



Tisham De - tisham.de08@imperial.ac.uk
Christos Gkekas - christos.gkekas08@imperial.ac.uk
Nikolas Pontikos - nikolas.pontikos08@imperial.ac.uk
Zhengzi Yi - zhengzi.yi08@imperial.ac.uk

MSc Bioinformatics and Theoretical Systems Biology
Centre for Bioinformatics
Imperial College London

April 8, 2009

Contents

1	Introduction	5
1.1	Project Description	5
1.2	Aims	6
1.3	Workplan	7
2	Background Biology	11
2.1	Phosphorylation	11
2.2	Kinase	13
2.3	Phosphatase	17
2.4	Substrate	18
3	Data Sources	21
3.1	PhosphoELM	21
3.2	NetworKIN	22
3.3	Phosida	24
3.4	PhosphoPOINT	24
3.5	PhosphaBase	26
3.6	KinBase	27
3.7	NetPath	27
3.8	TreeFam	28
3.9	Other resources	29
4	Implementation	31
4.1	Software Tools	31
4.1.1	MySQL	31
4.1.2	Python	31
4.1.3	Django-Python	32
4.1.4	Dreamweaver	33
4.1.5	PhyloWidget	33
4.2	Development Server	34
4.3	Database Schema	37
4.4	Data Input	39
4.5	Web Site	41

4.6	Database Search Tools	44
4.6.1	Kinase - Phosphatase Pairs	45
4.6.2	Simple Search	45
4.6.3	Advanced Search	47
4.6.4	SQL Query	47
4.6.5	Browse Data	50
4.6.6	Construct Phylogenetic Trees	51
4.6.7	Gene Families	51
5	Results & Future work	53
5.1	Outcomes	53
5.2	Have we met the requirements?	56
5.3	Future Plans	58

List of Figures

1.1	The work plan.	8
2.1	The effects of phosphorylation	12
2.2	The human kinome.	14
2.3	Insulin Receptor Tyrosine Kinase.	15
2.4	The evolution of kinases in different species.	16
2.5	The role of kinases and phosphatases in signalling pathways.	17
2.6	A phosphorylation cascade.	18
2.7	The frequencies of different kinds of phosphorylation sites.	19
3.1	Some of the data sources used in this project.	23
4.1	The PhyloWidget tool within KiPhoDB's website.	33
4.2	Software components of the development server.	35
4.3	The Entity Relationship (E/R) diagram of the KiPhoDB database.	38
4.4	The KiPhoDB website.	42
4.5	KiPhoDB's license	42
4.6	Search Tool for Finding Kinase-Phosphatase Pairs	45
4.7	The Simple Search tool.	46
4.8	The Advanced Search Tool.	48
4.9	The SQL Query Tool.	49
4.10	The Browse Gene Families Tool.	52

List of Tables

2.1	Kinase group classification.	15
3.1	The different categories of data in PhosphoPOINT.	25
3.2	The current contents of PhosphaBase.	26
3.3	The current contents of NetPath.	28
4.1	The structure of KiPhoDB's website.	43
5.1	The current contents of KiPhoDB.	55
5.2	Reactions per organism.	56
5.3	Pathways per organism.	56
5.4	10 top pathways for which we have the most reactions in human	57

Chapter 1

Introduction

1.1 Project Description

The KiPhoDB project started approximately three months ago with as main objective to develop a database that would enable scientists to retrieve information about kinases, phosphatases, the corresponding substrates as well as the reactions and pathways in which they are involved. Of course, this wealth of data requires a database which will be robust, stable and well designed. Therefore the creation of the database itself introduces many challenges that have to be properly addressed so that an appropriate database schema can be created. After the construction of the KiPhoDB database, one other challenging task is finding ways to insert information from various data sources. This task is difficult because every data source supplies its information in different data formats and therefore new techniques for data extraction have to be implemented each time. Finally, it is very important for every database to have a web site which will provide users with all necessary software tools to retrieve information from it. Thus we have created a very comprehensive and easy to use web site that enables users to exploit our database in every possible way.

Following is a list of all the different types of data that KiPhoDB is designed to store:

- **Kinases, Phosphatases & Substrates**

The database should contain all available information about protein kinases, protein phosphatases and their substrates found in a wide variety of species. We are not only interested in well known model organisms, such as *Homo sapiens*, *Mus musculus* and *E. coli* but we also want to store information about all available species. Using this information the scientific community will be able to compare the different protein kinases and phosphatases found in different species and hopefully draw some useful conclusions.

- **Gene Ontology Data & Enzyme Classification**

For every protein stored in the database, we want to include a lot of related information about it. Apart from the protein name, gene name and the amino acid sequence, we would also like to include Gene Ontology and Enzyme Classification information.

- **Pathway Data**

Many kinases and phosphatases are known to belong to certain biological pathways, mainly signalling pathways. It is very useful to know the pathways that a specific protein is involved in, because this information can help scientists identify the reactions that the aforementioned protein takes part in.

- **Evolutionary Data**

Many kinases, phosphatases and their substrates have paralogues and orthologues, which should also be stored in the database. Using this information it is possible to construct phylogenetic trees that visualize the evolution of similar proteins in different species from a common ancestor. Moreover, various interesting assumptions can be formed concerning the behaviour of novel kinase or phosphatase proteins based on the behaviour of their orthologues or paralogues.

1.2 Aims

There has been a lot of recent work on the kinome[1] but not so much on the phosphatome and the relationship between those two. Therefore, the need for a comprehensive database that will relate kinases, phosphatases and substrates has been a concern for many molecular biologists. Although there already exist many databases that contain information about kinases and substrates, there are surprisingly very few databases for phosphatases and none for linking these three kinds of molecules. The recent developments in mass spectrometry and high throughput data acquisition have led to the generation of a large amount of data concerning cellular pathways and cell signaling. The scientific community needs the appropriate software tools that will store these data in large databases and present them in a compact form. In this way, scientists will be able to view the data easily and efficiently, conduct their own experiments on the data and reach to the correct conclusions. With these objectives in mind, we decided to create KiPhoDB, the Kinase and Phosphatase DataBase.

The first and one of the main objectives that we have set for this database project is to provide users with all currently available information concerning the kinase and phosphatase interactome in various species. In more detail, KiPhoDB should contain all available information about kinases, phosphatases and substrates for a great variety of species, along with information about how these molecules are linked together to form certain reactions. For example, it has been scientifically proven that out of a total of 518 human kinases, only one third have known substrates and only approximately 14% have binding motifs which have been identified [2]. Moreover, the linking of kinases and phosphatases to known substrates is very difficult due to the limited experimental data at hand. Therefore the discovery of new kinase and phosphatase pairs that act on specific substrates is a very challenging task and our database will provide a safe storage for these data and all the software tools that will be required in order to mine them.

The second objective of KiPhoDB is to provide information about certain pathways

that these kinase and phosphatase reactions belong to. This information is very important because it allows scientists to obtain a better understanding of the roles that kinases and phosphatases play during the life cycle of a cell. Moreover, it is possible to target specific kinases or phosphatases which participate in signaling or metabolic pathways in order to positively affect the behaviour of certain cells and find possible cures to diseases.

Another objective of KiPhoDB is to provide evolutionary information about kinases and phosphatases. This can be achieved by comparing similar proteins (orthologues and paralogues) between various species, drawing the corresponding phylogenetic tree and reaching certain conclusions about the evolutionary distance of these molecules. This feature is very important in phylogenetic analysis of different enzymes and gives insight into the level of conservation of kinase and phosphatase domains and their activity across different species. Therefore KiPhoDB aims to provide an efficient, easy to use and flexible web environment that will enable users to construct phylogenetic trees on the fly.

1.3 Workplan

In order to achieve the aforementioned goals and deliver the product on time, we had to carefully plan our actions. The work plan that we followed is illustrated in Figure 1.1. The first step was to perform a thorough background investigation, read about and understand all the biological and computational aspects of the problem we had to solve. For this purpose, all necessary information was carefully collected and reviewed by the group members and regular meetings were held in order to discuss the results of our research and plan our next steps.

One of these steps was to locate all existing relevant databases that were available to the public and provided services similar to what we wanted to achieve. Since the work of Cohen, who in 2002 estimated that the number of proteins which get phosphorylated can be up to 50 [3], a lot of research and experimenting has followed. Several kinase, phosphatase and substrate databases have been published, such as Phospho.ELM [4], Phosida [5], Phospho-POINT [2] and networKIN [6], which provide useful online resources for phosphoproteins across various species. Protein post-translational modification by kinases and phosphatases is still an active area of research and therefore more data sources are constantly added to the list that was mentioned previously. A thorough survey of these resources was essential in order to make sure that our work would not be redundant and that it would provide the maximum benefits to the users. We reviewed them carefully and decided to use some of them as our primary data sources. Since there were multiple sources of data, we had to choose only the ones that were most significant and that offered high quality data in a format that could be easily read and parsed by a computer. All the data sources that we finally used are described in more detail in a subsequent chapter.

The next step involved choosing the software tools that we were going to use. A plethora of tools were proposed by the group members and we had to carefully examine each one of them, list its advantages, disadvantages and capabilities and finally decide on which ones to use. At the end we chose to use the MySQL relational database management system

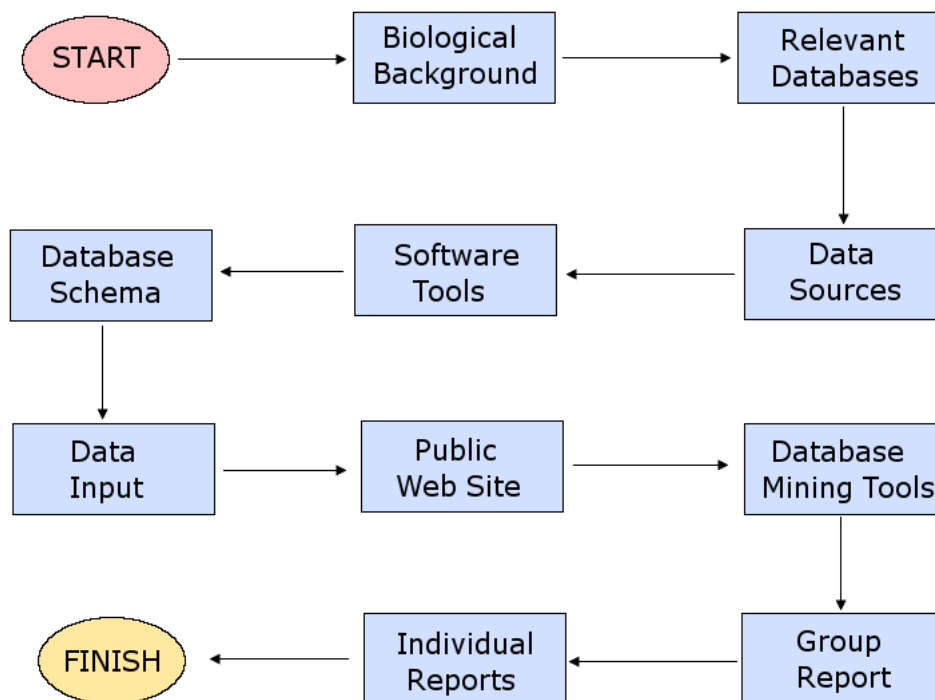


Figure 1.1: The work plan.

and the Django web framework for creating the web interface in Python. The reasons why we chose these tools, along with a brief description of each one of them, are provided in the next section.

After having a comprehensive list of all data sources and their details, we had to figure out ways of using them in order to map kinases and phosphatases to their corresponding substrates and find potential kinase-phosphatase pairs. Moreover, we had to think about and analyze the requirements that any interested researcher would have from our database. For this purpose we used the valuable advice of Prof. Michael Stumpf and Dr. Frances Turner. Based on the above analysis and taking into careful consideration the advice of our project supervisor, we created the database schema to be used for KiPhoDB. During the construction of any database, the development of the database schema is the single most important step, because it defines the nature of the information that can be stored to it. Therefore we had to spend a lot of time and effort on deciding the form of KiPhoDB, its tables and their data fields. Moreover, as we started inputting real data into the database, we often had to update and improve the database schema so that it would reflect the nature of the data and the needs of the users.

Once we had finalized the database schema, we focused on the creation of the public web site and the insertion of more data in the database. For this purpose, we created a logo for the project and all the web templates that were going to be displayed to the public user. Subsequently we had to link these web templates with the database, so that they could

draw information from it and present it to the users in a convenient and comprehensive format. Furthermore, we also created and integrated into the website various software components that enabled users to search the data in the database, perform SQL queries that were sent directly to the MySQL server, browse the data, administer the database and the web site and visualize phylogenetic trees. Automated scripts were also written in python, with the aim of keeping all database entries correct and up to date. More details about the software tools and components that we implemented can be found in later chapters.

Chapter 2

Background Biology

The main purpose of this chapter is to provide the reader with all the theoretical information about phosphorylation, kinases, phosphatases and substrates, which is essential in order to be able to understand the scope of this project, its aims and its implementation. Of course, this section does not provide a detailed description of all properties, functions and behaviour of the aforementioned molecules and reactions. If the reader wishes to gain a deeper understanding of the role that these molecules play in a living cell, we refer the reader to [7].

2.1 Phosphorylation

Phosphorylation is a chemical reaction in which a phosphate group (PO_4) is added to a biological molecule. This molecule can be a protein, a lipid or any other kind of organic molecule. Protein phosphorylation is one of the most common post-translational modifications of proteins and it plays a significant role in a wide variety of cellular processes in eukaryotic and prokaryotic cells, including cellular growth, differentiation and DNA repair [4]. In eukaryotic cells phosphorylation occurs mainly on Serine, Threonine and Tyrosine amino acids, whereas in prokaryotic cells it occurs mainly on Histidine, Arginine and Lysine residues.

In eukaryotic cells, phosphorylation tightly regulates the dynamic behaviour and decision processes of the cell. It can modify protein function by introducing conformational changes or by creating new binding sites for protein-protein interaction domains that recognize phosphorylated motifs and bind to them [6]. One example of a conformational change to the aquaporin SoPIP2;1 after phosphorylation is illustrated in Figure 2.1. Aquaporins are membrane proteins specialized in transporting water molecules across biological membranes. They are highly efficient and they are able to transport more than a billion water molecules per second. Most aquaporins in animals are always in an open state, allowing water to easily pass through them. On the contrary, plant aquaporins constantly change states according to signals, such as phosphorylation and pH, enabling plant cells to respond to drastic changes in their environment. Figure 2.1 shows how aquaporin SoPIP2;1, com-

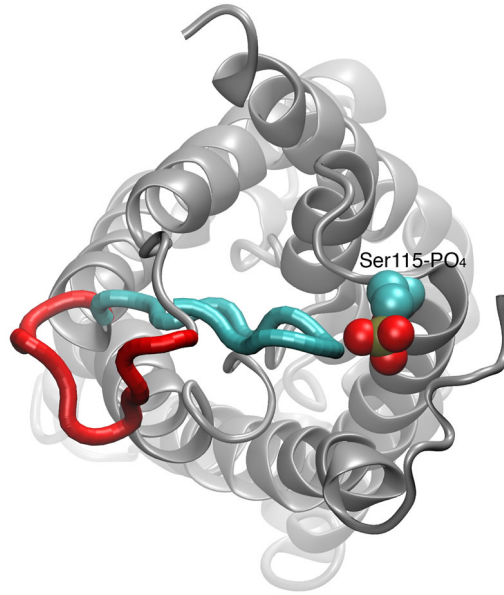


Figure 2.1: The effects of phosphorylation to aquaporin SoPIP2. (Courtesy Yi Wang)

monly found in spinach, can switch from the closed to the open state after phosphorylation. As the Serine residue (Ser115) undergoes phosphorylation, the loop moves away from the entrance of the water channel and water molecules are able to move freely through it [8]. The unphosphorylated state is shown in cyan, whereas the phosphorylated state is shown in red.

