Always program as if the person who will be maintaining your program is a violent psychopath that knows where you live.

– Martin Golding.

University of Alberta

PREDICTING HOMOLOGOUS SIGNALING PATHWAYS USING MACHINE LEARNING

by

Babak Bostan

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2009
Edmonton, Alberta

Examining Committee

Russell Greiner, Computing Science

Duane Szafron, Computing Science

Warren Gallin, Cell Biology

Robert Holte, Computing Science

To the best parents ever,

You are my life.

Abstract

Understanding biochemical reactions inside cells of individual organisms is a key factor for improving our biological knowledge. Signaling pathways provide a road map for a wide range of these chemical reactions that convert one signal or stimulus into another. In general, each signaling pathway in a cell involves many different proteins, each with one or more specific roles that help to amplify a relatively small stimulus into an effective response. Since proteins are essential components of a cell's activities, it is important to understand how they work – and in particular, to determine which of specie's proteins participate in each role. Experimentally determining this mapping of proteins to roles is difficult and time consuming. Fortunately, many individual pathways have been annotated for some species, and the pathways of other species can often be inferred using protein homology and the protein properties.

We present an automatic approach, PSP, that uses the signaling pathways in well-studied species to predict which proteins will serve which roles in less-studied species. Our machine learning approach creates a predictor that achieves a generalization F-measure of 78.2% when predicting protein roles in 11 different pathways across 14 different species. We also describe an evaluation method that suggests our prediction might be more accurate than this F-measure. This method makes predictions based on historical data, then evaluates the prediction based on new data that include more recent annotations of the proteins. This process revealed that our historical predictor was correct about many predictions that were considered to be wrong based on the historical data.

Acknowledgements

I would like to thank my supervisors Russell Greiner and Duane Szafron for their great support, valuable guidance, and helpful comments. I also gratefully acknowledge the insights from my colleagues in the Proteome Analyst team, including Paul Lu, Alex Ebhardt, Tadaaki Hiruki, Yifeng Liu and Chris Chen. This work was supported by Alberta Ingenuity Centre for Machine Learning (AICML) and Natural Sciences and Engineering Research Council of Canada (NSERC).

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Understanding chemical processes in living organisms is one of biology's main challenges. Even though the core chemistry of DNA[1] sequencing has not changed significantly in recent years, the advent of low-cost high-throughput DNA sequencing was a huge step toward this goal (Hall, 2007). These methods have significantly accelerated biological research and discovery by providing us with a huge source of information that can help us solve the complex puzzle of life. However, interpreting the vast amount of data represented by species' genomes cannot be done easily, due to their size and complexity, without efficient and accurate computational techniques. This dissertation presents a novel automated computational technique that effectively assists biologists to find another piece of the puzzle. Section 1.1 describes the biochemical pathways we are using in our task, which is given in Section 1.2.

1.1 Biochemical pathways

The cell is the basic functional unit and building block of all living organisms (Campbell and Reece, 2001). Each cell receives nutrients and information signals and carries out specialized functions. These functions, including growth, cell division and protein synthesis, require compounds (small molecules) and proteins. Proteins perform these functions through a series of simple interactions with other proteins and small-molecule substrates. Many of these simple interactions between

[1] See Campbell and Reece (2001) for description of basic biology terms.

proteins or compounds transforms one set of molecules into another, these are called *chemical reactions* (Campbell and Reece, 2001). A network of these chemical reactions when viewed together, forms a *pathway*. Many chemicals may be involved in each pathway, and the reactions between these chemicals can be quite elaborate. To better understand and accommodate research on pathways, biologists divide the pathways into different categories, which have different properties.

1.1.1 Metabolic pathway

Metabolism is managing the material and energy resources of the cell. It involves a step-by-step modification of the initial molecules to shape them into other products. This modification corresponds to a sequence of reactions that is called a *metabolic pathway*. In this process each reaction is controlled by some proteins, that are enzymes – *i.e.*, that accelerate a reaction without being consumed (Campbell and Reece, 2001). This property of the enzymes enables the cell to manage its chemical activities. When a cell needs more energy it uses some enzymes to release energy by breaking down complex organic matters to simpler compounds. By using other enzymes a cell can use the energy to construct components of the cell (Wallace *et al.*, 2001; Campbell and Reece, 2001).

Figure 1.1 shows a small sample of a metabolic pathway, synthesis and degradation of ketone bodies (KEGG, 2009b). Here rectangles refer to enzymes and circles refer to small molecules. This graph shows both synthesis and degradation of ketone bodies, which are consuming and releasing energy respectively. For example, this graph shows that Acetoacetate can be converted to Acetyl-CoA when the enzymes labeled 2.8.3.5 and 2.3.1.9 appear in the relevant environment, in sufficient quantity. This degradative process, which normally happens in the brain and heart when insufficient glucose is available, can produce energy. On the other hand, when the body breaks down fatty acids, it converts Acetyl-CoA to Acetoacetate and (R)-3-Hydroxybutyrate (as by-products). This process, which needs enzymes 2.3.3.10, 4.1.3.4 and 1.1.1.30, consumes energy. The concentration of enzymes can control which one of these processes is active at any time in a metabolic pathway.

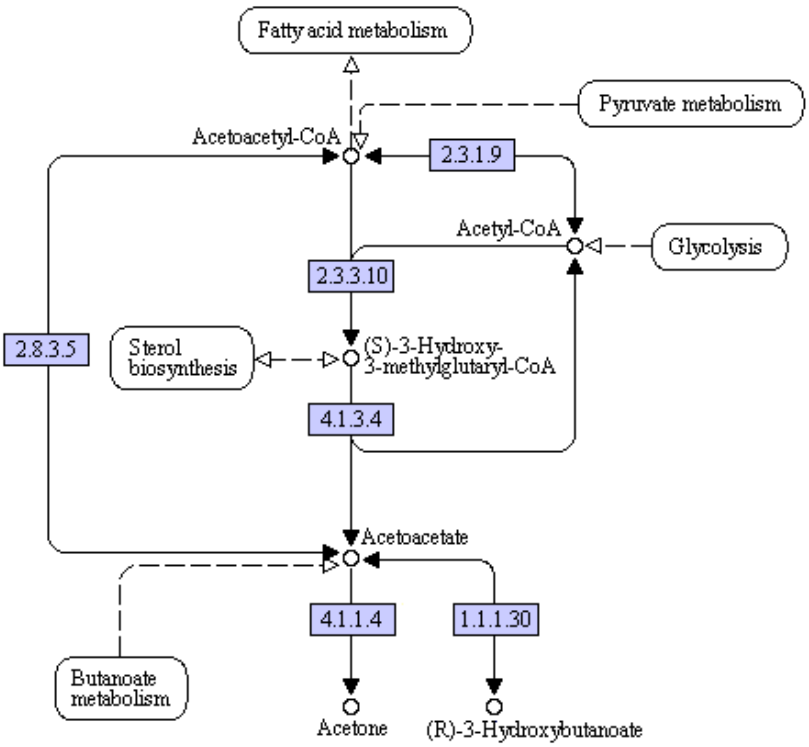

Figure 1.1: Synthesis and degradation of ketone bodies. Enzyme labels in this picture are examples of an enzyme classification (EC) number (Moss, 2009). The arcs show the reactions. (This figure is taken from KEGG, 2009b)

1.1.2 Signaling pathway

A series of chemical reactions can also enable communication between different parts of a cell or between one cell and another. These intracellular and intercellular communications, which play a crucial role in the life of a cell, are called cell signaling. These chemical reactions inside the cell can be connected (as a directed graph) to form a complicated network of reactions. This network of reactions is activated by receptors on the surface of the cell and includes secondary messenger molecules, proteins and other compounds.

This signaling process involves three stages: reception, transduction, and response. In the reception stage, the receptor proteins of a cell recognize a signal molecule. These signal molecules, which are usually too large to pass through the plasma membrane, normally bind to the receptor protein and change its shape. The

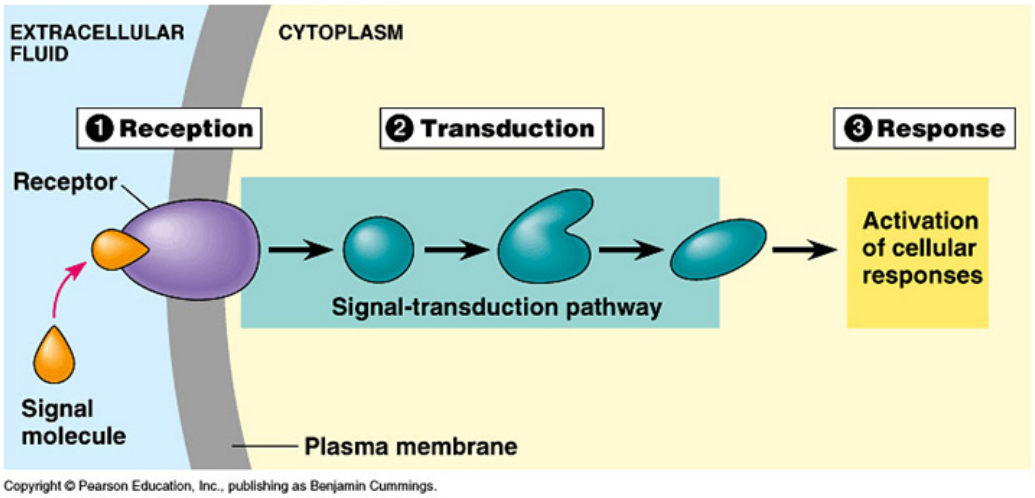

Figure 1.2: Cell signaling stages: 1- Reception 2- Transduction 3- Response. (This figure is taken from Campbell and Reece, 2001)

protein's changed shape starts a transduction stage by interacting with another cellular molecule or by causing the aggregation of two or more receptor molecules. The transduction stage sometimes occurs in a single step but more often requires a sequence of changes in a series of different molecules — a signal-transduction pathway. This stage is like falling dominoes, where the signal-activated receptor activates another protein, which activates another molecule, and so on. Multiple steps in this stage can amplify the signal, since some of the molecules in each step can activate multiple molecules, resulting in a large number of activated proteins at the end of the pathway. Ultimately, a signal-transduction pathway leads to the regulation of one or more cellular activities, which is the response stage (Campbell and Reece, 2001). Figure 1.2 shows these three stages. In this figure we can see how a signal molecule can activate a receptor in the plasma membrane of the cell and how this reception can start a sequence of changes inside the cell and finally cause cellular responses.

We refer to the sequence of changes that are involved in cell signaling as a *signaling pathway*. Figure 1.3 provides an example of the Notch signaling pathway in *H. sapiens*. Here one of the proteins that is involved in this pathway, the Notch protein, acts like a trigger (reception stage). When two cells make a direct cell-to-cell contact, this protein starts a series of reactions inside the cell (transduction stage)

that creates a signal inside the nucleus to alter gene expression (response stage). This mechanism controls multiple cell differentiation processes during embryonic and adult life.

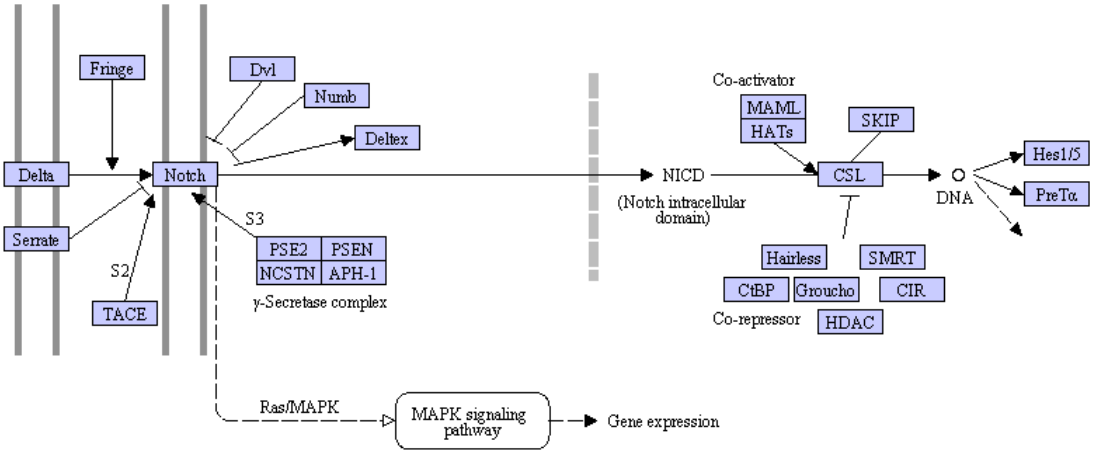

Figure 1.3: Notch signaling pathway (This figure is taken from KEGG, 2009a)

Understanding signaling pathways can help us to discover previously unknown aspects of cellular life and may provide useful information for improving health. For instance, many known diseases, including diabetes and many cancers, are caused by cellular abnormalities linked to signaling pathway malfunctions (Seifter et al., 2005). A better understanding of signaling pathways could lead to better treatments for these afflictions by aiding in drug design and development of other pathway interventions.

