VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

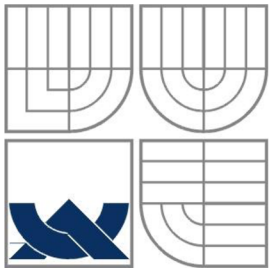
WEB SERVER FOR PROTEIN INTERACTION SEARCHING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

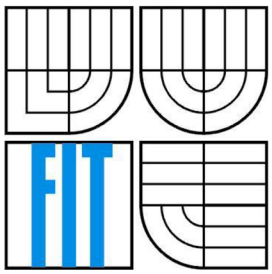
AUTOR PRÁCE
AUTHOR

Bc. MARTIN HALFAR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WEBOVÝ SERVER PRO ZJIŠŤOVÁNÍ INTERAKCÍ PROTEINŮ

WEB SERVER FOR PROTEIN INTERACTION SEARCHING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN HALFAR

VEDOUCÍ PRÁCE
SUPERVISOR

BURGET RADEK, Ing., Ph.D.

BRNO 2012

Abstrakt

Tato práce se zabývá zbůsoby, jimiž je možné získávat data z bioinformatických databází obsahujících data týkajících se interakcí mezi proteiny. Od souvislostí okolo vzniku bioinformatiky sloučením informatiky a biologie tato práce uvede čtenáře do problematiky přístupu k datům týkajících se interakcí mezi proteiny. Tato práce vysvětlí důvody vzniku IMEx konsorcia, jeho cíle a prostředky, kterými svých cílů dosahuje. IMEx konsorcium dalo vzniknout mnoha standardům, které usnadňují přístup k datům členů konsorcia a výměnu těchto dat mezi nimi. Jedním z výtvorů IMEx konsorcia je i webová služba PSICQUIC, která byla navržena s využitím architektonického stylu REST, a která je přístupná i pomocí protokolu SOAP. Obě tyto kategorie přístupů k webových službám jsou v rámci této práce studovány a na základě výsledků výzkumu je implementována aplikace pro získávání interakcí mezi proteiny z databází, jenž jsou členy IMEx konsorcia.

Abstract

This thesis deals with different possibilities, how to collect data from bioinforatics databases containing protein interaction data. Reader is put into context by introducing him problematics of emergence of bioinformatics by connecting two fields of human knowledge: biology and informatics. Then the reader will get acquainted with the importance of protein interactions and possible ways of retrieving protein interaction data from protein interaction databases. This thesis also elucidates the motivation for IMEx consortium existence. IMEx faciliattes access to data and data exchange between its members by issuing new standards and data formats. I a list of IMEx consortium sucecsses is also PSICQUIC web service. PSICQUIC is REST-compliant web service, which can be also accessed via SOAP protocol. Both REST and SOAP approaches are studied and compared in this thesis and on the basis of this research is implemented application for retreiving protein interaction data from PSICQUIC members‘ databases.

Klíčová slova

Protein-protein, interackce, webové služby, SOAP, REST,databáze, PSICQUIC.

Keywords

Protein-protein, interaction, web services, SOAP, REST, databases, PSICQUIC.

Web Server for Protein Interaction Searching

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Burgeta Radka, Ing., Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Martin Halfar
22.května 2012

Poděkování

Tímto bych rád poděkoval Ing. Ivaně Burgetové za cenné rady a všechny materiály, jenž mi k vypracovávání této práco poskytla.

Table of content

Table of content 1
Introduction 3
1 Bioinformatics – an introduction 5
1.1 Introduction 5
1.2 Relationship between biology and informatics 5
1.3 Importance of interactions in living organisms 6
1.4 Proteins 6
1.5 Pathway Informatics 6
1.6 Protein interactions in networks 7
1.7 Misleading protein interactions 8
1.8 Summary 9
2 Databases in bioinformatics research 11
2.1 Introduction 11
2.2 Database Structure 11
2.3 Database types 14
2.4 Online databases 15
2.5 Quality of data in databases 17
2.6 Availability 20
2.7 Summary 21
3 International Molecular Exchange Consortium 22
3.1 Introduction 22
3.2 HUPO PSI-MI 22
3.3 MIMIx guidelines 22
3.4 Literature curation 23
3.5 Development of the International Molecular Exchange consortium 24
3.6 Quality control 26
3.7 Data exchange 26
3.8 Summary 26
4 Web services and REST 28
4.1 Introduction 28
4.2 Web services 28
4.3 SOAP 29
4.4 REST architectural style 29
4.5 REST and SOAP comparison 32

4.6 Summary 34
5 PSICQUIC 35
5.1 Introduction 35
5.2 PSCICQUIC registry 37
5.3 PSICQUIC data availability 38
5.4 PSICQUIC REST access 38
5.5 MIQL flexible query 39
5.6 Summary 40
6 Application for protein interactions searching 41
6.1 Introduction 41
6.2 Simplicity 41
6.3 Novel interface 41
6.4 Customizable interface 41
6.5 Intelligence 42
6.6 Maintenance 42
6.7 Summary 42
7 Components of the application 43
7.1 Introduction 43
7.2 Simple interface 43
7.3 Easy interface 43
7.4 Results page 44
7.5 Help 44
7.6 Summary 44
8 Implementation 45
8.1 Introduction 45
8.2 Choice of programming langauge 45
8.3 Querying PSICQUIC 46
8.4 Working with simple initerface 46
8.5 Work with easy interface 47
8.6 Work with results page 47
8.7 Introducing other applications for protein interaction searching 50
8.8 Possible improvements 53
8.9 Summary 54
9 Conclusion 55

Introduction

Bioinformatics emerged from connection of biology and informatics. Its existence is important because of the need of biologists for faster data assimilation, more useful data visualization and to make the data saved in data storages more accessible. One of the most important goals of modern biology is to collect data about cells and create cellular model, which would serve for better understanding of processes in living organisms and thus to more efficient treatment of disease with high prevalence in population. The crucial part of building this model is collecting protein interactions. From protein interactions can be built large protein interactions networks from which can be derived new hypothesis for further testing. The goal of this thesis is to get acquainted with protein interaction databases and possible ways how to access their data and to implement server which will make the data available to user.

In the first chapter is described relationship between biology and informatics and how both these fields benefit from these connections. There is also explained the importance of protein interactions and their influence on the speed of discovery cycle in biology and what are the means of bioinformatics to help biologists in filtering incorrect data incurred by false positives or false negatives during experiments.