Phosphorylation events take place only in certain sites, namely phosphorylation sites, which mainly consist of the protein residue that undergoes phosphorylation and its neighbouring amino acids. These sites are commonly located in parts of a protein where the accessibility and structural flexibility are rather high, for example loops and hinges. This fact makes it easier for other biological molecules, such as kinases and phosphatases, to access them and catalyze the phosphorylation reaction [5]. Moreover, since the rate of mutation in these parts of a protein is comparatively high, the sequence regions around the residue subject to phosphorylation are evolving faster than the rest of the protein. Therefore it is difficult to correctly align phosphorylation sites found in homologous proteins and positively identify them.

A lot of effort has been invested by the scientific community to find phosphosites in proteins and annotate them, because this information offers insight to the various signalling pathways that each protein takes part in. Traditional methods for locating phosphorylation sites include Edman degradation chemistry on phosphopeptides and mutational analysis. These methods require large amounts of purified protein and they are relatively time consuming. The aforementioned disadvantages have led scientists to use modern Mass Spectrometry based methods to locate novel phosphosites in proteins and analyze protein post-translational modifications. These techniques offer higher sensitivity and speed and they are able to generate high quality data [4].

In previous years, thousands of phosphorylation sites have been identified using the methods and techniques described before. As a consequence, a lot of bioinformatics databases have been created in order to store all these data. Nevertheless, the information on which kinases act on these sites and the pathways that each protein is involved in is still missing. KiPhoDB aims to fill this gap and provide scientists with a valuable resource of information that can be easily accessed by the scientific community.

It has been estimated that almost one third of all proteins undergo phosphorylation at some point in their lifetime [9]. Each protein may contain one or more phosphorylation sites, which are regulated by one or more other proteins. The processes of phosphorylation and dephosphorylation of a protein are two ways to change its state from ‘activated’ to ‘deactivated’ and vice versa. These processes play a key role in signal transduction and in various other metabolic and cellular pathways. Therefore, the scientific community has adopted different approaches to elucidate and understand them. One such global approach is the field of ‘Phosphoproteomics’, which is dedicated to identifying and quantifying dynamic changes in phosphorylated proteins over time using mass spectrometry. These techniques are becoming increasingly important for the systematic analysis of complex phosphorylation networks.

The deregulation of protein phosphorylation and dephosphorylation in a particular pathway leads to various anomalies in cell function, which are responsible for the emergence of diseases. One such example is the p53 tumor suppressor protein [10], which is activated by phosphorylation and suppresses tumors by causing cell cycle arrest (Apoptosis). Thus phosphorylated p53 is the central defence of a cell against cancer. This highlights the importance of phosphorylation from a medical point of view (e.g. potential drug targets), apart from its relevant importance in molecular and cell biology.

In the following sections we will examine two different kind of molecules: Kinases and Phosphatases. Protein kinases are responsible for the phosphorylation of substrates, whereas protein phosphatases are involved in the dephosphorylation of substrates.

2.2 Kinase

Kinase is a type of enzyme that transfers phosphate groups from high energy donor molecules, such as ATP, to specific target molecules, namely substrates. They are also known as phosphotransferases and they can act on more than one phosphorylation sites on the surface of a given substrate. Kinases play central roles in various important cellular and metabolic networks (e.g. signal transduction, cellular growth and apoptosis), where they are often associated with the activation or deactivation of other proteins. Apart from proteins, kinases can also act in other small molecules, such as lipids, nucleotides, carbohydrates and others.

It is quite common for kinases to get phosphorylated by other kinase proteins or even catalyze their own phosphorylation (autophosphorylation). A lot of kinases are activated by phosphorylation on their ‘activation loop’, which is located at the center of most of them. Figure 2.3 illustrates this point. In this figure, both activated and deactivated

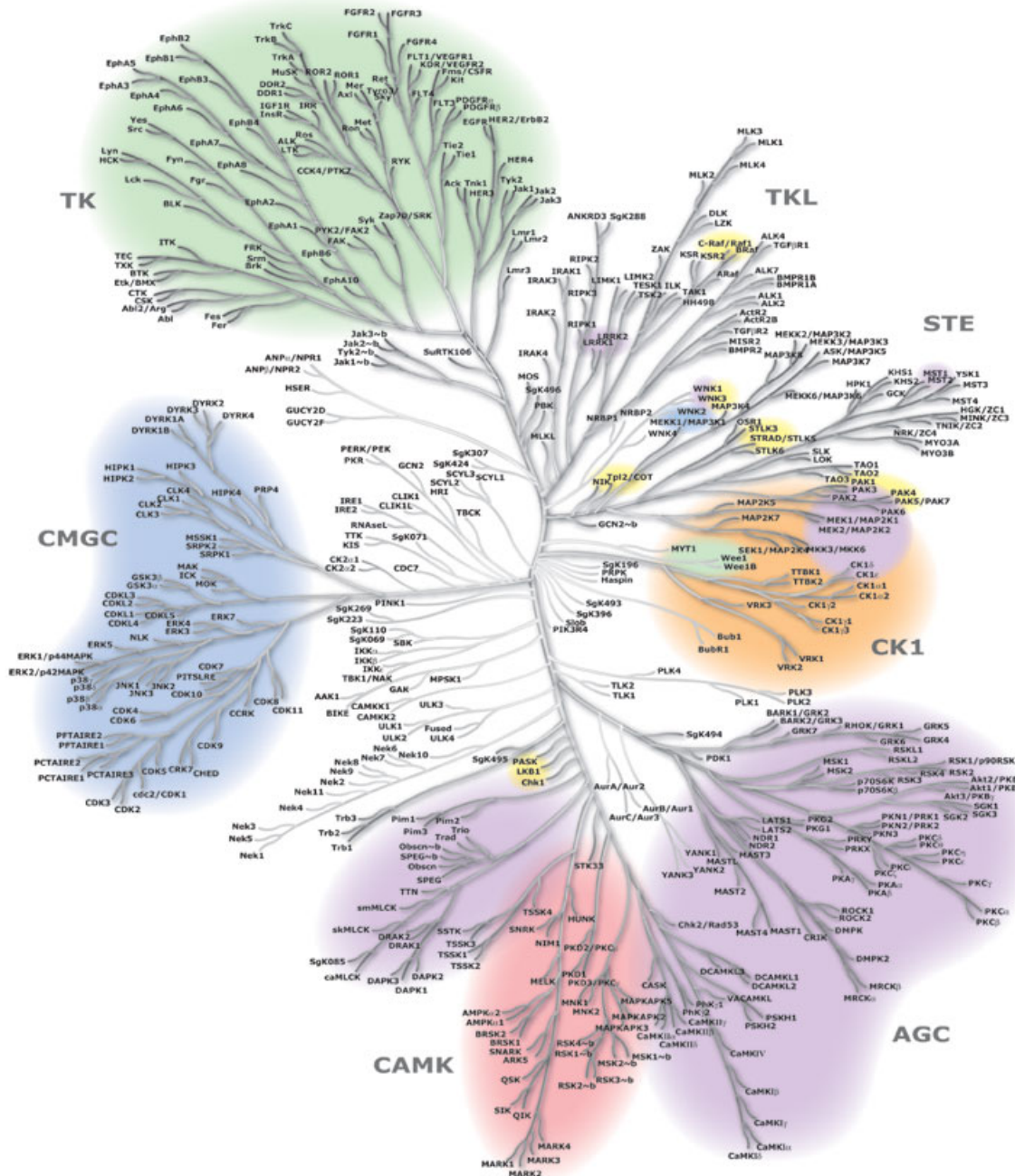


Figure 2.2: The human kinome.

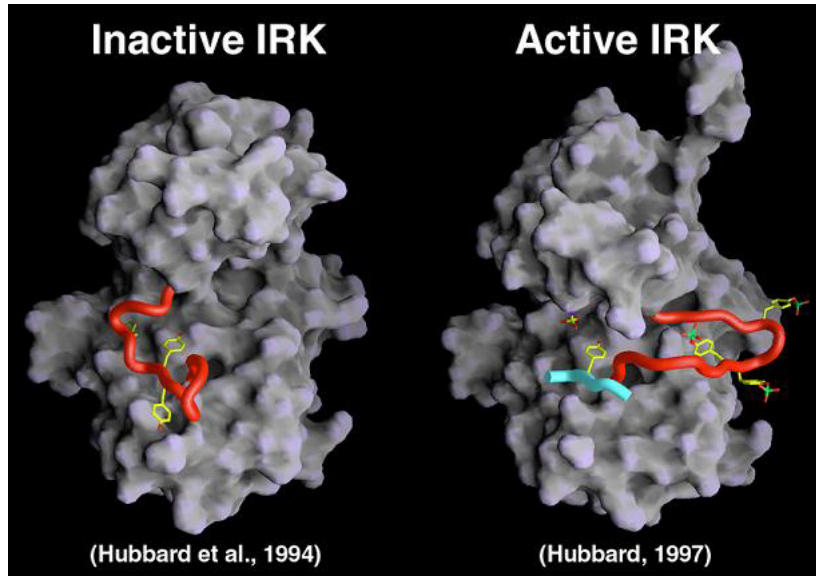


Figure 2.3: Activated and deactivated states of Insulin Receptor Tyrosine Kinase. (Courtesy of Stevan R.Hubbard)

states of the protein ‘insulin receptor tyrosine kinase’ are presented.

At the time of writing, up to 518 distinct kinases have been identified in humans [1]. However the scientific community does not know which of these kinases are responsible for which phosphorylation events. It has been reported [6] that the substrates and exact phosphorylation reactions of only approximately one third (35%) of the total number of human kinases are known. Of course this fraction is lower for other species that have not been so thoroughly researched and it is constantly decreasing as more and more phosphorylation sites are identified. Consequently there is a widening gap in our understanding of phosphorylation networks and signaling pathways, which is very difficult to close. One of KiPhoDB’s aims is to aid scientists in their efforts to investigate phosphorylation networks and help them close this knowledge gap.

Kinase Groups			
AGC	CAMK	CK1	CMGC
Other	STE	Tyrosine Kinase	RGC
Tyrosine Kinase-like	Atypical-PDHK	Atypical-Alpha	Atypical-RIO
Atypical-A6	Atypical-Other	Atypical-ABC1	Atypical-BRD
Atypical-PIKK			

Table 2.1: Kinase group classification [1].

Kinases are among the largest gene families in eukaryotes and according to [1] they can be classified in the kinase groups shown in Table 2.1. As shown in this table, there are seventeen major groups of kinases, which in turn have their own subcategories (families). This classification schema proves that kinases are enormously diverse. Figure 2.2 further illustrates this point by presenting the human kinome in the form of a tree. In this figure, all previously mentioned groups are shown painted with different colour, along with their subgroups. Mutations to these molecules may cause disease in human and other species and therefore kinases are attractive targets for drug design.

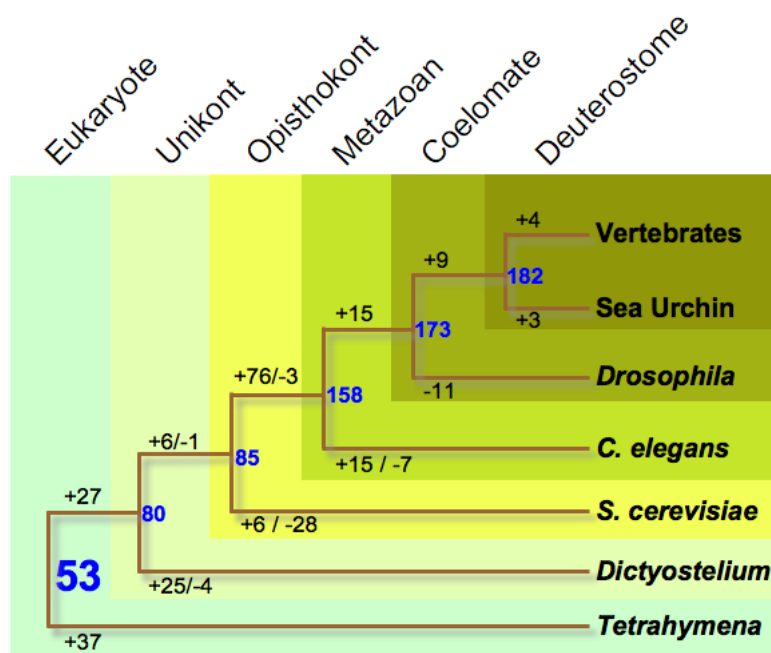


Figure 2.4: The evolution of kinases in different species (Courtesy of Manning et al).

The majority of protein kinases belong to a single superfamily and they share the same catalytic domain, namely ePK (eukaryotic Protein Kinase). Nevertheless, there are thirteen additional atypical protein kinase families (aPK - atypical Protein Kinases) which exhibit biochemical kinase activity, although they do not contain the ePK domain. However it has been reported [1] that these kinases have some common structural characteristics with the ePK domain. Whole kinome comparisons have enabled scientists to compare kinases across highly diverse species and have helped in the analysis of birth, spread, expansion and even death of kinases. Results from such analyses have shown that there were around 53 distinct kinase functions in the early common ancestor of all eukaryotes [11]. In the present day we find that there has been a lot of gain and loss in the kinase types across all species. This is clearly illustrated in Figure 2.4, which presents the evolution of kinases in a variety of species. As it is clearly shown in this figure, the most dramatic change is the emergence of metazoans with 76 classes of kinases, including among others the tyrosine kinase group. On the contrary, there have been only 28 classes in yeast which have also emerged from the

ancestral kinase groups. The evolutionary analysis of all kinases in form of trees curated at various databases can help understand the evolution of various metabolic pathways in many organisms.