1.2 The problem

Unfortunately, experimental approaches for investigating signaling pathways are extremely difficult due to the large number of proteins in each cell and a lack of information about which proteins are involved in each pathway and the roles that these proteins play in signaling. It would be very useful to have an automated system that can ease the process of discovering new signaling proteins and predicting their role. However, because of the complex nature and unknown aspects of biolog-

ical problems, implementing such a system can be very challenging.

In this dissertation, we provide an automated system that learns to predict which proteins in a less studied species play each role by using signaling pathways in well studied organisms. This dissertation makes following research contributions:

1. It shows that our Predict Signaling Pathway (PSP) system can accurately predict signaling pathways in less studied species based on signaling pathways in well-known organisms.

2. It demonstrates that various protein's properties can improve the accuracy of prediction over the basic methods.

3. It provides empirical results to show that our PSP system can achieve a generalization F-measure of 78.2 when predicting protein roles in 11 different pathways across 14 different species.

Chapter 2 briefly describes the classifiers along with the features that are used in our system. Chapter 3 describes the prediction process. Chapter 4 provides our evaluation method and experimental results. Finally, Chapter 5 gives a brief conclusion. The rest of this chapter describes some related work. (Many of these results has been published as Bostan *et al.* (2009).)

1.3 Related work

Many research projects use computational approaches to tackle biological system problems. Several research projects tackle biological system problems using information retrieval, entity recognition or information extraction. For example, the best-known biomedical information retrieval (IR) system, PubMed, uses IR methods to allow the user to retrieve all documents that contain certain combinations of terms. Information Hyperlinked over Proteins (IHOP) (Hoffmann and Valencia, 2004) is a web-based tool that allows the user to browse PubMed abstracts on the basis of the biomedical entities that they mention, using information extraction to automatically extract structured information from unstructured documents. GENIA

(Ohta *et al.*, 2002) is another system that uses text mining in the context of biological problems. This system uses natural-language processing to automatically extract information from abstracts in PubMed.

Our PSP, however, builds a classifier to tackle the problem. Computational approaches based on building classifiers, are very popular for addressing complex biological challenges. For example, Furey *et al.* (2000) used methods based on Support Vector Machines (SVMs) to identify genes useful in cancer diagnosis from micro-arrays. Guyon *et al.* (2002) used SVMs and recursive feature elimination to identify a small subset of genes from broad patterns of gene expression data, based on DNA micro-arrays, to produce a classifier suitable for genetic diagnosis and drug discovery. Ding and Dubchak (2001) predicted the structure of proteins using Neural Networks and SVMs. Several research groups including Park and Kanehisa (2003) and Lu *et al.* (2004) have used machine learning to produce classifiers that can predict the subcellular location of proteins. Many research projects are focused on predicting protein function. For example, Hishigaki *et al.* (2001) predict protein function from protein-protein interaction data, by predicting the function of a protein based on the function of its interacting protein – both direct and indirect interaction. Vázquez *et al.* (2003) use another approach to predict protein function using protein-protein interaction networks. They use sequence similarity to cluster the proteins into groups by minimizing the number of protein interactions among different functional categories. However, these projects can not be used for pathway prediction without use of expert biologists mainly because of their error rate and variety of functional categories.

Despite a growing volume of research that uses computational techniques to solve biological problems, there is a dearth of research about applying computational techniques to help us better understand *signaling pathways*. Some results have been published on using computational techniques to understand *metabolic pathways* – *e.g.*, Schilling *et al.* (1999) provide a good survey of this work up to 1999. There have only been a few recent contributions; *e.g.*, Ma and Zeng (2003) find shortest paths between metabolites, Pireddu *et al.* (2006) use machine learning to predict the role of proteins in *metabolic* pathways and the MetaCyc group (Caspi

et al., 2008) provide two databases of organism-specific metabolic pathways: some experimentally elucidated and some predicted (BioCyc).

However, none of these projects focus on *signaling pathways*. Previous work on signaling has been restricted to predicting *individual* signaling peptides and sorting signals (Nielsen *et al.*, 1999) or the effects of *single* genes on the overall functioning of signaling networks (Craven, 2002) or predicting *protein-protein interactions* (Yaffe *et al.*, 2001) that can be used to predict signaling pathways. While there are also many results that deal with a particular pathway or species, they each focus on one particular pathway or a subset of one pathway, and are not expandable to the other pathways or other parts of pathways – *e.g.*, Kim *et al.* (2004). The Panther group (Thomas *et al.*, 2003) also provide a database of pathways, both metabolic and signaling. Their system is a collection of proteins gathered by human experts on which they can use a Hidden Markov Model (HMM) to classify the functionality of novel protein sequences. While their computational approach has high coverage over mammalian protein-coding genes, it is not clear how to measure its accuracy because they do not make their specific prediction available, which makes it difficult to evaluate their system or compare it with other approaches.

The Predict Signaling Pathway (PSP) system presented in this dissertation uses approaches similar to Pireddu *et al.* (2006), but in the more complex domain of predicting *signaling* pathways. Both systems use homologous pathways and predict individual nodes in the graph structure. However, while Pireddu *et al.* use BLAST and HMM to predict which proteins (enzymes) will appear in *metabolic pathways*, our PSP uses a very different technique, using machine-learned classifiers, to predict proteins in *signaling pathways*. Moreover, we employ a "retrospective" analysis that suggests that PSP's predictions are highly effective in predicting proteins that have not yet been experimentally verified.

Fröhlich *et al.* (2008) provides another prediction system that predicts pathways using protein domains, which are each a subsequence of the protein's peptide string that is considered a functional unit that can evolve or act independently of the rest of the protein sequence. Fröhlich *et al.* predict whether a gene is involved in a particular pathway or not, but they do not provide a way to predict the specific *role* of each

protein within the pathways. While their system could determine whether a gene belongs to a subset of roles in these *signaling pathways*, they limit their predictions to annotated human genes that have domain signatures. Testing their system on 10 out of the 11 *human* signaling pathways available in the KEGG database (Kanehisa *et al.*, 2008), they obtained an F-measure of approximately 80%. Our PSP system, on the other hand, can predict pathways using the whole proteome of *any* species and can predict the exact role of each protein within *arbitrary* signaling pathways, with an overall F-measure of 90.4% over all 11 human signaling pathways available in KEGG.

Chapter 2

Background

Data classification techniques map each data instance to a class based on the values of its features. Finding the best mapping technique is the core of each classification system. Section 2.1 briefly describes the classifiers that are used in our system. Section 2.2 provides a method to evaluate the accuracy of the classifiers. The rest of this chapter (Section 2.3 through Section 2.6) briefly describes the features that are used in our system. A complete description of how these features are used in our system is given later in the dissertation (see Chapter 4).

2.1 Classifiers

The problem of finding useful patterns in the input data is a fundamental challenge. Machine learning's major focus is addressing this problem by providing a group of tools that automatically learn to recognize and use these patterns. A classifier is a mapping from each input sample to a finite number of discrete categories, which can be learned using a learner.

Learners can be divided into different categories, such as unsupervised and supervised. Unsupervised learning seeks to determine how the data is organized using only unlabeled samples. This approach is especially useful when the classes are unknown (or there is no class at all) or there is not enough labeled data. For example, in sequence analysis, clustering is used to group homologous sequences into gene families. On the other hand, supervised learners produce a classifier using pairs of input samples and correct class labels, called the training data.

The goal of supervised learning is to predict the best class for any valid input instance, after learning on training examples. To achieve this goal, the prediction process is divided into two different phases: training and testing. During the training phase, the learner develops a predictive model from the training data. In the subsequent testing phase, this trained predictive model is used to classify (that is, predict a label for) an unlabeled instance. In general, the classifier can be evaluated based on the number of correctly classifier samples and misclassified samples.

The goal of learning is to produce a predictive model that performs well on *unseen data* (*i.e.*, data that was not in the training sample). Hence, the predictive model should be trained in such a way that it avoids predictions based on specific properties of the training data. Otherwise, the performance on the training samples could be artificially high compared to the performance on unseen data. This problem, called overfitting, often occurs when the model is very complex in relation to the amount of available training data.

To accurately estimate the performance of the predictor, the labeled data can be divided into two exclusive subsets, a training-set and a test-set. The predictor then can be trained on all data to produce a predictor. An evaluation method then can be used to produce an estimate of the quality of that predictor.

There are many different classifiers that can be applied to each problem. However, determining a suitable classifier for a given problem is still more of an art than a science. Two of the most widely used classifiers are k-nearest neighbor and support vector machines. The next two sub-sections briefly describe these two classifiers because PSP uses support vector machines, and we later see k-nearest neighbor as another alternative to show how well PSPworks.

2.1.1 K-Nearest Neighbor

K-nearest neighbor is one of the simplest classifiers. This classifier classifies each test instance by a majority vote of its k nearest neighbors (Bishop, 2006). For example, as shown in Figure 2.1, by choosing $k = 3$ the new instance (shown as an octagon) would be classified as a circle.

Choosing a correct distance metric is one of the challenges when k-nearest

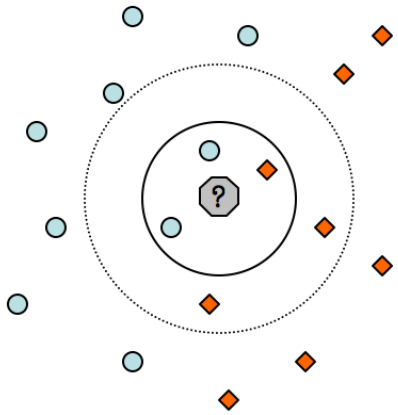

Figure 2.1: Using k-nearest neighbor to classify an unlabeled instance (octagon). Choosing $k = 3$ assigns circle to the unlabeled instance while choosing $k = 5$ classify this instance as a rhombus.

neighbor is being used. Normally, Euclidean distance can be used as the distance measure when the training examples are vectors in a multidimensional feature space. However in some problems features are not numerical, so other options should be considered unless these non-numerical features can be mapped to numerical values.

As Figure 2.1 shows, choosing the correct k can affect the performance of the system. For example, choosing $k = 5$, instead of $k = 3$, can change the prediction of the unlabeled instance to a rhombus. Generally, larger values of k reduce the effects of noise in the data. On the other hand, a smaller k is more useful when distinct boundaries between classes are required. However, the best choice of k depends upon the data and there is no rule to determine which k works best for a particular problem. Heuristic techniques such as cross-validation (see Section 2.2) can be used to select an appropriate k. The term *nearest neighbor algorithm* is used for the special case of k-nearest neighbor where $k = 1$. In this case the predicted class for each sample would be the label of the closest training example.

2.1.2 Support Vector Machine

Support vector machine (SVM) is a supervised learning method that is used for classification (Bishop, 2006). As we used SVM as the core of our system, this chapter

provides a short review of SVM. SVM classifiers belong to the family of linear separators, which divide the input data into two different classes by using a separating hyperplane. Constructing such a hyperplane can be challenging because the main goal of classifiers is correctly classifying the unlabeled data, not the labeled data. A data sample is linearly separable if there is a hyperplane that correctly classifies all the labeled instances. SVM divides the data into two classes using a hyperplane that maximises the *margin*, which is the distance from it to the nearest labeled data sample on each side – see Figure 2.2. In general such a hyperplane can minimize the misclassification of unlabeled data because it has the largest minimum distance to the labeled samples of both classes (Bishop, 2006).

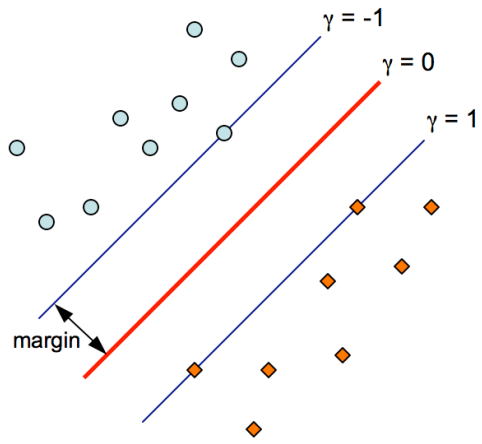

Figure 2.2: Support Vector Machine separates data by using a hyperplane that maximizes the margin.