The second chapter shows current possibilities of storing bioinformatics data into databases. It describes how informatics can make the cooperation between biologists from all over the world much easier (e.g. how to control used vocabulary, how quality of data can be controlled) and what effect it has on accessing gathered data.

In the second chapter is mentioned that a number of online databases is during last ten years constantly increasing. It is hard to maintain increasing number of databases that need to be connected to each other in order to exchange their data. Especially, when each of the databases can be of a different type, it can be difficult to agree on uniform interface, which would be easy to use and also accessible for everyone. The third chapter explains why the set of important online databases decided to create IMEX consortium and what steps the consortium took to make the data exchange between databases easier, how IMEx influenced curation process and thus ensured that information from all possible publications can be integrated and made available in online databases in a very short time.

Each one of the available online databases possesses own interface how to access its data. This is unfortunate for a user, which wants to access the data from several sources. He is forced to write new queries for each of the databases just to get the same information. The solution to that problem is utilizing Web services. In the chapter four two main categories of web services are introduced: REST-compliant Web services and Web services utilizing SOAP protocol. In this chapter are discussed their advantages and disadvantages and suggested the reasons of a bigger popularity of one of the mentioned solutions nowadays.

IMEx members' efforts led to definitions of new standards and manuals, which were widely accepted by community. In a list of their successes is also PSICQUIC described in chapter five. PSICQUIC is a web service, which facilitates data exchange between databases and access to the stored data by defining unified interface. PSICQUIC offers access via SOAP protocol or its REST-compliant web service. However, in the previous chapter were discussed SOAP and REST approaches and the conclusion was that both protocols are comparable. Nevertheless, the conclusion of the comparison is that SOAP protocol should be utilised only when one of its properties is needed; thus, in this chapter is PSICQUIC studied with regard for its REST-compliant web service.

The suggested properties for application retrieving protein interaction data are listed in the sixth chapter. The modern days offer plentiful examples of successful applications because of the ease of

use and simple interface. And there are also other applications with similar thinking. These applications will be introduced in chapter eight.

The seventh chapter put nearer the ideas, which will lead the new application implementation to its goal. There will be stated some of the basic parts of the final system and their purpose in the context of the whole application.

In the eight chapter is introduced the solution of how to access PSICQUIC REST-compliant web service in Ruby on Rails framework. The main parts of the implemented application will be introduced again, but this time already implemented with pictures from fully-functional application, connected to twenty-five online databases. Along with the new application are also introduced other applications that focus on the similar direction. At the end of the chapter will be suggested possible improvements to the implemented application also inspired by the observation of the other applications.

Conclusion will assess contribution of this work and suggest possible future sequel of this thesis.

1 Bioinformatics – an introduction

1.1 Introduction

According to Jacques Cohen [5], biology is not independent field of human knowledge but for a long time co-existed with other fields like chemistry or physics, which merged with biology into new fields called biochemistry or biophysics. Cooperation across different fields of human knowledge led to significant progress and brought vast amount of new discoveries in both of the cooperating fields. And because the partnership of biology with other fields was considered as fruitful and helped to explain a lot of new processes in living organisms, which were latterly found useful in various fields of biology, there arose also efforts to combine biology with informatics.

Bioinformatics helps in better understanding of the processes that are held within the living organisms in many ways. In this chapter will be discussed the influence of informatics on biology and how the cooperation of these fields speeds up looking for new hypotheses for new experiments. Importance of interactions in living organisms will be emphasized and ways how bioinformatics can utilize findings from biology to build interaction networks to study diseases with high prevalence in population. It is difficult in biology to perform high quality experiment and thus in biological data can appear incorrect results. There also will be introduced what means bioinformatics has for detecting the incorrect data.

1.2 Relationship between biology and informatics

As the technology is rapidly evolving, the biology gets better tools to examine structure of living organisms. Several new techniques have been evolved to help biologists to gather more accurate and detailed data. This massive amount of accumulated data about cells from various fields of modern biology (such as molecular or cellular) is needed to be processed. Thus, one of the biggest challenge for biologists is to aggregate already found data into a cellular model. This cellular model is supposed to create fundamental basis for medicine.

Bioinformatics, which evolved from computational biology and connects together biology and informatics, helps to process and interpret the new data that appeared in biology in recent years. Even though there is no agreed definition of computational biology and bioinformatics, basically could be said, that computational biology utilizes mathematical and computational approaches in relation to biology, whereas bioinformatics aims to transform biologic data into something more understandable and useful. Bioinformatics is more focused on data assimilation, visualization software and databases.

Cohen in his article [5] also claims that, as in the case of biochemistry and biophysics, the symbiosis with biology will bring some new thoughts and findings into informatics and thus inspire it and lead it to new improvements. The biology has got, for instance, increased demands on reliability and informatics has to respond to such demands and develop tools that will be able to fulfil such a requirements. Without understanding processes that are inseparable part of living organisms, it is almost impossible to make new discoveries on fields of cellular and molecular biology. Therefore, many new high-throughput methods of studying protein-protein interactions were developed and their processing can be highly computationally demanding. Therefore there also arise higher requirements for computational speed.

However, technological progress keeps up with increasing demands of bioinformatics. In article on Wikipedia [6] that listed all Intel microprocessors can be seen that Intel Pentium II which was introduced in 1997 consists of 7.5 million transistors and recently introduced Intel i5 has got 995 million transistors. Except these raising numbers of transistors also changes in architecture of processors ensures continuously growing performance of computer systems on the world. For example performance of the fastest systems is monitored by project TOP500 [7]. The project TOP500

started in 1993 with supercomputer Fujitsu Numerical Wind Tunnel with peak speed 124.5 GFLOPS and in year 2011 is the fastest system called Fujitsu K computer with 10.5 PFLOPS. This means that microprocessors became due to new technologies much faster and can offer performance that is able to help biologists with processing their data and thus become their capable assistants. Nevertheless, computational speed is not the only important part of informatics from which bioinformatics can have benefits.

Along with hardware, it is necessary to build software capable to utilize all the benefits of advanced hardware. Informatics helps to build up databases where biologists can store data from their experiments and develop new systems for sharing the newly gain knowledge. And in this thesis will be described some of the newly created standards for facilitating the biologists' work.

1.3 Importance of interactions in living organisms

Living cells consist of many parts. Some of them are bigger than the others, each one of these parts can differ in a shape or even in its structure. However, in every living organism there is something that connects all parts together, and that is interactions. None of living cells could exist without interactions. Such a cell without interactions would lose all of its capabilities to interact with its environment. And that would not be all. Losing all interactions would lead to certain disintegration of this cell, because interactions are of vital importance to every cell's physiological process [32].