2.3 Phosphatase

A phosphatase is an enzyme whose action is directly opposite to that of a kinase. Its main function is to remove a phosphate group from the given substrate, a process also known as dephosphorylation, which is the complement of the phosphorylation done by kinases. Phosphatases, together with kinases, are involved in important dynamic control and communication mechanisms inside a living cell. These mechanisms include metabolism, cell signaling, cell growth, muscle contraction, homeostasis and many more. It has also been proven that phosphatase malfunction may cause a wide variety of diseases in humans, such as diabetes, many types of cancer and obesity [12]. Therefore, as stated in previous sections, both protein kinases and protein phosphatases are important targets for molecular biology, pharmaceutical and medical research. For more information on the diverse roles that phosphatases play in living cells, we refer the reader to [13], where all presently known phosphatase functions are thoroughly reviewed.

For a long time, phosphatases were viewed by the scientific community as passive housekeeping enzymes. Yet in the last few years this view has started to change and now kinases and phosphatases are thought to be partners, both regulating the responses to certain signals. Recent studies indicate that kinases play a major role in controlling the timing and the amplitude of a signalling response, whereas phosphatases control mostly the rate and the duration of the response [14]. This behaviour of kinases and phosphatases is presented in Figure 2.5.

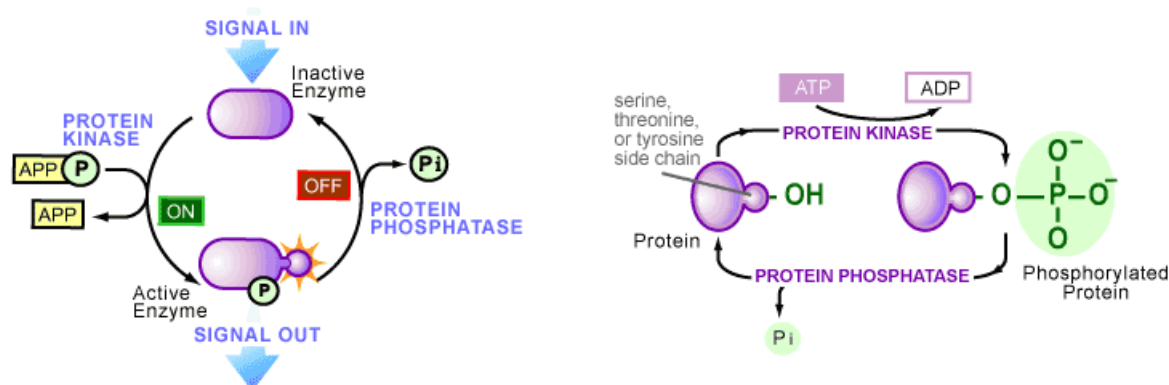


Figure 2.5: The role of kinases and phosphatases in signalling pathways (Courtesy of David Secko).

There are three main groups of protein phosphatases. The first group is called protein Ser/Thr phosphatase and it consists of phosphatases that dephosphorylate either Serine

or Threonine amino acids. This group is further divided into the following families: the large phosphoprotein phosphatase (PPP) family and the Mg^{2+} or Mn^{2+} dependent protein phosphatase family (PP2C). The second group is called protein Tyr phosphatases and it contains only phosphatases that dephosphorylate Tyrosine residues. Finally, the third group consists of Asp-based protein phosphatases [13].

It was stated in the previous section that human DNA encodes 518 protein kinases, 428 of which are known or predicted to phosphorylate Serine and Threonine residues, whereas only 90 of them phosphorylate Tyrosine. On the contrary there are only 147 protein phosphatases encoded in the human genome, 107 of which desphosphorylate Tyrosine amino acids and only 40 of them dephosphorylate Serine and Threonine residues. This discovery is very strange because, as we will see in more detail in the next section, more than 98% of the phosphorylation events take place on Serine and Threonine side chains. Therefore, the vast majority of phosphorylation events are dephosphorylated by only 40 of the 147 known phosphatases [13].

2.4 Substrate

Substrates are acted upon by kinases and phosphatases and they contain phosphorylation sites which bind a phosphate group to a particular amino acid. As it was clearly illustrated in the previous sections, kinases and phosphatases can themselves be substrates. For example, some kinase cascades have been discovered where each given kinase is itself a substrate to the previous kinase and at the same time phosphorylates the next kinase in the cascade.

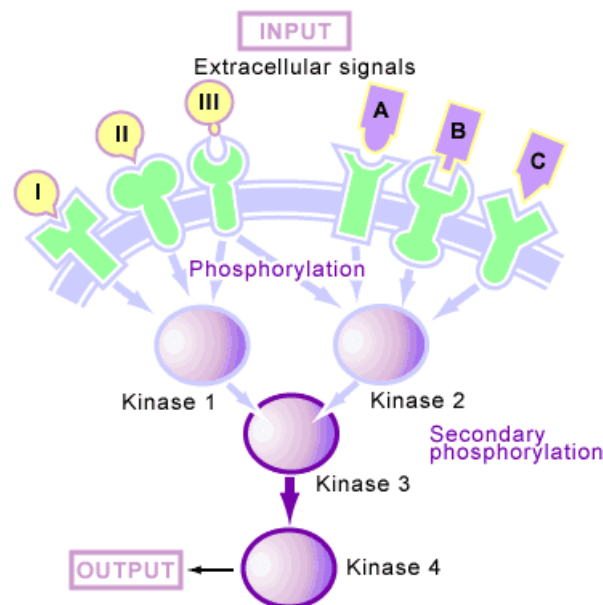


Figure 2.6: A phosphorylation cascade (Courtesy of David Secko).

One such cascade is presented in Figure 2.6. In this figure it is clearly illustrated how extracellular signals are received by specific transmembrane sensing proteins and how these proteins cause the phosphorylation of kinases. Subsequently these kinases phosphorylate other protein kinases leading to the creation of a phosphorylation cascade, which at the end transfers the signal to the appropriate receiver protein. The phosphorylation cascade continues to function until the corresponding protein phosphatases are activated and manage to shut it down.

It is important to note at this point that phosphorylation events happen to only three out of a total of twenty amino acids. These amino acids are Serine (Ser), Threonine (Thr) and Tyrosine (Tyr) and their frequencies for the human phosphoproteome are presented in Figure 2.7. The percentage for Serine is 86.4%, for Threonine is 11.8% and for Tyrosine is 1.8% [13]. Of course these percentages vary in different species.

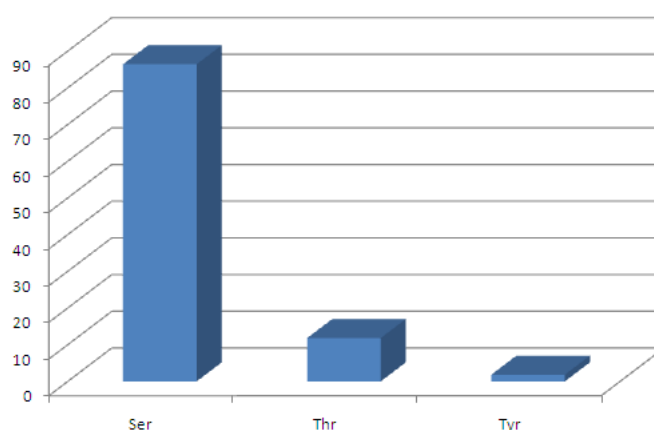


Figure 2.7: The frequencies of different kinds of phosphorylation sites.

It is widely known that phosphate groups are negatively charged and each one of them carries two negative charges. Therefore their addition to a protein through the process of phosphorylation may change the protein's structure and its overall behaviour. Nevertheless, once the protein gets dephosphorylated and the phosphate group is removed, the protein switches back to its original conformation. In most cases, these conformational changes result to changes in the protein's function and therefore phosphorylation and dephosphorylation can act as a molecular switch, turning a specific activity on and off. This mechanism of changing proteins' behaviour has many advantages: it is very rapid, taking only a few seconds, and it does not involve the destruction or creation of new proteins. Moreover, it is easily reversible and thus this fact makes the whole mechanism more flexible.

The advantages of the phosphorylation/dephosphorylation mechanism explain its extensive use as a control mechanism for various processes within a living cell. For example it has been proven that approximately 3% of the total number of proteins in yeast are kinases or phosphatases. Some of these enzymes are able to act on a wide variety of target proteins, while others are extremely specific and act only on a few substrates. These sub-

strates include other enzymes, ion channels, cell receptors, structural proteins and many more. Furthermore, the research in phosphorylation, kinases, phosphatases and substrates has been very active and intense the last few years, especially since 1992 when Fischer and Krebs, two pioneers on this field, received the Nobel Prize in medicine for their contribution [3].

Chapter 3

Data Sources

A wide variety of data sources were used in order to populate all KiPhoDB tables. In this section we will examine each external database that we used for this purpose by providing a brief description of it and listing its advantages and disadvantages. In addition to that, we will also explain which parts of the external data source were most useful and how we utilized them in order to achieve our target.

3.1 PhosphoELM

PhosphoELM [4] is a database that contains eukaryotic phosphorylation sites and it is available online at <http://phospho.elm.eu.org>. All database entries are manually curated and the data were collected both from published scientific literature and high-throughput data sets. It provides information not only about the exact phosphorylation sites of each protein, but also the kinases that are responsible for each post-translational modification. Moreover it includes numerous links to bibliographic references and a lot of additional information about interaction partners, structures, etc. From the above description it would appear that PhosphoELM is a good data resource that contains high quality and experimentally verified data and therefore should be one of the primary information sources for the creation of KiPhoDB.

Numerous bioinformatics projects have used the PhosphoELM data set in the past. Some of these are for example the kinase-specific prediction server GPS (Group-based Phosphorylation Scoring) [15], the literature mining rule-based program RLIMS-P [16], the database of tissue and phosphoprotein sub-cellular distribution PhosphoporegDB [17] and many more. This seems to indicate that the PhosphoELM database is widely accepted in the bioinformatics community as a reliable resource of phosphorylation site data and therefore it is safe to use it as one of the primary data sources for KiPhoDB.

We used the latest version of this database (version 8.1), which was released in December 2008. According to the release notes, this version contains 4384 protein entries (2166 Tyrosine, 13320 Serine and 2766 Threonine), with more than 18,000 phosphorylation sites. The vast majority of these proteins and phosphorylation sites come from two species:

human and mouse. The reason why these two species are prevalent in the PhosphoELM database is that they are both used extensively as model organisms in biological research.

One other significant characteristic of PhosphoELM is that for each phosphosite the database reports if the phosphorylation evidence originates from small-scale analysis (LTP - low-throughput) or large-scale experiments (HTP - high-throughput). LTP experiments typically focus on a limited number of proteins at a time, whereas HTP methodologies and techniques, such as mass spectrometry, examine a great number of proteins each time. In the PhosphoELM data set there is a small overlap between phosphosites identified by LTP and HTP experiments [4]. Therefore we had to be very careful while using the data to enrich the contents of KiPhoDB and take all necessary actions so that each phosphorylation site would be inserted only once in our database.

One final characteristic of PhosphoELM that is worth mentioning is that the kinase enzyme responsible for phosphorylation is known for only about 21% of substrates. The latest version of PhosphoELM, that we have used in this project contains more than 250 kinases, but the majority of them refer to a general category of kinase enzymes and not to a specific kinase protein of a certain species. As a consequence, although PhosphoELM is an excellent resource for substrates and exact phosphorylation sites, it provides very limited information concerning the exact kinase enzyme that is responsible for each phosphorylation. In order to obtain this kind of information and insert it into our database, we had to resort to other databases and data sources, which will be described in detail in the following sections.

3.2 NetworKIN

NetworKIN is a methodology for predicting *in vivo* kinase-substrate relationships. Its main aim is to find a systematic way to link experimentally verified phosphorylation sites to protein kinases. In order to achieve this, it exploits the fact that signaling proteins contain various catalytic or interaction domains and phosphorylation or binding motifs that play a major role in protein interactions. Additionally, it also utilizes the ability of certain kinase catalytic domains to phosphorylate particular sequence motifs and contextual information concerning the cooccurrence in the literature, physical association and coexpression of kinases and substrates. The project's website can be found at <http://networking.info> and it allows users to browse and search predictions made by the NetworKIN algorithm.

According to the NetworKIN publication [6], the algorithm augments motif-based prediction by taking into consideration the network of kinases and phosphoproteins, leading to an overall improvement in the accuracy with which phosphorylation networks can be build. In more detail, it tries to predict which protein kinases target experimentally identified phosphorylation sites by combining protein association networks with consensus sequence motifs. At first, the NetworKIN algorithm uses neural networks and PSMs (Position-specific Scoring Matrices) in order to determine which kinase family phosphorylates each phospho-site. Subsequently the STRING database is queried in order to construct a probabilistic protein network for every substrate, which contains a lot of contextual information

that can significantly improve the final prediction. This contextual information includes for example manually curated pathway data, mRNA expression studies, cooccurrence in abstracts and genomic context. At the end, the algorithm produces predictions based on sequence similarity searches against a database of kinase domain sequences. Further information about the exact function of the NetworkKIN algorithm can be found in [6].

We have chosen to use NetworkKIN as one of our data sources because it has a lot of advantages. One of these advantages is that it is able to capture both direct and indirect interactions between kinases and the corresponding substrates, which enables the algorithm to produce non-obvious predictions that would otherwise be very difficult to make. One other advantage of NetworkKIN is its relatively high accuracy, compared to other algorithms that use only consensus motifs. According to [6] the algorithm's accuracy is approximately 64%, which is a 2.5 fold improvement comparing to other algorithms offering a maximum accuracy of 25%. Finally, the third reason that led us into choosing NetworkKIN as one of our data sources is its close collaboration and integration with PhosphoELM. NetworkKIN has been applied to the PhosphoELM data set at the past, producing very accurate results and enabling scientists to construct an approximation of the whole phosphatase and kinase interaction network for humans. Therefore it is essential to include this information in the KiPhoDB database, as it will enrich its contents even further.

The major problem that we had to face while using the NetworkKIN data stems from the fact that this algorithm generates predictions which, although relatively accurate, can be false. Therefore we had to include additional fields in the KiPhoDB database that would store the probability of each prediction being true and thus give a measure of confidence to the database user. It is then the user's responsibility to accept or ignore the prediction. In addition to this measure, we also had to discard some of the data coming from the NetworkKIN database because their significance was rather low. We tried to keep only those data whose quality was high. In this way we tightly controlled the quality of data entering our database and thus we managed to keep the high standards that we initially aimed for.

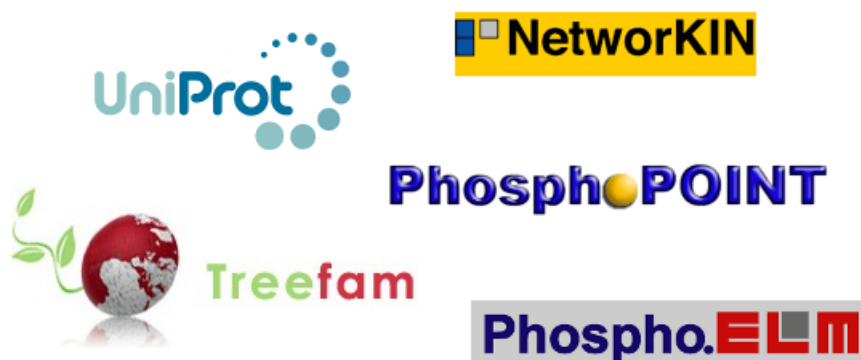


Figure 3.1: Some of the data sources used in this project.