The training data set contains N input instances $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N$, with corresponding target values $t_1, t_2, \cdots, t_N$ where $\vec{x}_i \in \Re^n$ is a feature vector and $t_n \in \{-1, +1\}$. The separating hyperplane, $\vec{w} \cdot \vec{x} + b$, requires a weight vector $\vec{w} \in \Re^n$ and a bias parameter $b \in \Re$. The support vector machine finds these parameters in the training phase, using input samples and their labels. After finding separating hyperplane in the training phase, when a new data sample $\vec{x}$ can be simply classified according to the sign of classification function $\gamma(\vec{x}) = \vec{w} \cdot \vec{x} + b$.

Linearly separable data

The separating hyperplane might not be unique as many values of $\vec{w}$ and b can classify the training data sample. Support vector machines choose these parameters to maximize the "margin", that is, the perpendicular distance of an instance $\vec{x}_n$ from the separating hyperplane. This hyperplane $\gamma(\vec{x}) = 0$ is given by following equation.

$$\frac{|\gamma(\vec{x}_n)|}{\| \vec{w} \|} = \frac{|\vec{w} \cdot \vec{x}_n + b|}{\| \vec{w} \|} \tag{2.1}$$

As Figure 2.2 shows this distance for at least three data samples is equal to the margin. For linearly separable data set, we can rewrite Equation 2.1 as follows:

$$\frac{t_n \gamma(\vec{x}_n)}{\| \vec{w} \|} = \frac{t_n(\vec{w} \cdot \vec{x}_n + b)}{\| \vec{w} \|} \tag{2.2}$$

Support vector machines seek the parameters $\vec{w}$ and b to maximize the margin, which is the perpendicular distance of the closest x_n from the hyperplane. Therefore, the required $\vec{w}$ and b can be found by solving the following optimization problem.

$$\arg\max_{\vec{w},b} \left\{ \min_n \left[\frac{t_n(\vec{w} \cdot \vec{x}_n + b)}{\| \vec{w} \|} \right] \right\} \tag{2.3}$$

As scaling $\vec{w} \rightarrow \kappa\vec{w}$ and $b \rightarrow \kappa b$ produce the same classifier, therefore without loosing the generality of the problem, we can set $t_n(\vec{w} \cdot \vec{x}_n + b) = 1$ for the closest instance. In this case, following constraints will be satisfied by all instances.

$$t_n(\vec{w} \cdot \vec{x}_n + b) \geq 1, \qquad n = 1, \cdots, N \tag{2.4}$$

Equation 2.3 simplifies to the following optimization problem.

$$\arg\max_{\vec{w},b} \Big(\min_n \left[\frac{1}{\| \vec{w} \|} \right] \Big) = \arg\max_{\vec{w},b} \Big(\frac{1}{\| \vec{w} \|} \Big) = \arg\min_{\vec{w},b} (\| \vec{w} \|) \tag{2.5}$$

subject to the constraints given by Equation 2.4.

Overlapping class distributions

In the previous sub-section we assumed the training data was linearly separable. In practice, however, the class distributions may overlap, in which case exact separation of data might not be possible. Therefore, we need to relax the hard margin constraints given by Equation 2.4 to allow the support vector machines to misclassify some of the training data samples.

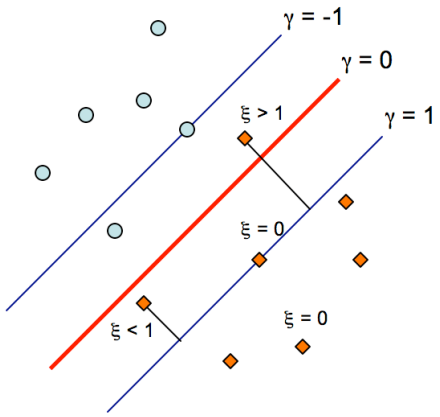

Figure 2.3: Illustration of the slack variables ξ_n.

To do this, we use *slack variables*, ξ_n, to allow the data samples to be on the wrong side of the hyperplane. So, Equation 2.4 should be replaced with

$$t_n(\vec{w} \cdot \vec{x}_n + b) \geq 1 - \xi_n, \qquad n = 1, \cdots, N \tag{2.6}$$

where slack variables are constrained to satisfy $\xi_n \geq 0$ for $n = 1, \cdots, N$. Here, each slack variable is associated with one training instance and set to zero for data samples that are inside the correct margin boundary and $\xi_n = |t_n - y(\vec{x}_n)|$ for the others. Figure 2.3 shows different values of slack variables for several data samples. To maximize the margins and also penalize the misclassification, we optimize the following optimization problem instead of Equation 2.5:

$$\arg\min_{\vec{w},b}(C\sum_{n=1}^{N}\xi_n + \frac{1}{2} \parallel \vec{w} \parallel^2) \tag{2.7}$$

where C controls the balance between the margin and misclassification penalty. To

solve this optimization problem we need a very powerful mathematical method, Lagrange Multiplier.

Lagrange Multiplier

Lagrange multipliers are used to find the stationary point of a function of several variables subject to one or more constraints. Consider finding the minimum of

$$f(\vec{w}, \vec{\xi}, b) = C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \parallel \vec{w} \parallel^2$$

subject to constraints given in Equation 2.6, which we rewrite in the following form:

$$g_n(\vec{w}, \xi_n, b) = t_n(\vec{w} \cdot \vec{x}_n + b) - 1 + \xi_n \geq 0, \qquad n = 1, \cdots, N \tag{2.8}$$

There are two possible cases, according to whether the constraint stationary point lies in the region $g_n(\vec{w}, \xi_n, b) > 0$ or on the boundary $g_n(\vec{w}, \xi_n, b) = 0$. In the former case, $g_n(\vec{w}, \xi_n, b)$ plays no role and so we need to find the stationary point of $f(\vec{w}, \xi_n, b)$, which can be found using the following equation:

$$\bigtriangledown f(\vec{w}, \xi_n, b) = 0 \tag{2.9}$$

In the latter case, we note that at any point on the constraint surface, the gradient $\bigtriangledown g(\vec{w}, \xi_n, b)$ will be orthogonal to the surface. In addition, because we are trying to maximize $f(\vec{w}, \xi, b)$, such a point must have the property that the vector $\bigtriangledown f(\vec{w}, \xi_n, b)$ also must be orthogonal to the constraint surface. Therefore, $\bigtriangledown f(\vec{w}, \xi_n, b)$ and $\bigtriangledown g(\vec{w}, \xi_n, b)$ are parallel. Figure 2.4 demonstrates a simplified version of the problem, where f and g have only two arguments, to clarify the case. Because f and g are parallel in the extremum point, there must exist a parameter λ, known as Lagrange multiplier, such that

$$\bigtriangledown f + \lambda \bigtriangledown g = 0 \tag{2.10}$$

Note Equation 2.10 holds for both cases, due to Equation 2.9

By defining $L \equiv f + \lambda g$, which is called the Lagrangian function, we can simply derive Equation 2.10 by setting $\bigtriangledown L = 0$. Therefore, we can find the minimum of

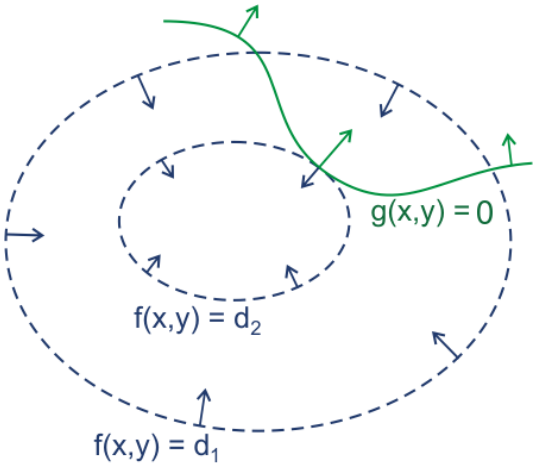

Figure 2.4: A geometrical picture of function $f(x, y)$ (dashed lines) and constraint $g(x, y) = 0$ (solid line). Arrows show their derivatives. At each stationary point of f, both derivatives are orthogonal to the constraint surface, which is a curve in this example. (This figure is taken from Bernat, 2009)

function $f(\vec{w}, \xi, b)$ subject to the constraints given in Equation 2.8 by finding the stationary point of $L(\vec{w}, \xi, b, \lambda)$ with respect to $\vec{w}, \vec{\xi}, b$ and λ.

$$L(\vec{w}, \xi, b, \vec{\lambda}) = C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \parallel \vec{w} \parallel^2 + \sum_{n=1}^{N} \lambda_n \{t_n(\vec{w}\vec{x} + b) - 1 + \xi_n\} \qquad (2.11)$$

where $\vec{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_N)$ and $\lambda_n \leq 0$ for $n = 1 \cdots N$. Notice the optimal λ will not have $\lambda_n > 0$ because we are looking for the minimum of function $f(\vec{w}, \xi, b)$. We also know that whether the stationary point lies on the border $g(\vec{w}, \xi, b) = 0$ or whether it lies in the region $g_n(\vec{w}, \xi, b) > 0$, the value of $\lambda g_n(\vec{w}, \xi, b)$ would be zero. So the solution to the problem of minimizing $f(\vec{w}, \xi, b)$ subject to $g(\vec{w}, \xi, b) \geq 0$ is obtained by optimizing Equation 2.11 subject to the following constraints:

$$g_n(\vec{w}, \xi, b) \geq 0 \quad n = 1 \cdots N \qquad (2.12)$$

$$\lambda_n \leq 0 \quad n = 1 \cdots N \qquad (2.13)$$

$$\lambda_n g_n(\vec{w}, \xi, b) = 0 \quad n = 1 \cdots N \qquad (2.14)$$

Non-linear classification

Equation 2.11 can be used to classify linearly separable data, however in many problems, linear separators in the feature space might not be useful. Figure 2.5(left) shows a sample distribution of training data in 2-dimentional feature space. There is no straight line that can separate the input data in this example. However, if we could transform the feature space to form a linearly separable distribution of data samples, such as Figure 2.5(right), a linear classifier could be used.

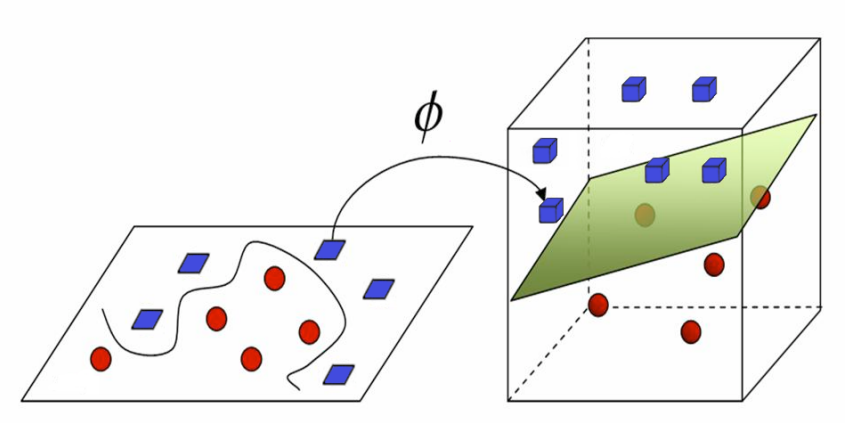

Figure 2.5: Using a kernel function. (left) Distribution of data samples in the input space. (right) Data samples after transforming the feature space by a kernel function ϕ. (This figure is taken from Takahashi, 2009)

A kernel function $\phi(\cdot)$ that transforms one feature space to another sometimes provides us with a powerful tool that can solve the separation problem. By replacing $\vec{x}_n$ with $\phi(\vec{x}_n)$ in all the above equations (Equation 2.1 to Equation 2.14), we can place the separating hyperplane in the transformed feature space. This approach, which is called the Kernel Method, is one of the main ideas in constructing effective support vector machines. Linear, polynomial, radial basis and sigmoid kernel functions are some of the simple kernel functions. However more complicated kernel functions can be used when it is needed. Selecting the best kernel function depends on the feature space and is very difficult for some problems. However, automatic approaches such as cross-validation can be used to determine which kernel function is probably best for any particular problem. We have used cross-validation as a part of PSP to choose between different kernel functions. Next section briefly

describe cross-validation.