Because molecular interactions plays important role in almost every physiological process, it is important to find about them as much as possible. Studying and finding all possible protein-protein interactions is only way how to understand all processes within a living cell. It will lead to better understanding of processes that happens in a human body and which can cause development of diseases; thus, to discovering new possibilities of disease treatment.

Although functional genomics is a field of molecular biology which focuses on studying protein-protein interactions, it would be almost impossible task to handle the amount of data produced by modern approaches to study protein-protein interactions without aid of bioinformatics and its tools for visualizations and its capabilities for automatic deriving of new hypothesis.

1.4 Proteins

Cell is complex system that consists of many components. These components can vary by size and shape. DNA can be designated as the bigger component with shape of right-handed double-helix, which consists of two right-handed polynucleotide chains coiled around the same axis. Information stored in DNA can be used to build smaller components of cell - proteins.

Proteins can have many functions - for example form the cell's membrane. If cell is placed in environment where conditions are suitable for live or even if it is situated in environment where cell is destined to die then proteins perform certain actions. And as the result of these actions the cell can consume and digest vicinal nutrients or produce some chemical substance that could raise survival prospects. These actions usually consist of many chemical interactions. A sequence of interactions between molecules in a cell that leads to some result – state change or creation of new product (such as creation of fat or protein) – is called biological pathway [33].

1.5 Pathway Informatics

With a technology that is accessible nowadays, the gained data from experiments is more accurate and the elements which participate on the studied part of cellular metabolism can be detected more precisely. It helps to reduce possibility of an error; thus helps to detect even small amount of studied

molecules and find new cell signalling or regulatory networks. These findings can be utilised within a medicine to fight with deceases such as Alzheimer's disease or cancer [2].

Pathway Informatics is the field of Bioinformatics that is relatively new. New types of data, more intelligent integration of new data into existing data and modern computational methods brought benefits in the form of many new discoveries such as new findings related with disease with high prevalence in population. Rapid development of the new experimental methods of investigating processes within cells created space within bioinformatics for biomolecular interaction and pathway analysis.

As was mentioned earlier, one of the biggest challenges in modern biology is to create accurate cellular model in which are described all processes that are performed in cells. One of the ways how bioinformatics helps to biologists nowadays in creation of such a model is the accelerating the discovery cycle of cell mapping experiments. This can be done by computational pathway and network analysis if they are included in earlier phases of planning these mapping experiments. The very important part of pathway informatics is information about protein-protein interactions.

1.6 Protein interactions in networks

Essentially, proteins' function depends on their interactions. Therefore, Arthur M. Lesk in Introduction to Bioinformatics likens proteins to "social animals" we need to study so we can decipher their complex relationships [3], and in connection with protein also Laura Bonetta recalls old adage "Show me your friends, and I'll know who you are" [15]. The biggest difference between interaction networks and pathways is that pathway cannot contain any loop and has to have a result, whether it is a new product or change of state.

Because there are many these so called "social animals" in a living cell, there has to be some control mechanism, which could gain control of them if they wanted to go rogue. It is important that when cell is placed into whatever environment, the right proteins are moved on their appropriate places in a cell, so they can execute actions they are programmed to do and thus ensure cell's survival. This system has to be robust so it can withstand stressful environment. On the other side, when it is in normal unchanging environment with sufficient amount of nutrients, it also needs to survive – in this scenario it would mean to get nutrients and stay stable.

Set of all pathways of chemical reactions in cell, which serve for transmission of energy or whole molecules, and which is called metabolism, can consist of nucleic acids, amino acids, sugars and also proteins. All these biological pathways are represented by sequential interactions leading to some result. In contrast to that networks do not have to be linear. They can form closed loops and be interlinked. Network that is made of these individual pathways can be "analysed utilizing mathematical apparatus dealing with graphs and flows and throughputs" [3].

One has to keep in mind that even pathways are human construct. And even though it may seem that cell consists of many organelles, which can be differentiating from each other under a microscope, pathways are part of much larger and fully connected interaction network.

Figure 1 shows how such a protein interaction network can look like. The nodes of this network are proteins and two proteins within this network are connected with edge which represents some protein-protein interaction. On the Figure is example of protein-protein interaction network. It is estimated, that there exist approximately 130,000 binary interactions, and only a small part of the whole set of all protein-protein interactions within a cell has been discovered so far. In a database that stores the interaction data called BioGRID, was listed 33,943 unique protein-protein interactions in December 2010 [16] and according to official BioGRID statistics from 2012 there were available already 65,392 unique protein-protein interaction [17]. The set of all protein-to-protein interactions within a cell is called *interactome*.

Such a protein-protein network, like on Figure 1, helps scientists to study and better understand cell functions and furthermore, similar networks can be utilized for studying diseases. Genes of such a disease can be put into protein-protein interaction network and then it will be easier for scientists to estimate disease's risks and sub sequential treatment – looking for the diseases weaknesses. This is the next step after a Human Genome Project: To find out how genes work in pathways; thus, how they participate on a development of a disease, what influence they have on the disease in its individual states. The most significant problem is that many of the biological pathways are transient. They happen only when some conditions are accomplished, sometimes only in one point in a time. That is why this task is much more difficult than Human Genome Project – unlike genome interactome is dynamic.

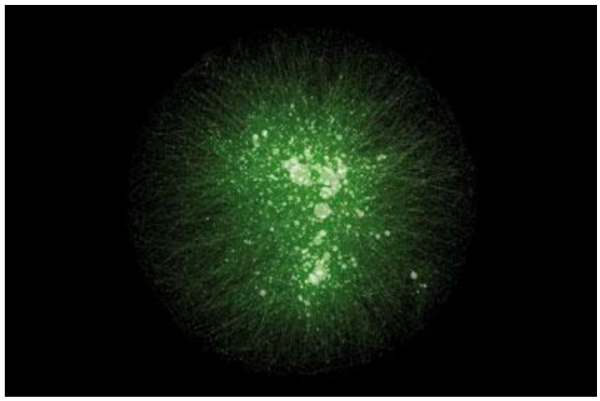


Figure 1 Example of protein-protein interaction network [15]

Besides metabolic interactions, there are also regulatory interactions, which serve for connecting proteins and metabolite concentrations. Together they create interaction networks. Interaction networks can be divided between:

- Physical network
- Logical network

The physical network is made of protein-protein and protein-nucleic acid complexes and the latter mentioned network is made of control cascades.