3.3 Phosida

Phosida is a phosphorylation site database which can be found at <http://www.phosida.com>. It contains numerous high-confidence phosphosites from a great variety of species, which have been positively identified in vivo using high resolution mass spectrometry techniques. As its creators argue in [5], the use of high resolution mass spectrometry has an estimated false positive rate of less than one percent. The previous statement proves that Phosida contains very high quality data and therefore should be used in order to populate the KiPhoDB database. Furthermore, Phosida also provides information about the evolution of phosphorylation sites by examining similar phosphoproteins in different eukaryotic and prokaryotic organisms. Finally, it also integrates a phosphosite prediction algorithm, which uses SVM (Support Vector Machines) to predict the existence of phosphorylation sites in non-annotated protein sequences.

The aforementioned phosphosite predictor takes advantage of the numerous in vivo identified phosphosites that the Phosida database contains in order to find novel phosphorylation sites that have not previously been identified. This process produces predictions that can be used to estimate a protein's role in a pathway or its involvement in a signaling cascade and design other biological experiments that will uncover its true functionality and validate or discard the predictions. Finally the prediction algorithm enables users to specify certain precision and recall levels so that the predictions will have a high probability of being true. Of course we did not use the predictor to insert data in our database, because we wanted KiPhoDB to contain only data that have been experimentally proven and not predicted phosphorylation sites or proteins.

Phosida was initially developed in order to store phosphosite information from large scale quantitative experiments and thus it contains many quantitative data which were not relevant to the purpose of this database project. For example, for each protein the database contains its molecular weight, its isoelectric point (pI) and many more information in addition to all phosphorylation sites that have been identified on it [5]. Moreover, for each phosphorylation site Phosida contains its predicted secondary structure, matching kinase motifs and conservation patterns. KiPhoDB utilizes only part of this information, because we did not want to include all this biological context but only the data that was most useful to the database users.

3.4 PhosphoPOINT

PhosphoPOINT is a human kinase interactome database, which aims to provide information and shed light on the interactome of 518 human protein kinases, their potential substrates and interacting partners. It integrates information from three existing phosphoprotein databases, namely PhosphoELM, HPRD and Swissprot, as well as approximately 400 manually curated kinase - substrate pairs. It is constantly updated as soon as new versions of the aforementioned databases are released. At the time of writing, it contains 4195 phosphoproteins with a total of 15,738 phosphorylation sites (10,937 Serine, 2,425 Thre-

online and 2,376 Tyrosine) and it is available at <http://kinase.bioinformatics.tw>. 7,843 of these phosphorylation sites come from high-throughput (HTP) experiments, 6,329 come from low-throughput (LTP) experiments and 679 come from both HTP and LTP screening. The above information will help us judge the quality of the data in the PhosphoPOINT database and decide whether we will use it as a data source for KiPhoDB or not.

In addition to kinase - substrate pairs for human kinases, the PhosphoPOINT database also provides all adequate annotation that helps scientists locate new kinase - substrate pairs using the available protein-protein interaction data sets. Moreover, Gene Ontology (GO) [18] cellular component information and external gene expression data sets are used to identify potential substrates for each kinase. Finally, PhosphoPOINT provides annotation to some of the amino acids located near each phosphorylation site, where a Single Nucleotide Polymorphism can cause the corruption of this site. This information can be very helpful and provide insight on how such alterations lead to the emergence of human disease. For example it has been found [2] that there are at least 64 phosphorylation sites whose change results to disease phenotypes such as schizophrenia and hypertension.

Category	Explanation
1	physically interacting proteins
2	interacting phosphoproteins
3	substrates for a biochemical reaction
4	substrates & interacting phosphoproteins

Table 3.1: The different categories of data in PhosphoPOINT.

According to [2], there are four kinds of links between kinases and their interacting substrates, as it is clearly illustrated in Table 3.1. All interactions available in the PhosphoPOINT database are classified into one of these categories. For the purpose of populating our database, we have chosen to use only categories three (substrates for a biochemical reaction) and four (substrates & interacting phosphoproteins). Categories one (physically interacting proteins) and two (interacting phosphoproteins) contain interacting proteins in general, which means that they may contain proteins that do not undergo phosphorylation but some other kind of interaction. Therefore we decided not to use categories one and two and limit the available dataset to the remaining two categories, which undoubtedly contain substrates that participate in phosphorylation reactions.

3.5 PhosphaBase

PhosphaBase was used in this project as a primary resource for phosphatase information and it can be found at <http://www.bioinf.manchester.ac.uk/phosphabase/>. It currently contains 11445 number of sequences belonging to 1112 organisms [12]. The exact contents of PhosphaBase are illustrated in more detail in Table 3.2. It is an ontology-driven database and the first public resource dedicated to protein phosphatases. The data found inside PhosphaBase have been collected from a great variety of biological sources, such as peer-reviewed literature and other publicly available biological databases (Uniprot, PDB, etc). Moreover, Gene Ontology terms have been used as a means of data extraction, in order to eliminate redundancy in the final data set.

Phosphatase Subfamily	Number of Sequences
Protein tyrosine phosphatases	2598
Dual-specificity phosphatases	1360
PTEN and Myotubularins	233
Serine/Threonine phosphatases	6370
Histidine phosphatases	47
Unclassified serine/threonine phosphatases	154
Unclassified tyrosine phosphatases	683
Total	11445

Table 3.2: The current contents of PhosphaBase.

In general, PhosphaBase contains very useful data of acceptable quality. Although the developers of PhosphaBase have put a lot of effort in eliminating redundancy in the data set, there are some cases where search results will contain duplicated information. This happens because no manual curation is performed on the database and therefore there is no way to delete duplicate entries that have been initially created due to the inability of computer programs to detect and prevent this kind of errors. In addition to that, some false positives have been reported when a protein listed in PhosphaBase is not actually a phosphatase. This can happen when a protein has a specific domain that is normally found on phosphatases but does not exhibit phosphatase behaviour. One positive fact though is that PhosphaBase does not contain any protein sequence fragments and therefore the redundancy problems that fragments may cause have been successfully avoided.

Taking under serious consideration the negative characteristics of PhosphaBase that were presented in the previous paragraph, we decided to include it as one of our data sources for phosphatases. This decision was mainly affected by the fact that there exist

very few databases for phosphatases and thus we had to use all available information for this kind of molecules. Nevertheless, we took all necessary steps in order to actively control the data inserted in KiPhoDB by performing regular checks and discarding those data that appeared to be redundant.

3.6 KinBase

KinBase is a database that contains data for all protein kinase genes found in various species. The species that are currently covered by KinBase are the following: Human, Mouse, Sea Urchin, Fruit Fly (*Drosophila melanogaster*), Nematode Worm (*C. elegans*), *Monosiga brevicollis*, Yeast (*S. cerevisiae*), *Dictyostelium* and *Tetrahymena*. KinBase is available at <http://kinase.com/kinbase/> and it is searchable by kinase group, family, sub-family, gene name and domain. Finally, the KinBase website integrates a BLAST server, which can be utilized in order to perform sequence similarity searches.

In this project we have used KinBase as a supplementary resource for kinase data. One characteristic of the KinBase database is that it contains information about pseudokinases. This category of kinases corresponds to about 10% of the total number of kinases in humans and they are predicted to be non-functional due to evolutionary changes in important amino acids of the protein sequence. Although pseudokinases are inactive and they do not function as proper kinase enzymes, they are thought to retain other roles in the cell that render them essential for its function [19]. Therefore we had to be very careful when using this data source and filter out all pseudokinases, because they are not relevant to our targets in this project and thus they should not be included into the KiPhoDB database.

3.7 NetPath

NetPath is a curated data source that contains valuable information about signal transduction pathways in humans. It can be found at <http://www.netpath.org> and its current contents are listed in Table 3.3. At the moment it contains a total of twenty pathways, which are freely available to download. Ten of these pathways concern immune signaling, whereas the remaining ten are cancer signaling pathways. The data contained in NetPath are mainly from humans, but the database contains information from other mammals too. Until now signaling pathways have been studied with focus on isolated members or individual reactions and no comprehensive view of the pathway as a whole was possible. Netpath's main characteristic is that it provides a global view of pathways, including many details of protein-protein interactions, enzyme catalysis, translocation of proteins etc. Moreover, its main source of data is HPRD, the Human Protein Reference Database.

We exploited this website by downloading the available NetPath data set in BioPAX format and trying to identify phosphorylation and dephosphorylation reactions between proteins in various pathways. Using this information we managed to pinpoint certain kinase-substrate and phosphatase-substrate pairs and identify kinases and phosphatases

Category	Contents
Curated Pathways	20
Molecules	963
Physical Interactions	1649
Genes Transcriptionally Regulated	6137
Transport	148
Enzyme Catalysis	729
PubMed Citations	10209

Table 3.3: The current contents of NetPath.

that act on the same substrate. However, as expected, the number of phosphatase-substrate pairs that were finally discovered, were far less than the kinase - substrate pairs.

3.8 TreeFam

TreeFam is a phylogenetic tree database which can be found at <http://www.treefam.org> [20]. It consists of two distinct parts. The first part is named Treefam-A and it includes manually curated phylogenetic trees for 1314 gene families. In contrast, the second part consists of automatically generated phylogenetic trees for 14351 families and it is called Treefam-B. A gene family can be thought of as a group of genes that have high sequence similarity because they have descended from a single gene in the last common ancestor. In order to populate the KiPhoDB database, we used the fourth version of Treefam which includes 25 fully sequenced animal genomes, along with four genomes from plant and fungal species.

Phylogenetic trees offered by TreeFam are very useful for scientists because they can be exploited in order to identify orthologous (originating from speciation) and paralogous (originating from duplication) genes. Therefore useful assumptions can be formed about the relationship between corresponding genes in different organisms, which can help us shed some light into the evolutionary history of genes and gain a deeper understanding of the evolution of organisms and their genomes. One other way to obtain this kind of information is by using BLAST matches (Inparanoid, KOG) or BLAST matches and synteny (HomoloGene, Ensembl-Compara). As the creators of TreeFam claim in [21], tree-based inference of paralogs and orthologs is a much more robust and accurate technique for many reasons. The primary one being that pair-wise BLAST scores are greatly affected by the difference of evolutionary rates between members of the same gene family. On the other hand, phylogenetic trees do not have this problem and they are much more informative

because they can visually illustrate the history of a whole gene family. Finally, another positive aspect of phylogenetic trees is that they offer scientists the opportunity to compare gene trees and species trees and draw useful conclusions.

We have used the TreeFam data source in order to obtain information about kinase and phosphatase gene families. As noted previously, TreeFam contains two separate data sets: TreeFam-A and TreeFam-B. The trees included in TreeFam-B are automatically generated and therefore they are often incorrect because of poor data quality and certain faults in the tree generation algorithms. For more details about the algorithms that TreeFam uses in order to automatically produce phylogenetic trees, infer orthologue or paralogue relationships and classify genes into families can be found in [20]. Since the data in TreeFam-B data set are prone to errors, we decided to only use TreeFam-A. Thus we only inserted manually curated trees in our database in the hope of conserving the relatively high quality standard of the data provided by KiPhoDB.

3.9 Other resources

Apart from the data sources that were described in previous sections, we also used a plethora of other resources in order to populate the different tables of our database. One of them was the Reactome [22] database, which contains curated core reactions and pathways in humans and can be found at <http://reactome.org>. We decided to use this resource because its data are manually curated by experts and specialized staff and therefore their quality is very high. Another valuable resource is dbPTM, a database that contains experimentally verified protein post-translational modifications from several databases. It also includes annotation of predicted post-translational modifications on proteins from SwissProt and can be found at <http://dbptm.mbc.nctu.edu.tw>. We used the information stored inside dbPTM by extracting the post-translational modifications that occur due to phosphorylation or dephosphorylation. Subsequently this data was inserted in the KiPhoDB database.

Our primary resource for pathway information was the NCI-Nature Pathway Interaction Database [23]. This database was created by the Nature Publishing Group and it is constantly reviewed by experts in the field. Therefore it contains high quality pathway data that we decided to input in the KiPhoDB database. Finally, the Kyoto Encyclopedia of Genes and Genomes (KEGG) [24] is a well known database that contains among others pathway and molecular interaction information. All KEGG data are manually curated and therefore KEGG proved to be an excellent resource for pathway and reaction information.

Chapter 4

Implementation

4.1 Software Tools

In order to meet the objectives that were set at the beginning of the project, we used a wide variety of software tools. These software tools include database management systems, web frameworks, web page design software, many visualization tools, as well as pieces of software that we have written ourselves. In the following paragraphs, the various software tools used and their properties will be illustrated.

4.1.1 MySQL

We used the MySQL relational database management system (RDMS) because of three main reasons. The first reason is that MySQL is an opensource project. Its source code is licenced under the GNU General Public Licence and therefore this gives us the right to download MySQL and use it free of charge. Moreover, MySQL is probably the most popular database management system for web applications and several high traffic websites (e.g. Wikipedia, Nokia, Google, Amazon.com and many others) use MySQL in order to store and retrieve data. Finally, MySQL provides a lot of command line and GUI tools for easy administration and coding.

4.1.2 Python

We chose to use the Python programming language for the development of the KiPhoDB web site because of the reasons presented bellow:

- Python is a powerful and widely used language, well suited for rapid development. It is pretty easy to pick up by people that do not have prior experience in computer programming, which enables them to learn the basics of this language and contribute to the development of the project in a relatively short period of time.
- Compared to other programming languages, such as C++ and Java, Python is far less verbose. It is certainly more high-level than C++ and it takes care of memory

management. Moreover, it does type checking at runtime rather than at compile time, which means that there is no need to declare types explicitly. Both C++ and Java do not offer this functionality.

- Python is truly object-oriented (even functions are objects) and it provides a very neat way of declaring and passing function arguments, making functions much more generic than in the aforementioned programming languages.
- Python excels at parsing (BeautifulSoup, re module for regular expressions) and HTTP access (httplib, urllib).
- Compared to other scripting languages such as Perl and Ruby, Python is also far more approachable and syntactically more aesthetic. Perl is weakly typed which can be a source of confusion for various reasons. Ruby would be an interesting alternative but it is unfortunately not widely used or well supported.
- Choosing Python allowed us to make use of the brilliant Django-Python framework for rapid database and web site development.
- One other useful feature of Python is the BioPython library, which provides many tools for automatic data extraction from numerous biological databases (UniProt, PubMed, and many more). Unfortunately, there were a lot of cases when BioPython did not have an appropriate interface for the specific databases we wished to use. In these cases we had to create our own custom interfaces and parsers, which was not a very difficult task due to the fact that Python is particularly well suited for these sort of jobs.

Based on the above discussion, we believe that Python was the ideal programming language for the development of the project.

4.1.3 Django-Python

Django is an open source framework for creating web applications. It is written in python and its source code is released under the BSD license, which gives us the right to download and use Django free of charge. Django's primary target is to ease the creation of database-driven websites. It can cooperate flawlessly with MySQL databases and offers an automatically generated administration interface, which allows users to log in and change entries in the database. For the aforementioned reasons, we chose to develop KiPhoDB using this framework. In more detail, we have set up a server using Linux, Apache, MySQL and SSH, which enables each team member to log in and work on the project. Finally, this server acts also as an experimental server which has helped us to experiment with python and Django in order to accomplish our goals.

4.1.4 Dreamweaver

In order to design the final website, we used the web development application "Dreamweaver". Dreamweaver is a software product from Adobe and it supports a great variety of web technologies, such as HTML, CSS, JavaScript, ASP and many more. Nevertheless, only the first two of these technologies were used for the implementation of the main website of KiPhoDB.

4.1.5 PhyloWidget

PhyloWidget is a web-based tool written in Java that enables users to visualize and manipulate phylogenetic tree data [25]. Its source code is released under the GNU General Public Licence, which gives us the right to use this application free of charge. We chose to use PhyloWidget and not one of the mainstream phylogenetic visualization and manipulation software tools (e.g. Phylip, PAUP, Mesquite, TreeView) because all these tools are ill-suited for integration with databases and online use. On the contrary, PhyloWidget offers online tree visualization capabilities, it integrates well with the web environment and it is very easy to configure and use.

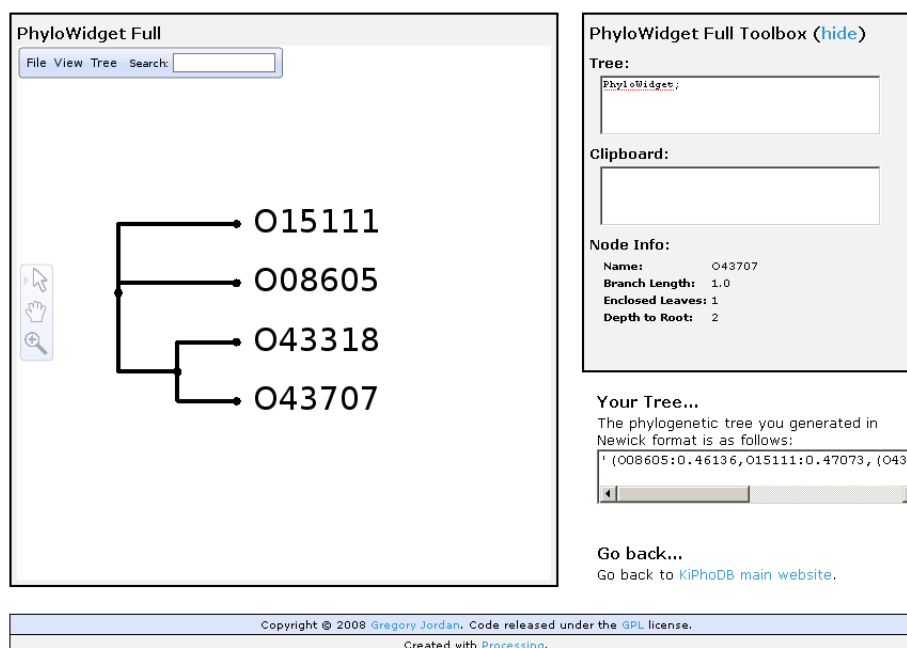


Figure 4.1: The PhyloWidget tool within KiPhoDB's website.

In more detail, PhyloWidget offers a simple and powerful user interface that allows users to navigate and manipulate phylogenetic trees by zooming in certain areas, changing the topology of the tree, calculating the distance between certain nodes, copying and pasting subtrees, rerooting the tree and editing the node label or branch length. When the user

clicks on any node, a contextual menu appears that enables users to execute any one of the aforementioned actions. Moreover, the web interface offers two text fields that are constantly updated and contain the Newick representation of the tree and the clipboard contents. This feature allows users to easily copy and paste tree string representations from other sources and programs. Finally, PhyloWidget is able to parse trees in many formats, such as Newick, NHX and Nexus, and it can output trees as images (JPEG and PNG) or PDF files.

One other reason that led us to choose PhyloWidget was its speed and responsiveness even when the program had to visualize a relatively large tree. For example, a tree with 3000 nodes can be displayed at 30 frames per second, allowing the user to easily navigate through it [25]. PhyloWidget is the ideal applet for online tree visualization because it is written in Java and therefore it is very fast and efficient and it can be easily and seamlessly integrated in the KiPhoDB web site.

4.2 Development Server

One of the primary tasks that we had to accomplish was to find a development server in order to set up our web site. This task was very important because we wanted to have a unique place where our web site and database would be stored, instead of having multiple copies of it in each group member's personal computer. If we had chosen to have multiple copies of the website, these copies would soon get out of sync and it would be very difficult in practice and time consuming to reconcile all the differences and produce a final output. On the contrary, setting up a development server offers the possibility of using a version control system, which enables each developer to download the changes that other developers have made, make his/her own contribution to the code and finally commit the new changes to the server.

In order to create a development server, we had to choose one out of the following two options. The first option was to contact the Bioinformatics Support Service of the Bioinformatics Centre or the Computing Department and ask them to set up a development server for us. Unfortunately, this means that we probably would have to wait many days or even weeks before we could get the development server and start using it. Of course this is a serious disadvantage, because the project had a very tight deadline and the time that we had to develop, test and debug it was limited. Therefore we could not waste any valuable time waiting for the construction of the development server from a third party.

One other crucial disadvantage of the aforementioned option is that we would not have any control over the server itself and the software that was installed in it. This means that we would have to contact the administrators of the server whenever we wanted to use an additional Python library or another program and ask them to download and install it on the server. Of course, such a procedure would introduce even more delays and waste of precious time. Moreover, the administrators would be very hesitant to install new software, because they would be concerned about its compatibility and the security issues related to it. Finally, we would have no guarantee concerning the uptime of the server.

The second option was to use the knowledge, skills and expertise of the group members in order to set up a development server of our own. We chose this option because it enabled us to avoid all previously mentioned negative aspects and save valuable time. In addition to that, using a self-made development server allowed us to have ultimate control over the hardware, software, compatibility issues and security issues that the creation of a server involves. The hardware and software components shown in Figure 4.2 and listed below were utilized in order to create our development server:



Figure 4.2: Software components of the development server.

- **Hardware**

The computer that we used to set up the server was Christos' laptop with the following hardware characteristics: 1248 MB RAM, 3.06 GHz Mobile Intel Pentium IV processor and 250GB HDD. The laptop was also connected to the internet through a home router and a high-throughput connection. Therefore we had to make all appropriate changes to the router in order to allow incoming traffic to certain ports and filter all other malicious internet packets.

- **Operating System**

We chose to use Arch Linux (Kernel version 2.6.27) instead of Windows XP or Vista because Linux is a much more stable and reliable operating system in comparison to Windows. Moreover, Arch Linux is a very flexible Linux rolling distribution, which means that it is not under a versioning system and the user can install cutting edge software at any given time without worrying about having an outdated version of the system. Finally, a user account was created for each member, which could be mainly used for programming purposes and for uploading all necessary files that allowed the server to operate flawlessly.

- **Database Management System**

As it was mentioned in previous sections, MySQL was the database management

system that we chose for the development of KiPhoDB. The latest version of MySQL was installed on the server for this purpose: Version 14.12, Distribution 5.0.75.

- **Django**

Django version 1.0.2 was also set up on the server. Moreover, we also had to install numerous python libraries that proved to be very useful for data mining, string matching, xml and html parsing, timing and many more (BeautifulSoup, BioPython, xlrd, etc).

- **HTTP Web Server**

Apache is a powerful web server that has been chosen to host KiPhoDB. The version used was 2.2.11 and we also installed various modules that enabled Apache to dynamically execute python scripts and display the results as html files (mod_python 3.3.1, Python 2.6, mod_ssl 2.2.11, etc).

- **SSH**

The SSH protocol is widely used for establishing a connection to a remote server and executing commands. We installed SSH to the server, so that the users could connect to it and either submit their work or execute various python scripts for updating the database.

- **Version Control**

The version control system we used is called SubVersion (SVN) and we installed the latest version of it, namely 1.6.0. This system proved to be very useful, because it provided a unified resource for the source code of KiPhoDB. Therefore each one of us was able to commit his/her work on the server so that all other group members could view, test, debug and build upon it. By using SVN we managed to avoid having multiple versions of the source code in our computers, which of course would make it very difficult to combine them at the end and form the final web site.

- **URL**

Of course a proper web server should have its own URL, which can be used by the web site's users in order to easily access it. The problem we had to face in this case was that the internet connection of the development server did not offer a static IP. On the contrary, each time the connection was lost or the router was restarted, a new IP address was assigned to the computer. Therefore we had to use dynamic DNS services, such as the ones offered free of charge from the company DynDNS.org, in order to book the following url: dbproject.dyndns.org.

In conclusion, the creation of our own MySQL and web server has proven to be a very wise decision, because of the reasons that were mentioned previously in this section. Furthermore, until now the development server has worked flawlessly and it has provided us with all necessary resources to work on the development, testing and debugging of KiPhoDB. At the time of writing, the server has been up and running for 34 days, 5 hours and 28 seconds, having an overall uptime of more than 99%. Nevertheless, once KiPhoDB

is complete, it should move to a proper production server, which will provide additional security features and functionalities. Moreover, the production server will have greater processing power and more memory, so that it can process simultaneous requests of many users.

4.3 Database Schema

Creating the KiPhoDB database schema was one of the most difficult parts of this project. This is because the database schema should reflect as accurately as possible the different biological structures and processes found in nature. Additionally, it should contain all necessary tables and fields that would provide adequate storage space for all available biological data. One other aspect of the Database Schema that made its creation a challenging task is that it should be normalized and strictly follow at least the third normal form. We believe that at the end we managed to create a very robust but yet flexible database schema that provides users with all useful information about kinases, phosphatases and their substrates. At this point we would like to acknowledge the support of Prof. Michael Stumpf and Dr. Frances Turner, who helped us significantly in the process of creating the database schema for KiPhoDB.

It was previously mentioned that, in general, a database schema should be normalized and follow at least the third normal form. The reason for this is that non normalized databases have many significant disadvantages compared to properly normalized databases. Therefore active steps must be taken in order to ensure that the database we create is properly normalized. Perhaps the most significant property of normalization is that it makes a database much more flexible and considerably less prone to errors. Moreover, it makes the data model more informative to the users, successfully avoids modification anomalies and minimizes the redesign efforts in cases when we want to expand the database. Based on the above discussion, we worked hard to produce a normalized database schema and we believe that at the end our efforts were successful.

The database schema of KiPhoDB is illustrated in Figure 4.3. As it is shown in this figure, KiPhoDB contains seventeen tables in total. The main table of our database is the *Protein* table, where all kinases, phosphatases and substrates are stored. Since all these three molecules are actually proteins, it is not advisable to create three different tables that will consist of exactly the same fields. Instead, it is much better to have one table where all this information will be stored and easily retrieved afterwards. All interactions between the proteins of this table are stored in another table named *Reaction*. As the name implies, the Reaction table stores information about reactions between pairs of proteins and various details about each reaction. These can be both phosphorylation and dephosphorylation reactions which can lead either to the activation or deactivation of the corresponding substrate. Furthermore, each reaction is considered to be part of a specific pathway and therefore a foreign key relationship exists between the Reaction and the *Pathway* tables. Needless to say that the Pathway table includes further information about pathways in different species.

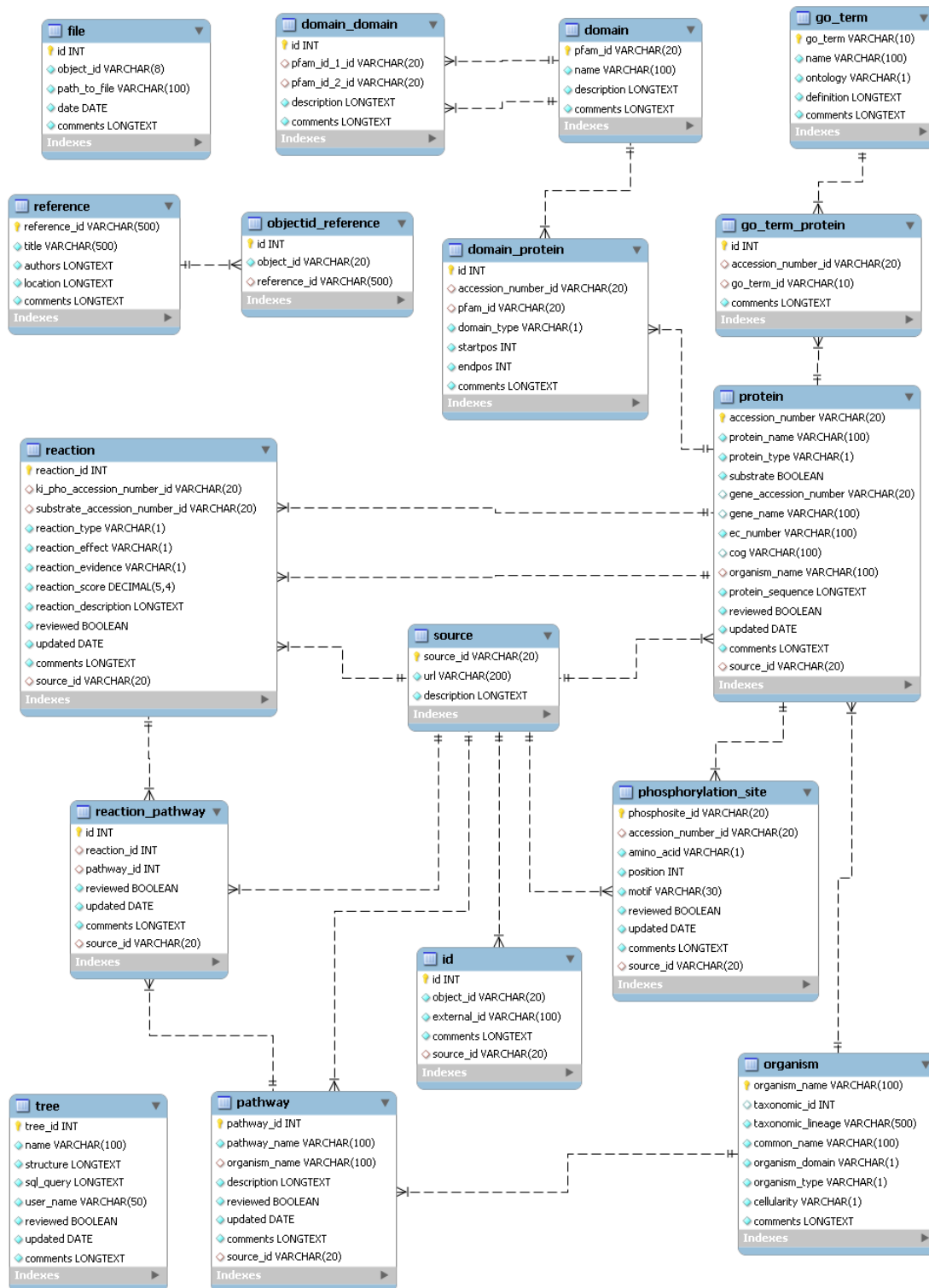


Figure 4.3: The Entity Relationship (E/R) diagram of the KiPhoDB database.