2.2 Cross-validation

Cross validation is a technique that estimates the practical accuracy of a predictor. In cross validation, the labeled data is partitioned into complementary subsets. One subset, the training set, is used to train the classifier while the other subset, called the testing set, is used for validation. Multiple rounds of cross-validation are performed using different partitioning to reduce the variability. The validation results then are averaged over the rounds. This technique is especially useful when the supply of data samples is limited or there is limited number of positive (or negative) samples. Using cross-validation gives us the opportunity to reduce bias due to poor partitioning of data samples (Bishop, 2006; Kohavi, 1995).

There are various types of cross-validation due to different partitioning techniques. Random sub-sampling splits the data randomly. The advantage of this method is that the proportion of the training and testing sets is not dependent on the number of rounds. However, some samples may never be selected for the testing set whereas others be selected several times. In K-fold cross-validation, each sample is selected for the testing set once. This technique partitions the data into K complementary subsets and in each round, one of these subsets is used as testing set and the rest are used as the training set. If K is equal to the number of data samples this technique is called leave-one-out cross-validation (Bishop, 2006). Figure 2.6 shows how K-fold cross-validation works for the case of $K = 4$.

2.3 Basic Local Alignment Search Tool

The similarity between two proteins can be determined by comparing their amino-acid sequences. Proteins that have the same amino-acid sequences serve the same functions, while most of the proteins that do not have similar amino-acid sequences have different functionalities. However, minor amino-acid sequence modifications might not hugely change protein functionality. Alignment is a way to take into account the effects of these modifications. We can define a distance measure be-

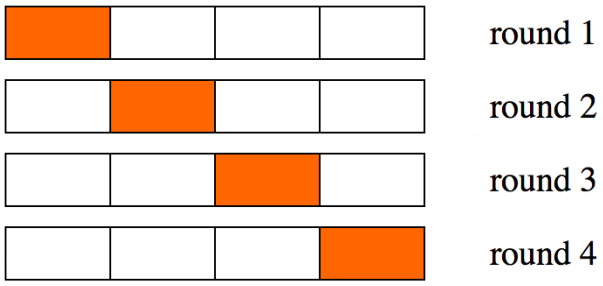

Figure 2.6: K-fold cross-validation for the case of $K = 4$. Each row demonstrates one round of cross-validation and shows the entire data set (large rectangle), which is partitioned into 4 complementary subsets. In each round one subset (shaded) is used as testing set and the other three subsets are used as a training set. The final result is the average of the accuracy of all four rounds. (This figure is taken from Bishop, 2006)

```
PKC-gamma:  HKQCVINVPSLCGMDHTEKRGRIYL--KAEVADEKLHVTV
            H++CV +VPSLCG+DHTE+RGR+ L  +A  +DE +H+TV
PRKCA :     HRRCVRSVPSLCGVDHTERRGRLQLEIRAPTSDE-IHITV
```

Figure 2.7: A part of alignment of PKC-gamma in M.musculus (P63318) and PRKCA in H. sapiens (Q7Z727). Upper and lower lines show amino acid sequences of PKC-gamma and PRKCA respectively. The center line shows the alignment. Each letter in this line shows an exact match and each "+" indicates a positive score substitution. Each "-" in the first sequence shows an insertion and each "-" in the second sequence shows a deletion. The other blanks in the middle line shows negative score substitution.

tween two different sequences of amino-acids by assigning cost values to possible sequence modifications, such as insertion, deletion and substitution. This distance metric can be used to define the similarity of two proteins – the lower the total cost of the modifications, the more similar the proteins. Figure 2.7 shows how using alignment can consider such changes.

Although modified proteins can be functionally different from the original ones, all modifications do not equally change the functionality of the proteins. Removing an amino acid from the sequence of amino acids of a protein p might totally change its functionality while inserting the same amino acid into the sequence of amino acids of p might not make a huge difference in its role. More generally, each insertion, deletion or substitution may change the functionality of the original

	C	S	T	P	A	G	⋯
C	9	-1	-1	-3	0	-3	
S	-1	4	1	-1	1	0	
T	-1	1	4	1	-1	1	
P	-3	-1	1	7	-1	-2	
A	0	1	-1	-1	4	0	
G	-3	0	1	-2	0	6	
⋮							⋱

Table 2.1: A part of BLOSUM62 matrix. Row and Columns are indexed by amino acids. Note how having no substitution (diagonal values) has a positive score and most of the other elements have negative or small values.

protein in different degrees, therefore different costs should be assigned for each of these mutations. In addition, all amino acids are not equally similar and the varying compatibility between amino acids needs to be considered when substitution cost is being calculated. Because of the difficulty of this challenge, scientists assume each mutation can affect the functionality of the protein independently of the other mutations. By using this simplifying assumption, the problem of varying amino acids compatibility can be solved by using a substitution cost matrix.

A substitution matrix assigns each possible substitution to a score-value which is used to calculate the total cost of alignment. In this matrix, substituting each amino acid with itself (having no mutation) has a relatively large positive value and most of the other substitutions have a small positive or a negative value. There are lots of substitution matrices that can be used as a measurement metric for similarity of amino acids, such as Point Accepted Mutation (Dayhoff *et al.*, 1978) or Block Substitution Matrix (BLOSUM) (Henikoff and Henikoff, 1992). There are several BLOSUM matrices based on different databases, each named with a number. BLOSUM with higher numbers are designed for comparing closely related sequences, while BLOSUM with lower numbers are designed for comparing less related sequences. Table 2.1 shows a part of BLOSUM62 substitution matrix.

Dynamic programing and FASTA (Pearson and Lipman, 1988) are two approaches that use alignment to compare primary biological sequence information — *e.g.*, the amino-acid sequences of different proteins. These algorithms compare one query sequence to another or to a database of sequences and identify the simi-

larity between these sequences and also can find the regions of similarity between them. This similarity is determined by comparing the amino-acid sequences of the proteins using alignment. However, the functionality of a protein is not sensitive to some modifications. Basic Local Alignment Search Tool (BLAST) (Altschul *et al.*, 1990) is another sequence comparison approach, which uses non-sensitivity of protein functionality to improve its running speed. This improvement is essential for biological projects when millions of comparisons should be made (Altschul *et al.*, 1990). This efficiency comes at a cost: while dynamic programming can guarantee the optimal alignments of the query and database sequences, BLAST cannot. FASTA is also more accurate than BLAST when the similarity of less similar sequences is calculated. However, in our problem, using BLAST is beneficial because the main focus of our system is to find the most similar sequences and BLAST and FASTA are equivalent for highly similar sequences and BLAST is faster than FASTA (Krawetz and Womble, 2003).

In general, BLAST takes as input a specific protein p and a database of proteins D, and returns a mapping, $\text{BLAST}_{p,D}$, from each protein $p' \in D$ to R, where $\text{BLAST}_{p,D}(p')$ is a measure of how similar p' is to p. The similarity result is dependent on the database, including its size. For each protein comparison, BLAST returns a vector of values, including a similarity score, percent identity and an *e-value*, where smaller e-values indicate higher similarity.

2.4 Sub-cellular Localization

Eukaryotic cells can be divided into functionally distinct parts, called sub-cellular localizations. Each protein inside the body of a eukaryotic organism belongs to one or more subcellular localizations. For example, nuclear proteins are located in the nucleus and plasma membrane proteins are inside the cell membrane. Knowledge of the sub-cellular localization of a protein is important for understanding the functionality of the protein. For example, as plasma membrane proteins are exposed to the outside of the cell, they are important for cell-cell communication or signaling. Some of the major sub-cellular localizations are nucleus, cytoplasm, peroxisome,

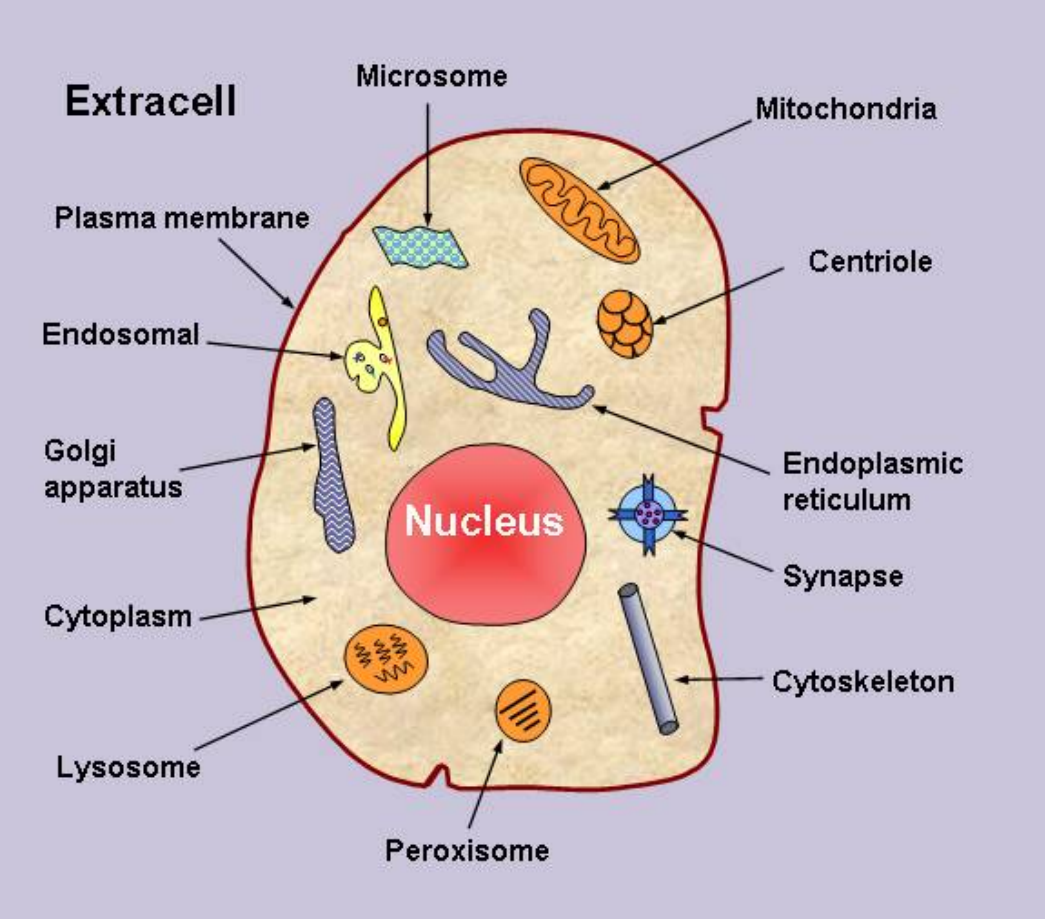

Figure 2.8: Schematic of sub-cellular components in a typical animal cell. (This figure is taken from Chou and Shen, 2009)

mitochondria, plasma membrane, lysosome, golgi apparatus, endoplasmic reticulum and extracellular space (Wallace *et al.*, 2001). Figure 2.8 provides a schematic of some of these sub-cellular components.

2.5 Transmembrane Regions

Proteins that span the membrane of a cell are called transmembrane proteins. These proteins have two major types: Alpha-helical and Beta-barrels. The former proteins, which form the major category of transmembrane proteins, are present in the inner membranes of bacterial cells and the plasma membrane of eukaryotes. The latter proteins are found in outer membranes and cell walls of some bacteria and in the membranes of mitochondria and chloroplasts (Wallace *et al.*, 2001). As this

dissertation only addresses eukaryotes, we ignore beta-barrel proteins and focus on alpha-helical proteins.

An alpha helix is a coiled conformation, resembling a spring, in which every backbone N-H group donates a hydrogen bond to the backbone C=O group of a near by amino acid (Wallace *et al.*, 2001). Not all amino-acids have the same potential to form an alpha helix structure. For example, methionine, alanine, leucine, uncharged glutamate, and lysine all have especially high helix-forming propensities, whereas proline, glycine and negatively charged aspartate have poor helix-forming propensities (Pace and Scholtz, 1998). There are some experimental methods for determining an alpha helix. However, there are some methods that can predict alpha helixes in given amino acid sequences. Ganapathiraju *et al.* (2008) present one of these approaches. These methods can be used to predict transmembrane domains.