As can be seen, there are many ways how to create protein interaction network. One of the ways is direct physical protein-protein interactions. And furthermore, it is possible to merge this protein interaction network with other types of information which will help to view the data from another perspective.

It is possible to create ultimate network with the all available knowledge, but could be counterproductive. It would add a complexity to such a network and the result would be unnecessarily distracting. Users, who wanted to study global properties of protein-protein interaction network, could find distracting that next to the relevant information they wanted to see, there was also information on electron dynamics [2]. Keeping some abstraction levels over the work data is therefore helpful instead of overwhelming user with too much information.

1.7 Misleading protein interactions

Experiments in biology can be influenced by many factors. Even small interference with samples in the beginning of the experiment can affect result. Thus, it is important that user has a means how to find out that the experiment was verified, or at least how much he can be assured that the result is correct.

There are basically two main groups of empirical methods that are looking for new interacting proteins. The first group searches for proteins that are physically interacting among each other – this approach is called binary approach. The second group deals with groups of proteins that may not physically interact with each other. The most used method to look for binary interactions is the yeast two-hybrid (Y2H) system, which has evolved since 1989, when an S. Fields and O. Song wrote an article about this method [18], and nowadays it is adapted to high-throughput screening.

False positives – interactions that did not happen – and *false negative* – interactions that happens but the experiment did not detect them – is a problem, which is closely related to almost every experiment in biology. After an experiment, one has to get rid of all stochastic interactions. When all non-specific interactions are removed, protein can end up with 4 – 5 partners or 30 – 50 partners if it is part of large complex. Because of elimination of both false negative and false positives, it is recommended to run more types of experiments with the same bait and prey (e.g. LUMIER and Y2H). But the result depends on every step of performed experiment and sometimes can happen that two laboratories do not get the same results because of different protocols. There is also a problem with definition of the term *interaction*. Proteins can for example interact when they are put next to each other in a test tube, but in a real cell environment they would not react. Or, two proteins can interact, but their interaction does not serve to any purpose in a cell. Basically, scientists can choose themselves how to work with the gained data and then it can be mixed with another type of information. There are many tools online, which can help with this information integration – such as GeneMania – which integrates information about proteins and genes. After a user enters gene's name into the system, the system shows him the list of genes with similar functionality or properties (e.g. expression), and also can show suggested interaction network. [18]

There are methods based on keeping a score of each protein-protein interaction to eliminate false positives. Into these methods belongs software platform called CompPASS. It gives a score to every interaction, evaluating its frequency, abundance and reproducibility [16]. In 2010, the CompPASS was utilised to reduce the list of 2,553 interacting proteins to 409. These 409 high-confidence proteins were interacting in 751 interactions.

Utilizing tools similar to CompPASS, one can be certain using only correct data. On the other hand, there is estimation that user utilizing these techniques to get rid of false positives also gets rid of 80% of the interactome [16].

Nowadays there is not technology that would identify every interaction that was not found by experiments. Thus, if one wants to have the best results, it is necessary to combine information from as many sources as possible – such as public databases, software for interaction identification with various settings over the sample of the interactome.

1.8 Summary

Bioinformatics was created by connecting together biology with informatics. This demand for forming this new field of knowledge arose from the increasing amount of data that is collected by new methods and approaches in biology and which needs to be analysed. Increasing computational speed and development of new software tools increases the speed of discovery cycle.

Although functional genomics is the part of biology that studies protein-protein interactions, bioinformatics has means of how to better utilize the data gained from especially high-throughput experiments. Bioinformatics helps scientists to visualize and faster assimilate new data and search for new hypotheses that can be afterwards confirmed or disproved.

Proteins and their interactions form the basis of life. Life would cease to exist without protein interactions and that is why it is important to study them. Studying interactome is after Human Genome Project the next step and Bioinformatics plays important role in it. It helps with handling the vast amount of data that comes from findings concerning protein-protein interactions. Protein

interaction networks serve for studying diseases such as Alzheimer's disease or cancer and cannot be built without knowledge of protein-protein interactions. Because there is much more interactions than genes in Human Genom, mapping the Human interactom is much more challenging process and because of the high number of existing interactions, it is necessary to use all informatics' means to process all the data and derive useful information from it.

Bioinformatics helps scientists to better understand processes within a living cell and find new ways how to treat diseases also by making all information available for every scientist on the world in online databases. However, data in database does not have to be always correct. It can come from incorrect experiments or experiments that were not correctly described and thus they were misunderstood. Though, Bioinformatics found solution how to deal even with this problem and online are available systems to evaluate the data, which can user utilize to gain confidence that the data he uses is correct.

2 Databases in bioinformatics research

2.1 Introduction

In year 1958, Frederick Snager became recipient of Nobel Prize in chemistry because of the research he had made in field of protein sequencing [1]. He was the first man ever, who sequenced protein and the centre of his attention was bovine insulin. Almost ten years after Frederick Snager's discovery [4] attempts to utilize computers in analysing these sequences appeared. In that time bioinformatics started to evolve by taking care of the data processing and data storage. However, it was not called bioinformatics right away.

Before biologists started to use computers, the old cards-in-a-box catalogues were used. These old databases did not have such sophisticated methods of data processing and the biologists moved to new computerized databases that could satisfy their needs more efficiently. That is because the database of biologic data is not only collection of data. On the top of it, it also has its structure and there are connections and dependencies that are hard to maintain, while the volume of data gained from sequencing grows rapidly.

In recent days, there exist hundreds of thousands of nucleotide sequences and over hundred thousand protein sequences [4]. As was mentioned before, without involving computer science into the process of gathering biologic data, it would be challenging task to maintain database of this size. Furthermore, the quantity of data has grown also in other fields of biomedical research. The modern high throughput experimental techniques generates massive amount of data and thus analogously, the utilization of databases has thus necessarily risen across all fields of modern biology. Due to existence of databases, which are all oriented at biology (whether the research is chemical or biomedical) the biologists have opportunity to link together information from these individual databases and create larger knowledge network.

In this chapter will be described properties of bioinformatics database. There will be introduced their possible structures, types, how databases gain new data, how is ensured their correctness and what scientist should be careful about while looking for bioinformatics database.

2.2 Database Structure

Database is not only storage for massive amount of data. It also aids analysing of the data and in order to serve well to this purpose, it has to have specific structure. According to the structure the databases can be classified into several categories such as flat-file, relational, object-oriented or distributed databases. [4] In following paragraphs will be each of these types slightly introduced and described.