For every protein in the Protein table, its phosphorylation sites, domains and GO terms are stored in different tables. This information is very important because a researcher needs to know the exact position in the amino acid sequence where the phosphorylation takes place, the various domains that a kinase/phosphatase/substrate contains and its GO term annotation. All these data are stored in tables *Phosphorylation Site*, *Domain* and *GO term*, whose contents are self explanatory. These three tables have strong interconnections with the Protein table and therefore the appropriate foreign key relationships have been defined in the database schema. The connection between the Protein table and the Domain and GO term tables is a many - to - many relationship and therefore additional tables have been created that implement these relationships. In this way our database is normalized and has all the advantages presented previously in this section. On the contrary, the connection between the Protein table and the Phosphorylation Site table is an one - to - one relationship and therefore the creation of a simple foreign key on the Phosphorylation site table has been sufficient.

In addition to the tables mentioned above, we have also created some tables that extend the functionality of our database. These tables are named *Reference*, *File*, *Tree*, *Organism* and *ID*, and store references, files, phylogenetic trees, organisms and external IDs respectively. Including this kind of information in our database is vital because in this way our database is enriched significantly and it also becomes more comprehensive.

4.4 Data Input

In order to populate the KiPhoDB database we had to utilize a great variety of biological data sources. All available data sources had to be thoroughly investigated, so that we could estimate the quality and quantity of data they offered. At the end we reviewed the available information about each source and decided to use only those that provided easily retrievable data of high quality. These sources and some of their main characteristics are summarized in the third chapter of this report. Here we will provide the reader with a brief description of the data extraction techniques that we used in order to retrieve data from those sources and successfully insert them in our database.

The most common way to extract data from a resource is to write a parser. The main aim of the parser is to read the data from a database, a flat file or the internet and convert these data into the appropriate format so that they can subsequently be easily inserted in the database. In essence, a parser is a tool that is used to change the data container, but leave the data themselves intact. Of course, the data from various sources were in different formats and therefore we could not use a single generic parser. On the contrary, we had to create one or more parsers for each data source. In the following, an overview of the various data formats that we had to deal with and the techniques that we used to extract interesting data are presented.

- **BioPAX**

BioPAX stands for Biological Pathway Exchange and it is an emerging format for representing molecular interactions, gene regulation networks and signalling pathways.

Comparing to other established formats, BioPAX offers a better representation of various physical states of the protein and contains more information about the protein itself (amino acid sequence, name, external references and other features). Furthermore it offers the ability to accurately represent gene regulation networks and gene expression regulators, such as transcription factors, microRNA, etc. Finally, the BioPAX format is able to include information about degradation substrates in various pathways and various genetic interactions which are important for mapping pathways. We encountered this format while manipulating the data from NetPath, a curated database for immune and cancer signalling pathways created by HPRD (Human Protein Reference Database).

- **HTML/XML**

The KiPhoDB database contains a lot of information from UniProt, which offers its data in HTML or XML format. These two formats are some of the simplest formats available and we had to create parsers that would extract the desired information and input it into our database. Fortunately there are a lot of libraries for the Python programming language that enable programmers to easily parse such files. One such library is DOM, which stands for Document Object Model and it provides us with all necessary methods for accessing and manipulating XML documents. DOM presents a tree view of the document under examination, with a root node, many child nodes and the leaves. In every case, the root node is used to read, write and modify each one of the child nodes. Our parser scripts made extensive use of the minidom Python library, which is a lightweight implementation of the DOM interface. Its main characteristics is its simplicity and flexibility compared to the full DOM library, which enabled us to quickly and easily learn how to use it and start writing the parsers without wasting any time.

One other library that we used for parsing HTML files is called BeautifulSoup. This is a very powerful Python library which offered us the ability to search for keywords in HTML documents using the appropriate regular expressions. Subsequently we were able to extract the required information from any HTML file and insert it into the appropriate table of our database. BeautifulSoup is ideal for screen scraping and it offers a wide variety of powerful data extraction features, which made it the ideal library for our purposes. However, the main drawback of BeautifulSoup is that it is much slower than other available HTML parsers, but overall it has many positive characteristics and this is the reason why we chose to use it. Its main use was to extract pathway data and subsequently update the appropriate tables in the database.

- **XLS**

Some data sources, for example PHOSIDA, offered their data in xls format. This format is used by ExcelTM, a very popular spreadsheet product from MicrosoftTM. In order to parse this kind of files we had to use the xlrd Python library, which allows developers to read, write and extract information from files having the xls format.

The main strength of this library is that it is written purely in Python and it can manage unicode encoded xls files. Moreover it offered us the possibility to read each sheet of the file separately and thus manage to get the required information quickly and easily.

- **TSV/CSV**

PhosphoELM and a few other data resources gave us flat files that contained information in TSV (Tab Separated Values) or CSV (Comma Separated Values) format. As the name implies, these files contained fields that were separated either by tabs or by commas. This kind of files can be easily imported in the database using Python's build-in split function, which returns a list with all the contents for each line. Therefore flat files in TSV or CSV format were not a particular problem for us and we were able to create all necessary parsers for them.

At this point it is useful to note that we also made extensive use of the Biopython library for Bioinformaticians. This library provides a wide variety of tools and modules that enable every Python programmer to acquire biological information from a variety of online sources. One of its main functionalities is that it can automatically connect to NCBI, EBI, UniProt and many more online resources of data, download information and make it available to the programmer. In addition to that, it also provides easy to use interfaces to common bioinformatics programs, such as NCBI Blast. It can perform sequence translation, transcription and weight calculations. For the purpose of this project we only used a subset of the BioPython features in order to automatically retrieve protein information from UniProt and update KiPhoDB's entries accordingly. Finally, we also used BioPython to fetch information and details from NCBI about each PubMed entry in our database.

4.5 Web Site

KiPhoDB's central web site is presented in Figure 4.4. As shown in the picture, the web site's design is simple but at the same time flexible and powerful. Its main purpose is to provide users with all necessary software tools in order to enable them to easily and effectively exploit all data contained in our database. Moreover it also includes a lot of information about the database itself, its contents and legal information about its usage. Finally, the web site aims to provide a pleasant and productive environment which users can use in order to find answers to their research questions.

Table 4.1 further illustrates the internal structure of the web site and the pages that it consists of. The table itself is pretty explanatory and it provides the URL of each page, its title and a brief description of its contents. Therefore in the following paragraphs we will only focus to some aspects of the website that deserve further explanation.

One of the aspects that should be mentioned and analyzed concerns the legal status of KiPhoDB. We have chosen to release KiPhoDB under the Creative Commons Attribution + Noncommercial + NoDerivs license, a graphical representation of which is shown in Figure

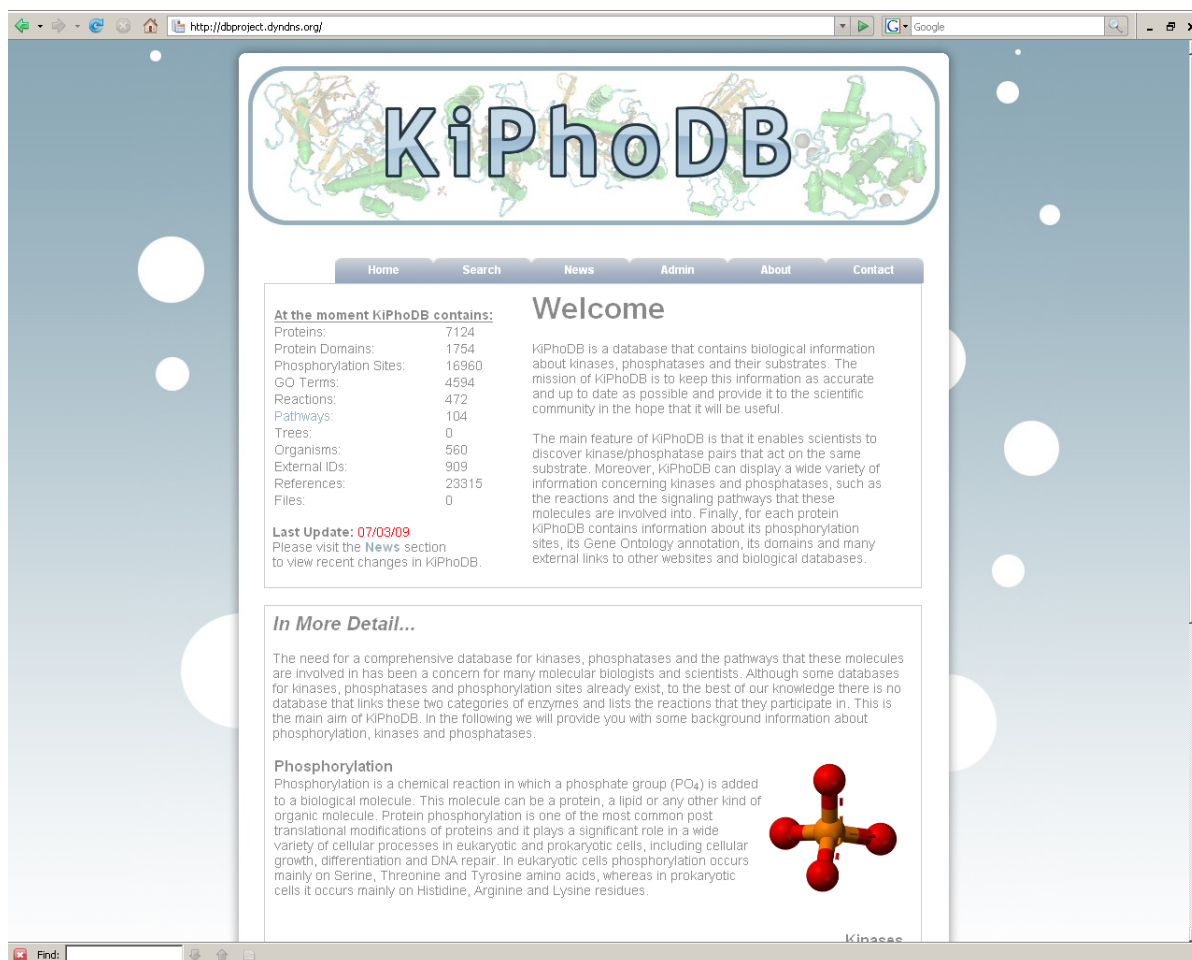


Figure 4.4: The KiPhoDB website.



Figure 4.5: KiPhoDB's license

URL	Name	Description
/index.html	Home	A welcome page which provides some statistics about the contents of KiPhoDB and theoretical information about kinases, phosphatases and substrates.
/search.html	Search	This page can be used to access the various database mining tools provided by KiPhoDB. These tools enable each user to extract information from the database or create phylogenetic trees based on KiPhoDB's data.
/news.html	News	All news, latest updates and releases concerning the KiPhoDB database or the web site are made public in this page.
/admin/	Administration	The administration interface of KiPhoDB. The user name and password is required in order for a user to have access to this part of the web site.
/about.html	About	This page provides information about KiPhoDB, its creators and the technologies used. Moreover, it provides a link that enables users to download the database and install it locally on their computers.
/contact.html	Contact	Legal information and contact details are provided in this page. In order to leave a comment, users can either send an email to the administrators or fill in and submit an online form.
/results.html	Results	When a user performs a simple search, all results are analytically displayed on this page.

Table 4.1: The structure of KiPhoDB's website.

4.5. This means that each user is free to copy, distribute and display the information stored into the KiPhoDB database and its public website under all legislations, provided that he/she will give credit to the creators of KiPhoDB. Nevertheless, users do not have the right to use the contents of KiPhoDB for commercial purposes or distribute a modified version of it. A user can do this only if he/she contacts the developers of KiPhoDB and gets a written permission to do so. On one hand, this license gives users all necessary permissions to use KiPhoDB free of charge and without violating any copyright laws, but on the other hand it restricts their ability to use KiPhoDB for commercial purposes or distribute a modified version without our prior written permission. These two characteristics of this license make it ideal for the purposes of KiPhoDB and therefore we decided to adopt it. Of course, similar licenses have commonly been used in a wide variety of Bioinformatics websites and databases.

Every database should ideally ask for user feedback in order to improve the quality of the data that it contains and the services that it offers to the public. We believe that user feedback is a very powerful tool that will help us improve KiPhoDB's services and data. Consequently we encourage users to report any errors they find and send us comments,

ideas and suggestions. This can be achieved either by sending an email to KiPhoDB’s email address or by completing an online form that we have set up solely for this purpose. The online form has the following fields: Name, email, URL and Comments. Once the form is completed and the button “Post” is pressed, the information is stored into the database itself and the administrators can view it at their own time.

One other characteristic of the website that should be mentioned is that it offers users the ability to download the entire KiPhoDB database and set up a copy of it in their local servers. This feature can be found in the “About” page, along with a short description of the database itself, the software technologies that have been used to create it and short biographies of the people behind this project. KiPhoDB data are offered in a single `.sql` file which was created directly from the MySQL database using the program *mysqldump*. The size of this file can be tens of Megabytes and this is the reason why we have to use compression in order to make it smaller. Once the file has been successfully downloaded from KiPhoDB’s web server and properly extracted, it is pretty straight-forward to import it in the local MySQL server and use it. This characteristic of the website is very important because it allows users to work on their own local copy of KiPhoDB instead of sending their queries to KiPhoDB’s server and thus increasing its load. Nevertheless, it should be noted that the sql file has to be updated regularly so that it can reflect all recent changes made to KiPhoDB.

The administration interface can be accessed through the “Admin” button and it is one of the most important parts of the website. Its main function is to allow authorised users to view, modify and delete the contents of our database, primarily for maintenance purposes. Public users are not allowed to perform such kind of operations, but only to view the contents of the database and combine the information stored in different tables in order to produce answers to their queries. On the contrary, a small number of individuals will have administrative access, which means that they will receive a certain user name and password combination in order to be able to use the administration interface. These individuals will be charged with the duty to routinely maintain the contents of the database, read the feedback provided by KiPhoDB users and take all appropriate steps to improve the data and the services provided. As part of the routine maintenance, which should be performed on a regular basis, we have created automated Python scripts that can update certain database tables in a matter of minutes. These scripts have been proven to be very useful and they have significantly simplified the task of maintaining the contents of KiPhoDB up-to-date.

4.6 Database Search Tools

The “Search” page of the website hosts all software tools that we created in order to enable every user to search the KiPhoDB database and find all necessary information. We believe that it is very important for every database to provide users with a plethora of powerful tools that will give them the opportunity to exploit the database in every possible way. Therefore we created seven distinct software tools, which will be examined in more detail

in the following sections. Each one of these tools has its own strengths and weaknesses and can be used for different purposes. Additionally, the user can combine two or more of these tools in order to get the information that he is looking for. In the next sections we will provide the reader with more details about each tool, its main features, strengths and weaknesses.

4.6.1 Kinase - Phosphatase Pairs

Kinase - Phosphatase Pairs

A kinase - phosphatase pair acts on the same substrate. Please enter the kinase or phosphatase accession number of your protein in the text box below. Make sure that the accession number belongs to a kinase or phosphatase, because otherwise there will be no results.

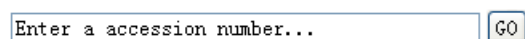


Figure 4.6: Search Tool for Finding Kinase-Phosphatase Pairs

A kinase – phosphatase pair is a combination of a protein kinase and a protein phosphatase that act on the same substrate. Finding kinase – phosphatase pairs is very interesting for biologists and one of the primary objectives of this project. The Kinase – Phosphatase Pairs Tool has one purpose: to enable users to identify kinase – phosphatase pairs based on a given kinase or phosphatase protein molecule. The user simply has to input the accession number of a protein kinase or protein phosphatase that he/she is interested in. Subsequently the tool performs a search in the Reaction table of the database and tries to find reactions that involve the given protein. In this way, all substrates can be positively identified and the corresponding kinase/phosphatase molecules can be found based on the common substrate. As a result, users will be provided by a list of kinase-phosphatase pairs along with their corresponding reactions. If the protein that the user has input is neither a kinase or a phosphatase, then this tool will not produce any results.

4.6.2 Simple Search

The main purpose of this tool is to enable users to perform simple searches in the database. It is clearly illustrated in Figure 4.7 that this tool consists of a text box, where the user can insert one or more keywords, nine checkboxes, where the user can select the tables that will be examined and one button to initiate the search. Once the button has been pressed, the server executes a script which searches all preselected tables of the database for the given keyword. The results of this script are subsequently displayed at a different page. Of course it is possible to select more than one check boxes in order to perform a keyword search in more than one tables. This will display all entries in the specific tables that contain this keyword.

Search KiPhoDB

Please write a keyword on the textbox below and press the button "GO". The website's search engine will perform a database search using the given keyword and it will display the final results on a new page. The search will be performed in all preselected fields: Protein, GO terms, Protein Domains, Phosphorylation Sites, Reactions, Pathways, Organisms, Trees, References and Files.

☐ Protein

☐ Domain

☐ Phosphorylation Site

☐ Reaction

☐ Pathway

☐ Organism

☐ Tree

☐ GO term

☐ Reference

Figure 4.7: The Simple Search tool.

For example, if a user wants to find how many pathways in the database are involved in signalling, he/she will check the "Pathway" check box, write the word "signalling" into the textbox and initiate a database search by clicking on the "GO" button. This will display a new page containing the results of the investigation, namely the pathways that contain the keyword "signalling". The results contain among others the Oxidative stress response pathway, the Opioid Signalling pathway and the Integrin signalling pathway. Clicking on any one of these pathways, the user can view more details about it: the organism that this pathway appears in, a brief description, some comments and the reactions that are involved in it. Clicking on any of the aforementioned reactions, the user can display more information about the reaction itself and the proteins that take part in it. For each one of these proteins, the user can also find out which other reactions these proteins participate in, the exact location of the phosphorylation sites in the amino acid sequence and much more information.

The example of the previous paragraph clearly illustrates the fact that the simple search tool provides an easy way to find proteins, domains, phosphorylation sites, pathways, reactions, etc related to a specific keyword. Moreover, the interconnections that exist between the various entities of KiPhoDB can help users investigate any potential connections between the objects of their queries and other objects in the database. This can be very useful when, for example, one is trying to identify which phosphatase dephosphorylates a substrate that was previously phosphorylated by a given kinase and vice versa. Using the database, one can create a list that contains all substrates that a kinase phosphorylates and then find the corresponding phosphatases by searching the Reaction table for the inverse reactions.

As a conclusion, the Simple Search tool is very useful in cases when a user wants to find out if a certain protein, phosphorylation site, pathway, etc exist in KiPhoDB and what other information the database has for this object. Unfortunately, this tool does

not support advanced database searches and finding answers to more complex queries by combining more than two database tables. For this purpose we have developed other more sophisticated tools, which we will describe in the following sections.

4.6.3 Advanced Search

The Advanced Search Tool extends the functionality of the Simple Search Tool by providing the user with more options and possibilities. Its main purpose is to provide visual aids to users in order to enable them to easily and quickly construct their own SQL Queries and run them on the MySQL server. The tool itself consists of an extensive list of all available tables in the database. Once the user selects one or more of these tables, these tables get activated and an extensive list of each table's fields appears. As shown in Figure 4.8, this list can be used in order to apply filters and determine the exact output of the query.

At the beginning, the user has to choose which fields from each table should be included in the final results. This can be achieved by checking the appropriate check boxes under the column "Display" that correspond to the fields of his/her choice. Subsequently the user has to define some filters that will restrict the output and display only the necessary information. If no filters are defined, the Advanced Search tool will display all information available, without applying any filters. In order to set up a filter, the user has to utilize the text field under the column "Filter" and next to the name of the field he/she is interested in.

In this paragraph we will examine a comprehensive example of the procedure mentioned previously. Let us assume that a user wants to display the accession numbers and names of all proteins that contain a domain with Pfam ID PF10584. In this case, he/she would have to follow this procedure: The first step would be to select the display check boxes located next to fields "Accession Number" and "Name" of the "Protein" table. Then he/she would have to set up a filter on the "Pfam ID" field of the "Domain" table by writing "PF10584" in the appropriate text box. Finally, after pressing the "Advanced Search" button at the bottom, the tool would construct an SQL query and display it to the user along with the results.

In overall, the Advanced Search tool provides users with a lot of flexibility, since it enables them to combine different filters and fields in order to get the desired results. Moreover it is ideally suited to users that do not have any previous experience with SQL, because it automatically constructs the SQL query for them based on their choices. As mentioned before, the output of this tool consists of both the generated SQL query and the results of running it on the server. If a user is not completely satisfied by the SQL query that this tool automatically generated for him, then he can modify it and run it again by using the SQL Query tool described in the following section.

4.6.4 SQL Query

As shown in Figure 4.9, the SQL Query Tool consists of a text area where the user can type in SQL commands and a button to execute the query. Once the user enters an appropriate

Advanced Search

Please select the one or more of the tables shown in the list below to activate it. Then you can choose which fields from each table you would like to include in the final results and which filters should be used to get the desired output. By clicking on the *Display* checkbox at the left side of each field you can select it or deselect it for displaying in the final output. Moreover, by using the text field *Filter* and the checkbox *NOT*, you can set up appropriate filters to get the output of your choice.

The Advanced Search tool provides you will a lot of flexibility, since it enables you to combine different filters and fields in order to get the desired results. The server translates your choices into an SQL query which is subsequently executed on the server. The SQL query itself and the output of its execution will be displayed in a new page. If you are not completely satisfied with the query, you can modify it and use it in the SQL Query tool below.

+ **Protein**

+ **Reaction**

+ **Pathway**

+ **Phosphorylation Site**

- **Domain**

Display	Field	Filter	NOT
<input type="checkbox"/>	PFAM ID	<input type="text"/>	<input type="checkbox"/>
<input type="checkbox"/>	Domain Name	<input type="text"/>	<input type="checkbox"/>
<input type="checkbox"/>	Description	<input type="text"/>	<input type="checkbox"/>
<input type="checkbox"/>	Comments	<input type="text"/>	<input type="checkbox"/>

+ **GO Term**

+ **Organism**

+ **Tree**

+ **Reference**

+ **Source**

Advanced Search

Clear all

Figure 4.8: The Advanced Search Tool.

SQL Query

Using the text box below you can enter your own SQL query, which will be subsequently executed in our MySQL server and the results will be presented in a separate html page. You can save this page either as a text file or as an html file, so that you can examine the results of the query in your own time. If there is an error in your SQL syntax, you will receive the exact MySQL error message so that you can correct your query. This text field works in the same way as an interactive MySQL prompt, but please use it with caution.

Enter your SQL query here...

Execute

Figure 4.9: The SQL Query Tool.

SQL command and presses the "Execute" button, the system reads the command and sends it to the KiPhoDB MySQL server for execution. When the execution of the query has finished, the tool retrieves the results and displays them to the user in a tabular format on a different html page. Of course, the results can be easily saved as an html or text file by making the appropriate selections on the menu of the browser that is being used. This feature enables users to further process the results with other external tools in order to get the information they are seeking.

There were many security challenges involved in the creation of this software tool. It is a fact that giving direct access to a MySQL prompt to the public user is not safe for the server and the database itself. Therefore active steps must be taken in order to secure the server and protect it from attacks launched by malicious users. The most important measure that we took for this purpose was to create a separate MySQL user account that had restricted access to the database. This account did not have the rights to delete or alter the database in any way and additionally it could not view some tables that contained information about the other accounts on the system. The only actions allowed for this account was to execute select statements only to the tables that contained biological information. This account was utilized in order to run all user generated scripts, effectively shielding the server from any attempts to alter the contents of the KiPhoDB database.

One other security issue that we had to face was the following: There are some SQL queries that combine information from multiple tables in order to produce a response. If the number of tables combined is large, the SQL query takes longer to execute. Therefore it is not safe for the server to allow users to combine information from various tables, because their query can take too much time to execute and restrict other users from using the server or even render it unresponsive to requests from other individuals. Although there are a variety of measures that can be employed in order to eliminate this thread, we chose not

to restrict the users' access to the database in any way. We want the user to be able to execute advanced queries and combine information for various tables in order to find answers to his questions. If we chose to take one or more of the aforementioned measures, this would not be possible. Therefore, the correct function of this tool is primarily based on the responsibility of the users. Nevertheless, if we suspect that someone is using this tool for non-authorized purposes, we will take immediate action in order to restore and ensure the well being of the database and the server.

In conclusion, this tool is very helpful for advanced users that need to have unlimited access to the database and execute their own queries. These queries may combine one or more tables to produce the desired response. In order for a user to be able to use this tool, adequate knowledge of the internal structure of the database is required. There are two ways to acquire this kind of knowledge. The first one is to use specific SQL queries that list the tables of the database and the fields that these tables consist of. For example queries such as "SHOW TABLES;" and "SHOW FIELDS FROM Protein;" can be utilized for this purpose. The second way is to view an image of the Entity - Relationship model of the database, which includes all tables, their fields and the foreign and primary keys of the database. A link to this image is located at the bottom of the Search page, along with some explanatory information about the database itself.

4.6.5 Browse Data

As the name implies, the Browse Data tool enables users to browse the data in our database by displaying the contents of different tables, fields that have the same characteristics and their interconnections. In order to construct this tool we have exploited the automatically generated Django application named "DataBrowse". This application inspects the database model and tables in order to dynamically create a rich and browsable web site that enables users to browse all data inside the KiPhoDB database. Moreover, it gives users the opportunity to classify data according to certain criteria and find entries that have similar characteristics.

The main drawback of this tool is that the DataBrowse application is relatively new and it is current under active development by the Django software team. It became available after Django version 1.0 and therefore there are a lot of improvements that still need to be implemented by the Django developers in order for DataBrowse to become a better tool that will offer users more functionality and easy of use. Nevertheless, we decided to include this tool and provide it to the KiPhoDB users. Although of course there is still a lot of room for improvement for DataBrowse, the tool in its present state can still offer some functionality that will help users to browse our data and get a general understanding about the contents and structure of our database. In addition to that, it is particularly useful in cases when a user wants to classify the available data according to specific fields. Therefore we took the decision to include it in the list of available software tools for data analysis.

4.6.6 Construct Phylogenetic Trees

This is one of the most important tools in the data analysis toolbox that KiPhoDB provides its users with. It enables users to perform multiple sequence alignments of proteins that already exist in the database and construct phylogenetic trees according to the results. The generated phylogenetic trees can subsequently be used to compare proteins and find groups of proteins that have similar structure and function. This can offer valuable insight into the evolution of various protein kinases and phosphatases and the evolutionary distance between them. Moreover it can help scientists identify orthologous and homologous genes and infer information about the structure and function of novel protein kinases or phosphatases based on similarity searches.

The tool consists of a text area where the user can insert the accession number of all proteins of interest and a button. The accession numbers should be entered in a comma separated format, for example O08605,O15111,O43318,O43707. Once the button “Generate” is pressed, the server performs a search in the KiPhoDB database in order to retrieve all amino acid sequences of the proteins of interest. Subsequently it runs an instance of the ClustalW program in the background in order to perform the multiple sequence alignment of these protein sequences. Based on the results of the multiple sequence alignment, ClustalW generates a phylogenetic tree in the well-known Newick format. Finally, the phylogenetic tree is visualized and displayed to the user through the PhyloWidget web application, which was described in detail in a previous section.

Of course it is quite difficult for a user to gather all accession numbers of interesting proteins and manually enter them in this field. Users need a more automated procedure that will enable them to quickly and easily generate phylogenetic trees with the minimum possible human intervention. In order to address this issue, we have improved this tool so that it can automatically accept results from queries performed by users. Once a user initiates a simple search, the accession numbers of the proteins appearing in the results are all gathered and inserted in this tool. Therefore a phylogenetic tree of the results of a protein search can be automatically generated, giving users the opportunity to compare the amino acid sequences of these proteins and form hypotheses about hidden relationships. Finally, each node of the produced phylogenetic tree contains information about the accession number, the organism and the name of the protein and therefore users can simply inspect the tree in order to get all the valuable information they need.

4.6.7 Gene Families

The Gene Families tool allows the user to search for different kinase or phosphatase gene families and display the corresponding evolutionary tree of the proteins that comprise each family. At the moment, the database includes data for about 121 gene families, 86 of which are kinase gene families and the remaining 35 are phosphatase gene families. All this data has been obtained from the Treefam data source, which was described in more detail in a previous section. Finally, this data involves only manually curated and not computer generated gene families.

Browse Gene Families

In this section you can browse or perform a search in the various Kinase or Phosphatase gene families

[Browse Kinase Families](#)

[Browse Phosphatase Families](#)

Search for a Gene Family

☐ Keyword☐ Gene Name☐ Treefam ID

Figure 4.10: The Browse Gene Families Tool.

The tool itself consists of two distinct components. The first component provides two links which can be used in order to display all kinase gene families or all phosphatase gene families in the database. Subsequently the user can click on the “View Tree” link for the family he/she is interested in and the corresponding evolutionary tree provided by Treefam will be displayed on screen. This tree has been pre-generated by the developers of Treefam and it is being accessed directly from their database. The second component of the Gene Family tool is a search box where the user can enter a keyword and search for a gene family that is relevant to it. Once the user clicks on the “GO” button, a results page appears containing all gene families that contain the given keyword. Finally the user can select to view the evolutionary tree of a gene family by clicking on the “View Tree” link as explained previously.

Chapter 5

Results & Future work

5.1 Outcomes

The KiPhoDB project had a lot of positive outcomes both for the members of the group and the scientific community. In the following list, the benefits for the scientific community are discussed in more detail. Later in this chapter we will also provide more details about the benefits for the group members, too.

- **KiPhoDB - A Unified Resource**

After the development of KiPhoDB, the scientific community has the opportunity to use a new data resource that contains valuable information about kinases, phosphatases, substrates, reactions, pathways, etc. We have developed the database and the website in such a way that KiPhoDB has now become a one-stop-shop for every researcher interested in this kind of information. All data stored in our database comes from a wide variety of data sources and therefore we managed to combine the positive aspects of every source that we used, while at the same time reject all negative characteristics.

- **Kinase - Phosphatase Pairs**

One of the main innovations of KiPhoDB is that it provides its users the opportunity to search the database and identify potential kinase and phosphatase pairs that act on the same substrate. Subsequently users can utilize this information to gain a deeper understanding and shed light on the various pathways that exist in living cells and involve this kind of molecules. Additionally, this information can be used to pinpoint potential drug targets, design new drugs and offer treatment to common diseases.

- **High Quality Data**

During the development of KiPhoDB we paid close attention to the quality of data entering the database. We constantly monitored the quality of information that we used in order to populate the tables of KiPhoDB and we managed to prevent low quality, automatically generated data from entering our database. This fact is

very important because it offers users the guarantee that each piece of information extracted from the database is manually curated.

- **Data Extraction**

The past three months we have worked hard not only to build the database and find the data to store in it, but also develop all necessary software tools that will enable users to easily and effectively extract valuable information from it. We firmly believe that a database is not very useful to end users if it does not offer these possibilities. Therefore we tried our best to develop a wide range of software tools that would offer the opportunity to exploit the data contained inside the database in every possible way. At the end we managed to create seven distinct data extraction tools, each one of which was specifically designed for a purpose. For a more detailed description of the function of each tool, its advantages and its disadvantages, we refer the reader to previous chapters of this report.

In Table 5.1 the current contents of our database are summarized. Of course the population of each object in the database will change over time as more proteins, reactions and pathways are added. Tables 5.2, 5.3 and 5.4 present further information about the contents of our database. Table 5.2 shows the number of phosphorylation and dephosphorylation reactions available in the database for every organism, whereas Table 5.3 presents the number of pathways in the database for every organism. Finally, Table 5.4 includes information about the phosphorylation and desphosphorylation reactions of the top ten pathways that the KiPhoDB database has the most reactions in human. It is immediately obvious from these tables that the scientific community has identified much less dephosphorylation reactions in comparison to phosphorylation reactions.

In the following we will provide the reader with a list of the benefits that the involvement with this project has offered the members of the group.

- **Biological Knowledge**

This project gave all members of the KiPhoDB team the opportunity to explore new scientific areas and acquire a lot of knowledge about phosphorylation, kinases, phosphatases, substrates, signalling pathways and many more. Furthermore we read many scientific publications and reviews, which enabled us to obtain a solid understanding about the challenges in the field of Bioinformatics and more specifically in the field of reactions and pathways. Reading many scientific papers and reviews gave us a clear understanding about many of the challenges in biology and how Bioinformatics can help solving them. The significance of kinases and phosphatases in various biological pathways and their association with diseases is one such example.

- **Databases**

Building databases is an important aspect for any Bioinformatician. Being involved in creating a database from scratch gave every member of the group valuable exposure about the whole process of building databases. Different members of the group were involved in creating different parts of the database, but in the end we all got a clear

Object	Population
Proteins	9263
Protein Domains	1741
Protein - Domain Relationships	11762
Domain - Domain Relationships	0
Phosphorylation Sites	39968
GO Terms	4257
Protein - GO Term Relationships	45801
Reactions	4387
Pathways	382
Reaction - Pathway Relationships	10384
Trees	0
Organisms	1141
Gene Families	121
External IDs	23733
References	25807
Files	0

Table 5.1: The current contents of KiPhoDB.

understanding of the whole process. This project also provided all the members an overview of the large number of existing biological public databases available today. It gave us an idea how bioinformatics tools are used to create such kind of databases.

- **Scripts**

For populating the database we utilized several python scripts for inputting of data. Each member was involved in writing scripts for extracting data from various sources which we had divided amongst ourselves. We learned about new libraries and functions like BeautifulSoup, minidom parser, etc for parsing data from various different types of format like CSV, BioPAX, Excel Sheets etc. We also learned how to fetch remote web pages on the fly, and parse data from them.

- **Undertaking Large Projects**

This group project was constructed from scratch in a span of only three months, which is relatively short time for building a high quality database. However we feel that KiPhoDB is a success and provides high quality and reliable data. This would not have been possible without close team work and strategic planning. In the end this has given all the members of the group the confidence to handle any large project

Organism Name	Phosphorylation Reactions	Dephosphorylation Reactions	Total
Homo sapiens	2946	75	3021
Mus musculus	1205	131	1336
Rattus norvegicus	11	13	24
Drosophila melanogaster	5	0	5
Oryctolagus cuniculus	1	0	1
Total	4168	219	4387

Table 5.2: Reactions per organism.

Organism Name	Number of Pathways
Homo sapiens	251
Mus musculus	95
Rattus norvegicus	30
Drosophila melanogaster	4
Oryctolagus cuniculus	3
Total	382

Table 5.3: Pathways per organism.

of this sort.

5.2 Have we met the requirements?

Below is a list of some of the queries initially presented by our supervisors to try and guide us in the design of the project and the current status in our system.

- What is the evolutionary distance between kinase x and kinase y?
We have provided a tool to display phylogenetic trees.
- What is the evolutionary distance between human kinase x and its orthologue in mouse?

Pathway	Phosphorylation Reactions	Dephosphorylation Reactions	Total
Cell Cycle, Mitotic	238	15	253
Signalling by NGF	218	1	219
MAPK signaling pathway	201	0	201
Insulin signaling pathway	134	0	134
Cell cycle	121	9	130
Signaling in Immune system	118	9	127
Focal adhesion	126	0	126
Prostate cancer	106	0	106
ErbB signaling pathway	105	0	105
BCR signaling pathway	92	0	92

Table 5.4: 10 top pathways for which we have the most reactions in human

- Which is the closest paralogue of phosphatase x?
We haven't really included paralogue information.
- What are the other members of the protein family to which kinase x belongs?
There is information about kinase and phosphatase families in our database provided by Treefam [21].
- What is the substrate(s) of kinase x?
For each protein we display the phosphorylation and dephosphorylation reactions it is involved in.
- What is the phosphorylation site of kinase x?
This information is retrieved from Phosphosite [26] to which the Uniprot [27] links.
- What is the kinase counter-part to phosphatase x?
There is a search tool to get kinase-phosphatase pairs.
- Show the pre-computed evolutionary tree to the substrate of kinase x?
We don't precompute evolutionary trees.
- Which kinases/phosphatases belong to pathway x?
Viewing a particular pathway will tell you which kinases and phosphatases belong to that given pathway.
- Which human kinases activate other kinases?
Yes a search can be limited to other kinases.

- Which phosphatases activate enzymes?
Yes we store EC numbers.
- Which kinases have known phosphate counter-parts?
- Which phosphatases have GO annotation x?
Yes it is possible to by GO annotation.
- What is the EC classification the substrates of kinase x?
If the EC classification is given on Uniprot [27] then it is stored in our database.

As you can see most of the requirements have been. The evolutionary data still leaves some room for improvement but as we store sequences and domains it is possible to infer that information from our data.

5.3 Future Plans

- Populate Domain-Domain interaction table. We have not been able to integrate data for the domain-domain interaction of any particular kinase substrate or phosphatase substrate pairs.
- Improve the interface. We have started using Ajax to make the website more interactive and user friendly and to display information with less latency. However one of the issues with Ajax is cross browser compatability especially between Internet Explorer and other browsers. The information can also be presented better i.e. perhaps more concisely and with more emphasis on the important bits.
- Improve the search tools. Again Ajax can be used to to provide suggestions as the user types in a search field. This would be implemented by using what the user has typed so far to make search partial matches.

Bibliography

- [1] G. Manning, DB Whyte, R. Martinez, T. Hunter, and S. Sudarsanam. The protein kinase complement of the human genome. *Science*, 298(5600):1912–1934, 2002.
- [2] C.Y. Yang, C.H. Chang, Y.L. Yu, T.C. Lin, S.A. Lee, C.C. Yen, J.M. Yang, J.M. Lai, Y.R. Hong, T.L. Tseng, K.M. Chao, and Huang C.Y. PhosphoPOINT: A comprehensive human kinase interactome and phospho-protein database. *Bioinformatics*, 24(16).
- [3] P. Cohen. The origins of protein phosphorylation. *Nature cell biology*, 4:E127–E130, 2002.
- [4] F. Diella, C.M. Gould, C. Chica, A. Via, and T.J. Gibson. Phospho. ELM: a database of phosphorylation sites update 2008. *Nucleic Acids Research*, 2007.
- [5] F. Gnäd, S. Ren, J. Cox, J. Olsen, B. Macek, M. Oroshi, and M. Mann. PHOSIDA (phosphorylation site database): management, structural and evolutionary investigation, and prediction of phosphosites. *Genome Biology*, 8(11):R250, 2007.
- [6] R. Linding, L.J. Jensen, G.J. Ostheimer, M.A.T.M. van Vugt, C. Jørgensen, I.M. Miron, F. Diella, K. Colwill, L. Taylor, K. Elder, et al. Systematic discovery of in vivo phosphorylation networks. *Cell*, 129(7):1415–1426, 2007.
- [7] John N. Abelson, Melvin I. Simon, Tony Hunter, Bartholomew M. Sefton. *Protein Phosphorylation, Parts A and B*. Academic Press, September 1991.
- [8] S. Trnroth-Horsefield, Y. Wang, K. Hedfalk, U. Johanson, M. Karlsson, E. Tajkhorshid, R. Neutze, and P. Kjellbom. Structural mechanism of plant aquaporin gating. *Nature*, 439(7077):688–94, 2006.
- [9] S.A. Johnson and T. Hunter. Kinomics: methods for deciphering the kinome. *Nature Methods*, 2:17–25, 2005.
- [10] F. Staib, A.I. Robles, L. Varticovski, X.W. Wang, B.R. Zeeberg, M. Sirotin, V.B. Zhurkin, L.J. Hofseth, S.P. Hussain, J.N. Weinstein, et al. The p53 tumor suppressor network is a key responder to microenvironmental components of chronic inflammatory stress, 2005.

- [11] G. Manning, G.D. Plowman, T. Hunter, and S. Sudarsanam. Evolution of protein kinase signaling from yeast to man. *Trends Biochem Sci*, 27(10):514–20, 2002.
- [12] KJ Wolstencroft, R. Stevens, L. Taberner, and A. Brass. PhosphaBase: an ontology-driven database resource for protein phosphatases, *Proteins (Band 58)*, Nr. 2. URL: <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>, pages 290–4, 2005.
- [13] G.B. Moorhead, L. Trinkle-Mulcahy, and A. Ulke-Leme. Emerging roles of nuclear protein phosphatases. *Nat Rev Mol Cell Biol.*, 8(3):234–44, 2007.
- [14] N. K. Tonks. Protein tyrosine phosphatases: from genes, to function, to disease. *Nat Rev Mol Cell Biol.*, 7(11):833–46, 2006.
- [15] F.F. Zhou, Y. Xue, G.L. Chen, and X. Yao. GPS: a novel group-based phosphorylation predicting and scoring method. *Biochemical and Biophysical Research Communications*, 325(4):1443–1448, 2004.
- [16] ZZ Hu, M. Narayanaswamy, KE Ravikumar, K. Vijay-Shanker, and CH Wu. Literature mining and database annotation of protein phosphorylation using a rule-based system. *Bioinformatics*, 21(11):2759–2765, 2005.
- [17] AR Forrest, D.F. Taylor, J.L. Fink, M.M. Gongora, C. Flegg, R.D. Teasdale, H. Suzuki, M. Kanamori, C. Kai, Y. Hayashizaki, et al. PhosphoregDB: the tissue and sub-cellular distribution of mammalian protein kinases and phosphatases. *BMC bioinformatics*, 7(1):82, 2006.
- [18] M. Ashburner, CA Ball, JA Blake, D. Botstein, H. Butler, JM Cherry, AP Davis, K. Dolinski, SS Dwight, JT Eppig, et al. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25, 2000.
- [19] J. Boudeau, D. Miranda-Saavedra, G.J. Barton, and D.R. Alessi. Emerging roles of pseudokinases. *Trends in Cell Biology*, 16(9):443–452, 2006.
- [20] Chen Z Coghlan A Coin LJ Guo Y Hrich?JK Hu Y Kristiansen K Li R Liu T Moses A Qin J Vang S Vilella AJ Ureta-Vidal A Bolund L Wang J Durbin R. Ruan J, Li H. TreeFam: 2008 Update. *Nucleic Acids Research*, 36:735–40, 2008.
- [21] Ruan J Coin LJ Hrich?JK Osmotherly L Li R Liu T Zhang Z Bolund L Wong GK Zheng W Dehal P Wang J Durbin R. Li H, Coghlan A. TreeFam: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Research*, 34:572–80, 2006.
- [22] Gillespie M Caudy M Croft D de Bono B Garapati P Hemish J Hermjakob H Jassal B Kanapin A Lewis S Mahajan S May B Schmidt E Vastrik I Wu G Birney E Stein L D’Eustachio P Matthews L, Gopinath G. Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res.*, 37:D619–D622, 2009.

- [23] Carl F. Schaefer, Kira Anthony, Shiva Krupa, Jeffrey Buchoff, Matthew Day, Timo Hannay and Kenneth H. Buetow. PID: The Pathway Interaction Database. *Nucleic Acids Res.*, 37:D674–D679, 2009.
- [24] Araki M. Goto S. Hattori M. Hirakawa M. Itoh M. Katayama T. Kawashima S. Okuda S. Tokimatsu T. Kanehisa, M. and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Res.*, 36:D480–D484, 2008.
- [25] Jordan G.E. and Piel W.H. PhyloWidget: web-based visualizations for the tree of life. *Bioinformatics*, 24(14):1641–1642, 2008.
- [26] P.V. Hornbeck, I. Chabra, J.M. Kornhauser, E. Skrzypek, and B. Zhang. Phospho-Site: a bioinformatics resource dedicated to physiological protein phosphorylation. *Proteomics*, 4(6), 2004.
- [27] A. Bairoch, R. Apweiler, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, et al. The universal protein resource (UniProt). *Nucleic Acids Research*, 33(Database Issue):D154, 2005.