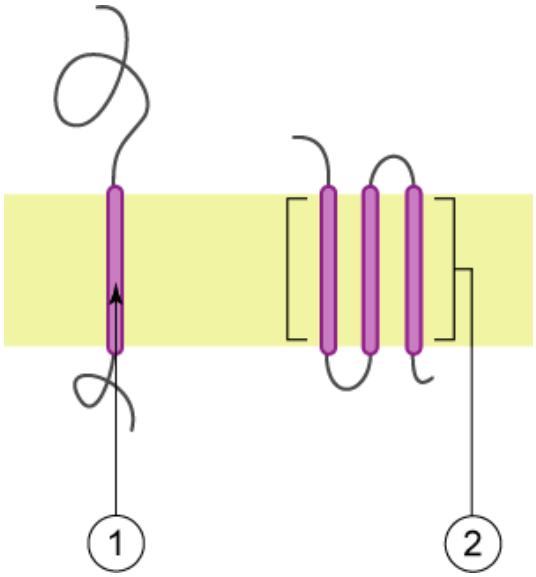

Figure 2.9: Schematic representation of transmembrane proteins: 1- A single transmembrane alpha-helix 2- An alpha-helical protein with three membrane regions. The membrane is represented in light brown. (This figure is taken from Wikipedia, 2009)

Predicting transmembrane helices enables prediction of which parts of the protein are inside the cell and which parts are outside and how many times the protein crosses the membrane. Each sub-sequence of amino acids that are completely inside the membrane is called a transmembrane region or simply a membrane region.

The number of membrane regions of the protein is the number of times that the protein crosses the membrane. Figure 2.9 shows two proteins, one with a single cross over the membrane and one with three membrane regions.

2.6 Signal Peptide

A signal peptide is sometimes referred to as a protein's "zip code". It is a short 3-60 amino acid long peptide chain that directs the transport of a protein. It targets a protein across the endoplasmic reticulum membrane in eukaryotes. By default these proteins are transported through the golgi apparatus and exported by secretory vesicles, but some of them stay in the endoplasmic reticlulum or the golgi or go back to the lysosomes (Emanuelsson *et al.*, 2007).

Although, there is no simple sequence for signal peptides, there are some predictors that predict signal peptide zones, such as Käll *et al.* (2007). These predictors are able to indicate whether a protein has a signal peptide or not. The output of a signal peptide predictor can be used as one of the features of the proteins for other classifications, such as its role in a signaling pathway.

Chapter 3

Pathway Prediction

The architecture of the entire pathway prediction system along with the functionalities of each sub-system are described in this chapter. Section 3.1 gives an overview of our prediction system. It describes how various sub-systems are connected to each other and how they work together. Section 3.2 shows how pathways, which are both inputs and outputs of our system, are modeled. This section also defines specialized terms that are used in the next sections. The rest of this chapter (Section 3.3 through Section 3.5) describes each individual sub-system in detail.

3.1 System architecture

Given a species' proteome (*i.e.*, the set of its proteins) and a library of known signaling pathways, the Predict Signaling Pathway (PSP) system predicts which of these proteins play which role in each of these signaling pathways for that species. Our PSP system has three sub-systems, Make Model Signaling Pathway (*MMSP*), Train Signaling Pathway Classifiers (*TSPC*) and Predict Signaling Pathway Roles (*PSPR*).

MMSP is a sub-system of PSP that receives a set, S, of known homologous pathway instances of various species, and creates a union model by merging them together. This union model, which is the output of *MMSP*, is used as the structure to predict pathway roles of query proteins (see Section 3.3). *TSPC* uses the proteomes of the species in S to train one classifier for each role of the union model created by *MMSP*. Each of these classifiers is responsible for predicting the proteins in one

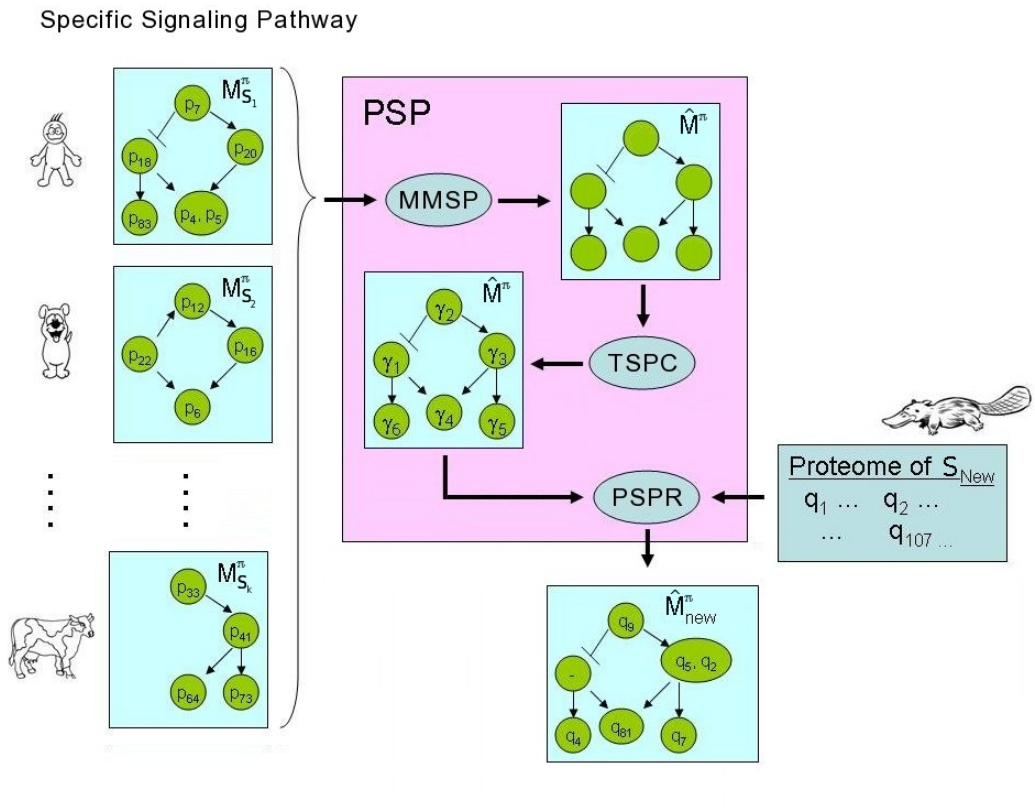

Figure 3.1: Overview of Predict Signaling Pathway (PSP): Given the M_S^π pathways of various species S, *MMSP* builds the union model $\hat{M}^\pi$. *TSPC* uses the proteomes P_S of these species (not shown) to train one classifier γ_ρ for each role ρ of $\hat{M}^\pi$ (corresponding to each node). Finally, *PSPR* uses these classifiers along with the proteome of a new species S_{New} to predict which protein(s) will qualify for each role in this π signaling pathway in this new species.

node of the pathway (see Section 3.4). *PSPR* is the last part of PSP, the prediction step. It uses trained classifiers along with the union model and proteome of a new species S_{new} to predict the pathway instance in S_{new}. Figure 3.1 gives an overview of the whole system.

3.2 Pathway Representation

In general, a pathway structure is a directed graph that describes the relations between proteins[1] in a signaling pathway. Each node of the graph represents a *role* of

[1]In general, these graphs can also include compounds – *i.e.*, small molecules. However, we ignore them for this dissertation, focusing on only the proteins.

the pathway and specifies a set of proteins that play that role. Each arc represents a relation (activation, inhibition, binding, etc.) between its source node and its target node. Figure 3.2 depicts a small part of the MAPK pathway structure in human, showing nine relation arcs of three different types. An activation arc "$\alpha \rightarrow \beta$" indicates that proteins in the source node α can activate proteins in the target node β; an inhibition arc "–|" indicates that proteins in the source node inhibit proteins in the target node; and a bind-to arc "–" means that any protein in the source node can bind to any protein in the target node. For example, Figure 3.2 shows that proteins in the SOS, RasGRP and PKC nodes can activate proteins in the Ras node, while proteins in the Gap1m and NF1 nodes can inhibit proteins in the Ras node.[2] We take our pathways from the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa *et al.*, 2008), but other sources could be used as long as they have an appropriate graph structure.

We represent signaling pathways using the following notation. Each pathway $M = \langle N, A \rangle$ is a graph where each node $n \in N$ has an associated "role", and a set of associated proteins P^n, and the arcs $A \subseteq N \times N$ are a subset of pairs of nodes, each labeled with a type $a(\langle n', n'' \rangle) \in \{$ activation, inhibition, phosphorylation, dephosphorylation, binding $\}$. M_S^π denotes the instance of the π pathway for the species S. For example, the MAPK pathway for *H. sapiens* is denoted $M_{H.sapiens}^{MAPK}$; it may be different from homologous pathways in other species — *e.g.*, $M_{H.sapiens}^{MAPK} \neq M_{M.mulatta}^{MAPK}$. The symbol n_S^ρ denotes the node with role ρ associated with species S; *e.g.*, $n_{H.sapiens}^{Ras}$ is the human node with the Ras role. A specific node n_S^ρ is identified with a single species S and it can appear in several pathway graphs of that species. However, a role can be the label of many nodes from many different species. *e.g.*, the $n_{H.sapiens}^{Raf1}$ node can appear in many different human pathways (here, it appears in $M_{H.sapiens}^{MAPK}$, $M_{H.sapiens}^{VEGF}$ and $M_{H.sapiens}^{Erb\beta}$), and there are many different Raf1-labeled nodes for different species: $n_{H.sapiens}^{Raf1}$, $n_{R.norvegicus}^{Raf1}$, etc.

We let $R_S^\pi = \{\rho \,|\, n_S^\rho \in N_S^\rho\}$ denote the set of roles that appear in pathway instance $M_S^\pi = \langle N_S^\pi, A_S^\pi \rangle$. For example, Ras is a member of $R_{H.sapiens}^{MAPK}$. P_S denotes

[2]Here we identify each node with its associated role. Hence, the "Ras node in human" refers to the node whose role is Ras, which we write $n_{H.sapiens}^{Ras}$; see Figure 3.3

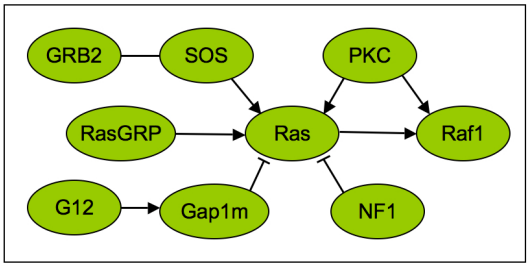

Figure 3.2: A small part of MAPK signaling pathway structure in human.

the proteome of species S and $P^{\rho} = \cup_S P_S^{\rho}$ denote the union of all the proteins associated with the role ρ across all available species. Figure 3.3 presents a summary of these, and other, terms.

3.3 *MMSP* Constructs the Model Pathway

There are many known cellular signaling pathways, including the eleven KEGG pathways used in this dissertation, each with its own function. However, these pathways vary over different species; *e.g.*, the VEGF pathway in *P. troglodytes* involves $|R^{VEGF}_{P.\ troglodytes}| = 30$ roles and 32 arcs, while the same pathway in *X. laevis* has only $|R^{VEGF}_{X.\ laevis}| = 27$ roles and 31 arcs. In fact, there are roles and arcs in *P. troglodytes* that are not in *X. laevis*, and vice versa. Our goal is to use the pathways of a set of studied species to predict the pathways of less known species. For example, we might use the VEGF pathways of both *P. troglodytes* and *X. laevis* to find the proteins participating in various roles in the VEGF pathway of a third species. This species might have some roles and arcs that correspond to only *P. troglodytes*, and other roles and arcs that correspond only to *X. laevis*. (In fact, the VEGF pathway in *H. Sapiens* indeed has some roles and arcs corresponding to only *P. troglodytes* and some other roles and arcs corresponding to only *X. laevis*.)

MMSP (a sub-system of PSP) starts by building a general model pathway by combining the pathway versions for a set of model species. This requires creating a "graph union" of the graphs for each species pathway. This approach not only creates a diverse set of roles and arcs in the pathway structure, but also increases the number of proteins associated with each role.

S: species; n: node; π: (signalling) pathway; ρ: role	
$M_S^\pi = \langle N_S^\pi, A_S^\pi \rangle$	instance of π pathway for the species S; graph structure involving nodes N_S^π and arcs A_S^π
$\mathcal{M}^\pi$	all models associated with pathway π, across species
$\hat{M}^\pi$	the union pathway for the π pathway
R_S^π	set of roles in pathway instance M_S^π
n_S^ρ $n_{\hat{M}}^\rho$	node with role ρ associated in species S; in model pathway $\hat{M}$
P P_S P^ρ P_S^ρ $P_{\hat{M}}^\rho$	all proteins ... across species; ... in species S; ... associated with role ρ; ... role ρ in species S; ... role ρ in model pathway $\hat{M}$
$\hat{P}_S^\rho$	for the set of proteins predicted to serve role ρ in species S
$\gamma_\rho(p)$	classifier associated with role ρ
PSP($\mathcal{M}^\pi$, P)	"Predict Signaling Pathway"
MMSP($\mathcal{M}^\pi$)	"Make Model Signaling Pathway"
TSPC($\hat{M}^\pi$, P)	"Train Signaling Pathway Classifiers"
PSPR($\hat{M}^\pi$, P_S)	"Predict Signaling Pathway Roles"

Figure 3.3: Glossary of Terms used

MMSP constructs the model pathway $\hat{M}$ (aka union pathway) by taking the union of all pathway instances. The model pathway has a node $n_{\hat{M}}^\rho$ for each role ρ occurring in any of the species models, whose associated proteins $P_{\hat{M}}^\rho$ are the union of all of the proteins associated with the same role in any species. $\hat{M}$ also includes an arc of type a between two nodes $n_{\hat{M}}^{\rho_1}$ and $n_{\hat{M}}^{\rho_2}$ if nodes with these two roles are connected by the same type of arc in any of the individual species.[3]

Figure 3.4 shows an example of the union of two trivial pathways, where each node n^ρ is labeled by its role ρ (on upper left bump) and shows the associated set of proteins P^ρ. The set of proteins in role b of the model pathway is the union of the set of proteins of role b in species A and B: $P_{\hat{M}}^b \coloneqq P_A^b \cup P_B^b$. The set of arcs of the union pathway is the union of the sets of arcs from the two pathways. Note that there is only *one* arc from $\langle n_{\hat{M}}^b, n_{\hat{M}}^a \rangle$, of type $\rightarrow$ corresponding to both $\langle n_A^b, n_A^a \rangle$ and $\langle n_B^b, n_B^a \rangle$.

[3]The same pair of nodes could be connected many times in the union pathway $\hat{M}^\pi$ if they appeared with different labels in different species.

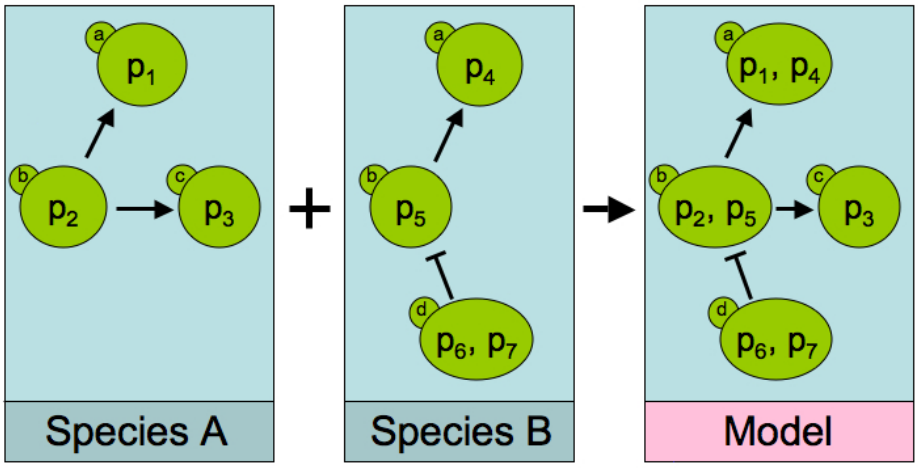

Figure 3.4: Building the model pathway. Two instances of a pathway in two different species A (left) and B (middle). The model pathway (right) is created by combining pathway instances in species A and B. The labels a, b, c, and d indicate the roles of each node; hence upper node in Species A is $n^a_{\hat{M}}$, with proteins $P^a_{\hat{M}} = \{p_1, p_4\}$. Similarly we see $P^b_A = \{p_2\}$ and $P^d_B = \{p_6, p_7\}$.

3.4 *TSPC* Learns a Set of Classifiers

After producing this model pathway $\hat{M}$, *TSPC* learns a set of classifiers, one for each of $\hat{M}$'s roles. Each of these role-specific classifiers $\gamma_\rho(p)$ predicts whether each protein p in a species plays the ρ role in the pathway for that species. We let γ_ρ:$P_S \rightarrow \{Y, N\}$ denote the classifier associated with role ρ, where $\gamma_\rho(p) = Y$ if the protein p plays role ρ in S and $\gamma_\rho(p) = N$ otherwise. For example, the $\gamma_a(\cdot)$ classifier for the role a (not shown in Figure 3.4) returns Y if it predicts that the protein plays role a in the pathway, and N otherwise.

There are many supervised learning methods that can learn a classifier from a data sample whose instances are each labeled either positive or negative. Like most standard classifiers, our γ_ρ's take as input a fixed-size vector of features to describe each instance (protein). We therefore compute a fixed set of features based on the primary amino-acid sequence of the protein. The first feature for each protein (with regard to the classifier $\gamma_\rho(\cdot)$) is a measure of the similarity between that protein and the most similar protein in the model pathway that is associated with role ρ. We use the BLAST algorithm (Altschul *et al.*, 1997) to compute this similarity measure.

For our computations, we set the database D (see Section 2.3) to be the set of proteins described in the KEGG website. We use only the *e-value*, where smaller values indicate a higher similarity: In particular, the first feature value for protein p, wrt classifier γ_ρ, is $e_\rho(p) = \min_{q \in P^\rho}\{e_{p,q}\}$ where

$$e_{p,q} = \text{BLAST}_{p,D_{KEGG}}(q) \tag{3.1}$$

is the *e-value* of protein q wrt protein p. In the training phase, we cannot use $e_\rho(p)$ because it is zero. Therefore we use the highest similarity between the protein p and the proteins in P^ρ that do not belong to the same species as p. For example, if p belongs to the *H. sapiens* we use the highest similarity between p and $P^\rho - P_{H.sapiens}$.

Although the first feature depends on the union model for the role as well as the protein, the other features depend only on the protein. The next nine features of protein p correspond to its subcellular locations. *TSPC* uses the Proteome Analyst system (Lu *et al.*, 2004) to predict in which of the nine cellular locations this protein does its main work: nucleus, cytoplasm, peroxisome, mitochondrion, plasma membrane, lysosome, golgi, endoplasmic reticulum and/or extracellular. Note that a protein can be in more than one location; hence *TSPC* uses 9 subcellular features (each a single bit) to encode this information. *TSPC* also uses the Phobius system (Käll *et al.*, 2007) to predict two more features for protein p: the number of membrane spanning regions (a non-negative integer) and whether it is a signal peptide or not (a bit). These features are relevant characteristics of roles in signaling pathways. For example, each signaling pathway should have one or more roles whose proteins each have a positive number of membrane spanning regions since the signal must pass through some cell membrane. All together, *TSPC* computes twelve features for each protein: the real-valued $e_\rho(p)$, nine binary subcellular values, the number of membrane regions and one binary feature that indicates if the protein p is a signal peptide or not.

For training examples, *TSPC* uses the model pathway $\hat{M}^\pi$ to obtain labeled positive instances — *e.g.*, for the role a in Figure 3.4, p_1 and p_4 serve as positive examples. For negative examples, we use all other proteins from the set of species

that were used to produce the model pathway. In this implementation, we consider $k = 14$ species $\{S_1, \cdots, S_k\}$ (Table 4.2), with proteome sizes varying between 7,126 proteins and 29,445 proteins, with a total of 278,201 proteins, $P = \cup_{i=1}^{k} P_{S_i}$ across all species. The number of proteins $|P_S^\rho|$ in any single node n_S^ρ varied from 1 to 45 across the 5,608 nodes in the 11 different pathways (of the 14 species) that we considered.

To train a $\gamma_\rho(\cdot)$ classifier for each role ρ, we must identify many training instances, both positive and negative. The most straightforward way to train a classifier for role ρ is to use a set of proteins, P^ρ, as positive examples and the complimentary set, $P - P^\rho$, as negative examples. However, this leads to a very imbalanced training set. Therefore *TSPC* used a quick cut-off on the *e-value* to define our negative training set, including only those proteins p where $e_\rho(p) < 10$ as negative training examples. This reduced the number of negative examples to approximately 1,000 instances, which creates more balanced training sets. The remaining negative training instances are the proteins that are most similar to the positive training instances, but do not play the appropriate role. Even though the training set is still unbalanced the results are reasonable.

For each of the 5,608 roles in the 11 union pathways, *TSPC* trained a Support Vector Machine (SVM) classifier (Bishop, 2006) using these labeled training instances. This system constructs a hyperplane that approximately separates the classes, by maximizing the margin between the two data sets. We used the `libsvm 2.86` implementation of SVM (Chang and Lin, 2001) with default settings and chose either linear or radial basis function kernels for each role, selecting the one with the larger "in-fold" training accuracy obtained by cross validation; see Section 4.2.

3.5 *PSPR* uses the Model Pathway to Make Predictions about Novel Proteins

PSPR is the component of PSP that uses the classifiers built by *TSPC*, within the model pathway $\hat{M}^\pi$, to predict which proteins of a given proteome P_S, from a

new species S, play which roles in this π pathway. For each role ρ in the model pathway $\hat{M}^\pi$, ***PSPR*** applies ρ's classifier γ_ρ to each protein in P_S to produce the set $\hat{P}_S^\rho = \{\gamma_\rho(p) = Y \mid p \in P_S\}$, which is the set of S's proteins predicted to play role ρ. For some roles, this set is empty. The "predicted pathway roles" $R_S^\pi = \{\, \rho \in \hat{M}^\pi \mid \hat{P}_S^\rho \neq \{\} \,\}$ include all roles ρ in $\hat{M}^\pi$ for which $\hat{P}_S^\rho$ is non-empty. We then let N_S^π be the associated nodes in this predicted pathway, with those roles. In addition, M_S^π inherits all arcs from $\hat{M}^\pi$ that connect nodes in M_S^π. That is, if $n_S^\alpha, n_S^\beta \in N_S^\pi$ and $\langle n_{\hat{M}}^\alpha, n_{\hat{M}}^\beta \rangle \in A_{\hat{M}}^\pi$ then $\langle n_S^\alpha, n_S^\beta \rangle$ is in A_S^π; moreover, it will have the same label: $a(\langle n_S^\alpha, n_S^\beta \rangle) = a(\langle n_{\hat{M}}^\alpha, n_{\hat{M}}^\beta \rangle)$.

Chapter 4

Experiments

Our experiments were based on the KEGG Pathway database, using the eleven pathways shown in Table 4.1, on the fourteen species shown in Table 4.2. As each pathway varies in size for different species, the second and third columns of Table 4.1 give the minimum and maximum number of roles appearing in each pathway across the different species. For evaluation purposes, we used two different versions of KEGG, one from 2006 and one from 2008. Table 4.1 contains summary data from both versions, with some new roles being discovered after 2006 are only included in the 2008 data and one role (in MAPK) being removed between 2006 and 2008. The information about the ErbB pathway is not included in the KEGG-06 section because this pathway was added to KEGG after 2006. For our data in all 11 pathways, the number of roles in each model pathway matched the maximum number of roles for that pathway. This was the case since there happened to be at least one species that had all of the roles, so its number of roles was equal to this maximum.

4.1 Evaluation

As shown in Figure 3.1, PSP takes as input a proteome P_S from a novel species S and a set of known pathways $\mathcal{M}^\pi = \{M^\pi_{S_1}, \ldots, M^\pi_{S_k}\}$, corresponding to the same π signaling pathway across multiple different species. Let $\hat{M}^\pi$ be the model pathway produced by ***MMSP***; M^π_S be the correct[1] signaling pathway for this species

[1]That is, "currently accepted"; see Section 4.3.

Table 4.1: For each signaling pathway π used: the minimum and maximum number of roles $|R^{\pi}_{S_i}|$ across the 14 species S_i, in both the 2006 and 2008 versions of the KEGG database. Note that ErbB did not appear in KEGG 2006.

	KEGG-08		KEGG-06	
Pathway	Min	Max	Min	Max
MAPK	26	124	21	125
Wnt	12	67	11	67
ErbB	12	60	–	–
TGF-beta	26	54	18	54
Calcium	18	43	10	41
Phosphatidylinositol	10	31	7	30
mTOR	4	29	4	29
VEGF	6	28	6	28
Jak-STAT	12	26	6	26
Notch	2	22	3	22
Hedgehog	2	18	4	18

S; and $R^{\pi} = R^{\pi}_{\hat{M}} \cup R^{\pi}_{S}$ be the union of the roles that appear in either $\hat{M}^{\pi}$ or M^{π}_{S}. Then PSP computes a set of predictions. For each role $\rho \in R$, ***PSPR*** predicts a set of proteins $\hat{P}^{\rho}_{S} \subset P_S$ that (appear to) qualify for this role. If this ρ is not in $R^{\pi}_{\hat{M}}$, then PSP sets $\hat{P}^{\rho}_{S}$:= $\{\}$. Let $\hat{\mathcal{P}}^{\pi}_{S} = \{\hat{P}^{\rho}_{S}\}_{\rho}$ be the entire collections of these protein-sets, one for each role. Similarly let $P^{\rho}_{S} \subset P_S$ is the true set of proteins associated with this role ρ, and $\mathcal{P}^{\pi}_{S}$:= $\{P^{\rho}_{S}\}_{\rho}$. We again set P^{ρ}_{S} := $\{\}$ if this ρ is not in $\hat{\mathcal{P}}^{\pi}_{S}$.

Ideally, if PSP worked perfectly, then $R^{\pi}_{\hat{M}}$ would match R^{π}_{S}, and for each role ρ, the predicted set $\hat{P}^{\rho}_{S}$ would exactly match the true set of proteins P^{ρ}_{S}. To compare $R^{\pi}_{\hat{M}}$ with R^{π}_{S}, we therefore compute their similarities over all of their roles, based on

$$\hat{q} = \bigcup_{\rho \in R^{\pi}_{\hat{M}},\ p \in \hat{P}^{\rho}_{S}} \langle \rho, p \rangle \qquad q = \bigcup_{\rho \in R^{\pi}_{S},\ p \in P^{\rho}_{S}} \langle \rho, p \rangle$$

which are each a set of pairs whose first component is the role and whose second is one of the proteins of that role. We then define the similarly between $R^{\pi}_{\hat{M}}$ and R^{π}_{S} based on the F-measure of the associated $\hat{q}$ and q:

$$F(\hat{q}, q) \quad = \quad \frac{2 \cdot |\hat{q} \cap q|}{|\hat{q}| + |q|} \quad = \quad \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

which uses

$$\text{Precision } = \frac{|\hat{q} \cap q|}{|\hat{q}|} \qquad \text{Recall } = \frac{|\hat{q} \cap q|}{|q|} \tag{4.1}$$

Note this F-measure ranges from 0 to 1, and is only 1 if $\hat{P}_S^\rho = P_S^\rho$ for all $\rho \in R^\pi$.

We use "leave out one species" cross-validation to estimate the accuracy of PSP. We start with $k = 14$ species $\{S_1, \ldots, S_k\}$, with known proteomes $P_i = P_{S_i}$ and let pathway M_i^j be the j^{th} pathway of the i^{th} species. Here, for each pathway M^j, for each species $i = 1..k$, we compute

$$\hat{M}_i^j \;=\; \hat{M}^j(P_i) \;=\; PSP(\{M_1^j, \ldots, M_{i-1}^j, M_{i+1}^j, \ldots, M_k^j\},\; P_i)$$

then compute the overall score – the (precision, recall, F-measure) triple – for each of the roles in $\hat{M}_i^j$ and M_i^j:

$$s_i^j \;=\; \text{Score}(M_i^j,\; PSP(\{M_1^j, \ldots, M_{i-1}^j, M_{i+1}^j, \ldots, M_k^j\},\; P_i))$$

Finally, for each species S_i with m known signaling pathways $\{M_i^1, \ldots, M_i^m\}$, we then compute the average triple $\text{ES}(S_i) \;=\; \frac{1}{m} \sum_{j=1}^{m} s_i^j$.

4.2 Empirical Results

Table 4.2 shows the results of our predictions, listing the average precision, recall and F-measure scores for each of the fourteen species ES(S_i). The "Total" row is the average of the 14 values. This table also includes the number of proteins in P_S and the total number of roles, over the pathways considered. We see that, in 12 of the 14 species, the average recall is over 0.85, meaning that PSP is able to find essentially all of the relevant proteins for the roles (average recall = 0.913); the average precision of 0.724 shows that it occasionally included a few too many proteins. Moreover, the F-measures of only the two "*" ed species — *i.e.*, *S. scrofa* (pig) and *D. rerio* (zebra fish) — are below 0.70; in both cases due to low precision (*i.e.*, many false positives). We discuss this result in Section 4.3.

Table 4.3 presents our results from another point of view. Here we categorized the results based on *pathways* instead of species — *i.e.*, this is the average over all the species for each of the pathways. This shows our prediction is accurate for

Table 4.2: From left to right: Species; Number of proteins in the species; Number of roles in each species over all considered pathways; Precision/Recall/F-measure of pathway predictions (averaged over all considered pathways).

Species	Proteins	Roles	Precision	Recall	F-measure
Homo sapiens [Man]	24200	500	0.877	0.938	0.904
Canis familiaris [Dog]	19807	474	0.847	0.914	0.877
Mus musculus [Mouse]	29445	501	0.825	0.923	0.868
Pan troglodytes [Chimpanzee]	25185	449	0.862	0.879	0.864
Macaca mulatta [Monkey]	23964	451	0.831	0.908	0.864
Gallus gallus [Chicken]	18115	433	0.857	0.870	0.861
Monodelphis domestica [Opossum]	19114	441	0.798	0.920	0.852
Rattus norvegicus [Rat]	26160	476	0.793	0.920	0.850
Ornithorhynchus anatinus [Duck-bill platypus]	16387	406	0.804	0.722	0.751
Xenopus laevis [African frog]	10623	322	0.606	0.940	0.732
Bos taurus [Bull]	22327	399	0.592	0.921	0.716
Xenopus tropicalis [Western frog]	8228	242	0.588	0.916	0.712
**Danio rerio* [Zebrafish]	27520	383	0.533	0.824	0.642
**Sus scrofa* [Pig]	7126	131	0.315	0.902	0.457
Total	278201	5608	0.724	0.893	0.782

Table 4.3: Precision/Recall/F-measure of pathway predictions (averaged over all considered species).

Pathway	Precision	Recall	F-measure
TGF-beta	0.816	0.955	0.871
mTor	0.781	0.896	0.821
ErbB	0.761	0.894	0.809
Jak-STAT	0.826	0.796	0.797
Phosphatidylinositol	0.716	0.920	0.785
Wnt	0.677	0.934	0.768
Notch	0.792	0.906	0.768
MAPK	0.695	0.897	0.762
VEGF	0.702	0.867	0.757
Hedgehog	0.648	0.936	0.754
Calcium	0.645	0.819	0.713
Total	0.724	0.893	0.782

almost all of the pathways and the overall high F-measure (seen in Table 4.2 for species) is not just due to some specific pathways.

Table 4.4 shows the effect of each of the features used by our $\gamma_\rho(\cdot)$ classifiers.

Table 4.4: Precision/Recall/F-measure of pathway predictions (averaged over all considered species and pathways). Each row provides the accuracy of the prediction after using the feature mentioned along with the features mentioned in upper rows.

Feature/approach	Precision	Recall	F-measure
e-value	0.696	0.854	0.752
+ *sub-cellular localization*	0.698	0.858	0.755
+ *membrane regions*	0.692	0.865	0.756
+ *signal peptide*	0.679	0.862	0.746
+ *kernel selection*	0.724	0.893	0.782

Table 4.5: (left) Number of arcs in each species over all considered pathways; (right) Precision/Recall/F-measure of pathway predictions calculated for *arcs* (averaged over all considered pathways)

Species	#Arcs	Precision	Recall	F-measure
C. familiaris [Dog]	492	1.000	0.995	0.997
M. musculus [Mouse]	443	1.000	0.960	0.980
H. sapiens [Man]	536	1.000	0.960	0.980
M. mulatta [Monkey]	536	0.967	0.977	0.972
M. domestica [Opossum]	460	0.965	0.967	0.966
R. norvegicus [Rat]	500	0.939	0.957	0.948
G. sallus [Chicken]	437	0.958	0.937	0.945
P. troglodytes [Chimpanzee]	458	0.951	0.888	0.913
D. rerio [Zebrafish]	346	0.840	0.999	0.911
B. taurus [Bull]	369	0.759	1.000	0.858
X. laevis [African frog]	233	0.735	1.000	0.847
O. anatinus [Duckbill platypus]	384	1.000	0.657	0.785
X. tropicalis [Western frog]	131	0.553	0.995	0.710
S. scrofa [Pig]	49	0.329	1.000	0.494
Total	5374	0.870	0.948	0.888

The first row, *e-value*, shows our predictive accuracy using only the single feature, *e-value*, which measures the highest similarity between the target protein and the proteins in this role. The values in the table are the average precision, recall and F-measures scores for all the fourteen species. The second row shows the effects of adding the 9 sub-cellular localization features; we see this slightly improves all three of the measures. The third row shows the effect of also adding the number of membrane spanning regions to the features — which makes essentially no difference. The values of the fourth row are obtained after adding signal peptide as the last feature of our classifier. While we can see that F-measure has dropped by

a small value, this is not statistically significant (1-sided paired t-test, $p \approx 0.90$). However, when combined with our final change (kernel selection), this feature turns out to give a higher F-measure than if it is not used. The final row represents our most accurate classifier. It shows the advantages of allowing *PSPR* to decide which kernel to use in the support vector machine: linear versus radial basis function (rbf). We see that this made an improvement to the F-measure, from 0.746 to 0.782 which is statistically significant — $p <$ 3E-05. Even though adding the "signal peptide" feature had not improved the F-measure (third "addition" of Table 4.4), we found that the average F-measure of the classifiers that exclude this feature (but include "kernel selection") is only 0.768, which is inferior to the classifiers that include it, at the 0.782 shown in Table 4.4. This is true in general: kernel selection helps increase our accuracy.

Note that this selection (linear or rfb) is done completely automatically, without any human input. Here, for each species S_i in Table 4.2, we remove S_i from the training set and run cross validation on the rest of the data (for the 13 other species). For each such fold, *PSPR* excludes species S_j from the training set (in addition to S_i) and for each role $\rho \in R^{\pi}_{\hat{M}}$ in each pathway π, *PSPR* trains two classifiers (SVM-linear and SVM-rbf) on the remaining $14 - 2 = 12$ species, and compares the accuracy of these classifiers on S_j. After repeating the process for all the species in the training set, *PSPR* calculate the average prediction accuracy for each of SVM-linear vs SVM-rbf for this role ρ in this pathway π, then selects the kernel function with the highest average performance value. The final classifier for that role ρ uses this kernel function. Across all classifiers, 1792 linear kernels and 5262 radial-basis functions kernels were selected.

The first row of Table 4.4 shows that running the SVM learner on the e-value, alone, gives a fairly high F-measure. This suggests two other, simpler approaches: First, we could just use this e-value directly to identify the proteins. Here, for each role ρ with associated proteins P^{ρ}, for each $q \in P^{\rho}$ we compute $e_{p,q}$ (Equation 3.1) with respect to each $p \in P_S$ for the proteome P_S of the novel species S, and simply set $\hat{P}^{\rho}_S = \{p \in P_S \mid \exists q \in P^{\rho},\ e_{p,q} <$ 1E-100$\}$ to be those proteins in the novel species with an e-value less than 1E-100 to some proteins in the model pathway.

(We used 1E-100 as the threshold as was empirically determined this was the best cut-off value in {1E-200, 1E-100, 1E-50, 1E-25, 1E-10, 1E-5}.) This produced an average leave-out-one-species F-measure (over the 11 pathways and 14 species) of only 0.650, which is 10.2% less than using SVM on the e-value alone (Table 4.4), and 13.2% worse than our best system. Second, we can view e-value as the basis for a nearest-neighbor classifier. For each protein $q \in P^\rho$, we find the $p \in P_S$ with the smallest e-value: $\text{nn}(q\,;\,P_S) = \arg\min_{p \in P_S}\{e_{p,q}\}$, then let $\hat{P}^\rho_S = \{\text{nn}(q\,;\,P_S) \mid q \in P^\rho\}$. The average F-measure here was 0.730, which was significantly lower than our best result (1-sided paired t-test, $p <$ 3E-3).

We also considered the multi-class classifier approach, of learning a single classifier for each pathway, that maps each protein to a role. The classifier returns one of $|R^\pi_{\hat{M}}|+1$ values for each protein (the extra "1" accounts for "none of the above"). However, this would force us to predict *(at most) one* role for each protein, rather than a set of roles. This is problematic as many proteins (1359 in our training set) belong to more than one role, which is why we could not use this approach.

Table 4.5 shows the average precision, recall and F-measure scores for predicting the *arcs* in the various pathways. Here, PSP includes an A^π_S arc in a predicted model if it predicts at least one protein for each end. For example, if it was seeking the Model pathway (from Figure 3.4) within the proteome of species C, P_C, we would include the "b $\rightarrow$ c" arc if at least protein from P_C qualified for the "b" role, and at least one P_C protein qualified for the "c" role – $\hat{P}^a_C$ and $\hat{P}^b_C$ are non-empty. (Note that we do not require that these qualifying proteins are correct.) This would be a false positive if the Model pathway of species C did not include this "b $\rightarrow$ c" arc.

While the focus of this system is predicting which proteins fill which roles of the pathways, Table 4.5 shows our system does accurately identify most of the arcs in the examined species as well as the proteins in the associated roles.

We see that PSP can effectively predict the roles, and arcs, of essentially all available signaling pathways in all species, except possibly the two marked with *'s in Table 4.2. However, the result for these two species may actually be better than they appear; see the next section.

Table 4.6: (left) Number of proteins in each species, P_S, and number of roles over all considered pathways, for each of the 2006 and 2008 versions of KEGG; Precision/Recall/F-measure of pathway predictions (average over all considered pathways): (middle) trained and tested on KEGG-06; (right) trained on KEGG-06 and tested on KEGG-08.

	KEGG-06		KEGG-08		KEGG-06 / KEGG-06			KEGG-06 / KEGG-08		
Species	Proteins	Roles	Proteins	Roles	Precision	Recall	F-measure	Precision	Recall	F-measure
S. scrofa	1,062	104	7,126	131	0.845	0.878	0.852	0.686	0.667	0.670
H. sapiens	25,719	440	24,200	500	0.885	0.814	0.842	0.879	0.805	0.835
M. musculus	30,172	436	29,445	501	0.839	0.832	0.827	0.839	0.816	0.820
R. norvegicus	26,259	379	26,160	476	0.691	0.893	0.776	0.776	0.881	0.822
**B. taurus*	22,854	218	22,327	399	0.345	0. 869	0.479	0.637	0.861	0.721
**C. familiaris*	19,808	155	19,807	474	0.205	0.857	0.318	0.812	0.838	0.821
**P. troglodytes*	21,825	139	25,185	449	0.186	0.653	0.262	0.840	0.634	0.715
Total (7 species)	147,699	1,871	154,250	2,930	0.563	0.827	0.615	0.783	0.788	0.773

4.3 Alternative Historical Evaluation

The predictive accuracies in Table 4.2 show that the precision is relatively low for two species (the "*"ed ones) — which are species that have not been extensively studied. We therefore wondered if PSP's accuracy for these species might actually be higher than these reported rates. That is, our F-measure scores are based on the (allegedly) "true" set of proteins associated with each role. However, many signaling pathways, especially those in understudied species, are not yet complete; researchers are still updating these pathways, typically by adding new proteins to roles. This means our evaluation may be wrong when it declares a predicted protein to be a false positive: *i.e.*, when PSP predicts a protein qualifies for a role ρ, but this protein is not in the current P_S^ρ. It is possible that PSP is actually correct as P_S^ρ is incomplete, in that this protein *should be* a member of P_S^ρ. Counting this protein as a false positive will (incorrectly) reduce PSP's precision score for this pathway of this species.

To test the possibility, we ran our PSP system on *historical* data: *i.e.*, we trained classifiers based on the 2006 version of KEGG (KEGG-06), and used these classifiers to make predictions on the (2006) proteomes of various species. The "KEGG-06/KEGG-06" columns in Table 4.6 provide these scores, when using the "2006 versions of the truth". (This involves only 7 of the 14 species, as the 2006 KEGG database did not include the data required for the other seven species.) Note especially the abysmal precision values for *C. familiaris*, *P. troglodytes* and *B. taurus* (the "*"ed ones). Our argument suggests this may be because, in 2006, these species had not been well annotated. If so, then we anticipate our predictions would better match the "2008 versions of the truth" — *i.e.*, the P_S^ρ for each role of these species based on KEGG-08.

The "KEGG-06 / KEGG-08" columns of Table 4.6 show the results of using KEGG-08 as ground truth to evaluate the predictions made by the "KEGG-06 classifier". We find a statistically significant improvement in the average precision (compared to KEGG-06/KEGG-06): from 0.615 to 0.773 — due largely to huge improvements in precision for those three species, coupled with minimal reductions

in recall. This shows that many of the predictions based on KEGG-06 were correct, even though they involved claims that were not included in KEGG-06. In total, KEGG-08 included 1,620 proteins that we considered false positives when we evaluated the predictions based on KEGG-06 data, that turned out to be true positives. This is why we suspect that many of the false-positives found using KEGG-08 may actually be correct — *i.e.*, that PSP's actual precision may be higher than the 0.782 found when training and testing on KEGG-08, as shown in Table 4.2. Note also that the annotations present in 2008 but not 2006 are very likely to be experimentally determined; if they were purely analytic, we suspect they would have been present in 2006. Hence, this "train on 2006, test on 2008" measure is tested on annotations that are probably more accurate than the alternative of just removing a random subset of the 2008 data.

Chapter 5

Conclusion

This dissertation provides a new technique for learning to predict signaling pathways in novel species, based on known signaling pathways in familiar species. This technique is completely automated – *i.e.*, it does not need human adjustments at any level and is based on various automatically-computed properties of proteins.

We have shown that our approach produces accurate predictions over all of the species and pathways we considered — *i.e.*, total precision, recall and F-measure of 0.724, 0.893 and 0.782 respectively. We have also used historical data to indicate why we think that the actual accuracy of our prediction might be even higher than reported here, due to incompleteness of the test sets. The webpage (http://cs.ualberta.ca/∼bioinfo/signaling) provides the complete set of these PSP's predictions; it will be interesting to see which of these predicted roles turn out to be correct. Moreover, our overall PSP system is expandable, as other species and other pathways can easily be added to the system. In addition, new features (perhaps, based on protein-protein interaction, or protein domains) may be used along with the described features to potentially increase the accuracy of its predictions. (Note that many of these results have been published as Bostan *et al.* (2009).)

Bibliography

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**(3), 403–410.

Altschul, S. F., Madden, T. L., Schffer, R. 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 Res*, **25**, 3389–3402.

Bernat, C. (2009). Lagrange multiplier - wikimedia commons. `http://commons.wikimedia.org/wiki/File:Lagrange_multiplier.png`.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bostan, B., Greiner, R., Szafron, D., and Lu, P. (2009). Predicting Homologous Signaling Pathways Using Machine Learning. *Bioinformatics*.

Campbell, N. A. and Reece, J. B. (2001). *Biology (6th edition)*. Benjamin Cummings.

Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker, M., Latendresse, M., Paley, S., Rhee, S. Y., Shearer, A. G., Tissier, C., Walk, T. C., Zhang, P., and Karp, P. D. (2008). The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway Genome Databases. *Nucleic Acids Research*, **36**(suppl_1), 623–631.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Chou, K.-C. and Shen, H.-B. (2009). Cell-ploc package. `http://www.csbio.sjtu.edu.cn/bioinf/Cell-PLoc/`.

Craven, M. (2002). The genomics of a signaling pathway: a kdd cup challenge task. *SIGKDD Explorations*, **4**, 97–98.

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978). A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, **5**(suppl 3), 345–351.

Ding, C. H. Q. and Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**, 349–358.

Emanuelsson, O., Brunak, S., von Heijne, G., and Nielsen, H. (2007). Locating proteins in the cell using targetp, signalp and related tools. *Nat Protoc*, **2**(4), 953–971.

Fröhlich, H., Fellmann, M., Sültmann, H., Poustka, A., and Beibbarth, T. (2008). Predicting pathway membership via domain signatures. *Bioinformatics*, **24**(19), 2137–2142.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**(10), 906–914.

Ganapathiraju, M., Balakrishnan, N., Reddy, R., and Klein-Seetharaman, J. (2008). Transmembrane helix prediction using amino acid property features and latent semantic analysis. *BMC Bioinformatics*, **9**(6).

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Mach. Learn.*, **46**(1-3), 389–422.

Hall, N. (2007). Advanced sequencing technologies and their wider impact in microbiology. *J Exp Biol*, **210**(9), 1518–1525.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, **89**(22), 10915–10919.

Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., and Takagi, T. (2001). Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast*, **18**(6), 523–531.

Hoffmann, R. and Valencia, A. (2004). A gene network for navigating the literature. *Nat Genet*, **36**(7).

Käll, L., Krogh, A., and Sonnhammer, E. L. (2007). Advantages of combined transmembrane topology and signal peptide prediction–the phobius web server. *Nucleic acids research*, **35**(Web Server issue).

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., and Yamanishi, Y. (2008). Kegg for linking genomes to life and the environment. *Nucleic Acids Research*, **36**(suppl_1).

KEGG (2009a). Kegg pathway: Notch signaling pathway - reference pathway (ko). `http://www.genome.jp/kegg-bin/show_pathway?org_name=ko&mapno=04330&mapscale=1.0`.

KEGG (2009b). Kegg pathway: Synthesis and degradation of ketone bodies - reference pathway (reaction). `http://www.genome.jp/kegg-bin/show_pathway?org_name=rn&mapno=00072&mapscale=1.0`.

Kim, J. H., Lee, J., Oh, B., Kimm, K., and Koh, I. (2004). Prediction of phosphorylation sites using SVMs. *Bioinformatics*, **20**(17), 3179–3184.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145.

Krawetz, S. A. and Womble, D. D. (2003). *Introduction to Bioinformatics: A Theoretical And Practical Approach*. Humana press.

Lu, Z., Szafron, D., Greiner, R., Lu, P., Wishart, D., Poulin, B., Anvik, J., Macdonell, C., and Eisner, R. (2004). Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, **20**(4), 547–556.

Ma, H. and Zeng, A.-P. (2003). Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*, **19**(2), 270–277.

Moss, G. P. (2009). Biochemical nomenclature committees. `http://www.chem.qmul.ac.uk/iupac/jcbn/`.

Nielsen, H., Brunak, S., and von Heijne, G. (1999). Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng.*, **12**(1), 3–9.

Ohta, T., Tateisi, Y., and Kim, J.-D. (2002). Genia corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of HLT*, San Diego, USA.

Pace, C. N. and Scholtz, J. M. (1998). A Helix Propensity Scale Based on Experimental Studies of Peptides and Proteins. *Biophysical Journal*, **75**(1), 422–427.

Park, K.-J. and Kanehisa, M. (2003). Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, **19**(13), 1656–1663.

Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America*, **85**(8), 2444–2448.

Pireddu, L., Szafron, D., Lu, P., and Greiner, R. (2006). The path-a metabolic pathway prediction web server. *Nucleic Acids Research*, **34**, 714–719.

Schilling, C. H., Schuster, S., Palsson, B. O., and Heinrich, R. (1999). Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era. *Biotechnol Prog*, **15**(3), 296–303.

Takahashi, N. (2009). Research of N. Takahashi. `http://www-kairo.csce.kyushu-u.ac.jp/~norikazu/research.en.html`.

Thomas, P. D., Campbell, M. J., Kejariwal, A., Mi, H., Karlak, B., Daverman, R., Diemer, K., Muruganujan, A., and Narechania, A. (2003). PANTHER: A Library of Protein Families and Subfamilies Indexed by Function. *Genome Research*, **13**(9), 2129–2141.

Vázquez, A., Flammini, A., Maritan, A., and Vespignani, A. (2003). Global protein function prediction from protein-protein interaction networks. *Nat Biotechnol*, **21**(6), 697–700.

Wallace, R. A., Sanders, G. P., and Ferl, R. J. (2001). *Biology: The science of life (4th edition)*. HarperCollins Publishers.

Wikipedia (2009). Transmembrane protein - wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Transmembrane_protein`.

Yaffe, M. B., Leparc, G. G., Lai, J., Obata, T., Volinia, S., and Cantley, L. C. (2001). A motif-based profile scanning approach for genome-wide prediction of signaling pathways. *Nature Biotechnology*, **19**(4), 348–353.