2.2.1 Flat-file databases

This form of database is the simplest and oldest one. Flat file database was utilised in the first computerized census in nineteenth century [4]. In spite of that, the flat file databases still have their place not only in today's database systems but also in operating systems (flat file databases are still commonly used in UNIX system environments [9]). The format of records in flat-file database can be distinguished by the length of records, which can be constant or variable. If the length of records varies, then in the database has to be specified delimiters that would separate individual records.

Figure 2 shows example of record stored in database in flat-file format from the GenBank sequence databank [4]. On the figure can be seen that individual records from tables are stored in files and the format of the files is as readable as would be the one in a paper form. More modern and more sophisticated methods of storing data in databases are more common because they offer possibility to

run complex queries over the stored data more efficiently. Although the work with small amount of data in a database utilizing flat-files can have its advantages, while the database grows, then the sequential access to its records stops to be able to cope with the data in reasonable amount of time, and thus it had to been replaced by more sophisticated techniques that will be mentioned later on. But as was mentioned above, the flat-files are used even in present day. Except utilizing flat-files exclusively as storage tool, they also can facilitate data distribution across different database systems.

Data transfer can be difficult because of existence of many database systems. Each database system can be released under various license terms, that can keep their own formats and if it is necessary, the content of database can be easily converted into flat-file format and on different system these files can be easily read. Flat-files can be replaced with Extensible Markup Language (XML) files that include the data and also description of the data. Drawback of utilizing XML can be larger resulting files.

LOCUS DEFINITION	**SCU49845 5028 bp DNA**
DEFINITION	Saccharomyces cerevisiae TCP-beta gene, partial cds, and Ax12p (AXL2) and Rev7p (REV7) genes, complete cds.
ACCESSION	U49845
VERSION	U49845.1 GI:1293613
KEYWORDS	.
SOURCE ORGANISM	Saccharomyces cerevisiae (baker's yeast) Saccharomyces cerevisiae Eukaryota; Fungi; Ascomycota; Sacharomycotina; Sacharomycetes; Sacharomycetales; Sacharomycetaceae; Sacharomyces.

Figure 2 Example of the beginning of a record from flat-file database GenBank sequence databank for gene TCP1-beta of Sacharomyces cerevisiae [4]

2.2.2 Relational databases

Usually the data is stored in relational database that conforms to relational model theory. Such a database structure consists of tables (also called files) that can contain records composed of fields (rows and columns). In terms of relational model theory the table is called relation and this relation is defined as set of tuples with the same set of attributes. [8] Attributes in table on Figure 3 are "Protein-code", "Protein-name", "Length" and "Species-origin". For the attribute "protein-code" is the first record "P1001" and the last probably will be "P9999". This set of all possible values for given attribute is called domain. Table can be understood to be composed of table header and body and in that case the table header is referred to as schema.

The tables can be connected together to larger logical complex and to do so, it is necessary to uniquely identify their particular rows and also ensure existence of the unambiguous identification of relationship in referring table's row. Constraints restrict the set of values that can appear for given attribute and thus basically further restrict the attribute's domain. With aid of constraints it is possible to define rules that have to be fulfilled for selected attribute – such as uniqueness. For each row in every table it is mandatory to have unique identifier that is called primary key that is used for unique defining a relationship within a database. In a table that wants to refer to another table foreign key that contains value of primary key of referred relation has to exist.

Protein-code	Protein-name	Length	Species-origin
P1001	Hemoglobin	145	Bovine
P1002	Hemoglobin	136	Ovine
P1003	Eye Lens Protein	234	Human
…			

Figure 3 Example table that represents the relational database model [4]

2.2.3 Object-oriented database

Object-oriented database is in object-oriented programming system and can be used with object oriented languages such as Ruby, Java or C++. While using relational database system, the only thing that is stored in database is data and code is kept apart. In an object-oriented data model, "the code and data are merged into a single indivisible thing – an object." [10] Thus, instead of separation of relevant information and storing them to different places, they are stored in one place – the data and also operations over the data are stored in database.

2.2.4 Distributed database

Distributed database can be physically located at many places, even though it is still controlled by one central database management system (e.g. Reciprocal Net – central database management in Indiana University but there are also nineteen other sites in the world that participate).

2.2.5 Data warehousing

Data warehousing solves different problem than distributed databases. While the latter has data on many places, in data warehousing data is processed and then integrated into central database. On Figure 4 can be seen that in every database runs program that extracts new data and sends them to program that collects them and then his task is to clean them and prepare for integration in data warehouse.

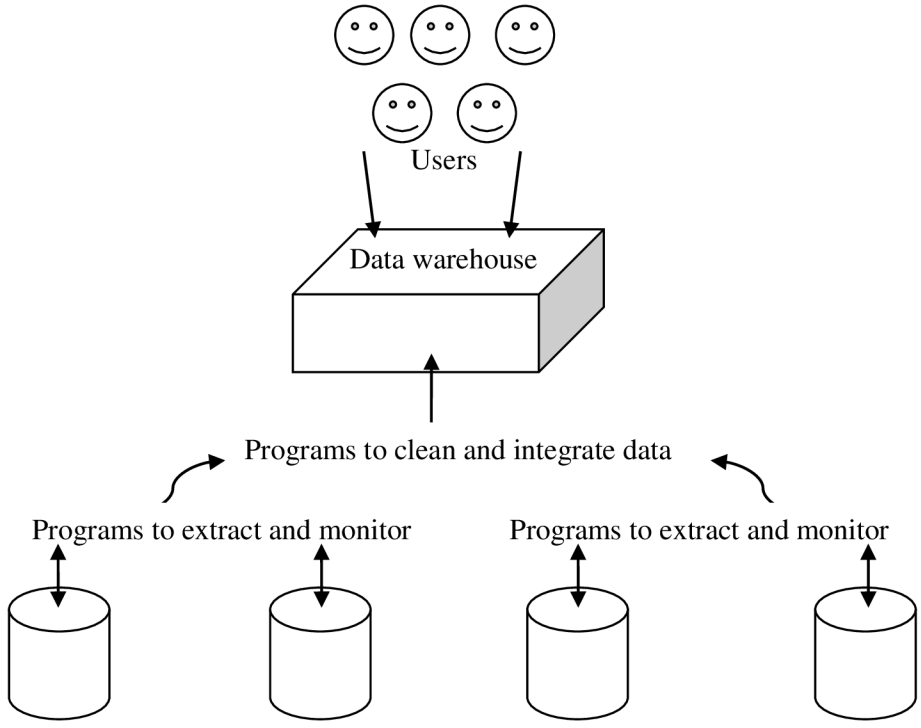


Figure 4 Schema of data warehousing [4]

For the listed types of databases does not necessarily apply, that the application of one type eliminates application of different type. For example, in Macromolecular Structure Database (MSD) at the European Bioinformatics Institute (EMBL-EBI) is utilized relational model for storing data in combination with warehouse model that integrates desirable information [4].

2.2.6 Database access

Databases are usually accessible via internet. They have implemented web-based interfaces and thus the users can connect to them and run queries over their data. The other possibility is the local access for users which offers faster access, better security and also offers better flexibility while designing specific query. Nevertheless, the first mentioned option (database accessible via internet) is preferred by majority of users [4] and that is why a lot of bioinformatics databases are available online, e.g. National Centre for Biotechnology Information (NCBI), EMBL-EBI, etc.

All available databases are usually oriented not only at their topic of interest. If, for example, the database deals with protein sequence, then most likely there will be accessible some links to relevant data from gene sequence data. If biologist is looking for such a combination of data, then he does not have to get acquainted with all online available databases, which cover his area of interest and learn about their structure and query them separately. The process in which all relevant databases would be queried could take a long time. In almost all databases links occur that lead to relevant information from another databases and thus can help quicken the process.

While choosing the right database to be queried, it is necessary to find the scope of data and the type of data that is stored in the database [2]. This is the only way how to avoid misusing chosen database or searching in wrongly chosen database for data that is not there.

2.3 Database types

Via internet there is accessible big number of databases containing data that can be duplicated, real or apparent. The term *data* is usually designated by the key information that is stored in database. Another important term, *annotation*, usually refers to additional information, such as interpretation of data, research citations or relevant links to related records of other databases.

Theoretically the smallest possible record in biological database consists of the key data and data's *identity* (i.e. what is examined subject and where it comes from) and name of author of the new record. Such a record would not be overly reliable; thus, to the record is usually included this so called annotation, which can consist of paper that published the data, other known facts, interpretation and more aspects that could raise reliability of the record. Some of the available databases offer programs that online analyse their data and so the user can decide better whether to take the data into an account or not.

Even though the databases usually concentrate on certain aspect of field of interest, within the mentioned annotation user can find in these databases also links and findings that may cause that the database will provide itself as better information resource than standard literature review.

2.3.1 Primary, secondary data and data reliability

In databases can be stored basically two main types of data: primary and secondary. As one could say, the secondary data is derived from primary. Primary data thus can be considered as more reliable, especially when it is supplemented with raw data from experimental results. There, of course, can appear exceptions - when the data comes from experiment, where experimental error occurred and thus the resulting data was inaccurate.

In previous paragraph was mentioned that the secondary data is derived from primary. For example, measures of sequence relatedness and similarity for multiple sequences can belong into the set of secondary data. And as one could assume, the data could appear inaccurate or incorrect after a certain time, once the sequence was completed by new primary data. Thus, the frequency, with which such a secondary data is updated, is crucial. And if the interval is too long then the data should not be considered as sufficiently reliable.

2.3.2 Ontology

Many biological databases are available via internet. The goal of bioinformatics is the aggregation of data from different sources for purpose of analysing them and getting new hypotheses to test. To accomplish this goal, it is necessary to follow certain standards in storing the data in databases, so the data could be processed by computers and occurrence of any ambiguities would be prevented.

If there was a scenario, in which two biologists got from experiments the same data, but saved them into database and described them by different terminology, the data would be unnecessarily complex for processing by tools of bioinformatics. Deriving new hypotheses would be very inefficient from such data, and all of that because of there would be missing some standard that would specify the used terms and which would define the relations between those terms. These formal and explicit specifications and definitions are called ontologies.

Ontology makes sharing of data among different fields of biology easier. For example, program Gene Ontology Tree Machine (GOTM) is used for "analysis and visualization of sets of interesting genes based on Gene Ontology hierarchies" [11]. This mentioned Gene Ontology (GO) is project that is held across many laboratories and its goal is to provide united and controlled vocabulary in which can be found terms and relations related with genes for all living organisms. Besides of GO there are also other projects that deals with ontologies such as Microarray Gene Expression Data (MGED), sequence ontology project (SOP) or multiple alignment ontology (MAO). Ontology concerning protein-protein interaction is called Molecular Interaction Ontology (PSI-MI).

Term Name	Definition
protein-protein	Interaction between a protein or peptide and a corresponding protein or peptide.
imported	The data has been imported into the database form an external resource.
spoke expansion	Complex n-ary data has been expanded to binary using the spoke model. This assumes that all molecules in the complex interact with a single designated molecule, usually the bait.
clustered	Binary pair is defined by multiple pieces of experimental evidence which have been clustered together.
smallmoleule-protein	Interaction between a small molecule and a corresponding protein or peptide.
internally-curated	Data has been directly curated into this database from the paper describing the experimental evidence
mimix curation	Paper has been curated to meet MIMIx specifications
evidence	Binary pair is defined by a single piece of experimental evidence.
rapid-curation	Minimal interaction data has been extracted from the paper
nucleicacid-protein	Interaction between a nucleic acid and a corresponding protein or peptide.
imex curation	Paper has been curated to full IMEx specifications
predicted	The interaction has been predicted using a specific algorithm.
biartite expansion	Complex n-ary data has been expanded to binary using the bipartite model. This assumes that all molecules in the complex interact with a single externally designated entity.
experimentally-observed	Data has been directly curated into the database from the paper describing the experimental evidence or by direct submission by the experimenter.

Figure 5 Definitions of tags listed in Figure 20 from Ontology Lookup service (OLS) (available at: http://www.ebi.ac.uk/ontology-lookup/)

2.4 Online databases

While looking for database that would fulfil one's requirements, it is necessary to utilize internet searching and also it is recommended to check the list of databases from Nucleic Acids Research (NAR). In year 2009 was NAR placed among one hundred the most influential journals in biology and medicine over the last one hundred years [12].

NAR is journal that every year, in the first issue releases papers that makes summary out of new and updated databases. This list is called "Molecular Biology Database Collection" and is available online as "2012 NAR Database Summary Paper". In year 2006 there were 858 records and in year 2012 there is already 1380 databases listed [13] (see Figure 6). On the internet version of this journal, the list can be sorted alphabetically or the records can be displayed in fifteen categories. These categories are:

- Nucleotide Sequence Databases
- RNA sequence databases
- Protein sequence databases
- Structure Databases
- Genomics Databases (non-vertebrate)
- Metabolic and Signaling Pathways
    - *Protein-protein interactions*
- Human and other Vertebrate Genomes
- Human Genes and Diseases
- Microarray Data and other Gene Expression Databases
- Proteomics Resources
- Other Molecular Biology Databases
- Organelle databases
- Plant databases
- Immunological databases
- Cell biology

As the list above shows, in category "Metabolic and Signaling Pathways" appears subcategory Protein-protein interactions (For list of all subcategories see Appendix A) which contains ninety-eight databases.

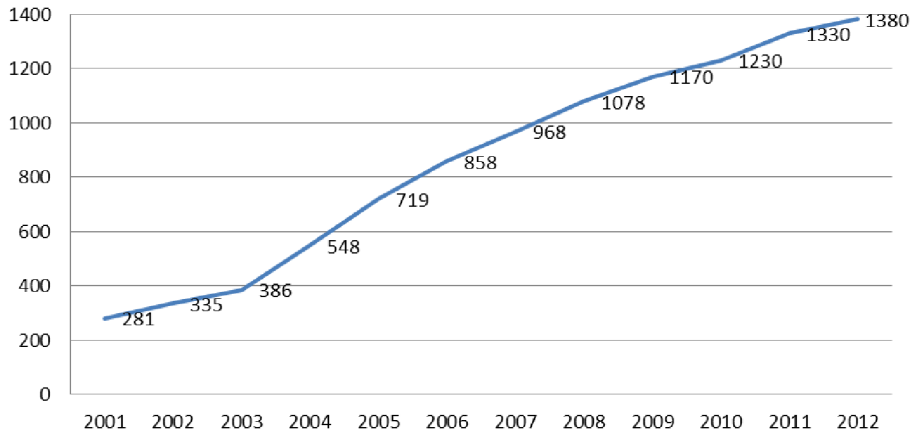


Figure 6 Growth of number of databases in Nucleic Acids Research online Molecular Biology Database Collection in ten years

2.4.1 Protein-protein interaction databases

Proteins can fulfil their function only if they can interact with other molecules or proteins, and thus they are integral part of biological networks. As was mentioned earlier, there are plenty of protein-protein interaction databases. Only in "2012 NAR Database Summary Paper" there are ninety-eight databases. Each of those databases has its advantages and disadvantages. For example Database of Interacting Proteins (DIP) according to [4] "contains rigorous criteria for evaluating of the reliability of each interaction", Molecular INTeraction database (MINT) contains also information about nucleic

acid and lipid interactions and Biomolecular Interaction Network (BIND) describes interaction at the atomic levels.

2.5 Quality of data in databases

The data stored in database is put into database by biologists, who usually add to the raw data also interpretations and, if it is relevant, then also links to records in another databases or further analysis. Although this could be one of the big advantages of online databases in comparison with, for example, literature reviews, this can also be considered as a drawback in case that the data from which analysis came from was not correct and came from experiment with inaccurate results. That is why it is desirable to provide database with system of determination of possible errors and level of uncertainty of stored data. Furthermore, availability of measurement of reliability is also desirable because some of the results of analysis can be less accurate than others. Even though users can insert notes about accuracy into annotation of entry, database itself can have means of how to determine reliability of inserted data.

It is hard for a user to assess data quality if he is not an expert. Thus, it is important to stress level of curation and validation that is held in a database. Nevertheless, if the data is not curated by team of experienced experts in biology, it still can be helpful. If one is expert, then he can go through the data and perform the curation himself. Furthermore, the majority of available data has origins in experiments executed with one of experimental high-throughput methods, such as comprehensive yeast two-hybrid system. This method creates many false negative and even false positives interactions. Also creating databases by utilizing text-mining (literature extraction techniques) are created databases to which one cannot completely trust, because the accurateness of the data can sometimes reach only up to 70% [2].

Automatic curation can eliminate data that is not invalid and manual curation can delay publication of experiment results. In some cases it is more helpful to make available data which was not curated just to offer scientists the possibility to study all available data. Even though the data is incorrect, it can still help to make a progress in their projects.

While utilising data from databases it is important to be aware of conditions in which respective data were obtained. If one of the components of cell was purified out of the cell and then was examined its behaviour, then there is possibility that the component will behave differently back in cell. Thus, even if the results of some experiment were correct, the inappropriate usage of this data can cause unnecessary problems and this applies also conversely, when the annotation is deficient and does not mention some of important facts that relates to the experiment.

With the quality of the data is also related data quantity. Some of available databases possess only hundreds of records, while others can possess hundreds of thousands of records. The cause of this situation can be just the difference in an age of two studied databases, but there can be also other reasons. Database could be, for instance, released as a demonstration of a new technology. This information among others can be found on statistics page of the database.

On Figure 7 below this paragraph is described the workflow of dealing with databases, where is also included some of the important information from all subchapters of the current chapter.

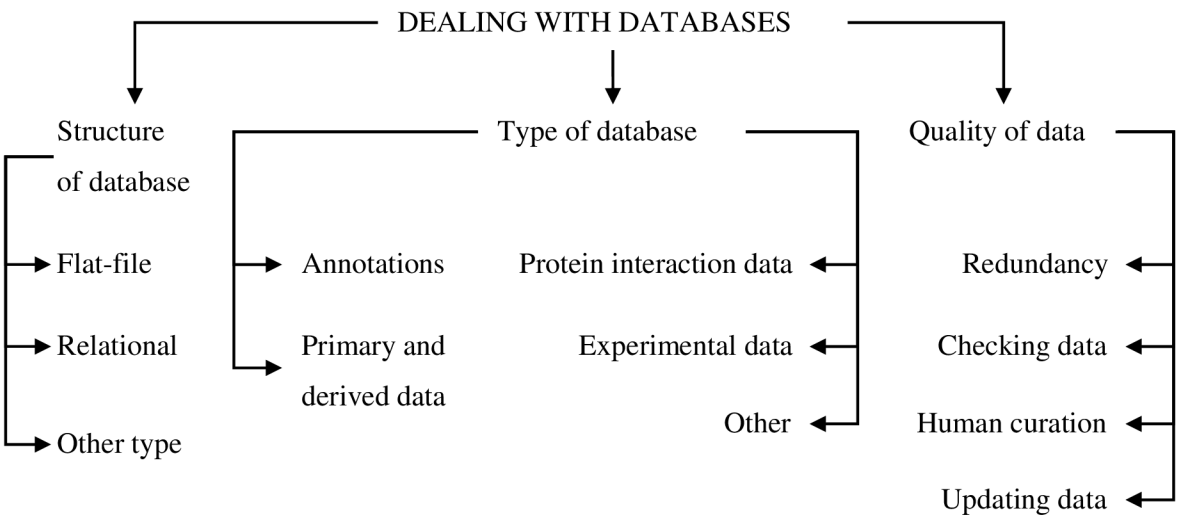


Figure 7 User should be aware of potential danger of working with inaccurate or incorrect data from bioinformatics databases. While deciding whether to work with particular database, it is necessary to identify structure of database, type of database and also quality of stored data. [4]

David B. Sears in his article Grand challenges in computational biology [14] pointed out three confounding principles in biology:

- for every rule exists exception
- for every biological phenomena exists nonlocal component
- every problem in computational biology is intertwined with another

From these three points can be deduced that in biology it can be challenging to create new hypotheses for testing even from the correct data, and therefore it is important to restrict occurrences of incorrect data. For the cause of preventing incorrect data to be found by biologists in online databases, manual curation or computer-based analysis can be done after entering data into database or sometimes even before data is stored.

2.5.1 Computer-based consistency check

Computers can aid the process of entering data into database by checking whether during the process an error occurred. The data, entered into database, is often inclinable to be faulty, because the process of entering data into database is driven by a human. And thus, there can appear not only typist's errors but also missing information. On the other hand, not all of missing information is important and sometimes it is necessary for human to have a look at the record and resolve the issue because annotations can provide further information about the data.

The system has to be aware of level of experimental uncertainty, such as the example on Figure 8, where can be seen the uncertainty, that can occur while identifying base on a certain position in DNA. When from the experiment it is not obvious which one of the bases is supposed to be in the sequence on the certain position, then it can be replaced by alternative Letter (e.g. there can appear M instead of "A or C" in database). Similar uncertainty can occur also in protein sequences.

There are many properties of a record that can be checked automatically, like bonding geometry in protein structures. In this case, biologists already know about the limits within the chain. There is, for example, limit for the main chain double bond between C and O [4]. Then, from the atomic coordinates of protein can be calculated lengths of bonds and if issue was found within the context of bond lengths, then the error can be either corrected, or there can be put a note into annotation with description of the found error. This error check can include many more properties such as bond angles, torsion angles, chirality, etc.

Letter	Base uncertainty
M	A or C
R	A or G
W	A or T
S	C or G
Y	C or T
K	G or T
V	A or C or G
H	A or C or T
D	A or G or T
B	C or G or T
X or N	Any nucleotide

Figure 8 Uncertainty in DNA sequences [4] In DNA sequence consists of only four bases: adenine (abbreviated A), cytosine (C), guanine(G) and thymine (T). When the experimental result is not decisive, then on the position of certain base can appear alternative letters.

Automatic check can search also for presence of cross-references to non-existent databases or non-existent records in existent databases. Even though automatic check cannot fix these kinds of issues, at least such records can be highlighted and manually repaired later. Furthermore, user can be noticed about missing information about experiment. Without certain information, the experiment cannot be reproduced or unambiguously interpreted. To avoid this, there exist standards like MIAME. MIAME stands for Minimum Information About a Microarray Experiment, and, as can already be clear from the title, it refers to microarrays, and it ensures that the experiment can be accurately reproduced.

Among other problems, among biologists were encountered different ways how to understand certain terms or different spelling of the same words or alternative names. The answer to this problem is called Ontologies that were described in Chapter 2.3.2. Utilizing ontologies, then it is easier to search in text and correct misspellings.

2.5.2 Computer-based data analysis

Increasing number of experiments generates data from experimental results in electronic form and thus, the important information is easier to include. In the initial analysis can be decided, whether to store data into a database or not. This decision can be made on the basis of accuracy with which the experiment was performed or its other parameters.

When the data is already in database, then there are countless methods how to analyse inserted data. These methods can include statistical analysis, identifying protein family, protein's likely function. Generally applies, that more general automatized analysis makes mistakes less often. But on the other hand, in the case that the analysis produces very general results then can happen that the results of the analysis will not be useful because of its vagueness. When the conclusions made by automatic analysis do not have evidence in experimental data, then in database they can be labelled as "hypothetical". This is often the case of gene prediction, where the method of predicting the genes is still relatively inaccurate [4].

2.5.3 Manual curation

The speed of manual data curation cannot compare to speed of computer-based data analysis. The computer-based analysis can be performed over the whole database in reasonable time, while the manual curation needs more time. However, the speed of computer based analysis is not always sufficient for all data. There are still tasks that cannot be performed by computer in the same quality as if it was performed by human. If there are a lot of experimental data that has been experimentally characterised, then it is more beneficial to use manual curation rather than computer-based analysis. The same applies also for experimental results which are related to other entries in databases.

Manual curation does not reach the same speed as automatic data analysis and therefore, it is supposed to be used mostly in cases where manual curation will bring some benefits. This is the case

when experimental evidence exists for data and when entered data can be in a relation with already existing data.

2.5.4 Redundancy in bioinformatics databases

In biology, it is common practice that some experiments are repeated several times, and sometimes the results can be ambiguous [5], or can be proven as incorrect or inaccurate by further tests, even though the results looked like they were correct from the beginning. Different results can be obtained also because of occurrence of mutation in previous cloned samples. For example, in sequence databases it is common practice to store records that are not exactly the same, but very similar, and differs usually because of the mutations. In case, those two records were the same, one of them would be redundant. Databases that are constructed in the way, that they do not have redundant records, are called nonredundant databases. The redundant database can develop even from combining of several databases whose data sets were originally nonredundant, but due to combining with other databases they happen to be redundant.

2.5.5 Importance of regularly updated database

Up to date data is important for successful project, and databases that are maintained more carefully can possess also higher quality information. Even though the old data sets can still be invaluable in a research, it is recommended to be aware of the age of the data. One should do a research about the age of currently studied dataset. Such information is usually to be found on a homepage of the concrete database or at the FTP sites where the database can be available.

Because of new findings, Bioinformatics rapidly evolves and so can structure of records in database or the data that is stored in database. In case of the most rapidly evolving fields, the changes can be made on regular basis. