge of cut-off values into account. Below these threshold values interactions or similarity data can be neglected. As a result, multiple thresholded datasets are created and combined together using an averaging or kernel fusion method (Kittler et al., 1998). The resulting combined network can then be queried for connected components using an efficient implementation of the UNION find algorithm (Tarjan, 1975). Connected components correspond to groups of nodes that are connected either by any or by all of the constituent network datasets, depending on the form of the weighted averaging used (**Figure 3.2, B** and **C**, respectively). Connected-component search has been widely used in grouping proteins into protein families or orthologous groups owing to its simplicity, scalability and biological soundness(Koonin et al., 2004). There are two distinct algorithmic approaches to this problem. The first approach requires an entire graph to be stored in computer's memory prior to detecting the clusters using either depth-first search or breadth-first search algorithms(Grimaldi, 1999). This approach also denoted as 'in core' is, however, memory-expensive particularly for large graphs of millions of nodes and/or edges ($O(E)$ or $O(N^2)$ space-complexity depending on the implementation, E=number of edges, N=number of nodes). The second, more memory-efficient approach (denoted as 'external-memory' from here on) does not need the entire graph to be stored in computer's memory but instead, the clusters are constructed gradually while reading-in the graph from a hard disk, hence saving a lot of memory space which can be achieved using family of UNION-FIND algorithms. In order to adapt this method to large heterogeneous datasets, we combined the thresholding, aggregation as well as connected component search into a single, memory and time efficient tool, Multi-Netclust that uses external-memory (Chiang, 1995) for matrix manipulations so that the size of the datasets is not a limiting factor.

The calculations are carried out in the following steps:

1) Pre-processing and filtering of the data: The data are transformed into a network/matrix format. Sequence data can be transformed into similarity matrices using a BLAST similarity, or a word-composition similarity. Some data, such as protein interaction data, are already in a network/matrix format.

To take into account the very different sizes of the feature space, matrices can be normalized (element-wise) in the following way,

$$\tilde{K}_{ij} = \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}} \qquad (1)$$

Here K is the matrix and i and j denote row and column of the matrix respectively. Filtering of the matrices involves the use of thresholds, below (or above) which the elements of the matrices are ignored.

2) Combination of network data: Let $\mathbf{K}_1$, $\mathbf{K}_2$, ...$\mathbf{K}_n$ be a set of normalized and filtered matrices, we can then make use of the following two methods to combine them.

The arithmetic mean,

$$K_{sum} = \frac{1}{n}\sum_{i=1}^{n} w_i K_i \,. \qquad (2)$$

Geometric mean,

$$K_{product} = \left(\prod_{i=1}^{n} (K_i)^{w_i}\right)^{\frac{1}{n}} . \qquad (3)$$

where, w_i is the weighting factor used for $\mathbf{K}_i$ and n denotes the number of matricies to be combined. The sum of weights (w) should be equal to 1. The weights provide a degree of freedom as one can assign more importance to one matrix than the other thus controlling the participation of each matrix in the decision process.

3) Finding connected components: An efficient, external-memory implementation of this algorithm is a key element of our program. This was achieved by implementing the asymptotically optimal UNION find algorithm variant with (nearly) linear time- and space-complexity (*O(E * alpha(E))* time-complexity in the worst-case scenario, *E*=number of edges,*alpha*=inverse Ackerman's function; *O(N)* space-complexity, *N*=number of nodes), hence enabling the analyses of very large data sets in almost real-time. The underlying algorithm has three abstract operations: (i) populate singletons, (ii) find group memberships, and (iii) merge groups sharing at least one member. The (preliminary) clusters are stored as rooted trees, which are then subjected to two post-processing steps. First, each tree is "compressed" so that all nodes (members) of a tree point directly to the root of that tree. Second, the resulting trees (clusters) are sorted by size and labelled by increasing integer values.

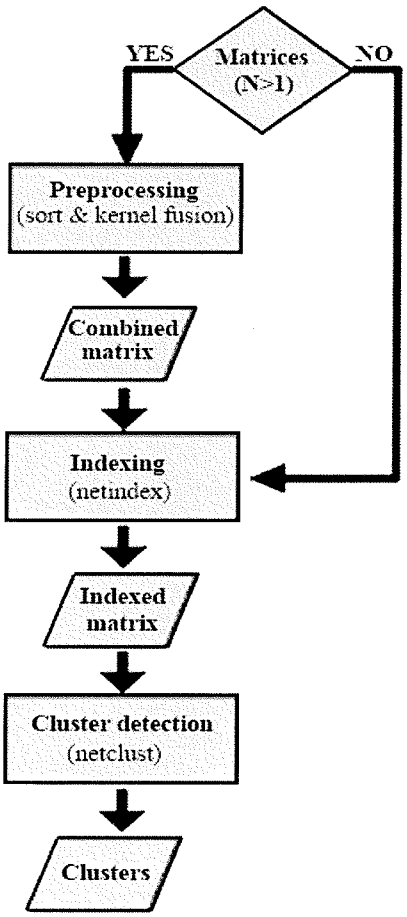


Figure 3.1: The principle of Multi-Netclust.

3.3. The Multi-Netclust Program

Multi-Netclust is a program that filters and combines biological network data and extracts connected groups. Multi-Netclust has been implemented in the C++ programming language and provided with a CGI interface written in Perl that serves for inputting the data.

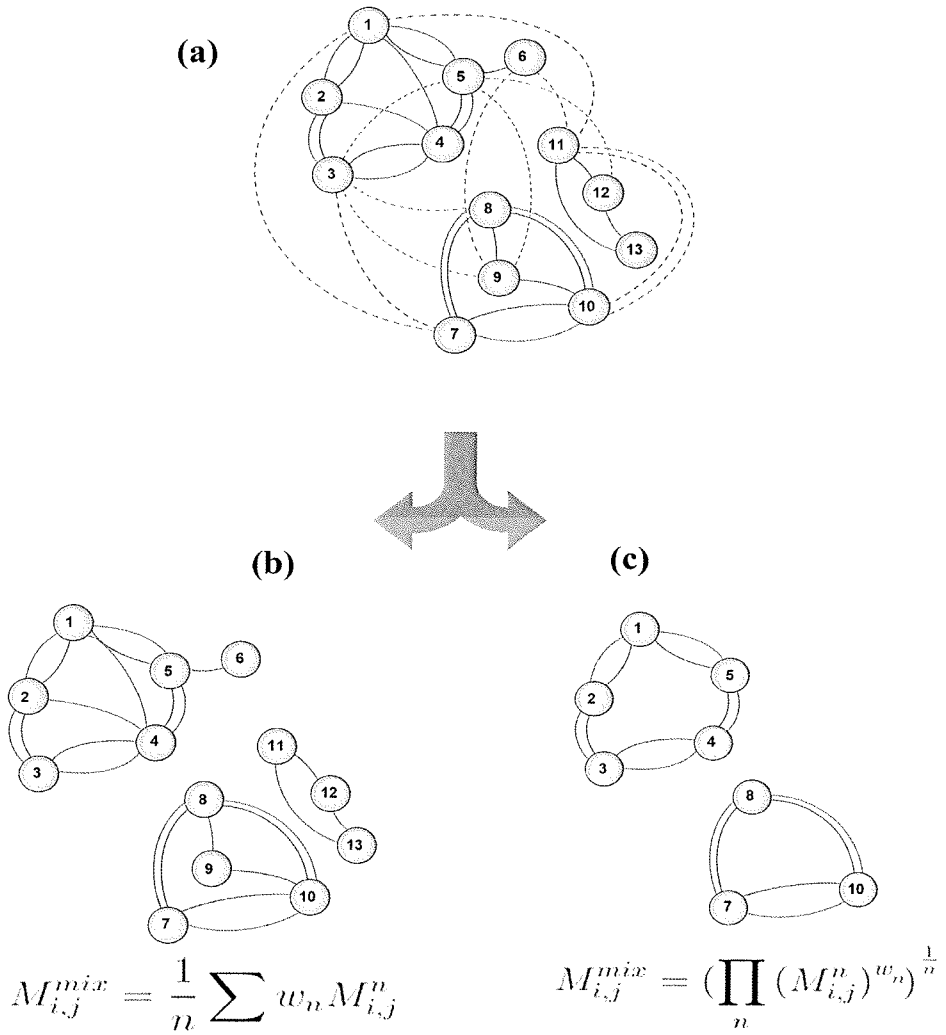


Figure 3.2: The principle of Multi-Netclust.
The principle of Multi-Netclust is illustrated on a two-parameter network or hypergraph (a) consisting of red and gray edges. Dotted lines denote edges that are below the respective threshold and hence are omitted from the networks. (b) Aggregation by weighted arithmetic averaging ("sum rule") gives connected components that are connected within either of the two networks. (c) Aggregation by weighted geometric averaging ("product rule") gives connected components

connected within both networks. M_{ij} denotes the value assigned to the edges, w is the weighting factor of the two matrices, and in the above example n=2.

The code, sample datasets, explanations, and performance data are available on the project's website http://www.bioinformatics.nl/netclust/. There is also a web-based application suitable to run smaller test-sets.

Input to the Multi-Netclust program are (un)directed weighted graphs (networks) in sparse matrix format along with weight and threshold/cutoff values associated with each network. The so-called sparse matrix format is used for the data which can be regarded as an edge list of a weighted graph. This representation was chosen in order to minimize the storage requirements as well as the time necessary to access matrix elements. The fact that a separate threshold value and combining weight can be assigned to each matrix, ensures that the user can control the participation of each kernel in the decision process and can thus control the sensitivity and specificity of results. For example, spurious similarities can be filtered out by choosing an appropriate cutoff value, which usually requires domain knowledge. Generally, a permissive cutoff may produce one large cluster while a strict cutoff may yield many singletons. Prior to the cluster detection, the input graph must be indexed using the netindex utility to speed-up the the downstream processing. The steps involved in the Multi-netclust workflow are schematically depicted in **Figure 3.1**. Moreover, the speed of the algorithm allows selecting the most appropriate weighting factor through a standard grid search. Data to the Multi-Netclust can be entered either via a CGI interface, or from the command line (**Figure 3.3**). The output is a list of the connected clusters given in a structured text format.

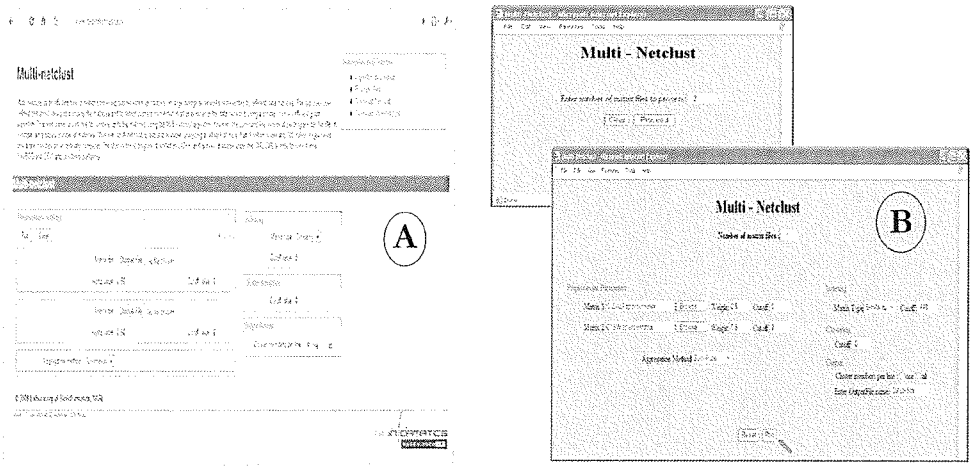


Figure 3.3: The Multi-Netclust.
(A) Screenshot of the Multi-Netclust web interface. **(B)** Screenshot of the Multi-Netclust form based interface.

3.4. Performance

The CPU-time of Multi-Netclust subsumes a) preprocessing time needed for reading-in the data, thresholding and aggregation (>99.9%), and b) finding the connected components and writing the results (0.1%). The below mentioned case studies gives the idea of how Multi-Netclust can be used and the improvement one gets upon combining information from multiple sources.

3.5. Application example 1

The task was to use Multi-Netclust to combine BLAST and DALI similarity data on a protein dataset and cluster proteins belonging to the same superfamily defined in the SCOP database (Andreeva et al., 2004).

Dataset and similarity matrices:

The validation dataset used for this experiment was taken from the Protein Classification Benchmark database (Sonego et al., 2007). It consists of 1357 proteins belonging to 24 superfamilies.

BLAST matrix:

An all against all comparison of protein sequences belonging to the above mentioned dataset was calculated using BLAST (Altschul et al., 1990) version 2.2.13. BLOSUM62 matrix was used with default parameters. The results were stored into a sparse matrix.

DALI matrix:

Structural similarity in the form of DALI similarity score (raw score) was calculated by the DALI-lite program for pairwise structure comparison and database searching, version 2.4.2 (Holm and Park, 2000). Results were stored into a sparse matrix.

Results:

The analysis took 4 seconds on a 2 GHz processor. The influence of thresholds on the purity of connected clusters is apparent from the data (**Table 3.1**).

Table 3.1: *Combining network data using the sum and product rule at different threshold levels.*

Dataset	Weight	Correct	Incorrect	Singletons
BLAST **(0)**	1	0	1356	1
DALI **(0)**	1	0	1352	5
BLAST **(0.1)**	1	66	1101	190
BLAST **(0.4)**	1	36	0	1321
DALI **(0.4)**	1	798	468	91
BLAST **(0.4)** + DALI **(0.4)**	0.5	803	469	85
BLAST **(0.1)** × DALI **(0.4)**	0.5	888	0	469

Numbers in parentheses denote the similarity cutoffs used. '×' and '+' refer to the product and sum aggregation rules, respectively. Correct = proteins connected only to members of the same SCOP superfamily, Incorrect = proteins connected to members of other SCOP superfamilies. Weight = the weighting factor assigned to each network.

An interesting example in this case-study is the Immunoglobulin superfamily (SCOP Superfamily Id b.1.1) which has 125 members in the benchmark dataset. Using DALI alone as an input, clusters them with the E set domains (SCOP Superfamily Id b.1.18), which is an "Early" Ig-like fold families possibly related to the immunoglobulin and/or fibronectin type III superfamilies. With BLAST, they are clustered with a number of other superfamilies whereas, the combination of the two (BLAST (0.1) * DALI (0.4)) made 94% of the group cluster correctly.

3.6. Application example 2

The task was to use Multi-Netclust to combine Smith-Waterman and DALI similarity data on a protein dataset and cluster proteins belonging to the same SCOP superfamily.

Dataset and kernel matrices:

The dataset used for this experiment was the same as in Example 1. Sequence similarity used for this example was calculated using the Smith-Waterman algorithm (Smith and Waterman, 1981). BLOSUM62 matrix was used with default parameters. Structural similarity in the form of DALI similarity score (raw score) was calculated by the DALI-Lite program for pairwise structure comparison and database searching, version 2.4.2 (Holm and Park, 2000). The results were stored into a sparse matrix.

Results:

The analysis took 6 seconds on a 2 GHz processor, the influence of thresholds on the purity of connected clusters is apparent from the data (**Table 3.2**).

Table 3.2: *Combining network data using the sum and product rule at different threshold levels*

Dataset	Weight	Correct	Incorrect	Singletons
DALI (0)	1	0	1352	5
SW (0)	1	0	1357	0
SW (251)	1	316	0	1041
SW (448)	1	74	0	1283
DALI (251)	1	56	1266	35
DALI (448)	1	336	782	239
SW (251) × DALI (251)	0.5	910	0	447
SW (448) + DALI (448)	0.5	843	0	514

Numbers in parentheses denote the similarity cutoffs used. '×' and '+' refer to the product and sum aggregation rules, respectively. Correct = proteins connected only to members of the same SCOP superfamily, Incorrect = proteins connected to members of other SCOP superfamilies. Weight = the weighting factor assigned to each network.

SCOP superfamily of NAD(P)-binding Rossmann-fold domains (SCOP Superfamily Id c.2.1), which has 149 members in the benchmark dataset, using Smith-Waterman alone as an input to the Multi-Netclust, clusters 30% of the group correctly. With DALI, they are clustered with a number of other superfamilies, whereas the combination of the two (SW * DALI) and (SW + DALI), correctly clustered 94% and 97% of the group, respectively.

3.7. Summary

In this chapter, I have described the Multi-Netclust program that allows one to filter and combine biological network data. The program builds on the user's knowledge on the data, for instance, the user has to define which data can be ignored, and what data-sources are more reliable than the others.

Multi-Netclust has two key components that I consider essential for time and memory efficient functioning: the use of sparse matrices and external memory-based programming of the connected component search.

Sparse matrices are the main data structures in large-scale scientific and engineering applications for representing linear systems of equations. Biological networks are usually huge with the number of non-interacting pairs far exceeding the number of interacting pairs. Rather than allocating space for every element in a matrix, sparse matrix data structures try to minimize the amount of memory used by only allocating memory for the non-zero elements and elements that are used directly by an algorithm.

The external memory-based, connected component search algorithm is a fast as compared to single-linkage based clustering methods and in-memory graph algorithms used for similar purposes within the bioinformatics community. The strength of Multi-Netclust is more obvious when we deal with large data that cannot be handled with other algorithms. For example, a dataset of 2,713,908 nodes and 781,328,458 edges took less than 5 minutes on a standard desktop processor. Of the other algorithms tested (see case studies on the website), only BLASTClust was able to handle a dataset of similar size, however its use is limited to BLAST similarity networks (and at greater expense of CPU time and memory required), whereas Multi-Netclust is generally applicable. To conclude, Multi-Netclust is an efficient preprocessing tool that can aid exploratory analyses of large biological networks using an ordinary computer. Specifically, the potential applications include any task where

network data of heterogeneous sources are to be combined, such as merging microarray and protein-protein interaction data, or combining gene ontology data with various similarity data. Constructing similarity matrices from these data may not be a trivial task, this is a major challenge for future applications of Multi-Netclust.

4. Comparing protein domain architectures assigned to protein sequences

One of the first steps in analysing proteins is to detect their constituent domains. Prediction of known domain types within a protein can be regarded as a special subcase of protein classification. Namely, while general classification methods assign a general, global descriptor (annotation) to an entire protein sequence, domain prediction methods use similar computational techniques to assign local descriptors to a specific segment of a protein sequence.

From our computational point of view, a domain type is a special kind of an identifier assigned to a segment of a protein sequence, such as an item in the feature-table of a sequence database like Swiss-Prot. Such an assignment has a scope, i.e. a starting and an end-point within the sequence, an attribute-identifier: "domain" and an identifier that can take any of the about 6-8 thousand domain names known today. A domain architecture is then the ensemble of such, non-overlapping segment annotations, that are defined. A few comments are in place: i) this is a general definition that covers not only domains and domain-architectures but any kind of sequence annotation. This description falls into the broad class of the Entity-Attribute-Value models and it can accommodate a large variety of annotated structures that are used in bioinformatics (such as protein or DNA structures, genome descriptions, etc., data not shown). ii) Non-overlapping annotations are the norm for the same identifier type. For instance, protein chains of a poly-protein are not supposed to overlap with each other, but a segment annotated as a particular domain can be the part of a protein chain. However, even some of the best curated sequence databases contain a few overlapping domain annotations.

Why is comparison of protein domain annotations different from the protein/domain classification tasks described in the previous chapter? Benchmarking machine learning algorithms on protein/protein domain sequences is a well defined problem wherein both the members of the positive/negative, train/test groups and the boundaries of the domain sequences are exactly defined. Assigning protein domain architectures to a large

dataset, like a proteome, is much less accurately defined. The first, trivial difference is that for domain-annotations, we need to determine the boundaries of the domain, and automated methods of domain annotation usually give slightly different results. Second, domain annotation often relies on carefully trained machine-learning algorithms (HMMs, sequence profiles) that were optimized by human intervention. As a result, assigning protein domain architectures is a complex process, which necessarily involves human judgement, so the process cannot be realistically repeated for statistical purposes, and it is generally believed that there is no "gold standard of truth". Consequently, it is not so straight-forward to answer, whether or not the annotation schemes of Pfam or SMART are "better" or "worse", and this is not our goal. What we can do, on the other hand, is to compare the results with each other, and ask the question whether or not the domain assignments of Pfam are more similar to Swiss-Prot or to SMART. In other words we can define comparison principles and numerical indices of similarity for annotated proteins and use them on a comparative basis.

For the purposes of this comparison, we selected a few domain architecture assignment schemes that represent different approaches:

Swiss-Prot/Uniprot (Boeckmann et al., 2003) is considered the most accurately curated database that contains domain-assignments curated by human experts.

SMART (Letunic et al., 2009) takes its domain-assignments partly from Swiss-Prot, partly from machine-learning algorithms, and adds a variety of further pieces of information, such as exon/intron boundaries checked by human experts. As a result, the SMART domain assignments can be considered a further refined version of the Swiss-Prot annotations. SMART has not only ready annotations but also prediction tools based on HMM which can predict protein domains from sequence.

Pfam (Sonnhammer et al., 1997) is a comprehensive collection of domain sequences that are used to train Hidden Markov Model (HMM) predictors. The Pfam website uses the trained HMM predictors to assign protein domains to sequence queries.

SBASE (Vlahovicek et al., 2005)is a collection of domain sequences collected from various other sequence databases, and the SBASE website uses sequence similarity search to pre-

-dict domains in proteins. The SBASE method of domain prediction does not include a training step and uses the BLAST 2.2 algorithm, i.e. a simple and computationally inexpensive method for domain assignment.

In this section we ask the question, how the protein domain architectures produced by the above approaches compare to each other. The approaches are vastly different in their scope as well as the amount of human intervention required for their maintenance, so the comparison can give us a feeling regarding how far one can go without investing human expertise. The additional question we are asking is how a method based on sequence similarity searching compares to more advanced methods of domain assignment.

4.1. Designing the assessment scheme

In order to define a framework for comparing protein domain architectures, we first have to define the criteria for considering two protein architectures as identical. We use a plausible hierarchy of statements:

a) Presence-absence level: This is the basic and most permissive level of assessing the quality of domain prediction. A protein that has a certain domain type and is predicted to have at least one is a TP (true positive). So, if we predict one single Ig-like domain in Titin (152 Ig-like domains) it counts already as a positive hit.

b) Composition level: This level checks the domain abundancy and assessing domain annotation methods at this level involves a higher level of stringency as opposed to the presence-absence level. A protein that has *n* number of copies of a certain domain type should be predicted as having at least the same number of domains in order to qualify as TP. If less domain copies are predicted, the prediction is considered to be incorrect.

c) Domain-boundary level: A domain assignment is considered to be in agreement between two annotations if the corresponding boundaries are within say within 5 amino acids. This is the most rigid test in this tier.

Of these, level a) is very permissive, all methods are expected to perform almost equally well at this level. On the other hand, it is not easy to find a gold standard for levels b) and

c), because the known collections differ in many details, the number of domains often differs, atypical domains are frequently missing, etc.

As mentioned earlier, due to the absence of a "gold standard" this assessment can be carried out only by comparison between two methods and detect the amount of match/agreement between the two schemes. When we compare two sets of annotations (say domains predicted by Pfam and domains annotated in the SMART database), we compare a large number of domain annotation pairs which leads to a large set of numerical values. In order to understand the results we need to summarize them and we can do in two ways:

i) Summary by domain type: We summarize identical/different predictions for each type of domain in the analysis. In this manner we will get a cumulative number that expresses, for instance that the EGF domain type is identically predicted in 60% of the cases, noting however that identity can be defined in terms of presence/absence, domain numbers or boundaries, as described above. Averaging the domain-type statistics will then provide a cumulative average of the database, in terms of domain types

ii) We can summarize identities within a protein: If all constituent domains of a protein are identically predicted by two annotation schemes, that is a true positive (TP). Again, identities can be considered at any of the 3 levels (a,b,c). Averaging the protein statistics will then provide a cumulative average of the database, in terms of protein architectures.

These definitions can be best understood by imagining protein domain annotations as a table in which proteins are the columns and domain types are the rows as shown in **Table 4.1**. In this hypothetical example the dataset contains four proteins (namely, C1S_HUMAN, BMP1_HUMAN, C1RL_HUMAN and OVCH1_HUMAN) that contain a few domain types only. "+" means that a given domain type is identically predicted by two annotation schemes being compared, "-" means that the annotations are different.

Table 4.1: Assessment of a protein domain prediction method by "Domain Type" and by "Protein Architecture".

Protein Architecture / Domain Type	C1S_HUMAN	C1RL_HUMAN	BMP1_HUMAN	OVCH1_HUMAN	% identities by domain type
CUB (IPR000859)	+	+	–	+	75
SUSHI/CCP (IPR000436)	–				0
EGF_CA (IPR001881)	+		–		50
Trypsin-like serine protease (IPR001254)	+	+		–	66
					Avg. identities for entire dataset by Domain Type = 47.75
% identities at protein architecture level	0	100	0	0	Avg. identities for entire dataset (Protein Architecture)= 25

"+" means that two annotations agree either a) at the presence/absence, or b) at domain abundance or c) domain boundary level. These identities can then be summed horizontally, to give average identities by domain type, whose values are written in the rightmost column. A protein's entire architecture (calculated vertically) is considered to be correct only if all its constituent domains are predicted by both the annotation schemes a) at the presence/absence, or b) at domain abundance or c) domain boundary level., so it can either be 100or 0., and these scores are written in the bottom row of the table. Finally, we can calculate dataset-averages by averaging the values in the bottom row and in the rightmost column, as shown by the arrows.

Comparing the two annotation schemes at the level of "Domain Type" would involve checking for the presence of elements in the column label for each element in the row label. This means checking the list of all those proteins that contain the particular domain type, for example in Table 3.1, three of the four proteins in the Swiss-Prot have an annotations for the CUB domain whereas the Trypsin–like serine protease is annotated by only two proteins (C1S_HUMAN and C1RL_HUMAN).

4.2. Designing and constructing the core dataset

To assess the correctness of predicted domain architectures, we manually curated a dataset of multi-domain proteins belonging to the human proteome. Prediction of multi-domain proteins poses to be a major challenge in protein classification mainly due to its higher-order organization. They represent a substantial fraction of the proteome: about 27% of proteins in bacteria and 39% of proteins in metazoans are multi-domain proteins (Tordai et al., 2005). Moreover, these proteins are involved in a plethora of functions (Ben-Shlomo et al., 2003; Lander et al., 2001; Miyata and Suga, 2001; Patthy, 2003). Recently, the establishment of several comprehensive databases of protein domains has been undertaken,

including Pfam (Sonnhammer et al., 1997), SUPERFAMILY(Wilson et al., 2009), SMART (Letunic et al., 2009), SBASE (Vlahovicek et al., 2005), InterPro (Hunter et al., 2009), and CDD (Marchler-Bauer et al., 2009). Using the testing procedure outlined above, we compare established collections of domain architectures (Pfam, SMART, Swiss-Prot) as well as compare them with results of simple similarity search (BLAST), using a subset of multi-domain proteins from the human proteome as an example.

The Multi-domain Protein dataset

The dataset used in the present work was constructed using SMART (Letunic et al., 2009), Pfam-A (Sonnhammer et al., 1997) and Swiss-Prot (Boeckmann et al., 2003) database.

Complete human protein sequences were extracted from the Swiss-Prot Release 57.9, yielding 20,272 protein sequences. SMART and Pfam domain architecture for these proteins were obtained by writing a client-server socket program in Perl. These three datasets (SMART, Pfam-A and Swiss-Prot) were then scanned for proteins containing at least two domains. Only those proteins were considered from the resulting data that were common to all the three sets, thus resulting in a common subset of 4011 proteins.

Mapping Protein domains and families

Evaluation and comparison of domain predictors becomes a complicated task due to the existence of several domain datasets/databases that sometimes conflict with each other (Liu and Rost, 2004). To achieve correspondence between the domain types defined by various annotation paradigms we used the InterPro (Hunter et al., 2009) as the basis for data integration. The InterPro database is an integrated resource for protein families, domains and functional sites diverse source.

Each domain type occurring in the three datasets was mapped to the corresponding InterPro Domain Signature. Mapping SMART and Pfam domain was a straightforward task as InterPro provides a direct mapping between the SMART/Pfam domains that are integrated into the InterPro and the corresponding InterPro domain signatures.

As there is no such mapping available between the domains in Swiss-Prot and the InterPro domain signatures, we approached this problem using an indirect method. We made use of the index of protein domains and families downloaded from

http://www.uniprot.org/docs/similar. This file enlists domains, repeats and zinc fingers along with the protein entries that contain the specified annotation. The domains, repeats and zinc fingers listed in this file are mapped to relevant PROSITE accession numbers. Since InterPro provides a mapping of the PROSITE profiles and patterns integrated into the InterPro, Swiss-Prot domains could be mapped to corresponding InterPro Ids.

To remove any discrepancy that might occur in the statistics due the various relationships existing between the InterPro entries, all the entries belonging to the PARENT/CHILD relationship were mapped to the PARENT.

Proteins sequences were broken down into domains based on the annotated domain boundaries and stored in a MySQL table. The dataset thus consisted of 4011 proteins with a total of 75105 annotations. A set of 278 domain types common to all three domain architecture collections was used as the representative dataset for domain types (***Appendix-B*, Table B1**).

In addition to the domain annotations that can be downloaded from various databases, I also retrieved predicted domain architectures using the default prediction parameters of SMART, Pfam and SBASE servers, using client-server socket programs written in Perl . I noted however a few differences since the server programs sometimes predicted domains in regions where the curated annotations contain predicted signal peptides (Nielsen et al., 1997), transmembrane regions (von Heijne, 1992) and coiled/coil regions (Lupas et al., 1991). So in order to keep predictions at an equal footing, I used the signal sequences, transmembrane and coiled-coil regions as calculated in the SMART database, and predicted the domains only in the remaining regions of the protein.

4.3. Comparing annotations

We calculated the agreement between two annotation schemes as briefly outlined about at Table 3.1. A more detailed description is found in ***Appendix-B.***

An important feature of our calculations is that we consider only those identities that occur in either or both the annotations compared. So if neither of the annotations contains a *CUB* domain, this fact is not counted as an identity. An interesting consequence of this fact is that our % identity is becomes analogous with the popular F-measure.

In practical terms, one can calculate pairwise comparisons between annotations and present the data in the form a similarity matrix whose rows and columns are the annotations to be compared, and the contents of the cells are the values of pairwise comparisons. One can calculate such matrices for individual domain types, for the averages of domain types, for individual protein architectures, and for the averages of all architectures.

For example, the prediction of the domain type, EGF_CA (EGF-like calcium-binding) between Swiss-Prot annotations (SW), SMART annotations (SMA), SMART predictions (SMP), Pfam prediction (PFP) and SBASE prediction (SBP) is shown in **Table 4.2.**

Table 4.2. *Comparison of various annotation and prediction schemes in terms of EGF_CA domain type assignment (presence absence level)*

	SW	SMA	SMP	PFP	SBP
SW	100				
SMA	93	100			
SMP	93	100	100		
PFP	80	77	77	100	
SBP	75.53	75.78	75.77	90.03	100

The values represent % identities between the various annotation schemes. Comparisons made at the presence/absence level, so identity is assigned if two proteins compared both contain an EGF domain. SW=Swiss-Prot annotation, SMA=SMART prediction, SMP=SMART Prediction, PFP=Pfam prediction, SBP=SBASE prediction

A comparison of the various annotations in terms of all domain types is shown in **Table 4.3.** The values in this similarity matrix are the average percent values calculated for all

domain types, i.e. this table is the average of 278 tables similar to **Table 4.2.** Finally, a cumulative comparison of annotations in terms of proteins is shown in **Table 4.4.**

Table 4.3 *Comparison of various annotation and prediction schemes in terms of all domain types (average identity values)*

		SW	SMA	SMP	PFP	SBP
Presence-Absence	**SW**	100				
	SMA	95.66	100			
	SMP	95.39	99.91	100		
	PFP	91.48	93.84	93.56	100	
	SBP	92.38	91.33	91.78	92.87	100
Composition	**SW**	100				
	SMA	94.82	100			
	SMP	94.46	99.83	100		
	PFP	86.26	90.12	94.65	100	
	SBP	91.5	90.78	91.38	92.09	100
Boundary	**SW**	100				
	SMA	50.26	100			
	SMP	50.66	100	100		
	PFP	43.04	48.47	50.51	100	
	SBP	50.10	61.49	64.07	61.79	100

The values represent % identities between the various annotation schemes. The values represent the average calculated for all the 278 domain types. SW=Swiss-Prot annotation, SMA=SMART prediction, SMP=SMART Prediction, PFP=Pfam prediction, SBP=SBASE prediction

Table 4.4 *Comparison of various annotation and prediction schemes in terms of protein architecture (average identity values)*

		SW	SMA	SMP	PFP	SBP
Presence-Absence	**SW**	100				
	SMA	64.5	100			
	SMP	64.7	95	100		
	PFP	58.5	60	59.7	100	
	SBP	59.43	64.59	64.65	62.58	100
Composition	**SW**	100				
	SMA	40.71	100			
	SMP	41.14	93.74	100		
	PFP	18.57	24.51	23.88	100	
	SBP	34.87	40.39	40.59	27.89	100
Boundary	**SW**	100				
	SMA	11.12	100			
	SMP	11.42	90.97	100		
	PFP	3.42	6.93	6.86	100	
	SBP	9.53	17.87	18.22	9.32	100

The values represent % identities between the various annotation schemes. The values represent the average calculated for all the 4011 proteins in the dataset. SW=Swiss-Prot annotation, SMA=SMART prediction, SMP=SMART Prediction, PFP=Pfam prediction, SBP=SBASE prediction.

From the above tables it is apparent that that the degree of similarity between various annotation schemes vary, with some annotations being more closer to each other than others. For example, the SMART annotations are near to those of Swiss-Prot. Interestingly, the BLAST-based predictions of SBASE are not farther from Swiss-Prot than Pfam predictions which are based on a much more sophisticated algorithm. The same trend is followed at all the levels of comparison, though the percentage of identities are in fact much lower at the composition and domain boundary level. These data thus confirm that the various annotation schemes are in good agreement, even though there can be discrepancies in smaller details, especially at the domain boundary level. For example, even SMART predictions differ from SMART annotations to some extent.

Finally, I present an example demonstrating how this statistics can be used to tune a predictor on an entire database. In the SBASE prediction scheme, there are a few tuneable thresholds that are used to reject weakly predicted domain assignments. These thresholds

are either database-wide (i.e. applied to all domain types), or group-specific that are different for each domain-type. **Table 4.5** shows how the inclusion of several weak thresholds increases the efficiency of the prediction what is manifested by the fact that the predictions come closer to the curated SMART annotations.

Table 4.5 *Tuning prediction parameters based on annotation-comparison*

	SMART annotations vs SBASE/BLAST predictions	% identities by protein architecture
1	No thresholds	15.47
2	Sequence coverage threshold (group average)	28.29
3	Score threshold (group average)	51.55
4	Score and sequence coverage (2 and 3)	56.59
5	Score, sequence, coverage and e-value	64.59

4.4. Summary

In this chapter, I presented a general method to compare sequence annotations. Briefly, I consider sequence annotations as a generalized form of a Swiss-Prot feature table, in which there can be an arbitrary number of assignment types (e.g. domains, exons, repeats, protein chains, etc.) and assignments can be overlapping. A specific example, protein architecture, contains the same assignment types ("domain") that are defined in a non-overlapping fashion. I outlined three levels to compare domain-assignments (presence/absence level, abundance level and domain/boundary level), and introduced two plausible ways to sum up the comparisons according to either domain-types or proteins, to obtain single qualitative indices for a complete dataset. I constructed a model dataset of human multi-domain proteins that contained 278 domain types that are present in Swiss-Prot, Pfam, SMART, SBASE, i.e. databases that employ very different principles of assignment, curation and/or prediction and compared the annotated architectures and the predictions between these data-sources One of the biggest problem of this comparison was to find an unequivocal mapping of domain names between data-sources. When this problem was resolved I found that there are minor but consistent differences even between curated annotations and

automated predictions coming from the same database. On the other hand there are tendencies that are apparent at all levels of comparison, for instance SMART annotations and predictions are closer to Swiss-Prot than SBASE and Pfam, but similarity based predictions are as close to Swiss-Prot even closer then HMM/based predictions. I also presented an example of how to tune prediction parameters based on the annotation/comparison principle. I wish to mention that the comparison method outlined here is general so it can be applied without modifications to entire genomes or other, sequentially aligned data.

And finally, even though a simple pairwise comparison of two annotations may not allow one to tell which one is better, a few comments are in place regarding the general tendencies apparent from the various comparisons shown in **Table 4.3** and **Table 4.4**. First, it is apparent that domain boundaries are not easily predicted, and even the high-quality, HMM-based predictors of Pfam do not do very well in terms of boundaries even though they tend to find the domains quite well. Second, the HMM-based predictors of SMART, that are trained on high quality, manually curated set of domain sequences, perform better and give predictions close to the manual annotations (in Swiss-Prot and in SMART). In other words, manual curation is still a prerequisite of accurate domain predictions. On the other hand, a simple similarity-based prediction could sometimes outperform Pfam HMM-s, which is somewhat surprising. Especially, since in these experiments I used the basic BLAST tool for similarity search(Altschul et al., 1990), not PSI-BLAST or any of the more refined similarity search tools. In other words, similarity-based annotations can be improved well beyond the performance shown in these comparisons.

5. Structural Analysis and Classification of an atypical EGF: A Case Study

5.1. Background

The Notch signalling pathway is an evolutionarily conserved, intercellular interaction mechanism, essential for proper embryonic development in all metazoan organisms. Originally identified in *Drosophila*, where the mutant allele gave rise to a notched wing, proteins of the Notch pathway have been studied extensively in flies, worms, and mammals. Thus unravelling a broad spectrum of roles of the Notch signalling in cell fate specification, patterning and morphogenesis through effects on differentiation, proliferation, neurogenesis, miogenesis, hematopoiesis, survival and apoptosis (Bray, 2006; Fiuza and Arias, 2007; Lewis, 1998).

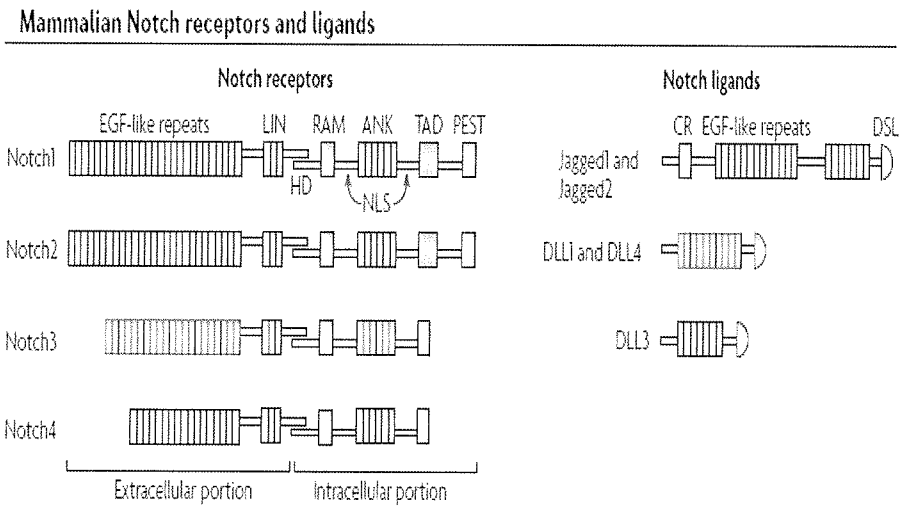


Figure 5.1: Notch receptors.
There are four mammalian Notch receptors (Notch1–Notch4 and five Notch ligands, Jagged-1, Jagged-2, Delta-like 1 (DLL1), DLL3 and DLL4 (Osborne and Minter, 2007).

The cell contact based pathway in Notch signaling occurs as a result of interaction between a ligand on a signal sending cell and a receptor on a signal receiving cell. In mammals, four different Notch receptors (NTC1, NTC2, NTC3, NTC4) have been identified, which bind to five canonical Notch ligands belonging to two distinct families: homologues of *Drosophila* delta protein (DLL1, DLL3, DLL4) and homologues of *Drosophila* Serrate, Jagged-1 and -2 (JAG1, JAG2) (Beatus and Lendahl, 1998) (**Figure 5.1**).

Notch Receptors

Notch receptors are non-covalently assembled heterodimeric membrane-spanning glycoproteins involved in transducing specific extracellular signals to the nucleus upon ligand binding. Notch receptors are synthesized, as large pre-proteins comprised of extracellular and intracellular domains (**Figure 5.2**).

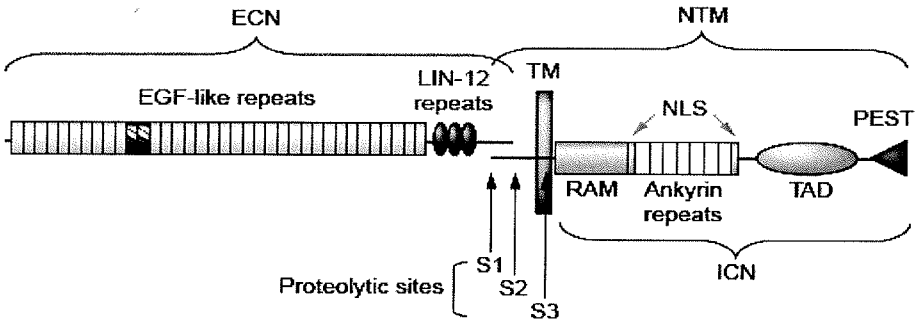


Figure 5.2: Domain organization of Notch receptors.
Human Notch1 (NTC1) is shown as an example. Proteolytic cleavage by furin at site S1 produces two subunits, ECN and NTM, which remain non-covalently associated at the cell surface. EGF-like modules 11 and 12, implicated in ligand binding in Drosophila Notch, are shaded. S2 and S3 identify the sites of proteolytic cleavage induced upon activation by the ligand. ICN, intracellular domain of Notch; NLS, nuclear localization signal; PEST, proline, glutamate, serine, threonine rich sequence; TAD, transactivation domain; TM, transmembrane.

The polypeptide encoded by Notch gene is proteolytically cleaved in the Golgi during the transport to the cell surface, to give an extracellular (ECN) and a transmembrane subunit (NTM). The ECN part of this receptor contains an array of ~29-36 EGF tandem repeats, followed by three LIN-12 repeats that maintain the heterodimeric structure of the functional receptor by disulphide bridges (Sanchez-Irizarry et al., 2004). Proper folding of EGF like repeats have been shown to be Ca^{2+} dependant (Rao et al., 1995) and it has also been shown that EGF-like repeats 11 and 12 are necessary for ligand binding (Rebay et al., 1991). The intracellular region of the NTM contains several functional domains, which includes a RAM domain, followed by seven ankyrin repeats, a TAD domain, and a PEST region. The ankyrin repeat region is the most highly conserved portion of the Notch (Stifani et al., 1992) and functionally significant mutations that map to the ANK repeats show that it is essential for signalling (Kopan et al., 1994; Lieber et al., 1993; Rebay et al., 1993; Roehl et al., 1996).

Jagged and Delta ligands

Ligands for Notch are members of the DSL (Delta, Serrate, Lag-2) family of transmembrane proteins. All the ligands of the DSL (Delta/Serrate/Lag-2) family share the same architecture (Letunic, et al., 2004) (**Figure 5.3**). They are type I membrane spanning proteins composed of a N-terminal, cysteine rich region that includes a DSL domain, a variable number of EGF-like repeats, a transmembrane segment, and a relatively short (~100-150 amino acids) cytoplasmic tail. Ligands of the Jagged group (JAG1 and JAG2) have also a juxtmembrane additional region that is not present in the Delta group ligands. In mammals, the ligands are expressed in all embryonic tissues.

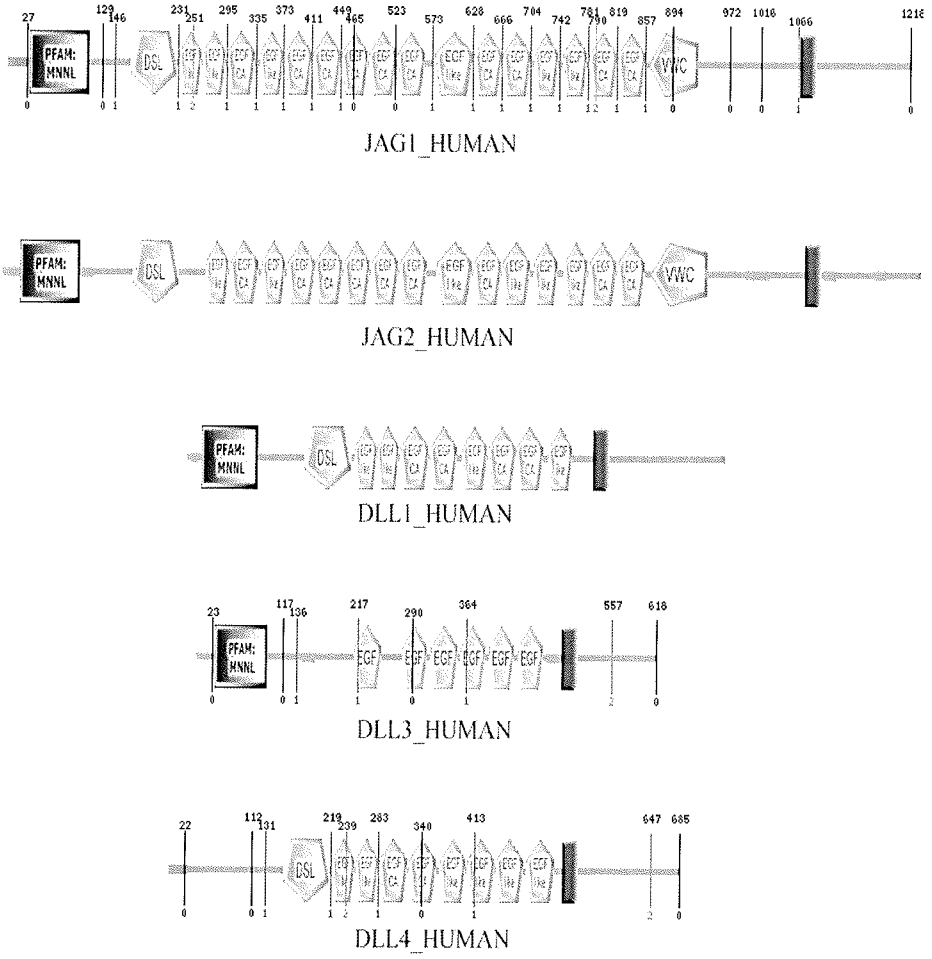


Figure 5.3: Domain architecture of human Notch ligands as depicted by SMART. **MNNL,** N-terminal region of Notch ligands (Pfam); **DSL,** Delta/Serrate/lag-2 domain; **EGF-like** - epidermal growth factor (EGF) domain, unclassified subfamily; **EGF_Ca** - Calcium-binding EGF-like domain; **VWC** - von Willebrand factor (VWF) type C domain; the transmembrane region is shown as a blue rectangle; low-complexity regions in magenta.

Jagged-1

Jagged-1, one of the five Notch ligands identified in man, is a single pass type I membrane protein with a large extracellular region made of a poorly characterized N-terminal region, a DSL (Delta/Serrate/Lag-2) domain, a series of 16 epidermal growth factor (EGF) tandem repeats, and a cysteine-rich juxtamembrane region (**Figure 5.4**). The DSL domain, together with the first two atypical EGF repeats constitutes Jagged-1 receptor binding region (Cordle et al., 2008; Shimizu et al., 1999). Binding of Jagged-1 to Notch receptors triggers a cascade of proteolytic cleavages (Weinmaster, 2000) that eventually leads to the release of the intracellular part of the receptor from the membrane, its translocation to the nucleus, and the activation of transcription factors (Allman et al., 2002; Iso et al., 2003).

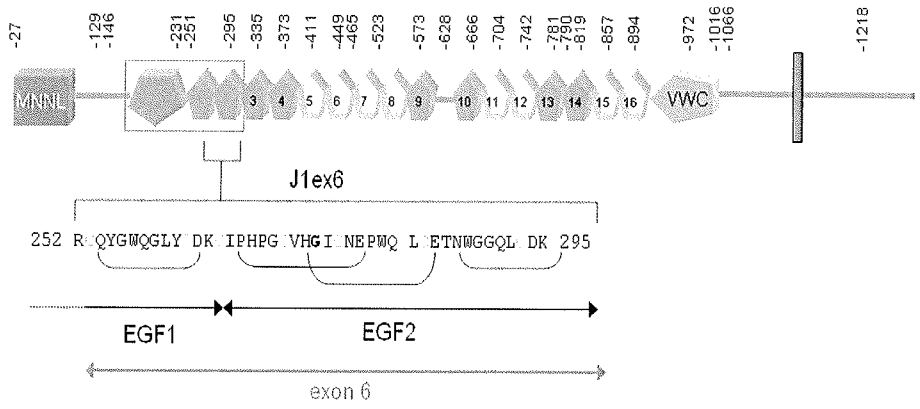


Figure 5.4: Domain architecture of human Jagged-1.
MNNL, N-terminal domain of Notch ligands; DSL, Delta/Serrate/Lag-2 domain; EGF domains (green) are numbered progressively; potential calcium binding EGF domains are in lighter green; VWC, von Willebrand factor type C domain; the transmembrane segment is shown as a blue bar; the receptor binding region is marked in red. Amino acid number of exon boundaries are shown on top. The amino acid sequence of J1ex6 and the disulfide bond connectivities are also shown.

Figure 5.4 shows the domain architecture of the protein encoded by the human jagged-1 gene. It was shown that exon 6 of the *JAG1* gene encodes an autonomously folding with a disulfide bond topology typical of EGF repeats (Guarnaccia et al., 2004) .

Early on in 1978 it was proposed that exons encode "folded protein units", emphasizing the role of a correct folding process to produce functional proteins or domains (Blake, 1978). Recent advances in genome sequencing, domain classification, and 3D structure determination confirmed this hypothesis: a strong correlation between exon boundaries and

predicted domain boundaries has been found in nine eukaryotic genomes, the correlation becoming stronger with the increasing genome complexity (Liu and Rost, 2004). Such a high correlation lead to the suggestion that in certain cases exon boundaries can be used to predict domain limits more accurately (Liu et al., 2005). In particular, a survey of domain repeats in seven metazoan species showed that there is a very good correspondence between exons and EGF repeats (0.93 exon/repeat on the average) (Bjorklund et al., 2006). This does not hold true for exon 6 of the *JAG1* gene as it encodes not only EGF2 but also a part of EGF1.

Recently the solution structure of the peptide encoded by exon 6 (J1ex6) of the *JAG1* gene was determined by ^{1}H-NMR spectroscopy by the Protein Structure lab at ICGEB. This provided me with the opportunity to apply various bioinformatics based methods for the classification and structural analysis of this newly determined structure. Moreover, EGF repeats are widespread in extracellular proteins and hundreds of missense mutations have been identified and associated with several genetic diseases. The fact that structural grounds of these disorders been investigated only in a few cases, we carried out a systematic and comprehensive analysis of mutations found in epidermal growth factor repeats.

5.2. Methods

Sequence and Structure based classification of J1ex6

Epidermal growth factor, EGF-like domains are extracellular protein modules approximately 30-40 amino acid residues in length and stabilized by six Cysteine residues that form three disulphide bonds (Campbell and Bork, 1993).

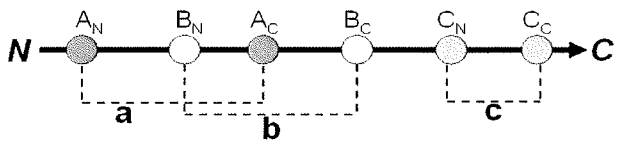


Figure 5.5: Disulfide signature of the EGF motif.
Disulfide signature of the EGF motif showing the cross-linked arrangement of half-cystines in the primary structure and their connectivity

The Cysteine connectivities within EGF are the first to the third, the second to the fourth, and the fifth to the sixth or (1-3, 2-4, 5-6: Savage et al., 1973) also defined by the $A_NB_NA_CB_CC_NC_C$ annotation (**Figure 5.5**).

Depending on the location of these half-Cystines in the structure, it was proposed that EGF domains can be divided in two structural groups, namely the human EGFs (hEGF) and C1r-like EGFs (cEGF)(Wouters et al., 2005) as described in **Table 5.1**.

Table 5.1: *Summary of properties of three-disulphide EGF types. [Source: Wouters et al., 2005]*

hEGF	cEGF
• 8-9 amino acids in c_N-c_C loop	• 10+ amino acids in c_N-c_C loop
• 4+ residues in b_N-a_C loop	• Subtype 1 has 4+ residues in b_N-a_C loop
	• Subtype 2 has 3 residues in b_N-a_C loop
• 6 residues between tandem domains of ssame type	• 5 residues between tandem domains

With the aim of classifying the EGF2 of the *J1ex6* either as hEGF or as cEGF, we tried a two-way approach, using both the sequence and structure information at hand. **Figure 5.6** depicts the approach followed.

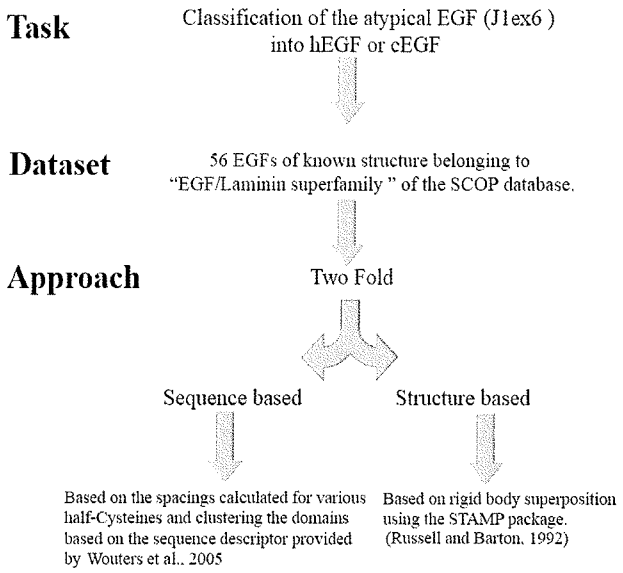


Figure 5.6: Classification of J1ex6.
Depiction of the methodology used for the classification of atypical EGF of J1ex6 using sequence and structure information.

56 domain sequences belonging to the EGF/Laminin superfamily was taken from the SCOP95 dataset of the Protein Benchmark Collection(Sonego et al., 2007). This formed the dataset used for the sequence-based classification of the atypical EGF. A complete list of the domains used in this study can be found in ***Appendix C*, Table C1.**

A Perl script was used to categorize the 56 domains into cEGF or hEGF based on the sequence descriptor (Wouters et al., 2005). Various frequency distributions of the Cysteine loop lengths were plotted (like A_N - A_C, B_N - B_C and so on).

For the structure based classification, the same superfamily from the SCOP95 dataset was used but only those domain structures were considered for this study which were also present in the EGF-type module of the PALI database (Gowri et al., 2003). PALI (Phylogeny and ALIgnment of homologous protein structures (Release 2.6)), comprises of protein structural families in which every member is structurally aligned pairwise with every other member in that family and multiple structural alignment of all members in the family is also available. The alignments were made using STAMP (Russell and Barton, 1992) which encodes a rigid body superposition. This dataset comprised of 36 domain structures.

Similar to PALI, the STAMP (STructural Alignment of Multiple Proteins) program (Russell and Barton, 1992) was chosen for the construction of structural superpositions. STAMP is a package for the alignment of protein sequence based on 3D structure. It provides not only multiple alignments and the corresponding "best-fit" superimpositions based on structural equivalences, but also a systematic and reproducible method for assessing the quality of such alignments. STAMP makes extensive use of the Smith-Waterman algorithm(Smith and Waterman, 1981). This widely used algorithm allows fast determination of the best pair through a matrix containing a numerical measure of the pairwise similarity of each position in one sequence to each position in another sequence. At the heart of the method is the Argos and Rossmann (Rossmann and Argos, 1976) equation for expressing the probability of equivalence of residue structural equivalence.

STAMP needs an initial alignment to start from. The SCAN method of STAMP suite was used to obtain an initial set of superimpositions followed by the final rigid body superposition by STAMP. Similar to PALI, the structure based phylogenetic tree was constructed using the KITSCH program from the PHYLIP package of programs (version 3.5) (Felsenstein, 1995). The input to this program was a Structural Distance Metric (SDM) (Johnson et al., 1990) calculated for every pairwise alignment. SDM combines RMSD and number of equivalent Cα atoms between two proteins and is calculated as,

$$SDM = -100 * \log\left(w1 * PFTE + w2 * SRMS\right),$$

where ,

$$w1 = 1 - PFTE + 1 - SRMS \,/\, 2,$$

$$w2 = \left(PFTE + SRMS\right) / 2,$$

$$PFTE = \frac{\text{No. of topologic ally equivalent residues}}{\text{Length of the smallest protein}},$$

$$SRMS = 1 - \left(RMSD / 3.0\right)$$

Analysis of the Exon/Intron organization of the J1ex6 and the extent of its conservation

Orthologues of Human JAG1

Amino acid sequences of orthologues of human *JAG1* gene were retrieved from ENSEMBL Release 50 (Hubbard et al., 2009). This comprised of sequences from 26 different species ranging from fishes to primates, namely, primates (5) and non-primate mammals (15), birds (1), amphibians (1), and fishes (4).

The protein sequences were then broken down into peptides encoded by exons using the colour coding scheme used by ENSEMBL (Hubbard et al., 2009) to depict peptides encoded by different exons (**Figure 5.7**).

Figure 5.7: Peptides encoded by different exons as shown in ENSEMBL.
The ENSEMBL database demarcates peptides encoded by exons using a colour coded scheme. So, the first exon corresponds to the black amino acids, the second exon corresponds to the blue sequence, the third exon is in black and so on.

Identification and Retrieval of Orthologs of Notch Ligands

Swiss-Prot (Boeckmann et al., 2003) was searched for all proteins containing EGF repeats, entries for which the exon/intron organization is annotated in the ENSEMBL (Hubbard et al., 2009) were collected. The amino acid sequences for these proteins were then broken down into segments corresponding to exons, and a BLAST search was performed with the sequence encoded by Jagged-1 exon 6. Apart from the usual orthologs of Jagged-1, namely, *JAG2, DLL1, DLL4* we identified the non-canonical Notch ligands *DLK1* and *DLK2* sharing high sequence similarity with *JAG1*.

Following this, all amino acid sequences annotated in ENSEMBLE as orthologues to *JAG1, JAG2, DLL1, DLL4, DLK1,* and *DLK2* were collected, and broken down into segments corresponding to exons. A detailed list of these genes along with the

corresponding Ensemble Id and the domain boundaries can be found in ***Appendix C,*** **Table C2.**

Analysis of disease associated mutations in Epidermal Growth factors

Disease associated mutation in EGF domains

Egf-like domains with disease associated mutation were obtained after searching the SwissProt database (Boeckmann et al., 2003) Release 55.6 for proteins using the keywords EGF-LIKE, DISEASE MUTATION and HUMAN. This resulted in a total of 325 disease-associated mutations from 105 EGF domains in 21 different proteins (***Appendix C,*** **TABLE C4**)

Polymorphism in EGF domains

For the neutral mutations, a search using keywords EGF-LIKE, VARIANT and HUMAN was used to retrieve an initial dataset which was then filtered using a Perl script to obtain the neutral mutations (polymorphisms). This dataset consisted of a total of 67 polymorphisms from 64 EGF domains in 38 proteins (***Appendix C*****,Tables C5**).

For both the above cases, only EGF domains with three-disulfide bonds were considered, thus discarding the laminin and integrin-like EGF domains, which have one additional disulfide bond. An in-house Perl script was used to extract the EGF-Like domains according to the domain boundary information provided in Swiss-Prot.

Reference Dataset

As a reference dataset, we used a collection of all disease-associated mutations described in the MIM database (Hamosh et al., 2005) and annotated in Swiss-Prot. This dataset comprises a total of 4236 mutations from 436 genes, regardless of protein function, cellular localization and domain type (Vitkup et al., 2003).

Grantham Binning of Mutations

Disease-associated and neutral mutations in EGF domains were analyzed in terms of the Grantham score (Grantham, 1974) associated with every mutation type. The Grantham

score is a composite measure of the chemical distance between two amino acid types based on assessment of chemical dissimilarity between residues. It takes into account the molecular volume, polarity and side-chain composition of amino acid pairs. Grantham scores are in the range 5–215, with a higher number reflecting less conservative changes. Scores in the Grantham matrix were divided in ascending order into 9 bins each of size 25. Diseased and neutral mutations were binned according to their frequency/occurrence into these Grantham bins.

Positional Analysis of Mutations

Disease-associated and neutral mutations were mapped onto the sequence of EGF domains using a Perl script. The sequence was then broken down into seven windows, w1to w7, based on half cystines (with w1 comprising the N-terminal residues, w2 to w6 comprising the residues delimited by disulfide bonds half-cystines, and w7 the C-terminal linker residues). Mutations were then counted in each of these seven windows. This mutation frequency was normalized by the average number of residues in each window and was then plotted.

Mutation Impact Plot

To compare the frequency of each disease-associated mutation type observed in EGF domains with that in the reference dataset, all disease mutations of the type $AA_i \rightarrow X$, where X is any amino acid, were collected, summed up for each amino acid type AA_i, and divided by the number of occurrences of AA_i, to obtain a normalized mutation frequency F_i for the EGF domain dataset and f_i for the reference dataset. The ratio F_i / f_i between these two frequencies was plotted for each amino acid type. To account for the very different size of the two datasets, the number of observed mutations in the reference dataset was first downscaled to the size of the EGF dataset.

5.3. Results

Classification of EGF repeats based on disulphide bond topology

Sequences belonging to the EGF/Laminin superfamily were categorised into cEGF and hEGF using the $A_NB_NA_CB_CC_NC_C$ annotation to describe the disulfide bond topology, where A_NA_C, B_NB_C, C_NC_C are the three disulfides. Based on the various sequence descriptors (Wouters et al., 2005), 29 out of the 56 members belonged to the hEGF whereas 27 belonged to cEGF. These two groups display different lengths of the C_N-C_C loop, of the B_N-A_C loop, and of the linker connecting two EGFs of the same type.

A comparison between different spacings in EGF2 of J1ex6 and in a set of 56 EGFs of known structure (**Figure 5.8**) shows that J1ex6 can be clustered together with the hEGFs for certain characteristics, such as the length of the C_N-C_C loop (8 residues), while for others it clusters neither with cEGFs nor with hEGFs. Notably, the B_N-B_C loop (10 residues) is shorter than in cEGFs (most frequently 12–13 residues) and in hEGFs (14 residues or more), as well as the total spacing between the first and the last halfcystine (A_N-C_C loop, 27 residues vs. 30 or more in other EGFs) and the linker between EGF1 and EGF2 (2 residues, vs. 5 or 6 in cEGFs and hEGFs, respectively). Overall, this makes the EGF2 of J1ex6 rather more constrained than both cEGFs and hEGFs.

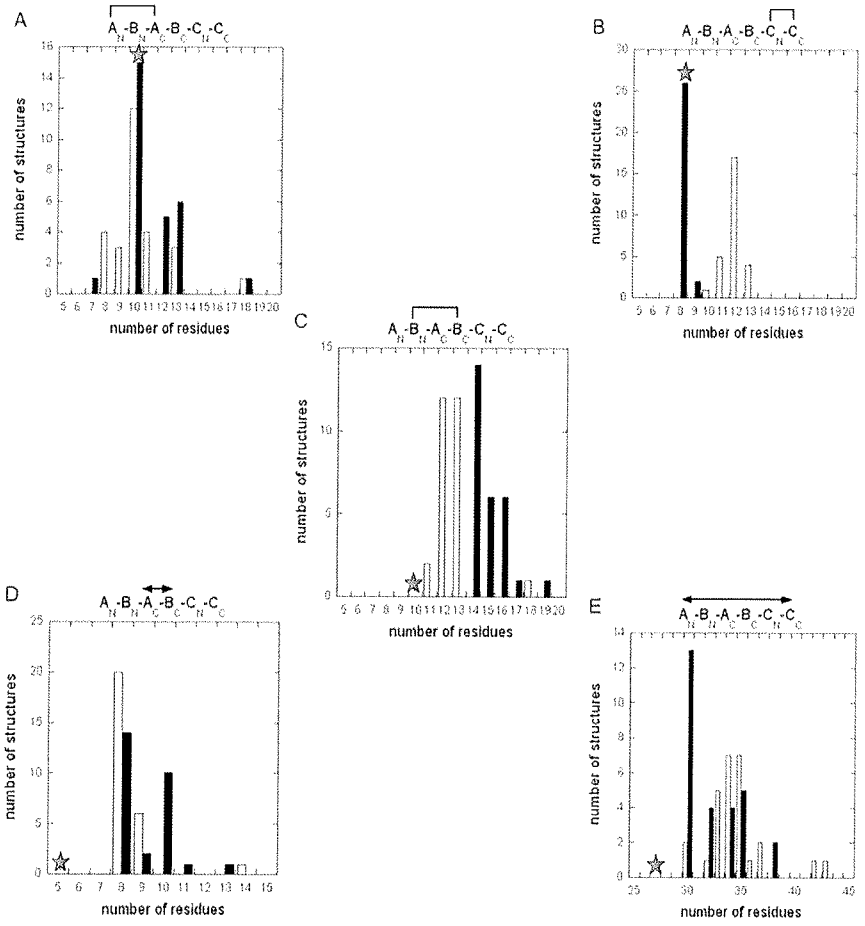


Figure 5.8: Classification of EGF repeats based on sequence descriptors.
Different spacings calculated for a dataset of 56 structures classified as cEGF (empty bars) or hEGF (filled bars); spacings in EGF2 are marked by an asterisk.

Similar ambiguity in placing the atypical EGF into either cEGF or hEGF was observed in the case of structure-based classification. **Figure 5.9** shows an unrooted dendrogram, which was constructed, based on the structure dissimilarity matrix obtained from the structure-based alignment of the domains belonging to the members of the EGF/Laminin superfamily and the EGF2 of J1ex6. This structure based tree shows similar ambiguity in grouping the atypical EGF of the J1ex6 as it neither clusters with the cEGF (depicted by pink colour) nor with the hEGF (blue).

Moreover, an exhaustive search of structural databases with this EGF of J1ex6 structure did not produce any hit with a significant score.

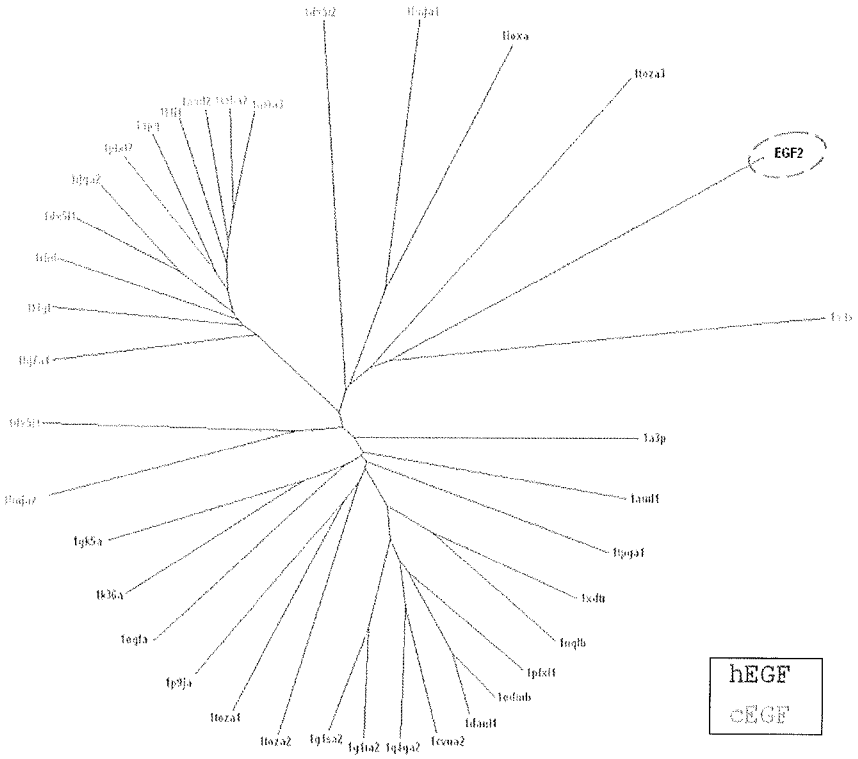


Figure 5.9: Classification of EGF repeats based on structure information.
Phylogenetic tree showing the clustering of the members of the EGF/Laminin superfamily. The members belonging to the hEGF have been coloured in blue and those belonging to the cEGF have been coloured pink. The structure based sequence alignment produced by the STAMP package can be seen in ***Appendix C,* Figure C2.**

Recently, the crystal structure of the region encompassing the DSL and the first three EGF repeats of Jagged-1 has been reported (Cordle et al., 2008) (PDB: 2VJ2). A comparison between the solution structure of J1ex6 and the structure of the same region in the X-ray structure shows a good agreement in the tracing of the backbone. As can be seen in **Figure 5.10**, the crystal structure of the DSL/EGF1-3 modules (Cordle et al., 2008), shows the presence of a kink between EGF1 and EGF2 in an otherwise linear, rod-like structure. Because this construct crystallized as a dimer with several interchain contacts, it can be questioned if packing forces are responsible for the bending of the chain. On the other hand, the good agreement between the crystal structure and the solution structure, in particular in the N-terminal overhang, despite the reduced structural context, suggests that the kink is actually a structural feature that might have some functional relevance.

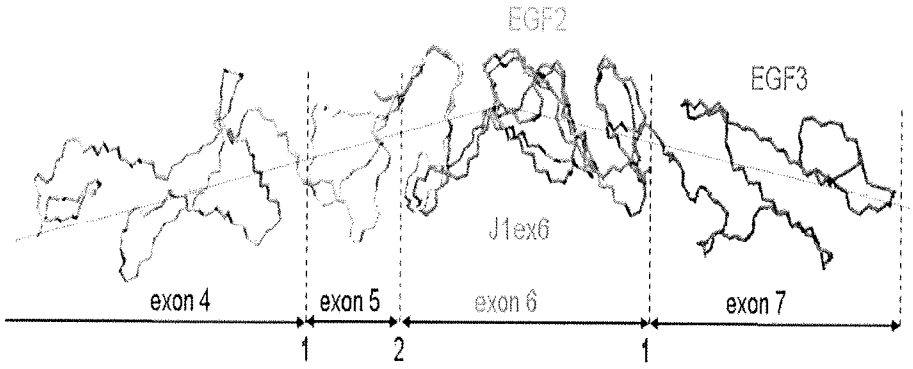


Figure 5.10: Structure of the receptor binding region.
The X-ray model the DSL domain and the first three EGFs of Jagged-1 (PDB: 2VJ2) superimposed on the solution structure of J1ex6 (PDB: 2KB9, first model, in red); exon boundaries and phases are also shown.

This made us to question if the dephasing of exon boundaries with respect to predicted domain boundaries in the region comprising these two atypical EGF repeats in the J1ex6 is accidental, or does there underlie any common evolutionary origin.

Exon/intron arrangement in this region of the JAG1 genes is very well conserved throughout evolution

The analysis of the exon/intron organization of human JAG1 orthologues in 26 different species including primates (5), non-primate mammals (15), birds (1), amphibians (1), and fishes (4) revealed that the exon/intron arrangement in the region encoding the J1ex6 of *JAG1* genes was found to be very well conserved throughout evolution, with a single exon encoding the C-terminal region of EGF1 and the complete EGF2 (**Figure 5.11**).

ERROR: timeout
OFFENDING COMMAND: timeout

STACK:

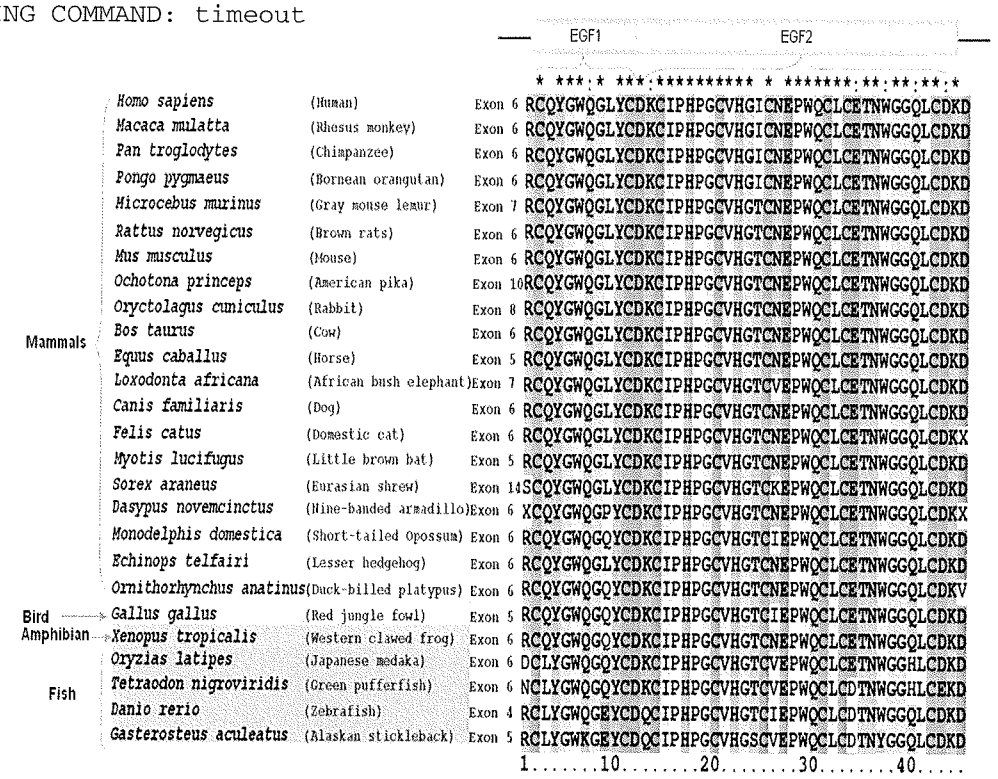
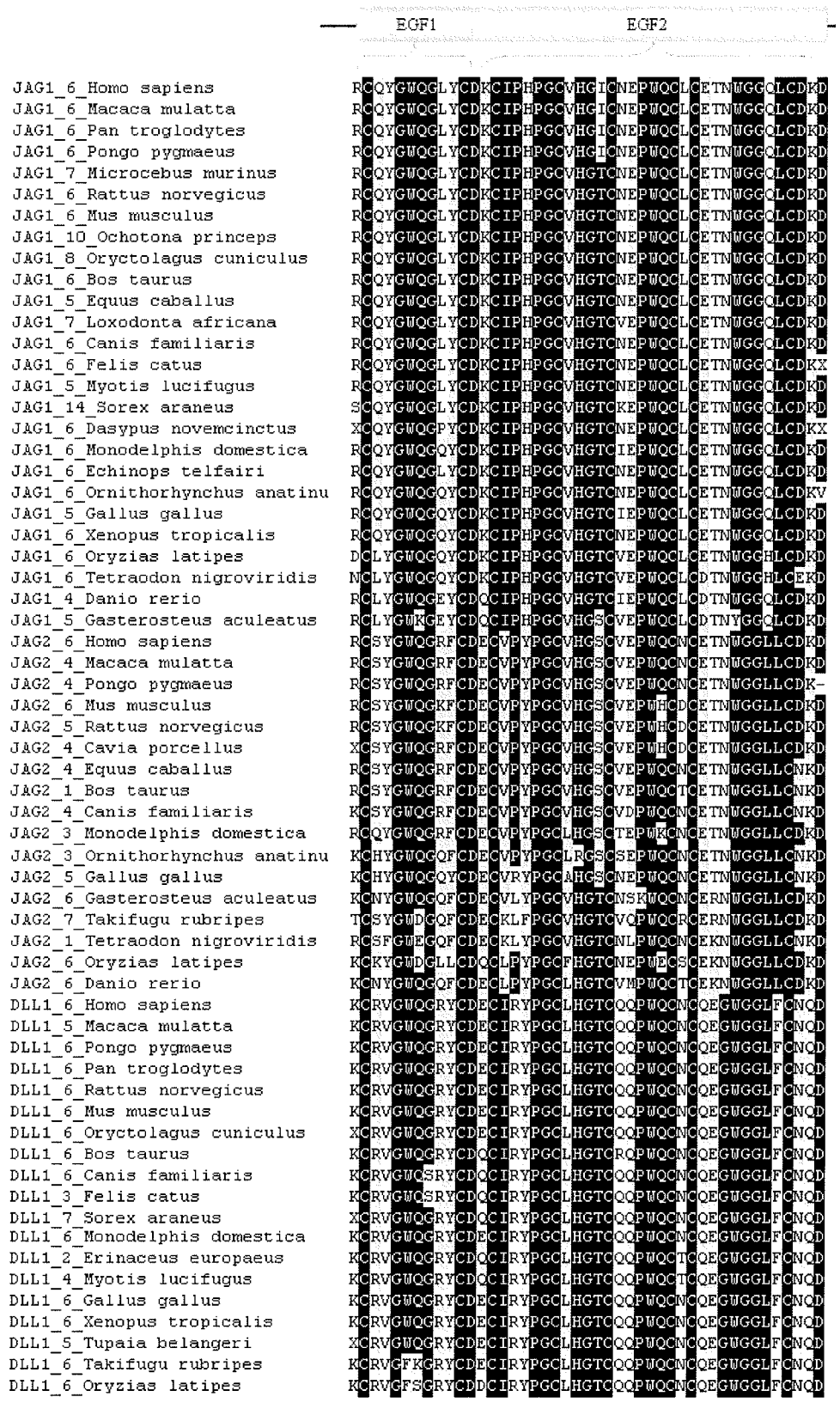


Figure 5.11: Multiple sequence alignment of the polypeptides encoded by exon 6 of human JAG1 and its orthologues in 26 different species using CLUSTAL-W.

Exon/Intron organisation preserved in homologues of Notch ligands

The extension of this analysis to all homologues of Notch ligands showed that the same exonic organization is found not only in *JAG1* but also in the *JAG2, DLL1, DLL4, DLK1,* and *DLK2* gene families, for a total of 112 genes in species varying from fishes to primates as seen in **Figure 5.12**. Usually, exon 6 (or its equivalent) is flanked by a phase 2 and a phase 1 intron on the 5' and 3' ends, respectively.


EGF1
EGF2
JAG1_6_Homo sapiens
JAG1_6_Macaca mulatta
JAG1_6_Pan troglodytes
JAG1_6_Pongo pygmaeus
JAG1_7_Microcebus murinus
JAG1_6_Rattus norvegicus
JAG1_6_Mus musculus
JAG1_10_Ochotona princeps
JAG1_8_Oryctolagus cuniculus
JAG1_6_Bos taurus
JAG1_5_Equus caballus
JAG1_7_Loxodonta africana
JAG1_6_Canis familiaris
JAG1_6_Felis catus
JAG1_5_Myotis lucifugus
JAG1_14_Sorex araneus
JAG1_6_Dasypus novemcinctus
JAG1_6_Monodelphis domestica
JAG1_6_Echinops telfairi
JAG1_6_Ornithorhynchus anatinu
JAG1_5_Gallus gallus
JAG1_6_Xenopus tropicalis
JAG1_6_Oryzias latipes
JAG1_6_Tetraodon nigroviridis
JAG1_4_Danio rerio
JAG1_5_Gasterosteus aculeatus
JAG2_6_Homo sapiens
JAG2_4_Macaca mulatta
JAG2_4_Pongo pygmaeus
JAG2_6_Mus musculus
JAG2_5_Rattus norvegicus
JAG2_4_Cavia porcellus
JAG2_4_Equus caballus
JAG2_1_Bos taurus
JAG2_4_Canis familiaris
JAG2_3_Monodelphis domestica
JAG2_3_Ornithorhynchus anatinu
JAG2_5_Gallus gallus
JAG2_6_Gasterosteus aculeatus
JAG2_7_Takifugu rubripes
JAG2_1_Tetraodon nigroviridis
JAG2_6_Oryzias latipes
JAG2_6_Danio rerio
DLL1_6_Homo sapiens
DLL1_5_Macaca mulatta
DLL1_6_Pongo pygmaeus
DLL1_6_Pan troglodytes
DLL1_6_Rattus norvegicus
DLL1_6_Mus musculus
DLL1_6_Oryctolagus cuniculus
DLL1_6_Bos taurus
DLL1_6_Canis familiaris
DLL1_3_Felis catus
DLL1_7_Sorex araneus
DLL1_6_Monodelphis domestica
DLL1_2_Erinaceus europaeus
DLL1_4_Myotis lucifugus
DLL1_6_Gallus gallus
DLL1_6_Xenopus tropicalis
DLL1_5_Tupaia belangeri
DLL1_6_Takifugu rubripes
DLL1_6_Oryzias latipes

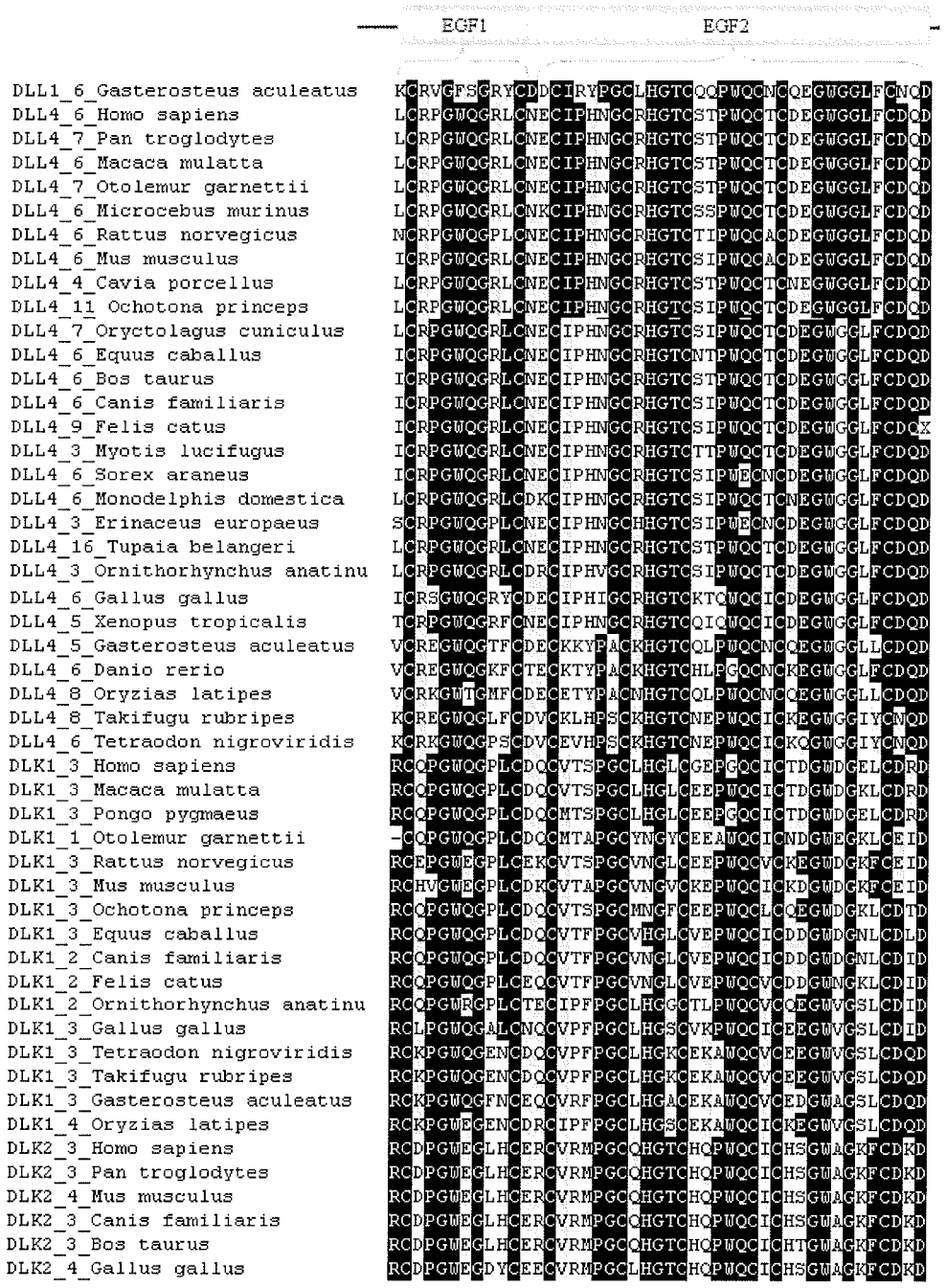


Figure 5.12: Multiple sequence alignment of the polypeptides encoded by exon 6 of human JAG1 and its homologues in different species.
All amino acid sequences annotated in ENSEMBLE as orthologues to *JAG1, JAG2, DLL1, DLL4, DLK1,* and *DLK2* were collected, broken down into segments corresponding to exons, and searched using BLAST with the sequence encoded by exon 6 of human Jagged-1; hits were then aligned using CLUSTAL-W.

With all this evidence, one can speculate that the particular exon structure in this region is dictated by the folding and structural requirements and that this atypically short EGF2 repeat might require the N-terminal extension for its correct folding

It is worthwhile to mention here that three were exceptions found while studying the exon/intron conservation and all of them occurring in lower organisms. **Figure 5.13** displays the outliers having a different exon/intron organization. In *Drosophila* Delta (DL_DROME) exon 6 is encoding not only the C-terminal region of EGF1 and the entire EGF2 but also the following EGFs; in *C. elegans* APX1 (APX1_CAEEL) a single exon is encoding both EGF1 and EGF2; whereas in zebrafish Delta-like B (DLLB_DANRE) a single exon is encoding three EGFs (1-3).

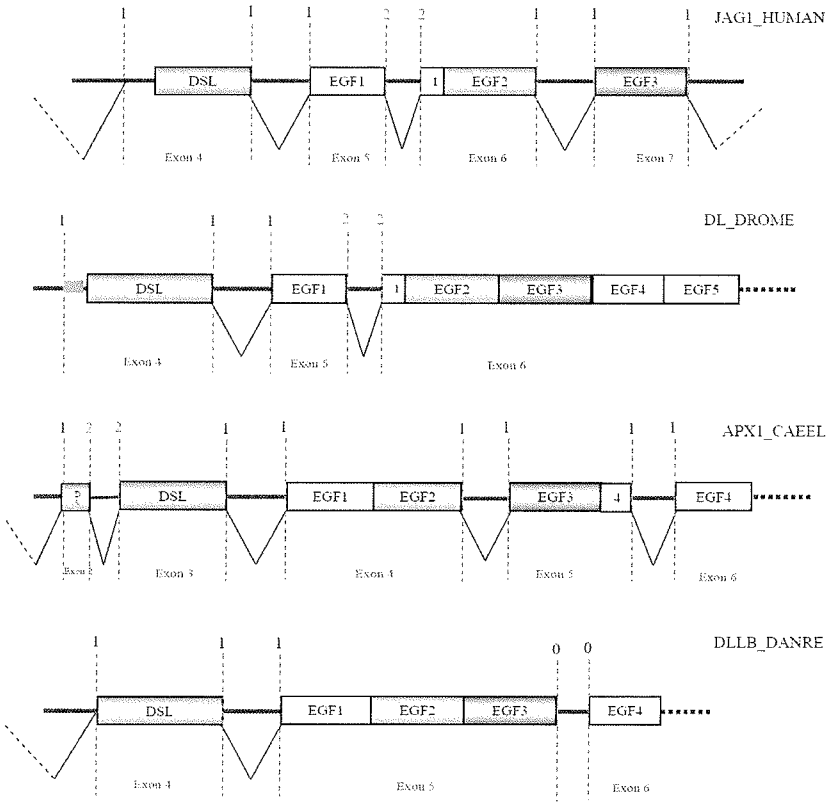


Figure 5.13: Exon/intron organization in outliers.
Diagrams showing exon/intron organization, intron phase, and domain architecture for the three outliers., *Drosophila* Delta (DL_DROME), *C.elegans* (APX1_CAEEL) and Zebrafish (DLLB_DANRE).

The sequence pattern of J1ex6 EGF2 is unique

The very low tolerance of J1ex6 to amino acid substitution at position 274 lead us to investigate whether the sequence pattern associated with EGF2 is found in other proteins. A pattern search in swiss-prot (http:// www.expasy.org/prosite/) produced 22 hits, which, surprisingly, are all related to Notch ligands in different organisms. In this dataset, G at position 274 was found to be absolutely conserved.

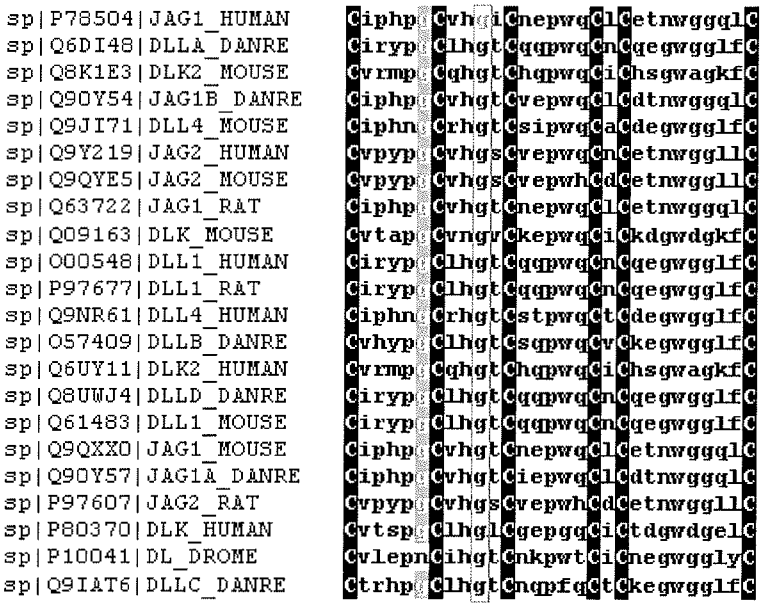


Figure 5.14: Multiple sequence alignment of sequences obtained from Swiss-Prot for a pattern associated with EGF2.
All the 22 sequences are related to Notch ligands in different organisms and Glycine at position 274 (marked in red) is absolutely conserved.

Extending the pattern search to trEMBL, we obtained 115 hits. A plot of Shannon entropy shows that, apart from Cysteines, there are only two additional positions that display no variability at all, the first corresponds to G274 in the Jagged-1 sequence, and the second to G290 (**Figure 5.15**). Thus supporting the idea that, in this specially constrained type of EGF, position 274 is not tolerant to substitution.

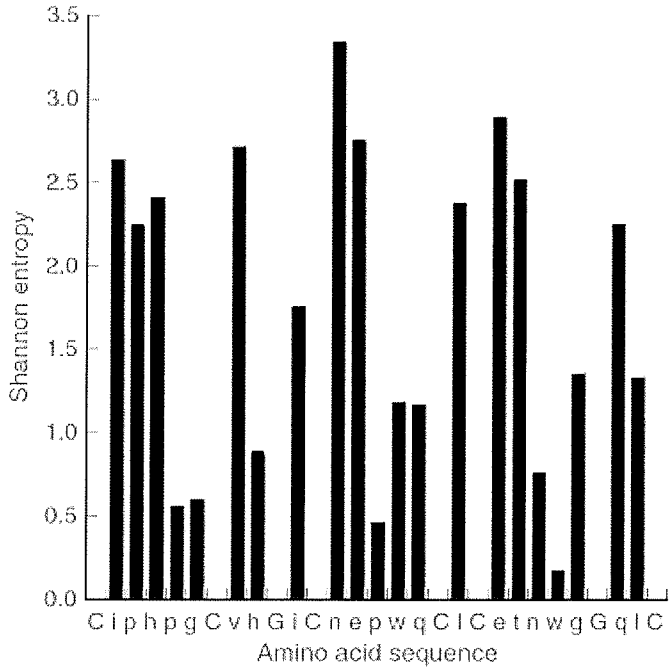


Figure 5.15: Shannon Entropy Plot.
Sequence variability in a set of 115 EGFs matching the pattern {C-X(5)-C-X(4)-C-X(5)-C-X-C-X(8)-C} plotted as Shannon entropy versus position. Values for the Shannon entropy can vary from zero (no variability) to a maximum of 5.3. The amino acid sequence of Jagged-1 EGF2 (residues 265–293) is shown on the x-axis; amino acids in capital letters are totally conserved.

Global analysis of disease-associated missense mutations in EGF containing proteins

Because the EGF domain is one of the most common structural building blocks in extracellular proteins (Campbell and Bork, 1993; Wouters et al., 2005), we decided to undertake a global analysis of disease-associated missense mutations found in EGF containing proteins with the idea of providing an important window to understanding human disease associated mutations.

To study the effect of change of physico-chemical properties of a specific amino acid, we compared the disease-associated and neutral mutations in terms of the chemical distance, as measured by the Grantham score (Grantham, 1974) (**Figure 5.16**). As mentioned in the methods section, the Grantham scale scores substitutions based on the based on the chemical dissimilarity between residues which can be viewed as a distance between two

amino acids. The greater the distance, the less similar the amino acids are, and the less exchangeable they become during evolution.

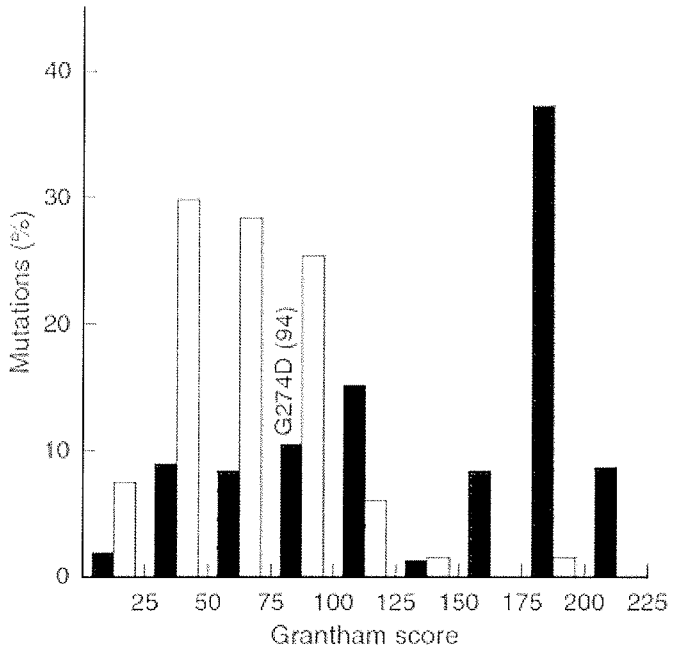


Figure 5.16: Physico-chemical analysis of mutations.
The percentage of disease-associated mutations (black bars) and polymorphism-related mutations (gray bars) are plotted versus their corresponding Grantham score.

As can be seen in **Figure 5.16**, we found that polymorphism-related mutations follow an almost bell-like distribution centred on relative small values of the Grantham score, whereas disease- associated mutations show an uneven distribution. Overall, it can be concluded that mutations with a high Grantham score are highly likely to be disease associated, but the contrasting case is not true, at least for EGF domains, suggesting that the chemical distance is not the only determinant.

Next we tried to identify positions in the EGF scaffold that are most sensitive to mutations. This type of analysis, however, turned out to be problematic because of the very high variability in the amino acid sequence of EGF domains and in the length of the loops, which together make both sequence and structural alignments unreliable for this purpose. We thus decided to carry out this type of analysis on a coarser basis, dividing the sequence of EGFs into seven windows, w1 to w7, and partitioning mutations accordingly (**Figure 5.17**). Polymorphism-related mutations show a relatively homogeneous distribution over

the sequence, whereas disease-related mutations are mainly found in w1, w3, w4 and, to a minor extent, in w6. The relatively high frequency of disease associated mutations in the N-terminal region most likely has no specific structural explanation, but is rather related to the strict requirement of specific amino acids (D / N) necessary for calcium coordination in calcium-binding EGF domains.

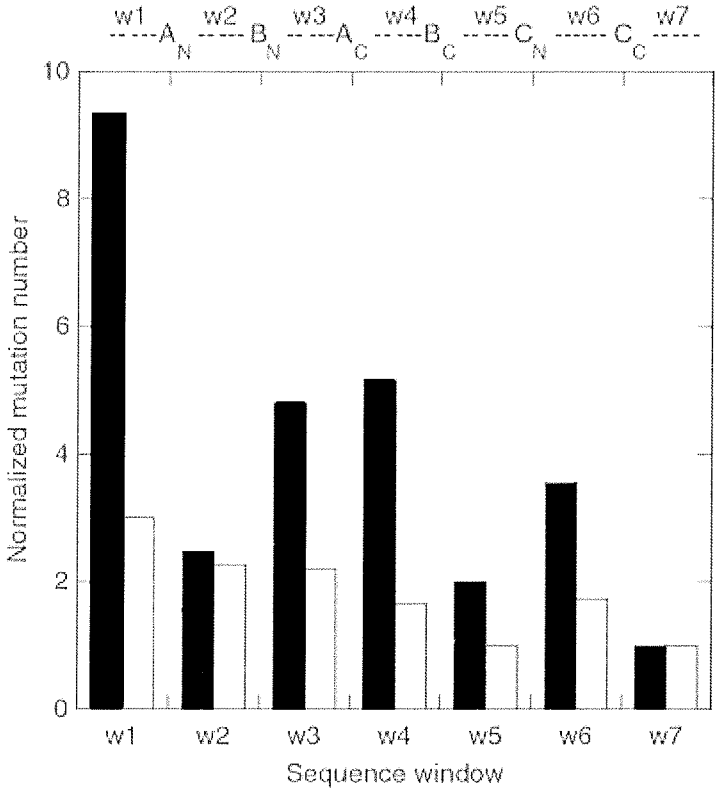


Figure 5.17: Positional analysis of mutations.
Disease-associated (black bars) and polymorphism-related (gray bars) mutations in EGF domains were partitioned according to their position in windows w1 to w7 and normalized for the average window size. Mutations involving cysteine were not considered. The six half-cystines are named according to the $A_NB_NA_CB_CC_NC_C$ annotation.

On the other hand, mutations in w3 and w4 are more likely to disrupt the two-strand antiparallel β-sheet that is the main (and sometimes the only) secondary structure element in EGF domains, or to involve residues that are required for the correct formation of the interface between two consecutive EGF repeats.

A separate positional analysis of Cysteine mutations, which are all disease-associated, showed that they are equally distributed, with no significant prevalence of the six positions.

It is known that by far the most frequents disease associated mutations found in EGF domains involve Cysteines, suggesting the high impact of this mutation in causing a disease. We wanted to see whether in the EGF there are any other amino acids like the Cysteine which when mutated has a higher impact in causing a disease. For this we calculated the frequency of each disease associated mutation type in EGF domains with that in the reference dataset (**Figure 5.18**).

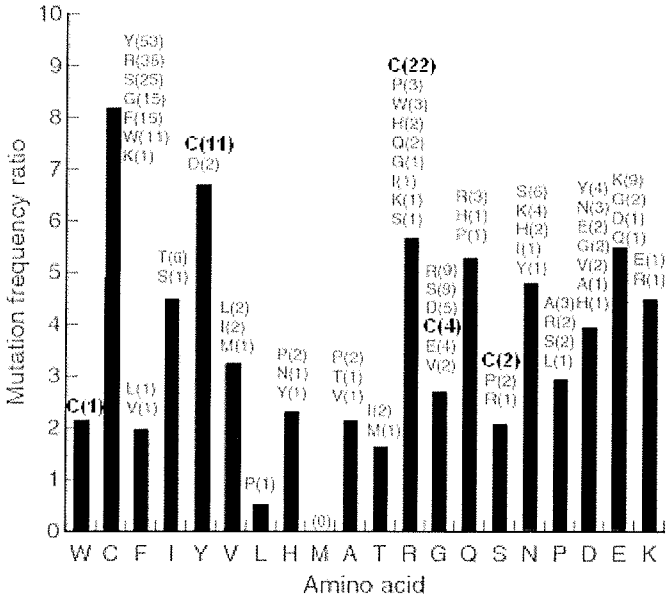


Figure 5.18: Disease-associated mutations in EGF domains.
The ratio between disease-associated mutation frequencies in EGF domains and the reference data set is plotted for each amino acid type. Amino acid types are shown in order of flexibility, as defined previously (Vihinen et al., 1994), from the least flexible (W) to the most flexible (K). The resulting amino acid and the number of occurrences for each mutation (in parenthesis) are shown above each bar. Mutations involving cysteines are shown in bold.

Although normalization drastically reduces the weight of mutations involving Cysteine, it is apparent that mutations either removing (C → X) or introducing a Cysteine (X → C, similar to Y → C and R → C) still have a great impact on EGF domains. This effect can be easily explained by the structural requirements of EGF domains, which, lacking a true hydrophobic core, are mainly stabilized by the three disulfide bridges. On the other hand, the introduction of an additional Cysteine is likely to scramble the oxidative folding of EGF domains in vivo. Oxidation of Cysteines to yield disulfide bonds is the most studied but not the only post-translational modification found in EGF domains (Harris and Spellman, 1993).β-hydroxylation of aspartate and asparagine, as well as different types of N- and O-

glycosylation, has been reported. Although the role of β-hydroxylation remains elusive, correct O-glucosylation and O-fucosylation of serine/threonine residues has been shownto be required for correct signaling mediated by Notch receptors (Haines and Irvine, 2003; Stanley, 2007).

5.4. Summary

In eukaryotic genomes, there is an overall very good correspondence between exon boundaries and predicted domain limits (Bjorklund et al., 2006; Liu and Grigoriev, 2004; Liu et al., 2005). Here, we have reported a case study where this correspondence is not fulfilled. Although it can be argued that in some instances domain limits cannot be defined precisely, this is not the case of EGF repeats, which are clearly recognizable by a very specific pattern of the three disulfide bonds and by the spacing between half-cystines. In this case study, the overall correspondence is maintained, with exons 5 and 6 encoding EGF1 and 2, but exon and domain boundaries are clearly out of phase, with exon 5 encoding a truncated EGF with only four half-cystines and exon 6 encoding the C-terminal half of EGF1 and the entire EGF2.

How can these results be reconciled with the experimental finding that exon 6 of human *JAG1* is encoding an autonomously folding and structural unit? Although from the statistical point of view this may be one of the rare instances where the 1:1 correspondence between exons and EGF repeat does not hold, the question remains if this has any structural or functional significance. It is possible that the particular exon structure in this region is dictated by folding and structural requirements. In this specific case, the constraints in the atypically short EGF2 repeat might require the N-terminal extension as an internal chaperone and a docking template to drive the correct folding. Moreover, as mentioned earlier, the presence of a kink at the interface between EGF1 and EGF2 observed in the crystal structure of the Jagged-1 region comprising the DSL domain and the first three EGF repeats (**Figure 5.10**) (Cordle et al., 2008) might not be accidental and may be required for correct binding to Notch receptors. In calcium binding EGFs, which are connected by a fairly long linker, the relative orientation of two adjacent domains is mainly determined by the geometric constraints imposed by the coordination of the calcium ion. In EGF1-2, the same objective is achieved by drastically reducing the length of the

linker region and encoding the C-terminal part of EGF1 and EGF2 in a single, conserved exon.

The G274D mutation in EGF2 of Jagged-1 despite occurring within the same window (w3 in **Figure 5.17**) and at a position that is structurally equivalent to G1127 in fibrillin-1 and G106 in factor IX (**Figure 5.19**), appears to affect folding in a more drastic way as compared to the G1127S mutation in fibrillin-1 and the G106S mutation in factor IX (Whiteman et al., 1998; Whiteman et al., 2001). Again, this can be attributed to the higher constraints in the structure of this atypical EGF, as indicated by the shorter B_N-B_C loop (10 residues, compared to 13 in fibrillin-1 and 14 in factor IX) and spacing between the first and last half-cystines (the A_N–C_C distance is 27 residues in Jagged-1 EGF2, compared to 35 in fibrillin-1 and 30 in factor IX) and supported by the observation that glycine at that position is totally conserved in Notch ligands (**Figure 5.15**).

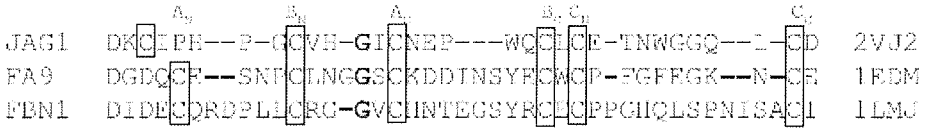


Figure 5.19: Structural alignment.
Multiple sequence alignment based on the structural alignment of EGF2 from Jagged-1 (*JAG1*; PDB code: 2VJ2), cbEGF1 from factor IX (FA9, PDB code: 1EDM) and cbEGF13 from fibrillin-1 (FBN1; PDB code: 1LMJ). Despite some discrepancy in the N-terminal region, half-cystines (boxed) and the mutated glycines (in bold) are aligned. Structure comparison was made using STAMP .

It is possible that the G274D mutation, introducing a larger charged amino acid, is more disrupting than a G → S mutation (a difference of 94 in the Grantham score, compared to 56 for a G → S mutation; **Figure 5.16**). However, the misfolding of the G274S and G274A mutants supports the hypothesis that no amino acid other than glycine can be accommodated at that position, regardless of the substitution type.

Additional missense mutations reported for exon 6 of JAG1 and expected to induce an amino acid replacement include G256S in EGF1 (Warthen et al., 2006), P269L (Ropke et al., 2003), C271R (Warthen et al., 2006), C284F (Boyer-Di Ponio et al., 2007; Boyer et al., 2005; Crosnier et al., 1999), and W288C (Boyer et al., 2005; Crosnier et al., 1999) in EGF2. All these six missense mutations share a common feature; they occur at residues that are either completely (positions 256, 271, 274 and 284) or very highly (positions 269 and 288)

conserved in the amino acid sequence (**Figure 5.15**). When considering all the 17 missense mutations occurring in the 16 EGF repeats of Jagged-1, ten involve either the replacement or the introduction of a cysteine, and are thus likely to be structurally disrupting (**Figure. 5.18**). Previously reported mappings of mutations over the Jagged-1 sequence (Crosnier et al., 1999; Ropke et al., 2003; Warthen et al., 2006) did not indicate the presence of any hot spot of critical region. Such mapping, however, was performed considering all types of possible mutations, including premature termination, and partitioning them over the 26 exons of the *JAG1* gene. Taking into account only missense mutations, which are likely to be more informative with respect to structural changes, and partitioning them over domains, rather than exons, it appears that the segment comprising the N-terminal domain, the DSL and the first two EGFs is most sensitive to missense mutations (**Appendix C** Figure C1 and Table C2). This is consistent with the DSL/EGF1-2 region being involved in receptor binding(Cordle et al., 2008; Shimizu et al., 1999) and points to a key role of the N-terminal domain. From this map, it can be speculated that two additional regions, one extending over EGF12–14 and the other including the von Willebrand factor type C domain, might also play a yet unidentified structural or functional role.

6. Discussion And Conclusions

Advances in experimental methods have generated, and continue to generate, enormous volumes of biological data that present significant storage, retrieval and analysis challenges (Slonim 2002).

Two factors dominate current developments in molecular biology :

1. the increasing amount of highly complex emperical data - in particular molecular and genetic data.
2. successful application of the data to biomedical research requires carefully and continuously curated and accurately annotated databanks.

Annotation has thus become a challenging and arguably the limiting, component of the whole enterprise. Most of current data interpretation ("data annotation") tasks are carried out by classifier algorithms. The annotations are then curated by experts and are added to the databases which are not only the most visible products of bioinformatics, but also the predominant form of representing biological knowledge today. This is a new situation that calls for new informatics approaches and as outlined in the preface, the subjects of my research has been focused towards that fit this changing scenery.

i) Even though automated classification methods (machine learning algorithms) are routinely used in most annotation pipelines, there are few methods that enable one to compare the efficiency of classification algorithms in bioinformatics tasks. How can we benchmark a data-interpretation method? I approach this subject via the analysis of the protein classification problem and the development of a benchmark database of 6405 classification tasks, applicable to test structural and functional annotation of proteins. This database contains sequence and 3D structure data organized into predefined positive/negative train/test groups in such a way that a known subclass is taken as the test group. This s achieved by building a set of classification tasks (positive train, positive test, negative train, negative test groups) for a protein database that has a category hierarchy (such as protein domains databases, protein family databases, phylogenetic hierarchies etc.).Although the idea exploiting the hierarchical structure of protein is not novel, the concepts such as `family' and

`superfamily', which were first introduced by Dayhoff (1976), are still valid for the ever-expanding protein universe.This subdivision – termed supervised cross-validation allows one to get a realistic estimate regarding a predictor's performance with respect to to hitherto unseen subclasses of the known classes and is applicable essentially to any datasets wherein the objects are classified in a hierarchical manner. This is a more stringent test than the generally known cross-validation principles used in the practice of machine learning. I illustrate the use of the Benchmark Collection by developing an algorithm based on a Committee of Classifiers.

ii) How can we compare complete annotations, such as domain architectures predicted by various prediction algorithms? This task is different from a simple classification problem as a complete domain architecture – or rather, a complete set of protein architectures annotated within a proteome- is a set of manually curated data, so the annotation process cannot be realistically repeated for statistical purposes. And since there are no gold standards, we can ask questions how different annotations relate to each other, or how they compare with manually curated annotations. I approached this problem by developing a general framework of comparison principles and numerical indices of similarity by which I could compare various protein domain annotation schemes. I show that similarity-based domain prediction performs as well, sometimes even better than generative models based on learning algorithms.

iii) When human database annotators identify groups, they do this by combining information from various sources such function, structure, protein interaction, in other terms, from large datasets of heterogeneous data. Human experts use high-level, and critical knowledge for this task. How can we integrate similarity data obtained from various data-sources by computational tools? One of the most general schemes to represent data-similarities is called a similarity space that can be represented as a network/matrix of similarities. I developed Multi-Netclust, a straightforward algorithm that can combine similarity data from different sources using a straightforward mixing algorithm borrowed from kernel

fusion. To our knowledge, there is no such freely available tool in bioinformatics that lets user to combine information (represented by similarity matrix) coming in from various data sources. This approach still uses human expertise, but instead of the high level background knowledge, the users only need to have an estimate of the background noise level. I showed that this approach can lead not only to better recognition but a substantial data compression as well.

iv) Finally, how do we apply these principles to practical problems? I carried out the structural analysis and classification of the newly determined ^{1}H-NMR solution structure of an epidermal growth factor (EGF) domain encoded by exon 6 of the *JAG1* gene. Apart from classifying this newely determined structure of the EGF domain we wanted to address a well-defined question, do the exon boundaries in *Jagged-1* coincide with the boundaries of the defined structural unit? This is relevant because the dogma says that multiple domain proteins having evolved through a process of exon duplication and shuffling. Exons that don't correspond to predicted domains are relatively infrequent, and such examples have not widely been studied. I found that this domain has an atypical structure and is encoded by an atypical exon/intron arrangment which is conserved throughout evolution. Moreover, since EGF domain is one of the most common structural building blocks in extracellular proteins, I carried out a systematic and comprehensive analysis of mutations found in EGF domains and showed that specific residue requirements for folding, structural integrity and correct post-translational processing may provide a rationale for most of the disease-associated mutations.

v) And finally, in the various comparisons I have used the examples of one tecchnique, similarity based prediction of protein domains. This is a seemingly simple method as compared to learning algorithms (generative models) such as profiles or HMM. However, as we complement similarity based predictions with simple thresholds, we convert this method into a discriminative model which in turn, performs nearly as well as the more sophisticated, generative models.

BIBLIOGRAPHY

BIBLIOGRAPHY

Ágoston, V., Kaján, L., Carugo, O., Hegedüs, Z., Vlahovicek, K. and Pongor, S. (2005) *Concepts of similarity in bioinformatics*. IOS Press, Amsterdam.

Allman, D., Punt, J.A., Izon, D.J., Aster, J.C. and Pear, W.S. (2002) An invitation to T and more: notch signaling in lymphopoiesis, *Cell*, 109 Suppl, S1-11.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool, *J Mol Biol*, 215, 403-410.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool, *Journal of molecular biology*, 215, 403-410.

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

Andreeva, A., Howorth, D., Brenner, S.E., Hubbard, T.J., Chothia, C. and Murzin, A.G. (2004) SCOP database in 2004: refinements integrate structure and sequence family data, *Nucleic acids research*, 32, D226-229.

Andreeva, A., Howorth, D., Chandonia, J.M., Brenner, S.E., Hubbard, T.J., Chothia, C. and Murzin, A.G. (2008) Data growth and its impact on the SCOP database: new developments, *Nucleic acids research*, 36, D419-425.

Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, C., Redaschi, N. and Yeh, L.S. (2004) UniProt: the Universal Protein knowledgebase, *Nucleic acids research*, 32, D115-119.

Attwood, T.K. (2000) The role of pattern databases in sequence analysis, *Brief Bioinform*, 1, 45-59.

Attwood, T.K., Bradley, P., Flower, D.R., Gaulton, A., Maudling, N., Mitchell, A.L., Moulton, G., Nordle, A., Paine, K., Taylor, P., Uddin, A. and Zygouri, C. (2003) PRINTS and its automatic supplement, prePRINTS, *Nucleic acids research*, 31, 400-402.

Bach, F.R., Lanckriet, G.R.G. and Jordan, M.I. (2004) Multiple Kernel Learning, Conic Duality, and the SMO Algorithm, *Proceedings of the 21 st International Conference on Machine Learning, Banff, Canada, 2.*

Bairoch, A. (1991) PROSITE: a dictionary of sites and patterns in proteins, *Nucleic acids research*, 19 Suppl, 2241-2245.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A. and Nielsen, H. (2000) Assessing the accuracy of prediction algorithms for classification: an overview, *Bioinformatics (Oxford, England)*, 16, 412-424.

Baron, M., Norman, D.G., Harvey, T.S., Handford, P.A., Mayhew, M., Tse, A.G., Brownlee, G.G. and Campbell, I.D. (1992) The three-dimensional structure of the first EGF-like module of human factor IX: comparison with EGF and TGF-alpha, *Protein Sci*, 1, 81-90.

Basu, M.K., Carmel, L., Rogozin, I.B. and Koonin, E.V. (2008) Evolution of protein domain promiscuity in eukaryotes, *Genome research*, 18, 449-461.

Beatus, P. and Lendahl, U. (1998) Notch and neurogenesis, *Journal of Neuroscience Research*, 54, 125 - 136.

Ben-Hur, A. and Noble, W.S. (2005) Kernel methods for predicting protein-protein interactions, *Bioinformatics (Oxford, England)*, 21 Suppl 1, i38-46.

Ben-Shlomo, I., Yu Hsu, S., Rauch, R., Kowalski, H.W. and Hsueh, A.J. (2003) Signaling receptome: a genomic and evolutionary perspective of plasma membrane receptors involved in signal transduction, *Sci STKE*, 2003, RE9.

Berman, H.M., Battistuz, T., Bhat, T.N., Bluhm, W.F., Bourne, P.E., Burkhardt, K., Feng, Z., Gilliland, G.L., Iype, L., Jain, S., Fagan, P., Marvin, J., Padilla, D., Ravichandran, V., Schneider, B., Thanki, N., Weissig, H., Westbrook, J.D. and Zardecki, C. (2002) The Protein Data Bank, *Acta crystallographica*, 58, 899-907.

Bishop, C.M. (1995) Neural Networks for Pattern Recognition, *Oxford University Press*, chapter 7.

Bjorklund, A.K., Ekman, D. and Elofsson, A. (2006) Expansion of protein domain repeats, *PLoS Comput. Biol.*, 2, e114.

Blake, C. (1978) Do genes-in-pieces imply proteins-in-pieces?, *Nature*, 273, 267.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S. and Schneider, M. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, *Nucleic acids research*, 31, 365-370.

Bork, P. (1992) The modular architecture of vertebrate collagens, *FEBS Lett*, 307, 49-54.

Bork, P. and Koonin, E.V. (1996) Protein sequence motifs, *Curr Opin Struct Biol*, 6, 366-376.

Boyer-Di Ponio, J., Wright-Crosnier, C., Groyer-Picard, M.T., Driancourt, C., Beau, I., Hadchouel, M. and Meunier-Rotival, M. (2007) Biological function of mutant forms of JAGGED1 proteins in Alagille syndrome: inhibitory effect on Notch signaling, *Hum. Mol. Genet.*, 16, 2683-2692.

Boyer, J., Crosnier, C., Driancourt, C., Raynaud, N., Gonzales, M., Hadchouel, M. and Meunier-Rotival, M. (2005) Expression of mutant JAGGED1 alleles in patients with Alagille syndrome, *Hum. Genet.*, 116, 445-453.

Bray, S.J. (2006) Notch signalling: a simple pathway becomes complex, *Nat Rev Mol Cell Biol*, 7, 678-689.

Breiman, L. (2001) Random forests, *Machine Learning*, 45 5-32.

Brin, S. and Page, L. (1998) The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, 30, 107-117.

Bru, C., Courcelle, E., Carrere, S., Beausse, Y., Dalmar, S. and Kahn, D. (2005) The ProDom database of protein domain families: more emphasis on 3D, *Nucleic acids research*, 33, D212-215.

Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X. and Chen, Y.Z. (2003) SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence, *Nucleic acids research*, 31, 3692-3697.

Campbell, I.D. and Bork, P. (1993) Epidermal growth factor-like modules, *Curr. Opin. Struct. Biol.*, 3, 385-392.

Chandonia, J.M., Hon, G., Walker, N.S., Lo Conte, L., Koehl, P., Levitt, M. and Brenner, S.E. (2004) The ASTRAL Compendium in 2004, *Nucleic acids research*, 32, D189-192.

Chen, F., Mackey, A.J., Stoeckert, C.J., Jr. and Roos, D.S. (2006) OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups, *Nucleic acids research*, 34, D363-368.

Chiang, Y.J., Goodrich, M.T., Grove, E.F., Tamassia, R., Vengroff, D.E. and Vitter, J.S. (1995) External-Memory Graph Algorithms, *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95).Society for Industrial and Applied Mathematics, Philadelphia, PA*, 139-149.

Cordle, J., Johnson, S., Tay, J.Z., Roversi, P., Wilkin, M.B., de Madrid, B.H., Shimizu, H., Jensen, S., Whiteman, P., Jin, B., Redfield, C., Baron, M., Lea, S.M. and Handford, P.A. (2008) A conserved face of the Jagged/Serrate DSL domain is involved in Notch trans-activation and cis-inhibition, *Nat. Struct. Mol. Biol.*, 15, 849-857.

Corpet, F., Gouzy, J. and Kahn, D. (1998) The ProDom database of protein domain families, *Nucleic acids research*, 26, 323-326.

Crosnier, C., Driancourt, C., Raynaud, N., Dhorne-Pollet, S., Pollet, N., Bernard, O., Hadchouel, M. and Meunier-Rotival, M. (1999) Mutations in JAGGED1 gene are predominantly sporadic in Alagille syndrome, *Gastroenterology*, 116, 1141-1148.

Dong, Q.W., Wang, X.L. and Lin, L. (2006) Application of latent semantic analysis to protein remote homology detection, *Bioinformatics (Oxford, England)*, 22, 285-290.

Doolittle, R.F. (1995) The multiplicity of domains in proteins, *Annual review of biochemistry*, 64, 287-314.

Doolittle, R.F. and Bork, P. (1993) Evolutionarily mobile modules in proteins, *Sci Am*, 269, 50-56.

Duda, R.O., Hart, P.E. and Stork, D.G. (2000) Pattern Classification., *Wiley Interscience. 2nd Edition.*

Egan, J.P. (1975) *Signal Detection Theory and Roc Analysis*. Academic Press.

Egan, J.P. (1975) Signal Detection theory and ROC Analysis, New York.

Enright, A.J., Kunin, V. and Ouzounis, C.A. (2003) Protein families and TRIBES in genome sequence space, *Nucleic acids research*, 31, 4632-4638.

Felsenstein, J. (1995) PHYLIP (Phylogeny Inference Package) Version 3.57c., *Department of Genetics, University of Washington, Seattle, WA.*

Finn, R.D., Tate, J., Mistry, J., Coggill, P.C., Sammut, S.J., Hotz, H.R., Ceric, G., Forslund, K., Eddy, S.R., Sonnhammer, E.L. and Bateman, A. (2008) The Pfam protein families database, *Nucleic acids research*, 36, D281-288.

Fiuza, U.M. and Arias, A.M. (2007) Cell and molecular biology of Notch, *J Endocrinol*, 194, 459-474.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition.* Academic Press, New York.

Gaspari, Z., Vlahovicek, K. and Pongor, S. (2005) Efficient recognition of folds in protein 3D structures by the improved PRIDE algorithm, *Bioinformatics (Oxford, England)*, 21, 3322-3323.

Gouzy, J., Corpet, F. and Kahn, D. (1999) Whole genome protein domain analysis using a new method for domain clustering, *Computers & chemistry*, 23, 333-340.

Gowri, V.S., Pandit, S.B., Karthik, P.S., Srinivasan, N. and Balaji, S. (2003) Integration of related sequences with protein three-dimensional structural families in an updated version of PALI database, *Nucleic acids research*, 31, 486-488.

Grantham, R. (1974) Amino acid difference formula to help explain protein evolution, *Science (New York, N.Y.*, 185, 862-864.

Greene, L.H., Lewis, T.E., Addou, S., Cuff, A., Dallman, T., Dibley, M., Redfern, O., Pearl, F., Nambudiry, R., Reid, A., Sillitoe, I., Yeats, C., Thornton, J.M. and Orengo, C.A. (2007) The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution, *Nucleic acids research*, 35, D291-297.

Gribskov, M., McLachlan, A.D. and Eisenberg, D. (1987) Profile analysis: detection of distantly related proteins, *Proceedings of the National Academy of Sciences of the United States of America*, 84, 4355-4358.

Gribskov, M. and Robinson, N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching, *Computers & chemistry*, 20, 25-33.

Grimaldi, R.P. (1999) *Discrete and Combinatorial Mathematics.*

Guarnaccia, C., Pintar, A. and Pongor, S. (2004) Exon 6 of human Jagged-1 encodes an autonomously folding unit, *FEBS Lett.*, 574, 156-160.

Haines, N. and Irvine, K.D. (2003) Glycosylation regulates Notch signalling, *Nat. Rev. Mol. Cell Biol.*, 4, 786-797.

Hamosh, A., Scott, A.F., Amberger, J.S., Bocchini, C.A. and McKusick, V.A. (2005) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders, *Nucleic Acids Res.*, 33, D514-517.

Harris, R.J. and Spellman, M.W. (1993) O-linked fucose and other post-translational modifications unique to EGF modules, *Glycobiology*, 3, 219-224.

Hastie, T., Tibshirani, R. and Friedman, J.H. (2003) *The Elements of Statistical Learning.*

Heger, A., Wilton, C.A., Sivakumar, A. and Holm, L. (2005) ADDA: a domain database with global coverage of the protein universe, *Nucleic acids research*, 33, D188-191.

Henery, R.J. (1994) Classification. In Michie, D., Spiegelhalter, D.J. and Taylor, C.C. (eds), *Machine Learning, Neural and Statistical Classification.New York: Ellis Horwood.*

Henikoff, J.G., Greene, E.A., Pietrokovski, S. and Henikoff, S. (2000) Increased coverage of protein families with the blocks database servers, *Nucleic acids research*, 28, 228-230.

Holm, L. and Park, J. (2000) DaliLite workbench for protein structure comparison, *Bioinformatics (Oxford, England)*, 16, 566-567.

Holm, L. and Sander, C. (1994) The FSSP database of structurally aligned protein fold families, *Nucleic acids research*, 22, 3600-3609.

Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L., Cox, T., Cuff, J., Curwen, V., Down, T., Durbin, R., Eyras, E., Gilbert, J., Hammond, M., Huminiecki, L., Kasprzyk, A., Lehvaslaiho, H., Lijnzaad, P., Melsopp, C., Mongin, E., Pettett, R., Pocock, M., Potter, S., Rust, A., Schmidt, E., Searle, S., Slater, G., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Stupka, E., Ureta-Vidal, A., Vastrik, I. and Clamp, M. (2002) The Ensembl genome database project, *Nucleic acids research*, 30, 38-41.

Hubbard, T.J., Aken, B.L., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P., Clarke, L., Coates, G., Fairley, S., Fitzgerald, S., Fernandez-Banet, J., Gordon, L., Graf, S., Haider, S., Hammond, M., Holland, R., Howe, K., Jenkinson, A., Johnson, N., Kahari, A., Keefe, D., Keenan, S., Kinsella, R., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Megy, K., Meidl, P., Overduin, B., Parker, A., Pritchard, B., Rios, D., Schuster, M., Slater, G., Smedley, D., Spooner, W., Spudich, G., Trevanion, S., Vilella, A., Vogel, J., White, S., Wilder, S., Zadissa, A., Birney, E., Cunningham, F., Curwen, V., Durbin, R., Fernandez-Suarez, X.M., Herrero, J., Kasprzyk, A., Proctor, G., Smith, J., Searle, S. and Flicek, P. (2009) Ensembl 2009, *Nucleic acids research*, 37, D690-697.

Hunter, S., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Das, U., Daugherty, L., Duquenne, L., Finn, R.D., Gough, J., Haft, D., Hulo, N., Kahn, D., Kelly, E., Laugraud, A., Letunic, I., Lonsdale, D., Lopez, R., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Mulder, N., Natale, D., Orengo, C., Quinn, A.F., Selengut, J.D., Sigrist, C.J., Thimma, M., Thomas, P.D., Valentin, F., Wilson, D., Wu, C.H. and Yeats, C. (2009) InterPro: the integrative protein signature database, *Nucleic acids research*, 37, D211-215.

Iso, T., Kedes, L. and Hamamori, Y. (2003) HES and HERP families: multiple effectors of the Notch signaling pathway, *Journal of cellular physiology*, 194, 237-255.

Jaakkola, T., Diekhans, M. and Haussler, D. (1999) Using the Fisher kernel method to detect remote protein homologies, *Proceedings /... International Conference on Intelligent Systems for Molecular Biology ; ISMB*, 149-158.

Jaakkola, T., Diekhans, M. and Haussler, D. (2000) A discriminative framework for detecting remote protein homologies, *J Comput Biol*, 7, 95-114.

Jean-Philippe Vert, H.S., and Tatsuya Akutsu (2004) Local alignment kernels for biological sequences, *Kernel Methods in Computational Biology, Cambridge, MA*.

Johnson, M.S., Sali, A. and Blundell, T.L. (1990) Phylogenetic relationships from three-dimensional protein structures, *Methods in enzymology*, 183, 670-690.

Kersey, P.J., Lawson, D., Birney, E., Derwent, P.S., Haimel, M., Herrero, J., Keenan, S., Kerhornou, A., Koscielny, G., Kahari, A., Kinsella, R.J., Kulesha, E., Maheswari, U., Megy, K., Nuhn, M., Proctor, G., Staines, D., Valentin, F., Vilella, A.J. and Yates, A. (2010) Ensembl Genomes: extending Ensembl across the taxonomic space, *Nucleic acids research*, 38, D563-569.

Kittler, J., Hatef, M., Duin, R.P. and Matas, J.G. (1998) On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 226-239.

Kocsor, A., Kertesz-Farkas, A., Kajan, L. and Pongor, S. (2006) Application of compression-based distance measures to protein sequence classification: a methodological study, *Bioinformatics (Oxford, England)*, 22, 407-412.

Koonin, E.V., Fedorova, N.D., Jackson, J.D., Jacobs, A.R., Krylov, D.M., Makarova, K.S., Mazumder, R., Mekhedov, S.L., Nikolskaya, A.N., Rao, B.S., Rogozin, I.B., Smirnov, S., Sorokin, A.V., Sverdlov, A.V., Vasudevan, S., Wolf, Y.I., Yin, J.J. and Natale, D.A. (2004) A comprehensive evolutionary classification of proteins encoded in complete eukaryotic genomes, *Genome biology*, 5, R7.

Kopan, R., Nye, J.S. and Weintraub, H. (1994) The intracellular domain of mouse Notch: a constitutively activated repressor of myogenesis directed at the basic helix-loop-helix region of MyoD, *Development*, 120, 2385-2396.

Krogh, A., Brown, M., Mian, I.S., Sjolander, K. and Haussler, D. (1994) Hidden Markov models in computational biology. Applications to protein modeling, *Journal of molecular biology*, 235, 1501-1531.

Kuzniar, A., Lin, K., He, Y., Nijveen, H., Pongor, S. and Leunissen, J.A. (2009) ProGMap: an integrated annotation resource for protein orthology, *Nucleic acids research*, 37, W428-434.

Lanckriet, G.R., De Bie, T., Cristianini, N., Jordan, M.I. and Noble, W.S. (2004b) A statistical framework for genomic data fusion, *Bioinformatics (Oxford, England)*, 20, 2626-2635.

Lanckriet, G.R.G., Cristianini, N., Bartlett, P.L., El Ghaoui, E. and Jordan, M.I. (2004a) Learning the Kernel Matrix with Semi-Definite Programming., *The Journal of Machine Learning Research*, 27–72.

Lander, E.S., Linton, L.M., Birren, B., Nusbaum, C., Zody, M.C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., Meldrim, J., Mesirov, J.P., Miranda, C., Morris, W., Naylor, J., Raymond, C., Rosetti, M., Santos, R., Sheridan, A., Sougnez, C., Stange-Thomann, N., Stojanovic, N., Subramanian, A.,

Wyman, D., Rogers, J., Sulston, J., Ainscough, R., Beck, S., Bentley, D., Burton, J., Clee, C., Carter, N., Coulson, A., Deadman, R., Deloukas, P., Dunham, A., Dunham, I., Durbin, R., French, L., Grafham, D., Gregory, S., Hubbard, T., Humphray, S., Hunt, A., Jones, M., Lloyd, C., McMurray, A., Matthews, L., Mercer, S., Milne, S., Mullikin, J.C., Mungall, A., Plumb, R., Ross, M., Shownkeen, R., Sims, S., Waterston, R.H., Wilson, R.K., Hillier, L.W., McPherson, J.D., Marra, M.A., Mardis, E.R., Fulton, L.A., Chinwalla, A.T., Pepin, K.H., Gish, W.R., Chissoe, S.L., Wendl, M.C., Delehaunty, K.D., Miner, T.L., Delehaunty, A., Kramer, J.B., Cook, L.L., Fulton, R.S., Johnson, D.L., Minx, P.J., Clifton, S.W., Hawkins, T., Branscomb, E., Predki, P., Richardson, P., Wenning, S., Slezak, T., Doggett, N., Cheng, J.F., Olsen, A., Lucas, S., Elkin, C., Uberbacher, E., Frazier, M., Gibbs, R.A., Muzny, D.M., Scherer, S.E., Bouck, J.B., Sodergren, E.J., Worley, K.C., Rives, C.M., Gorrell, J.H., Metzker, M.L., Naylor, S.L., Kucherlapati, R.S., Nelson, D.L., Weinstock, G.M., Sakaki, Y., Fujiyama, A., Hattori, M., Yada, T., Toyoda, A., Itoh, T., Kawagoe, C., Watanabe, H., Totoki, Y., Taylor, T., Weissenbach, J., Heilig, R., Saurin, W., Artiguenave, F., Brottier, P., Bruls, T., Pelletier, E., Robert, C., Wincker, P., Smith, D.R., Doucette-Stamm, L., Rubenfield, M., Weinstock, K., Lee, H.M., Dubois, J., Rosenthal, A., Platzer, M., Nyakatura, G., Taudien, S., Rump, A., Yang, H., Yu, J., Wang, J., Huang, G., Gu, J., Hood, L., Rowen, L., Madan, A., Qin, S., Davis, R.W., Federspiel, N.A., Abola, A.P., Proctor, M.J., Myers, R.M., Schmutz, J., Dickson, M., Grimwood, J., Cox, D.R., Olson, M.V., Kaul, R., Raymond, C., Shimizu, N., Kawasaki, K., Minoshima, S., Evans, G.A., Athanasiou, M., Schultz, R., Roe, B.A., Chen, F., Pan, H., Ramser, J., Lehrach, H., Reinhardt, R., McCombie, W.R., de la Bastide, M., Dedhia, N., Blocker, H., Hornischer, K., Nordsiek, G., Agarwala, R., Aravind, L., Bailey, J.A., Bateman, A., Batzoglou, S., Birney, E., Bork, P., Brown, D.G., Burge, C.B., Cerutti, L., Chen, H.C., Church, D., Clamp, M., Copley, R.R., Doerks, T., Eddy, S.R., Eichler, E.E., Furey, T.S., Galagan, J., Gilbert, J.G., Harmon, C., Hayashizaki, Y., Haussler, D., Hermjakob, H., Hokamp, K., Jang, W., Johnson, L.S., Jones, T.A., Kasif, S., Kaspryzk, A., Kennedy, S., Kent, W.J., Kitts, P., Koonin, E.V., Korf, I., Kulp, D., Lancet, D., Lowe, T.M., McLysaght, A., Mikkelsen, T., Moran, J.V., Mulder, N., Pollara, V.J., Ponting, C.P., Schuler, G., Schultz, J., Slater, G., Smit, A.F., Stupka, E., Szustakowski, J., Thierry-Mieg, D., Thierry-Mieg, J., Wagner, L., Wallis, J., Wheeler, R., Williams, A., Wolf, Y.I., Wolfe, K.H., Yang, S.P., Yeh, R.F., Collins, F., Guyer, M.S., Peterson, J., Felsenfeld, A., Wetterstrand, K.A., Patrinos, A., Morgan, M.J., de Jong, P., Catanese, J.J., Osoegawa, K., Shizuya, H., Choi, S. and Chen, Y.J. (2001) Initial sequencing and analysis of the human genome, *Nature*, 409, 860-921.

Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J.A., Armananzas, R., Santafe, G., Perez, A. and Robles, V. (2006) Machine learning in bioinformatics, *Briefings in bioinformatics*, 7, 86-112.

Lees, J., Yeats, C., Redfern, O., Clegg, A. and Orengo, C. (2010) Gene3D: merging structure and function for a Thousand genomes, *Nucleic acids research*, 38, D296-300.

Leslie, C., Eskin, E. and Noble, W.S. (2002) Mismatch string kernels for SVM protein classification, *Advances in Neural Information Processing Systems*.

Leslie, C., Eskin, E. and Noble, W.S. (2002) The spectrum kernel: a string kernel for SVM protein classification, *Pac Symp Biocomput*, 564-575.

Letunic, I., Doerks, T. and Bork, P. (2009) SMART 6: recent updates and new developments, *Nucleic acids research*, 37, D229-232.

Lewis, J. (1998) Notch signalling and the control of cell fate choices in vertebrates, *Semin Cell Dev Biol*, 9, 583-589.

Li, W. and Godzik, A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, *Bioinformatics (Oxford, England)*, 22, 1658-1659.

Liao, L. and Noble, W.S. (2003) Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships, *J Comput Biol*, 10, 857-868.

Lieber, T., Kidd, S., Alcamo, E., Corbin, V. and Young, M.W. (1993) Antineurogenic phenotypes induced by truncated Notch proteins indicate a role in signal transduction and may point to a novel function for Notch in nuclei, *Genes Dev*, 7, 1949-1965.

Lindahl, E. and Elofsson, A. (2000) Identification of related proteins on family, superfamily and fold level, *Journal of molecular biology*, 295, 613-625.

Liu, J. and Rost, B. (2004) Sequence-based prediction of protein domains, *Nucleic acids research*, 32, 3522-3530.

Liu, M. and Grigoriev, A. (2004) Protein domains correlate strongly with exons in multiple eukaryotic genomes--evidence of exon shuffling?, *Trends Genet.*, 20, 399-403.

Liu, M., Wu, S., Walch, H. and Grigoriev, A. (2005) Exon-domain correlation and its corollaries, *Bioinformatics (Oxford, England)*, 21, 3213-3216.

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

Marchler-Bauer, A., Anderson, J.B., Chitsaz, F., Derbyshire, M.K., DeWeese-Scott, C., Fong, J.H., Geer, L.Y., Geer, R.C., Gonzales, N.R., Gwadz, M., He, S., Hurwitz, D.I., Jackson, J.D., Ke, Z., Lanczycki, C.J., Liebert, C.A., Liu, C., Lu, F., Lu, S., Marchler, G.H., Mullokandov, M., Song, J.S., Tasneem, A., Thanki, N., Yamashita, R.A., Zhang, D., Zhang, N. and Bryant, S.H. (2009) CDD: specific functional annotation with the Conserved Domain Database, *Nucleic acids research*, 37, D205-210.

Marchler-Bauer, A. and Bryant, S.H. (2004) CD-Search: protein domain annotations on the fly, *Nucleic acids research*, 32, W327-331.

Melvin, I., Ie, E., Kuang, R., Weston, J., Stafford, W.N. and Leslie, C. (2007) SVM-Fold: a tool for discriminative multi-class protein fold and superfamily recognition, *BMC bioinformatics*, 8 Suppl 4, S2.

Mi, H., Dong, Q., Muruganujan, A., Gaudet, P., Lewis, S. and Thomas, P.D. (2010) PANTHER version 7: improved phylogenetic trees, orthologs and collaboration with the Gene Ontology Consortium, *Nucleic acids research*, 38, D204-210.

Mitchell, T.M. (1997) Machine Learning. In. McGraw-Hill, New York.

Miyata, T. and Suga, H. (2001) Divergence pattern of animal gene families and relationship with the Cambrian explosion, *Bioessays*, 23, 1018-1027.

Mottl, V., Tatarchuk, A., Sulimova, V., Krasotkina, O. and Seredin, O. (2007) Combining Pattern Recognition Modalities at the Sensor Level Via Kernel Fusion In, *Lecture Notes in Computer Science*.

Murvai, J., Vlahovicek, K., Barta, E. and Pongor, S. (2001) The SBASE protein domain library, release 8.0: a collection of annotated protein sequence segments, *Nucleic acids research*, 29, 58-60.

Murvai, J., Vlahovicek, K., Szepesvari, C. and Pongor, S. (2001) Prediction of protein functional domains from sequences using artificial neural networks, *Genome research*, 11, 1410-1417.

Murzin, A.G., Brenner, S.E., Hubbard, T. and Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures, *Journal of molecular biology*, 247, 536-540.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of molecular biology*, 48, 443-453.

Nielsen, H., Engelbrecht, J., Brunak, S. and von Heijne, G. (1997) Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites, *Protein engineering*, 10, 1-6.

Nikolskaya, A.N., Arighi, C.N., Huang, H., Barker, W.C. and Wu, C.H. (2006) PIRSF family classification system for protein functional and evolutionary analysis, *Evolutionary bioinformatics online*, 2, 197-209.

Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B. and Thornton, J.M. (1997) CATH--a hierarchic classification of protein domain structures, *Structure*, 5, 1093-1108.

Osborne, B.A. and Minter, L.M. (2007) Notch signalling during peripheral T-cell activation and differentiation, *Nat Rev Immunol*, 7, 64-75.

Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. and Chothia, C. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods, *J Mol Biol*, 284, 1201-1210.

Patthy, L. (1999) *Protein Evolution.* Blackwell Science Ltd., London, Edinburgh, Maiden MA.

Patthy, L. (2003) Modular assembly of genes and the evolution of new functions, *Genetica*, 118, 217-231.

Pavlidis, P., Weston, J., Cai, J. and W.S., N. (2002) Learning gene functional classifications from multiple data types., *Journal of Computational Biology*.

Pearson, W.R. (1990) Rapid and sensitive sequence comparison with FASTP and FASTA, *Methods in enzymology*, 183, 63-98.

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

Pollack, J.D., Li, Q. and Pearl, D.K. (2005) Taxonomic utility of a phylogenetic analysis of phosphoglycerate kinase proteins of Archaea, Bacteria, and Eukaryota: insights by Bayesian analyses, *Molecular phylogenetics and evolution*, 35, 420-430.

Pongor, S., Skerl, V., Cserzo, M., Hatsagi, Z., Simon, G. and Bevilacqua, V. (1993) The SBASE domain library: a collection of annotated protein segments, *Protein engineering*, 6, 391-395.

Pongor, S., Skerl, V., Cserzo, M., Hatsagi, Z., Simon, G. and Bevilacqua, V. (1993) The SBASE protein domain library, release 2.0: a collection of annotated protein sequence segments, *Nucleic acids research*, 21, 3111-3115.

Pruitt, K.D., Tatusova, T., Klimke, W. and Maglott, D.R. (2009) NCBI Reference Sequences: current status, policy and new initiatives, *Nucleic acids research*, 37, D32-36.

Rao, Z., Handford, P., Mayhew, M., Knott, V., Brownlee, G.G. and Stuart, D. (1995) The structure of a Ca(2+)-binding epidermal growth factor-like domain: its role in protein-protein interactions, *Cell*, 82, 131-141.

Rebay, I., Fehon, R.G. and Artavanis-Tsakonas, S. (1993) Specific truncations of Drosophila Notch define dominant activated and dominant negative forms of the receptor, *Cell*, 74, 319-329.

Rebay, I., Fleming, R.J., Fehon, R.G., Cherbas, L., Cherbas, P. and Artavanis-Tsakonas, S. (1991) Specific EGF repeats of Notch mediate interactions with Delta and Serrate: implications for Notch as a multifunctional receptor, *Cell*, 67, 687-699.

Rice, J.C. (1994) Logistic regression: An introduction, *Thompson,B.(ed.), Advances in social science methodology.*, 3, 191-245.

Rice, P., Longden, I. and Bleasby, A. (2000) EMBOSS: the European Molecular Biology Open Software Suite, *Trends Genet*, 16, 276-277.

Roehl, H., Bosenberg, M., Blelloch, R. and Kimble, J. (1996) Roles of the RAM and ANK domains in signaling by the C. elegans GLP-1 receptor, *Embo J*, 15, 7002-7012.

Ropke, A., Kujat, A., Graber, M., Giannakudis, J. and Hansmann, I. (2003) Identification of 36 novel Jagged1 (JAG1) mutations in patients with Alagille syndrome, *Hum. Mutat.*, 21, 100.

Rossmann, M.G. and Argos, P. (1976) Exploring structural homology of proteins, *Journal of molecular biology*, 105, 75-95.

Russell, R.B. and Barton, G.J. (1992) Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels, *Proteins*, 14, 309-323.

Saigo, H., Vert, J.P., Ueda, N. and Akutsu, T. (2004) Protein homology detection using string alignment kernels, *Bioinformatics (Oxford, England)*, 20, 1682-1689.

Sanchez-Irizarry, C., Carpenter, A.C., Weng, A.P., Pear, W.S., Aster, J.C. and Blacklow, S.C. (2004) Notch subunit heterodimerization and prevention of ligand-independent proteolytic activation depend, respectively, on a novel domain and the LNR repeats, *Mol Cell Biol*, 24, 9265-9273.

Sayers, E.W., Barrett, T., Benson, D.A., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., DiCuccio, M., Edgar, R., Federhen, S., Feolo, M., Geer, L.Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D.J., Madden, T.L., Maglott, D.R., Miller, V., Mizrachi, I., Ostell, J., Pruitt, K.D., Schuler, G.D., Sequeira, E., Sherry, S.T., Shumway, M., Sirotkin, K., Souvorov, A., Starchenko, G., Tatusova, T.A., Wagner, L., Yaschenko, E. and Ye, J. (2009) Database resources of the National Center for Biotechnology Information, *Nucleic acids research*, 37, D5-15.

Schultz, J., Milpetz, F., Bork, P. and Ponting, C.P. (1998) SMART, a simple modular architecture research tool: identification of signaling domains, *Proceedings of the National Academy of Sciences of the United States of America*, 95, 5857-5864.

Selengut, J.D., Haft, D.H., Davidsen, T., Ganapathy, A., Gwinn-Giglio, M., Nelson, W.C., Richter, A.R. and White, O. (2007) TIGRFAMs and Genome Properties: tools for the assignment of molecular function and biological process in prokaryotic genomes, *Nucleic acids research*, 35, D260-264.

Shavik, J., Hunter, L. and Searls, D. (1995) Introduction., *Machine Learning*, 5-10.

Shimizu, K., Chiba, S., Kumano, K., Hosoya, N., Takahashi, T., Kanda, Y., Hamada, Y., Yazaki, Y. and Hirai, H. (1999) Mouse jagged1 physically interacts with notch2 and other notch receptors. Assessment by quantitative methods, *J. Biol. Chem.*, 274, 32961-32969.

Sigrist, C.J., Cerutti, L., de Castro, E., Langendijk-Genevaux, P.S., Bulliard, V., Bairoch, A. and Hulo, N. (2010) PROSITE, a protein domain database for functional characterization and annotation., *Nucleic acids research*, 38, D161-166.

Sigrist, C.J., De Castro, E., Langendijk-Genevaux, P.S., Le Saux, V., Bairoch, A. and Hulo, N. (2005) ProRule: a new database containing functional and structural information on PROSITE profiles, *Bioinformatics (Oxford, England)*, 21, 4060-4066.

Silverstein, K.A., Shoop, E., Johnson, J.E., Kilian, A., Freeman, J.L., Kunau, T.M., Awad, I.A., Mayer, M. and Retzel, E.F. (2001) The MetaFam Server: a comprehensive protein family resource, *Nucleic acids research*, 29, 49-51.

Simon, G., Paladini, R., Tisminetzky, S., Cserzo, M., Hatsagi, Z., Tossi, A. and Pongor, S. (1992) Improved detection of homology in distantly related proteins: similarity of adducin with actin-binding proteins, *Protein sequences & data analysis*, 5, 39-42.

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences, *Journal of molecular biology*, 147, 195-197.

Sonego, P., Kocsor, A. and Pongor, S. (2008) ROC analysis: applications to the classification of biological sequences and 3D structures, *Briefings in bioinformatics*, 9, 198-209.

Sonego, P., Pacurar, M., Dhir, S., Kertesz-Farkas, A., Kocsor, A., Gaspari, Z., Leunissen, J.A. and Pongor, S. (2007) A Protein Classification Benchmark collection for machine learning, *Nucleic acids research*, 35, D232-236.

Sonnenburg, S., Rätsch, G., Schäfer, C. and Schölkopf, B. (2006) Large Scale Multiple Kernel Learning, *Journal of Machine Learning Research*, 7.

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

Sonnhammer, E.L. and Kahn, D. (1994) Modular arrangement of proteins as inferred from analysis of homology, *Protein Sci*, 3, 482-492.

Stanley, P. (2007) Regulation of Notch signaling by glycosylation, *Curr. Opin. Struct. Biol.*, 17, 530-535.

Stifani, S., Blaumueller, C.M., Redhead, N.J., Hill, R.E. and Artavanis-Tsakonas, S. (1992) Human homologs of a Drosophila Enhancer of split gene product define a novel family of nuclear proteins, *Nature genetics*, 2, 119-127.

Tarjan, R.E. (1975) Efficiency of a Good But Not Linear Set Union Algorithm, *Journal of the ACM (J.ACM)*, 22, 215 - 225

Tatusov, R.L., Fedorova, N.D., Jackson, J.D., Jacobs, A.R., Kiryutin, B., Koonin, E.V., Krylov, D.M., Mazumder, R., Mekhedov, S.L., Nikolskaya, A.N., Rao, B.S., Smirnov, S., Sverdlov, A.V., Vasudevan, S., Wolf, Y.I., Yin, J.J. and Natale, D.A. (2003) The COG database: an updated version includes eukaryotes, *BMC bioinformatics*, 4, 41.

Thornton, J.M., Orengo, C.A., Todd, A.E. and Pearl, F.M. (1999) Protein folds, functions and evolution, *J Mol Biol*, 293, 333-342.

Tordai, H., Nagy, A., Farkas, K., Banyai, L. and Patthy, L. (2005) Modules, multidomain proteins and organismic complexity, *The FEBS journal*, 272, 5064-5078.

UniProt (2010) The Universal Protein Resource (UniProt) in 2010, *Nucleic acids research*, 38, D142-148.

Vapnik, V.N. (1998) Statistical Learning Theory, *Wiley, New York*.

Vert, J.P., Qiu, J. and Noble, W.S. (2007) A new pairwise kernel for biological network inference with support vector machines, *BMC bioinformatics*, 8 Suppl 10, S8.

Vihinen, M., Torkkila, E. and Riikonen, P. (1994) Accuracy of protein flexibility predictions, *Proteins*, 19, 141-149.

Vitkup, D., Sander, C. and Church, G.M. (2003) The amino-acid mutational spectrum of human genetic disease, *Genome Biol.*, 4, R72.

Vlahovicek, K., Kajan, L., Agoston, V. and Pongor, S. (2005) The SBASE domain sequence resource, release 12: prediction of protein domain-architecture using support vector machines, *Nucleic acids research*, 33, D223-225.

von Heijne, G. (1992) Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule, *Journal of molecular biology*, 225, 487-494.

Warthen, D.M., Moore, E.C., Kamath, B.M., Morrissette, J.J., Sanchez, P., Piccoli, D.A., Krantz, I.D. and Spinner, N.B. (2006) Jagged1 (JAG1) mutations in Alagille syndrome: increasing the mutation detection rate, *Hum. Mutat.*, 27, 436-443.

Watanabe, S. (1985) *Pattern Recognition: Human and Mechanical.* Wiley, New York.

Weinmaster, G. (2000) Notch signal transduction: a real rip and more, *Current opinion in genetics & development*, 10, 363-369.

Weston, J., Elisseeff, A., Zhou, D., Leslie, C.S. and Noble, W.S. (2004) Protein ranking: from local to global structure in the protein similarity network, *Proceedings of the National Academy of Sciences of the United States of America*, 101, 6559-6563.

Whiteman, P., Downing, A.K., Smallridge, R., Winship, P.R. and Handford, P.A. (1998) A Gly --> Ser change causes defective folding in vitro of calcium-binding epidermal growth factor-like domains from factor IX and fibrillin-1, *J. Biol. Chem.*, 273, 7807-7813.

Whiteman, P., Smallridge, R.S., Knott, V., Cordle, J.J., Downing, A.K. and Handford, P.A. (2001) A G1127S change in calcium-binding epidermal growth factor-like domain 13 of human fibrillin-1 causes short range conformational effects, *J. Biol. Chem.*, 276, 17156-17162.

Wilson, D., Madera, M., Vogel, C., Chothia, C. and Gough, J. (2007) The SUPERFAMILY database in 2007: families and functions, *Nucleic acids research*, 35, D308-313.

Wilson, D., Pethica, R., Zhou, Y., Talbot, C., Vogel, C., Madera, M., Chothia, C. and Gough, J. (2009) SUPERFAMILY--sophisticated comparative genomics, data mining, visualization and phylogeny, *Nucleic acids research*, 37, D380-386.

Wouters, M.A., Rigoutsos, I., Chu, C.K., Feng, L.L., Sparrow, D.B. and Dunwoodie, S.L. (2005) Evolution of distinct EGF domains with specific functions, *Protein Sci.*, 14, 1091-1103.

Wu, C.H., Huang, H., Nikolskaya, A., Hu, Z. and Barker, W.C. (2004) The iProClass integrated database for protein functional analysis, *Computational biology and chemistry*, 28, 87-96.

Zweig, M.H. and Campbell, G. (1993) Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine, 39, 561-577.

APPENDICES

APPENDIX A

This appendix contains the screenshot for a cast-matrix as mentioned in *Chapter-2.*

ID	Archaea_Crenarchaeot	Archaea_Euryarchaeota	Bacteria_Actinobacterida	Bacteria_Firmicutes	Bacteria_Proteobacteria	Eukaryota_Alveolata	Eukaryota_E
PGK_METTH	1	3	4	4	4	4	4
PGK_METJA	1	3	4	4	4	4	4
PGK_ARCFU	1	3	4	4	4	4	4
PGK_METKA	1	3	4	4	4	4	4
PGK_PYRFU	1	3	4	4	4	4	4
PGK_HALSA	1	3	4	4	4	4	4
PGK_PYRAB	1	3	4	4	4	4	4
PGK_PYRHO	1	3	4	4	4	4	4
PGK_METAC	1	3	4	4	4	4	4
PGK_METMA_sp	1	3	4	4	4	4	4
PGK_HALVA	1	3	4	4	4	4	4
PGK_SULSO	3	1	2	2	2	2	2
PGK_SULTO	3	1	2	2	2	2	2
PGK_PYRAE	3	1	2	2	2	2	2
PGK_AERPE	3	1	2	2	2	2	2
PGKC_ALCEU	4	4	1	1	3	4	4
PGK_RALSO	4	4	1	1	3	4	4
PGK_NEIMA	4	4	1	1	3	4	4
PGK_FUSNN	4	4	1	1	3	4	4
PGKT_THEMA_1	4	4	1	1	3	4	4
PGK_DEIRA	4	4	1	1	3	4	4
PGK_THETH_sp	4	4	1	1	3	4	4
PGK_TREPA_sp	4	4	1	1	3	4	4
PGK_BORBU	4	4	1	1	3	4	4
PGK_CHLTE	4	4	1	1	3	4	4

Figure A1: Screenshot of a cast-matrix from the Protein Classification Benchmark for the 3PGK (Pollack et al., 2005).

APPENDIX B

This appendix contains the description of the three metrices: *Presence/Absence*, *Composition* and *Boundary* designed for comparing annotations, as used in *Chapter-4*. It also contains a list of the 278 domain types (provided as InterPro accession numbers along with the entry Id) used for the study.

The following lines give a description of scheme used for carrying out comparison between two annotation methods at the level of "Domain Type" and "Protein Architecture".

Comparison of different domain prediction methods using "Domain Type"

1) Presence / Absence:

   TP (correct) = Sum of the number of proteins predicted by both the annotations methods
   Incorrect = Sum of the number of proteins that are only predicted by either of the two annotation methods.

2) Composition / Number:

   TP (correct) = Sum of the number of cases when the abundance of proteins having a particular domain type is correctly predicted by both the annotation methods.
   Incorrect = Sum of the number of cases when the abundance of proteins having a particular domain type does not match between the two annotation methods being compared.

3) Boundary:

   TP = Sum of all those cases where the boundaries predicted by the two methods for the domain type in question either matches or falls within a set tolerance level.
   Incorrect =Otherwise.
   * The tolerance level was set to ±10 for this study.

Comparison of different domain prediction methods using "Protein Architecture"

1) Presence / Absence:

   TP = Gets a score 1, if all domain types predicted by both the methods for a particular protein match, neither more nor less.

   Incorrect = Gets a score 1, if for a particular protein, either of the two annotation methods being compared predicts (i) extra domain type or (ii) fails to predict any. TP becomes 0 in this case.

2) Composition / Number:

TP = Gets a score 1 if the number of all domain types predicted by the two annotation methods match.
Incorrect =Otherwise. (TP becomes 0 in this case)

3) Boundary:

TP = Gets a score 1 if, for a particular protein, all the domains boundaries predicted by the two methods being compared either match or fall within a set tolerance level.
Incorrect: Gets a score 1 otherwise. (TP becomes 0 in this case)
* The tolerance level for this study was set to ±10.

Table B1: List of InterPro Ids along with their description.

InterPro Id	Description
IPR000010	Proteinase inhibitor I25, cystatin
IPR000014	PAS
IPR000020	Anaphylatoxin/fibulin
IPR000033	Low-density lipoprotein receptor, class B (YWTD) repeat
IPR000034	Laminin B type IV
IPR000048	IQ calmodulin-binding region
IPR000058	Zinc finger, AN1-type
IPR000061	SWAP/Surp
IPR000082	SEA
IPR000083	Fibronectin, type I
IPR000095	PAK-box/P21-Rho-binding
IPR000157	Toll-Interleukin receptor
IPR000195	RabGAP/TBC
IPR000197	Zinc finger, TAZ-type
IPR000203	GPS domain
IPR000219	Dbl homology (DH) domain
IPR000225	Armadillo
IPR000237	GRIP
IPR000242	Protein-tyrosine phosphatase, receptor/non-receptor type
IPR000270	Octicosapeptide/Phox/Bem1p
IPR000294	Gamma-carboxyglutamic acid-rich (GLA) domain
IPR000313	PWWP
IPR000315	Zinc finger, B-box
IPR000327	POU-specific
IPR000372	Leucine-rich repeat-containing N-terminal domain
IPR000433	Zinc finger, ZZ-type
IPR000436	Sushi/SCR/CCP
IPR000449	Ubiquitin-associated/translation elongation factor EF1B, N-terminal
IPR000467	D111/G-patch
IPR000504	RNA recognition motif, RNP-1
IPR000519	P-type trefoil
IPR000555	Mov34/MPN/PAD-1
IPR000569	HECT

IPR000571	Zinc finger, CCCH-type
IPR000601	PKD
IPR000644	Cystathionine beta-synthase, core
IPR000651	Ras-like guanine nucleotide exchange factor, N-terminal
IPR000679	Zinc finger, GATA-type
IPR000716	Thyroglobulin type-1
IPR000727	Target SNARE coiled-coil domain
IPR000772	Ricin B lectin
IPR000782	FAS1 domain
IPR000800	Notch domain
IPR000857	MyTH4 domain
IPR000859	CUB
IPR000861	HR1-like rho-binding repeat
IPR000867	Insulin-like growth factor-binding protein, IGFBP
IPR000884	Thrombospondin, type 1 repeat
IPR000885	Fibrillar collagen, C-terminal
IPR000900	Nebulin 35 residue motif
IPR000904	SEC7-like
IPR000906	ZU5
IPR000909	Phospholipase C, phosphatidylinositol-specific, X domain
IPR000953	Chromo domain
IPR000961	AGC-kinase, C-terminal
IPR000967	Zinc finger, NF-X1-type
IPR000980	SH2 motif
IPR000998	MAM
IPR000999	Ribonuclease III
IPR001007	Von Willebrand factor, type C
IPR001012	UBX
IPR001025	Bromo adjacent homology (BAH) domain
IPR001054	Adenylyl cyclase class-3/4/guanylyl cyclise
IPR001060	Fps/Fes/Fer/CIP4 homology
IPR001073	Complement C1q protein
IPR001092	Helix-loop-helix DNA-binding domain
IPR001101	Plectin repeat
IPR001156	Peptidase S60, transferrin lactoferrin
IPR001158	DIX
IPR001164	Arf GTPase activating protein
IPR001180	Citron-like
IPR001202	WW/Rsp5/WWP
IPR001206	Diacylglycerol kinase, catalytic domain
IPR001212	Somatomedin B
IPR001222	Zinc finger, TFIIS-type
IPR001251	Cellular retinaldehyde-binding/triple function, C-terminal
IPR001300	Peptidase C2, calpain
IPR001313	Pumilio RNA-binding repeat
IPR001357	BRCT
IPR001368	TNFR/CD27/30/40/95 cysteine-rich region
IPR001370	Proteinase inhibitor I32, inhibitor of apoptosis
IPR001374	Single-stranded nucleic acid binding R3H
IPR001401	Dynamin, GTPase domain
IPR001452	Src homology-3 domain
IPR001478	PDZ/DHR/GLGF
IPR001487	Bromodomain

IPR001496	SOCS protein, C-terminal
IPR001507	Endoglin/CD105 antigen
IPR001562	Zinc finger, Btk motif
IPR001606	ARID/BRIGHT DNA-binding domain
IPR001607	Zinc finger, UBP-type
IPR001609	Myosin head, motor domain
IPR001623	Heat shock protein DnaJ, N-terminal
IPR001650	DNA/RNA helicase, C-terminal
IPR001680	WD40 repeat
IPR001711	Phospholipase C, phosphatidylinositol-specific, Y domain
IPR001736	Phospholipase D/Transphosphatidylase
IPR001752	Kinesin, motor domain
IPR001762	Blood coagulation inhibitor, Disintegrin
IPR001763	Rhodanese-like
IPR001774	Delta/Serrate/lag-2 (DSL) protein
IPR001781	Zinc finger, LIM-type
IPR001810	Cyclin-like F-box
IPR001846	Von Willebrand factor, type D
IPR001849	Pleckstrin homology
IPR001876	Zinc finger, RanBP2-type
IPR001878	Zinc finger, CCHC-type
IPR001881	EGF-like calcium-binding
IPR001895	Guanine-nucleotide dissociation stimulator CDC25
IPR001909	Krueppel-associated box
IPR001965	Zinc finger, PHD-type
IPR002004	Polyadenylate-binding protein/Hyperplastic disc protein
IPR002035	Von Willebrand factor, type A
IPR002049	EGF-like, laminin
IPR002108	Actin-binding, cofilin/tropomyosin type
IPR002110	Ankyrin repeat
IPR002121	Helicase/RNase D C-terminal, HRDC domain
IPR002172	Low density lipoprotein-receptor, class A (cysteine-rich) repeat
IPR002181	Fibrinogen, alpha/beta/gamma chain, C-terminal globular
IPR002219	Protein kinase C-like, phorbol ester/diacylglycerol binding
IPR002223	Proteinase inhibitor I2, Kunitz metazoa
IPR002350	Proteinase inhibitor I1, Kazal
IPR002404	Insulin receptor substrate-1, PTB
IPR002466	Adenosine deaminase/editase
IPR002483	Splicing factor PWI
IPR002498	Phosphatidylinositol-4-phosphate 5-kinase, core
IPR002558	I/LWEQ
IPR002589	Appr-1-p processing
IPR002653	Zinc finger, A20-type
IPR002713	FF domain
IPR002867	Zinc finger, C6HC-type
IPR002889	Carbohydrate-binding WSC
IPR002913	Lipid-binding START
IPR003018	GAF
IPR003034	DNA-binding SAP
IPR003103	Apoptosis regulator, Bcl-2 protein, BAG
IPR003105	SRA-YDG
IPR003107	RNA-processing protein, HAT helix
IPR003109	GoLoco motif
IPR003110	Phosphorylated immunoreceptor signaling ITAM

IPR003112	Olfactomedin-like
IPR003114	Phox-associated domain
IPR003116	Raf-like Ras-binding
IPR003119	Saposin type A
IPR003123	Vacuolar sorting protein 9
IPR003124	Actin-binding WH2
IPR003126	Zinc finger, N-recognin
IPR003127	Sorbin-like
IPR003128	Villin headpiece
IPR003306	WIF domain
IPR003309	Transcription regulator SCAN
IPR003347	Transcription factor jumonji/aspartyl beta-hydroxylase
IPR003349	Transcription factor jumonji, JmjN
IPR003409	MORN motif
IPR003533	Doublecortin
IPR003593	ATPase, AAA+ type, core
IPR003597	Immunoglobulin C1-set
IPR003605	TGF beta receptor, GS motif
IPR003607	Metal-dependent phosphohydrolase, HD domain
IPR003609	Apple-like
IPR003618	Transcription elongation factor S-II, central domain
IPR003619	MAD homology 1, Dwarfin-type
IPR003644	Na-Ca exchanger/integrin-beta4
IPR003645	Follistatin-like, N-terminal
IPR003650	Orange
IPR003656	Zinc finger, BED-type predicted
IPR003659	Plexin/semaphorin/integrin
IPR003877	SPla/RYanodine receptor SPRY
IPR003886	Nidogen, extracellular domain
IPR003890	MIF4G-like, type 3
IPR003892	Ubiquitin system component Cue
IPR003903	Ubiquitin interacting motif
IPR004012	RUN
IPR004018	RPEL repeat
IPR004043	LCCL
IPR004087	K Homology
IPR004092	Mbt repeat
IPR004148	BAR
IPR004155	PBS lyase HEAT-like repeat
IPR004170	WWE domain
IPR004172	L27
IPR004179	Sec63 domain
IPR004182	GRAM
IPR004274	NLI interacting factor
IPR005018	DOMON related
IPR005112	dDENN
IPR005113	uDENN
IPR005533	AMOP
IPR005607	BSD
IPR005824	KOW
IPR006021	Staphylococcal nuclease (SNase-like)
IPR006164	DNA helicase, ATP-dependent, Ku type
IPR006561	DZF

IPR006567	PUG domain
IPR006569	RNA polymerase II, large subunit, CTD
IPR006573	NEUZ
IPR006588	Peptide N glycanase, PAW domain
IPR006594	LisH dimerisation motif
IPR006597	Sel1-like
IPR006600	Pogo transposase / Cenp-B / PDC2, DNA-binding HTH domain
IPR006624	Beta-propeller repeat TECPR
IPR006634	TRAM/LAG1/CLN8 homology domain
IPR006652	Kelch repeat type 1
IPR007330	MIT
IPR007728	Pre-SET domain
IPR007842	HEPN
IPR008145	Guanylate kinase/L-type calcium channel
IPR008152	Clathrin adaptor, alpha/beta/gamma-adaptin, appendage, Ig-like subdomain
IPR008197	Whey acidic protein, 4-disulphide core
IPR008211	Laminin, N-terminal
IPR008936	Rho GTPase activation protein
IPR008942	ENTH/VHS
IPR008957	Fibronectin, type III-like fold
IPR008973	C2 calcium/lipid-binding domain, CaLB
IPR008974	TRAF-like
IPR008976	Lipase/lipooxygenase, PLAT/LH2
IPR008979	Galactose-binding domain-like
IPR008984	SMAD/FHA domain
IPR008985	Concanavalin A-like lectin/glucanase
IPR008993	TIMP-like, OB-fold
IPR009003	Serine/cysteine peptidase, trypsin-like
IPR009057	Homeodomain-like
IPR009060	UBA-like
IPR009071	High mobility group, superfamily
IPR009072	Histone-fold
IPR010895	CHRD
IPR010919	SAND-like
IPR010993	Sterile alpha motif homology
IPR011009	Protein kinase-like domain
IPR011024	Gamma-crystallin related
IPR011029	DEATH-like
IPR011129	Cold shock protein
IPR011333	BTB/POZ fold
IPR011705	BTB/Kelch-associated
IPR011991	Winged helix-turn-helix transcription repressor DNA-binding
IPR011993	Pleckstrin homology-type
IPR012337	Polynucleotidyl transferase, ribonuclease H fold
IPR012989	SEP domain
IPR013083	Zinc finger, RING/FYVE/PHD-type
IPR013106	Immunoglobulin V-set
IPR013517	FG-GAP
IPR013694	Vault protein inter-alpha-trypsin
IPR013723	Ataxin-1/HBP1 module (AXH)
IPR013806	Kringle-like fold
IPR013980	Seven Cysteines
IPR014012	Helicase/SANT-associated, DNA binding
IPR014021	Helicase, superfamily 1/2, ATP-binding domain

IPR014710	RmlC-like jelly roll fold
IPR014720	Double-stranded RNA-binding-like
IPR014756	Immunoglobulin E-set
IPR014853	Conserved-cysteine-rich domain
IPR015880	Zinc finger, C2H2-like
IPR015898	G-protein, gamma-like subunit
IPR015919	Cadherin-like
IPR015943	WD40/YVTN repeat-like-containing domain
IPR016024	Armadillo-type fold
IPR016035	Acyl transferase/acyl hydrolase/lysophospholipase
IPR016135	Ubiquitin-conjugating enzyme/RWD-like
IPR016137	Regulator of G protein signalling superfamily
IPR016146	Calponin-homology
IPR016177	DNA-binding, integrase-type
IPR016187	C-type lectin fold
IPR017448	Speract/scavenger receptor related
IPR017868	Filamin/ABP280 repeat-like
IPR017923	Transcription factor IIS, N-terminal
IPR017956	AT hook, DNA-binding motif
IPR018159	Spectrin/alpha-actinin
IPR018249	EF-HAND 2
IPR018392	Peptidoglycan-binding lysin domain
IPR018487	Hemopexin/matrixin, repeat
IPR018501	DDT domain superfamily
IPR018502	Annexin repeat
IPR019734	Tetratricopeptide repeat
IPR019955	Ubiquitin supergroup
IPR020067	Frizzled-like domain
IPR020850	GTPase effector domain, GED
IPR020858	Serum albumin-like
IPR020864	Membrane attack complex component/perforin (MACPF) domain

APPENDIX C

This appendix contains various data related to *Chapter-5 of this thesis.*

Table C1: List of domains belonging to the EGF/Laminin superfamily of the SCOP95 dataset of the Protein Benchmark Collection(Sonego et al., 2007).

Domain Name	SCOP ID	EGF category
1edmb_	g.3.11.1	H
1rfnb_	g.3.11.1	C
1pfx11	g.3.11.1	H
1pfx12	g.3.11.1	C
1danl1	g.3.11.1	H
1klil_	g.3.11.1	C
1g1sa2	g.3.11.1	H
1g1ta2	g.3.11.1	H
1kigl_	g.3.11.1	C
1autl1	g.3.11.1	H
1autl2	g.3.11.1	C
1q4ga2	g.3.11.1	H
1cvua2	g.3.11.1	H
1a3pa_	g.3.11.1	H
1egfa_	g.3.11.1	H
1gk5a_	g.3.11.1	H
1nqlb_	g.3.11.1	H
1p9ja_	g.3.11.1	H
1k36a_	g.3.11.1	H
1ioxa_	g.3.11.1	H
1xdtr_	g.3.11.1	H
1adxa_	g.3.11.1	C
1dx5i1	g.3.11.1	C
1dx5i2	g.3.11.1	C
1dx5i3	g.3.11.1	C
1lmja1	g.3.11.1	C
1lmja2	g.3.11.1	C
1nt0a3	g.3.11.1	C
1szba2	g.3.11.1	C
1apqa_	g.3.11.1	C
1tpga1	g.3.11.1	H
1hj7a1	g.3.11.1	C
1ijqa2	g.3.11.1	C
1toza1	g.3.11.1	H
1toza2	g.3.11.1	H
1toza3	g.3.11.1	H

Table C2: List of missense mutations found in *JAG1* and associated with Alagille syndrome

NT	19	L37S, I99S, L40P, V45L, N53D, R65M, F75S, C78S, L79H, C82R, C82Y, I120N, P123S, A127T, P129R, I152T, A155P, P163R
D&L	12	Y181N, R184C, R184G, R184H, R184L, C187S, C187Y, R203K, C220F, W224C, C229G, C229Y
EGF1	3	C234Y, R250G, G256S
EGF2	5	P269L, C271R, G274D, C284F, W288C
EGF3	0	
EGF4	0	
EGF5	1	G336R
EGF6	1	C439F
EGF7	0	
EGF8	1	N504S
EGF9	0	
EGF10	0	
EGF11	0	
EGF12	2	V690D, C693Y
EGF13	3	C714Y, C731S, C740R
EGF14	1	C753R
EGF15	0	
EGF16	0	
VWC	6	R889Q, C902S, H908Q, C911Y, S913R, L921P
JTXM	2	R937Q, 1055VR1056G
TM	0	
IC	1	R1213Q

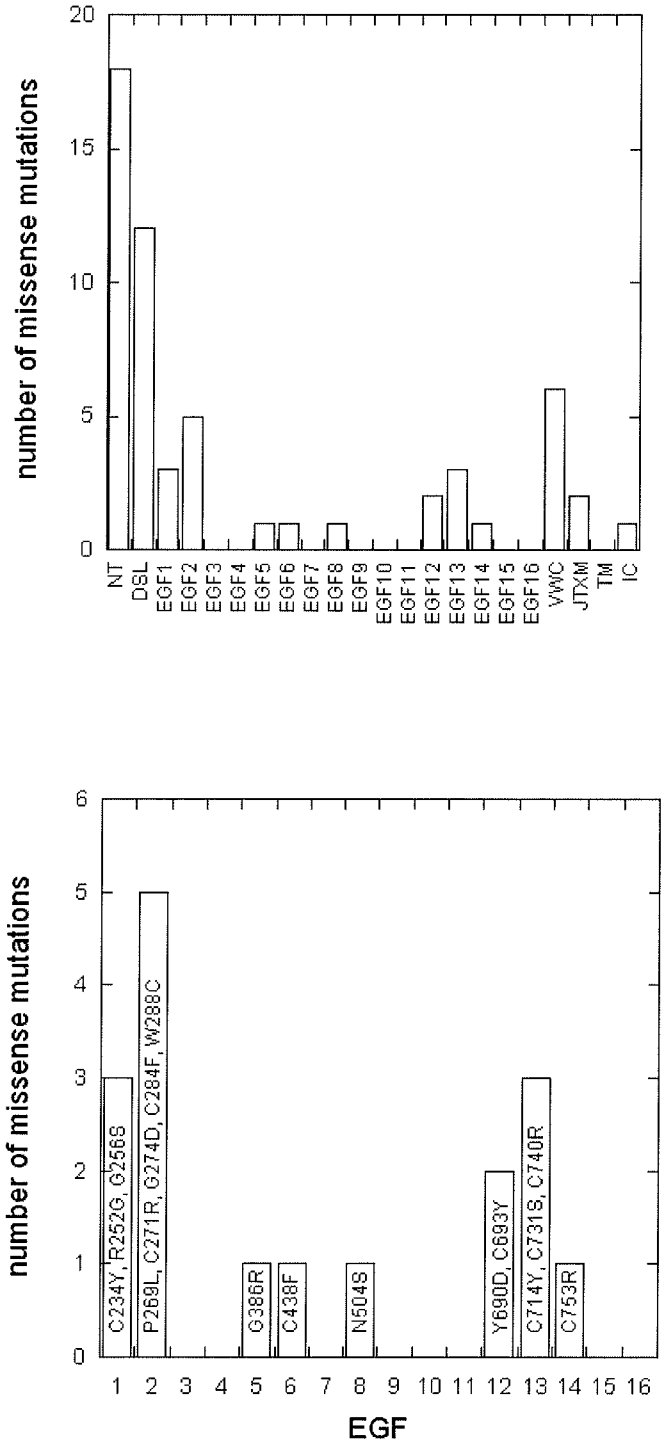
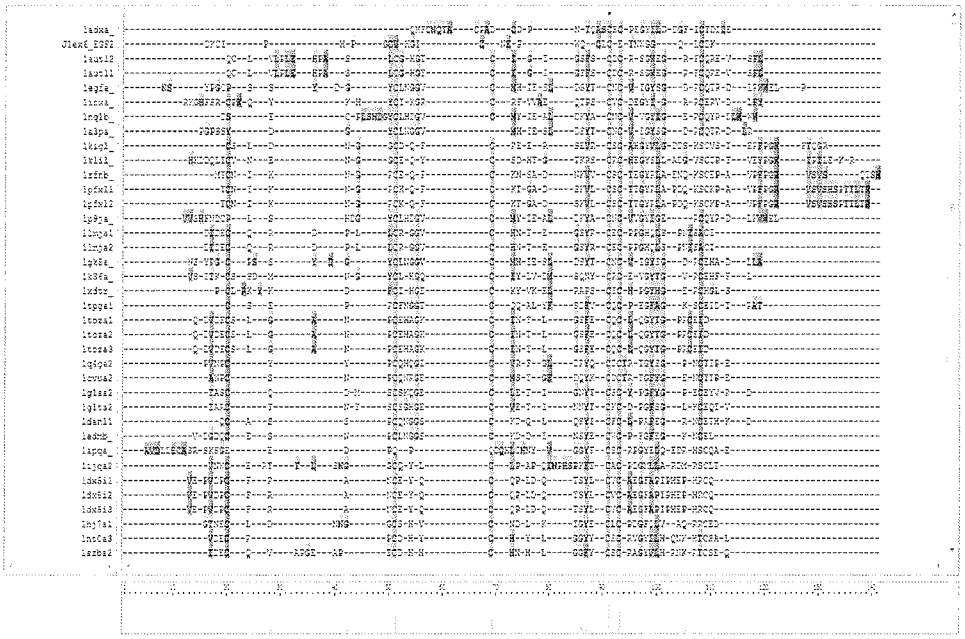


Figure C1: Plot of missense mutations found in *JAG1* and associated with Alagille syndrome



C2: Multiple sequence alignment based on the structural alignment of EGF2 from Jagged-1 (JAG1; PDB code: 2VJ2) and 56 domains belonging to the EGF/Laminin superfamily of the SCOP95 dataset of the Protein Benchmark Collection(Sonego et al., 2007).

Table C3: List of genes used for the multiple sequence alignmentof the polypeptides encoded by exon 6 of human *JAG1*

Gene	Exon No	Species	Ensemble ID	From	To
JAG1	6	Homo sapiens	ENSG00000101384	255	299
JAG1	6	Macaca mulatta	ENSMMUG00000018116	255	299
JAG1	6	Pan troglodytes	ENSPTRG00000013250	255	299
JAG1	6	Pongo pygmaeus	ENSPPYG00000010708	255	299
JAG1	7	Microcebus murinus	ENSMICG00000005608	255	299
JAG1	6	Rattus norvegicus	ENSRNOG00000007443	255	299
JAG1	6	Mus musculus	ENSMUSG00000027276	255	299
JAG1	10	Ochotona princeps	ENSOPRG00000008390	229	273
JAG1	8	Oryctolagus cuniculus	ENSOCUG00000002884	254	298
JAG1	6	Bos Taurus	ENSBTAG00000012817	255	299
JAG1	5	Equus caballus	ENSECAG00000012993	229	273
JAG1	7	Loxodonta Africana	ENSLAFG00000018468	256	300
JAG1	6	Canis familiaris	ENSCAFG00000005627	255	299

JAG1	6	Felis catus	ENSFCAG00000002195	255	299
JAG1	5	Myotis lucifugus	ENSMLUG00000009421	228	272
JAG1	14	Sorex araneus	ENSSARG00000005039	252	296
JAG1	6	Dasypus novemcinctus	ENSDNOG00000007593	255	299
JAG1	6	Monodelphis domestica	ENSMODG00000004910	254	298
JAG1	6	Echinops telfairi	ENSETEG00000001437	255	299
JAG1	6	Ornithorhynchus anatinus	ENSOANG00000008425	254	298
JAG1	5	Gallus gallus	ENSGALG00000009020	229	273
JAG1	6	Xenopus tropicalis	ENSXETG00000002340	255	299
JAG1	6	Oryzias latipes	ENSORLG00000000972	256	300
JAG1	6	Tetraodon nigroviridis	ENSTNIG00000016644	254	298
JAG1	4	Danio rerio	ENSDARG00000030289	129	173
JAG1	5	Gasterosteus aculeatus	ENSGACG00000004493	233	277
JAG2	6	Homo sapiens	ENSG00000184916	266	310
JAG2	4	Macaca mulatta	ENSMMUG00000001276	127	171
JAG2	4	Pongo pygmaeus	ENSPPYG00000006203	127	170
JAG2	6	Mus musculus	ENSMUSG00000002799	264	308
JAG2	5	Rattus norvegicus	ENSRNOG00000013927	220	264
JAG2	4	Cavia porcellus	ENSCPOG00000008419	127	171
JAG2	4	Equus caballus	ENSECAG00000006609	129	173
JAG2	1	Bos Taurus	ENSBTAG00000007319	1	45
JAG2	4	Canis familiaris	ENSCAFG00000018401	127	171
JAG2	3	Monodelphis domestica	ENSMODG00000014707	107	151
JAG2	3	Ornithorhynchus anatinus	ENSOANG00000007869	104	148
JAG2	5	Gallus gallus	ENSGALG00000011696	235	279
JAG2	6	Gasterosteus aculeatus	ENSGACG00000007522	259	303
JAG2	7	Takifugu rubripes	ENSTRUG00000000042	263	307
JAG2	1	Tetraodon nigroviridis	ENSTNIG00000012383	1	45
JAG2	6	Oryzias latipes	ENSORLG00000017877	259	303
JAG2	6	Danio rerio	ENSDARG00000021389	258	302
DLL1	6	Homo sapiens	ENSG00000198719	247	291
DLL1	5	Macaca mulatta	ENSMMUG00000021144	197	241
DLL1	6	Pongo pygmaeus	ENSPPYG00000017189	247	291
DLL1	6	Pan troglodytes	ENSPTRG00000018824	310	354
DLL1	6	Rattus norvegicus	ENSRNOG00000014667	246	290
DLL1	6	Mus musculus	ENSMUSG00000014773	246	290
DLL1	6	Oryctolagus cuniculus	ENSOCUG00000013290	246	290
DLL1	6	Bos Taurus	ENSBTAG00000031476	247	291
DLL1	6	Canis familiaris	ENSCAFG00000004094	197	241

DLL1	3	Felis catus	ENSFCAG00000004661	108	152
DLL1	7	Sorex araneus	ENSSARG00000007078	126	170
DLL1	6	Monodelphis domestica	ENSMODG00000005607	259	303
DLL1	2	Erinaceus europaeus	ENSEEUG00000007260	21	65
DLL1	4	Myotis lucifugus	ENSMLUG00000005071	130	174
DLL1	6	Gallus gallus	ENSGALG00000011182	254	298
DLL1	6	Xenopus tropicalis	ENSXETG00000022525	249	293
DLL1	5	Tupaia belangeri	ENSTBEG00000014817	131	175
DLL1	6	Takifugu rubripes	ENSTRUG00000006183	249	293
DLL1	6	Oryzias latipes	ENSORLG00000010606	249	293
DLL1	6	Gasterosteus aculeatus	ENSGACG00000016131	250	294
DLL4	6	Homo sapiens	ENSG00000128917	317	361
DLL4	7	Pan troglodytes	ENSPTRG00000006937	316	360
DLL4	6	Macaca mulatta	ENSMMUG00000014541	243	287
DLL4	7	Otolemur garnettii	ENSOGAG00000001215	244	288
DLL4	6	Microcebus murinus	ENSMICG00000005797	243	287
DLL4	6	Rattus norvegicus	ENSRNOG00000014011	244	288
DLL4	6	Mus musculus	ENSMUSG00000027314	244	288
DLL4	4	Cavia porcellus	ENSCPOG00000011383	111	155
DLL4	11	Ochotona princeps	ENSOPRG00000000813	318	362
DLL4	7	Oryctolagus cuniculus	ENSOCUG00000010756	243	287
DLL4	6	Equus caballus	ENSECAG00000013434	243	287
DLL4	6	Bos Taurus	ENSBTAG00000010361	243	287
DLL4	6	Canis familiaris	ENSCAFG00000009401	243	287
DLL4	9	Felis catus	ENSFCAG00000014721	316	360
DLL4	3	Myotis lucifugus	ENSMLUG00000004545	111	155
DLL4	6	Sorex araneus	ENSSARG00000005952	244	288
DLL4	6	Monodelphis domestica	ENSMODG00000000198	244	288
DLL4	3	Erinaceus europaeus	ENSEEUG00000014146	110	154
DLL4	16	Tupaia belangeri	ENSTBEG00000010805	240	284
DLL4	3	Ornithorhynchus anatinus	ENSOANG00000012601	61	105
DLL4	6	Gallus gallus	ENSGALG00000008514	243	287
DLL4	5	Xenopus tropicalis	ENSXETG00000021584	231	275
DLL4	5	Gasterosteus aculeatus	ENSGACG00000005896	218	262
DLL4	6	Danio rerio	ENSDARG00000070425	236	280
DLL4	8	Oryzias latipes	ENSORLG00000016743	250	294
DLL4	8	Takifugu rubripes	ENSTRUG00000012962	236	280
DLL4	6	Tetraodon nigroviridis	ENSTNIG00000010969	233	277
DLK1	3	Homo sapiens	ENSG00000185559	46	90

DLK1	3	Macaca mulatta	ENSMUSG00000040856	46	90
DLK1	3	Pongo pygmaeus	ENSPPYG00000006146	46	90
DLK1	1	Otolemur garnettii	ENSOGAG00000004875	1	44
DLK1	3	Rattus norvegicus	ENSRNOG00000019584	46	90
DLK1	3	Mus musculus	ENSMUSG00000040856	46	90
DLK1	3	Ochotona princeps	ENSOPRG00000012622	45	89
DLK1	3	Equus caballus	ENSECAG00000012122	46	90
DLK1	2	Canis familiaris	ENSCAFG00000017925	42	86
DLK1	2	Felis catus	ENSFCAG00000000458	22	66
DLK1	2	Ornithorhynchus anatinus	ENSOANG00000005731	27	71
DLK1	3	Gallus gallus	ENSGALG00000011244	51	95
DLK1	3	Tetraodon nigroviridis	ENSTNIG00000017282	46	90
DLK1	3	Takifugu rubripes	ENSTRUG00000009231	48	92
DLK1	3	Gasterosteus aculeatus	ENSGACG00000009397	41	85
DLK1	4	Oryzias latipes	ENSORLG00000014546	47	91
DLK2	3	Homo sapiens	ENSG00000171462	49	93
DLK2	3	Pan troglodytes	ENSPTRG00000018198	49	93
DLK2	4	Mus musculus	ENSMUSG00000047428	93	137
DLK2	3	Canis familiaris	ENSCAFG00000001858	49	93
DLK2	3	Bos Taurus	ENSBTAG00000005850	84	128
DLK2	4	Gallus gallus	ENSGALG00000010386	46	90

Table C4: List of disease-associated missense mutations found in EGF repeats

Swiss-Prot name	EGF	Mutation	Disease
CFC1_HUMAN	EGF-like	R112C	Visceral heterotaxy (MIM:605376)
CREL1_HUMAN	EGF-like1	P162A	AVSD2 susceptibility
CREL1_HUMAN	EGF-like2; calcium-binding(Potential)	T311I	AVSD2 susceptibility
CREL1_HUMAN	EGF-like2; calcium-binding(Potential)	R329C	AVSD2 susceptibility
CRUM1_HUMAN	EGF-like3	F144V	Leber congenital amaurosis type 8 (MIM:604210)
CRUM1_HUMAN	EGF-like4; calcium-binding(Potential)	A161V	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like4; calcium-binding(Potential)	V162M	Pigmented paravenous chorioretinal atrophy (MIM:172870)
CRUM1_HUMAN	EGF-like5; calcium-binding(Potential)	C195F	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like6; calcium-binding(Potential)	C250W	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like10	C423Y	Leber congenital amaurosis type 8 (MIM:604210)

CRUM1_HUMAN	EGF-like10; calcium-binding(Potential)	Y433C	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like11	C480G	Leber congenital amaurosis type 8 (MIM:604210)
CRUM1_HUMAN	EGF-like11	C480R	Leber congenital amaurosis type 8 (MIM:604210)
CRUM1_HUMAN	EGF-like12	C681Y	Leber congenital amaurosis type 8 (MIM:604210)
CRUM1_HUMAN	EGF-like13	C891G	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like13	N894S	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like13	C902Y	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like13	G919S	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like16; calcium-binding(Potential)	C1181R	Retinitis pigmentosa type 12 (MIM:600105)
CRUM1_HUMAN	EGF-like16; calcium-binding(Potential)	G1205R	Leber congenital amaurosis type 8 (MIM:604210)
CRUM1_HUMAN	EGF-like17	C1218F	Leber congenital amaurosis type 8 (MIM:604210)
CRUM1_HUMAN	EGF-like19; calcium-binding(Potential)	N1317H	Leber congenital amaurosis type 8 (MIM:604210)
CRUM1_HUMAN	EGF-like19; calcium-binding(Potential)	C1321S	Leber congenital amaurosis type 8 (MIM:604210)
CRUM2_HUMAN	EGF-like1	V97L	in a patient with Leber congenital amaurosis
CRUM2_HUMAN	EGF-like2; calcium-binding(Potential)	P116L	in a patient with Leber congenital amaurosis
CRUM2_HUMAN	EGF-like4; calcium-binding(Potential)	E187D	in a patient with Leber congenital amaurosis
CRUM2_HUMAN	EGF-like7; calcium-binding(Potential)	A351T	Retinitis pigmentosa (MIM:268000)
DLL3_HUMAN	EGF-like4	G385D	Autosomal recessive spondylocostal dysostosis type 1 (MIM:277300)
FA7_HUMAN	EGF-like1; calcium-binding(Potential)	S120P	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like1; calcium-binding(Potential)	C121F	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like1; calcium-binding(Potential)	L125P	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like1; calcium-binding(Potential)	Y128C	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like1; calcium-binding(Potential)	R139K	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like1; calcium-binding(Potential)	R139Q	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like1; calcium-binding(Potential)	R139W	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like2	C151S	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like2	E154K	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like2	G157C	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like2	G157S	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like2	G157V	Coagulation factor VII deficiency
FA7_HUMAN	EGF-like2	Q160R	Coagulation factor VII deficiency
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	D93G	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	Q96P	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	C97S	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	P101R	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	C102R	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	G106D	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	G106S	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	C108S	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	D110N	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	I112S	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-	N113K	Recessive X-linked hemophilia B (MIM:306900)

	binding(Potential)		
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	Y115C	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	C119F	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	C119R	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	E124K	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	G125E	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	G125R	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like1; calcium-binding(Potential)	G125V	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	C134Y	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	I136T	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	G139D	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	G139S	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	C155F	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	G160E	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	Q167H	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	S169C	Recessive X-linked hemophilia B (MIM:306900)
FA9_HUMAN	EGF-like2	C170F	Recessive X-linked hemophilia B (MIM:306900)
FBLN5_HUMAN	EGF-like3; calcium-binding(Potential)	V184L	Age-related macular degeneration type 3 (MIM:608895)
FBLN5_HUMAN	EGF-like3; calcium-binding(Potential)	R103Q	Age-related macular degeneration type 3 (MIM:608895)
FBLN5_HUMAN	EGF-like3; calcium-binding(Potential)	I169T	Age-related macular degeneration type 3 (MIM:608895)
FBLN5_HUMAN	EGF-like4; calcium-binding(Potential)	S227P	Autosomal recessive cutis laxa type I (MIM:219100)
FBN1_HUMAN	EGF-like1	C89F	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like1	C111R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like2	S115C	Isolated ectopia lentis (MIM:129600)
FBN1_HUMAN	EGF-like2	R122C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like2	C123Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like2	C129Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like3	C154S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like3	C166F	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like3	C166S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like3	C177R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like6	C476G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like7; calcium-binding	D490Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like7; calcium-binding	C504F	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like8; calcium-binding	C541Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like8; calcium-binding	R545C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like8; calcium-binding	N548I	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like8; calcium-binding	G560S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like8; calcium-binding	C570Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like9; calcium-binding	C587Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like9; calcium-binding	G592D	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like9; calcium-binding	C596Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like9; calcium-binding	C598W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like10; calcium-binding	R627C	Marfan syndrome (MIM:154700)

FBN1_HUMAN	EGF-like10; calcium-binding	C628K	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like10; calcium-binding	Y635C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like10; calcium-binding	R636I	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like10; calcium-binding	C637Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like10; calcium-binding	C652S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like11; calcium-binding	D723A	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like11; calcium-binding	D723V	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like11; calcium-binding	C734F	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like11; calcium-binding	Y746C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like11; calcium-binding	C748Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like11; calcium-binding	C750G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like12; calcium-binding	C776G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like12; calcium-binding	C776Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like12; calcium-binding	C781R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like12; calcium-binding	C781Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like13; calcium-binding	C816S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like13; calcium-binding	C832Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like14; calcium-binding	E913G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like14; calcium-binding	C921G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like14; calcium-binding	C926R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	K1043R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	C1044Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	I1048T	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	C1053R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	C1055G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	C1055W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	C1055Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	G1058D	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like15; calcium-binding	G1058GC	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like16; calcium-binding	D1072G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like16; calcium-binding	E1073K	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like16; calcium-binding	C1074R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like16; calcium-binding	C1086W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like16; calcium-binding	Y1101C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	D1113V	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	C1117G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-	C1117Y	Marfan syndrome (MIM:154700)

	binding		
FBN1_HUMAN	EGF-like17; calcium-binding	G1127S	in a mild form of inherited weakness of elastic tissue that predisposes to ascending aortic aneurysm and dissection later in life
FBN1_HUMAN	EGF-like17; calcium-binding	V1128I	in a patient with mitral valve prolapse
FBN1_HUMAN	EGF-like17; calcium-binding	C1129Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	N1131Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	R1137P	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	C1140Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	C1153S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	C1153R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like17; calcium-binding	C1153Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like18; calcium-binding	D1155N	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like18; calcium-binding	R1170H	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like18; calcium-binding	R1170G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like18; calcium-binding	C1171W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like18; calcium-binding	N1173K	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like19; calcium-binding	E1200G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like19; calcium-binding	Y1219C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like19; calcium-binding	C1223Y	Marfan syndrome (MIM:154700) & Shprintzen-Goldberg craniosynostosis syndrome (MIM:182212)
FBN1_HUMAN	EGF-like20; calcium-binding	C1242Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like20; calcium-binding	C1249S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like20; calcium-binding	Y1261C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like20; calcium-binding	Y1261D	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like20; calcium-binding	C1265R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like20; calcium-binding	C1278S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like21; calcium-binding	C1284G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like22; calcium-binding	E1325Q	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like22; calcium-binding	C1333S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like22; calcium-binding	C1333R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like22; calcium-binding	A1337P	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like22; calcium-binding	C1339Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like23; calcium-binding	E1366K	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like23; calcium-binding	C1374S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like23; calcium-binding	N1382S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like23; calcium-binding	C1389R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like23; calcium-binding	C1402R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like24; calcium-binding	D1404Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like24; calcium-	P1424A	Marfan syndrome (MIM:154700)

	binding		
FBN1_HUMAN	EGF-like24; calcium-binding	P1424S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like24; calcium-binding	C1429S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like25; calcium-binding	G1475E	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like25; calcium-binding	G1475S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like26; calcium-binding	C1513R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like27; calcium-binding	C1610G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like27; calcium-binding	C1631G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like28; calcium-binding	C1663R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like28; calcium-binding	C1663Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	C1770F	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	R1790P	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	R1790H	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	C1791R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	C1791Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	C1793W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	G1796E	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	C1806S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like29; calcium-binding	C1806Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like30; calcium-binding	C1833S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like30; calcium-binding	C1835Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like30; calcium-binding	P1837S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like31; calcium-binding	C1876Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like31; calcium-binding	T1887I	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	N1893K	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	C1895R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	C1900Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	H1909T	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	R1915S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	R1915C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	C1928G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	C1928R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like32; calcium-binding	C1928Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like33; calcium-binding	C1971Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like34; calcium-binding	C1977Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like34; calcium-binding	C1998Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like35; calcium-binding	C2038Y	Marfan syndrome (MIM:154700)

FBN1_HUMAN	EGF-like36; calcium-binding	D2127E	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like36; calcium-binding	C2131Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like36; calcium-binding	C2142Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like36; calcium-binding	N2144S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like36; calcium-binding	C2151W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like36; calcium-binding	P2154R	Isolated ectopia lentis (MIM:129600)
FBN1_HUMAN	EGF-like36; calcium-binding	A2160P	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like38; calcium-binding	C2221F	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like38; calcium-binding	C2221G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like38; calcium-binding	C2221S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like38; calcium-binding	N2223H	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like39; calcium-binding	C2251R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like39; calcium-binding	C2258R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like39; calcium-binding	I2269T	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like39; calcium-binding	R2282W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like40; calcium-binding	C2307S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like41; calcium-binding	C2406Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like41; calcium-binding	C2442W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like42; calcium-binding	E2447K	Isolated ectopia lentis (MIM:129600)
FBN1_HUMAN	EGF-like42; calcium-binding	Y2474C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like43; calcium-binding	C2489R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like43; calcium-binding	C2500R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like43; calcium-binding	C2500Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like43; calcium-binding	C2511R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like44; calcium-binding	C2535W	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like44; calcium-binding	G2536R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like45; calcium-binding	E2570K	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like45; calcium-binding	C2571R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like45; calcium-binding	C2581F	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like45; calcium-binding	I2585T	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like45; calcium-binding	C2592S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like45; calcium-binding	C2605R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like45; calcium-binding	C2605Y	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like46; calcium-binding	E2610K	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like46; calcium-binding	G2618R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like46; calcium-binding	H2623P	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like46; calcium-binding	N2624K	Marfan syndrome (MIM:154700)

FBN1_HUMAN	EGF-like46; calcium-binding	G2627R	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like46; calcium-binding	Y2629C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like47; calcium-binding	C2652G	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like47; calcium-binding	C2663S	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like47; calcium-binding	G2668C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like47; calcium-binding	R2680C	Marfan syndrome (MIM:154700)
FBN1_HUMAN	EGF-like47; calcium-binding	R2680P	Marfan syndrome (MIM:154700)
FBN2_HUMAN	EGF-like16; calcium-binding	D1114H	Congenital contractural arachnodactyly (MIM:121050)
FBN2_HUMAN	EGF-like16; calcium-binding	C1141F	Congenital contractural arachnodactyly (MIM:121050)
FBN2_HUMAN	EGF-like19; calcium-binding	C1252W	Congenital contractural arachnodactyly (MIM:121050)
FBN2_HUMAN	EGF-like19; calcium-binding	C1252Y	Congenital contractural arachnodactyly (MIM:121050)
FBN2_HUMAN	EGF-like23; calcium-binding	C1433S	Congenital contractural arachnodactyly (MIM:121050)
HMCN1_HUMAN	EGF-like6; calcium-binding(Potential)	Q5345R	Age-related macular degeneration type 1 (MIM:603075)
JAG1_HUMAN	EGF-like4; calcium-binding(Potential)	G386R	Alagille syndrome type 1(MIM:118450)
JAG1_HUMAN	EGF-like5; calcium-binding(Potential)	C438F	Alagille syndrome type 1(MIM:118450)
JAG1_HUMAN	EGF-like7; calcium-binding(Potential)	N504S	Alagille syndrome type 1(MIM:118450)
JAG1_HUMAN	EGF-like11; calcium-binding(Potential)	Y690D	in biliary atresia; extrahepatic
JAG1_HUMAN	EGF-like11; calcium-binding(Potential)	C693Y	Alagille syndrome type 1(MIM:118450)
JAG1_HUMAN	EGF-like12	C714Y	Alagille syndrome type 1(MIM:118450)
JAG1_HUMAN	EGF-like12	C731S	Alagille syndrome type 1(MIM:118450)
JAG1_HUMAN	EGF-like12	C740R	Alagille syndrome type 1(MIM:118450)
JAG1_HUMAN	EGF-like13	C753R	Alagille syndrome type 1(MIM:118450)
LDLR_HUMAN	EGF-like1	C318Y	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like1	H327Y	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like1	C329Y	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like1	C338S	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like1	D342N	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like2; calcium-binding(Potential)	D356Y	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like2; calcium-binding(Potential)	Q366R	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like2; calcium-binding(Potential)	C368R	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like2; calcium-binding(Potential)	C379Y	Familial hypercholesterolemia (MIM:143890)
LDLR_HUMAN	EGF-like3	D700E	Familial hypercholesterolemia (MIM:143890)
LRP6_HUMAN	EGF-like2	R611C	Autosomal dominant coronary artery disease type 2 (MIM:610947)
MATN3_HUMAN	EGF-like1	T303M	Susceptibility to hand osteoarthritis (MIM:607850)
MATN3_HUMAN	EGF-like1	C304S	Spondyloepimetaphyseal dysplasia bowed-legs type (MIM:608728)
NOTC2_HUMAN	EGF-like11; calcium-binding(Potential)	C444Y	Alagille syndrome type 2 (MIM:610205)
NOTC3_HUMAN	EGF-like1	C49Y	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like1	W71C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like2	R90C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like2	R110C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)

NOTC3_HUMAN	EGF-like3	R133C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like3	R141C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like3	C146R	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like3	R153C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like4; calcium-binding(Potential)	R169C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like4; calcium-binding(Potential)	G171S	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like4; calcium-binding(Potential)	G171C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like4; calcium-binding(Potential)	R182C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like4; calcium-binding(Potential)	C185R	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like5	C212S	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like5	R213W	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like5	C222G	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like5	C224Y	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like6; calcium-binding(Potential)	Y258C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like13; calcium-binding(Potential)	C542Y	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like14; calcium-binding(Potential)	R558C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like14; calcium-binding(Potential)	R578C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like18	R728C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like25	R985C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like26	R1006C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like26	R1031C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like31	R1231C	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
NOTC3_HUMAN	EGF-like32	C1261R	Cerebral autosomal dominant arteriopathy with subcortical infarcts & leukoencephalopathy(MIM:125310)
PERT_HUMAN	EGF-like; calcium-binding(Potential)	D796Y	Congenital hypothyroidism due to dyshormonogenesis type 2A (MIM:274500)
PERT_HUMAN	EGF-like; calcium-binding(Potential)	E799K	Congenital hypothyroidism due to dyshormonogenesis type 2A (MIM:274500)
PERT_HUMAN	EGF-like; calcium-binding(Potential)	C808R	Congenital hypothyroidism due to dyshormonogenesis type 2A (MIM:274500)
PERT_HUMAN	EGF-like; calcium-binding(Potential)	V839I	Congenital hypothyroidism due to dyshormonogenesis type 2A (MIM:274500)
PROC_HUMAN	EGF-like1	H108N	Protein C deficiency (MIM:176860)
PROC_HUMAN	EGF-like1	G109R	Protein C deficiency (MIM:176860)
PROC_HUMAN	EGF-like1	G114R	Protein C deficiency (MIM:176860)
PROC_HUMAN	EGF-like1	F118L	Protein C deficiency (MIM:176860)
PROC_HUMAN	EGF-like2	G145R	Protein C deficiency (MIM:176860)
PROC_HUMAN	EGF-like2	C147Y	Protein C deficiency (MIM:176860)
PROC_HUMAN	EGF-like2	H149P	Protein C deficiency (MIM:176860)
PROC_HUMAN	EGF-like2	S161R	Protein C deficiency (MIM:176860)
PROS_HUMAN	EGF-like2; calcium-binding(Potential)	K196E	Protein S deficiency (MIM:176880)
PROS_HUMAN	EGF-like4; calcium-binding(Potential)	N258S	Protein S deficiency (MIM:176880)
UROM_HUMAN	EGF-like2; calcium-binding(Potential)	C77Y	Familial juvenile hyperuricemic nephropathy (MIM:162000)
UROM_HUMAN	EGF-like2; calcium-binding(Potential)	G103C	Medullary cystic kidney disease 2 (MIM:603860)
UROM_HUMAN	EGF-like3; calcium-binding(Potential)	C126R	Familial juvenile hyperuricemic nephropathy (MIM:162000)

UROM_HUMAN	EGF-like3; calcium-binding(Potential)	N128S	Familial juvenile hyperuricemic nephropathy (MIM:162000)
UROM_HUMAN	EGF-like3; calcium-binding(Potential)	C148W	Medullary cystic kidney disease 2 (MIM:603860) OR Familial juvenile hyperuricemic nephropathy (MIM:162000)
UROM_HUMAN	EGF-like3; calcium-binding(Potential)	C148Y	Familial juvenile hyperuricemic nephropathy (MIM:162000)

Table C5. List of neutral mutations found in EGF repeats.

Swiss-Prot name	EGF	Mutation
CREL2_HUMAN	EGF-like 2; calcium-binding (Potential)	S295A
CREL2_HUMAN	EGF-like 2; calcium-binding (Potential)	E325G
CRUM2_HUMAN	EGF-like 3; calcium-binding (Potential)	G159A
DLL3_HUMAN	EGF-like 1	L218P
DNER_HUMAN	EGF-like 6	P433L
EGFL6_HUMAN	EGF-like 1	E66K
EGFL6_HUMAN	EGF-like 3	R164C
EGFL7_HUMAN	EGF-like 2; calcium-binding (Potential)	V153I
EGFLA_HUMAN	EGF-like 2	H576N
EGF_HUMAN	EGF-like 3	R431K
EGF_HUMAN	EGF-like 6	M842T
EGF_HUMAN	EGF-like 9	D981E
EMR1_HUMAN	EGF-like 1	A57T
EMR1_HUMAN	EGF-like 3; calcium-binding (Potential)	S140R
EMR1_HUMAN	EGF-like 4; calcium-binding (Potential)	D174N
EMR1_HUMAN	EGF-like 5; calcium-binding (Potential)	N254S
FA10_HUMAN	EGF-like 2	A152T
FA12_HUMAN	EGF-like 2	P207A
FAT3_HUMAN	EGF-like 1	S3812G
FAT4_HUMAN	EGF-like 1	K3828E
FAT4_HUMAN	EGF-like 2; calcium-binding (Potential)	S3873N
FBLN4_HUMAN	EGF-like 5; calcium-binding (Potential)	I259V
FBN1_HUMAN	EGF-like 17; calcium-binding	P1148A
FBN3_HUMAN	EGF-like 19; calcium-binding	S1293N
FBN3_HUMAN	EGF-like 28	R1806Q
FBN3_HUMAN	EGF-like 29; calcium-binding	N1869K
FBN3_HUMAN	EGF-like 30; calcium-binding	L1904P
FBN3_HUMAN	EGF-like 31; calcium-binding	P1958H
FBN3_HUMAN	EGF-like 44; calcium-binding	D2610E
HABP2_HUMAN	EGF-like 1	V90I
HEG1_HUMAN	EGF-like 2; calcium-binding (Potential)	M1039T
JAG2_HUMAN	EGF-like 8	E501K
LDLR_HUMAN	EGF-like 2; calcium-binding (Potential)	A391T
LRP1_HUMAN	EGF-like 2; calcium-binding (Potential)	N166D
LRP2_HUMAN	EGF-like 3	G669D
LTBP4_HUMAN	EGF-like 5; calcium-binding (Potential)	R635G
LTBP4_HUMAN	EGF-like 6; calcium-binding (Potential)	P679A
LTBP4_HUMAN	EGF-like 8; calcium-binding (Potential)	T787A
LYAM3_HUMAN	EGF-like	G179R
MASP2_HUMAN	EGF-like; calcium-binding	H155R
MATN2_HUMAN	EGF-like 3	E356K
NID1_HUMAN	EGF-like 2	Q669R
NOTC3_HUMAN	EGF-like 12; calcium-binding (Potential)	P496L
NOTC3_HUMAN	EGF-like 30; calcium-binding (Potential)	V1183M
NOTC4_HUMAN	EGF-like 5; calcium-binding (Potential)	P204L
NOTC4_HUMAN	EGF-like 5; calcium-binding (Potential)	P206L
NOTC4_HUMAN	EGF-like 6	S244L
NOTC4_HUMAN	EGF-like 6	D272G
NOTC4_HUMAN	EGF-like 8; calcium-binding (Potential)	T320A
NOTC4_HUMAN	EGF-like 13; calcium-binding (Potential)	G534S

NOTC4_HUMAN	EGF-like 22	K851R
NPNT_HUMAN	EGF-like 3	Q159H
NPNT_HUMAN	EGF-like 5; calcium-binding (Potential)	V234I
NT2NL_HUMAN	EGF-like 3	S67P
NT2NL_HUMAN	EGF-like 5; calcium-binding (Potential)	T158I
NT2NL_HUMAN	EGF-like 6	T196S
SCUB1_HUMAN	EGF-like 9; calcium-binding (Potential)	G398R
SLIT3_HUMAN	EGF-like 2	E994G
SVEP1_HUMAN	EGF-like 4; calcium-binding (Potential)	L1330M
SVEP1_HUMAN	EGF-like 6; calcium-binding (Potential)	K1416Q
SVEP1_HUMAN	EGF-like 9	T3562M
TENA_HUMAN	EGF-like 13	Q539R
TENA_HUMAN	EGF-like 15	V605I
TRBM_HUMAN	EGF-like 6; calcium-binding (Potential)	A473V
TSP4_HUMAN	EGF-like 3; calcium-binding (Potential)	A387P
TSP4_HUMAN	EGF-like 4	A420V
VLDLR_HUMAN	EGF-like 1	E379K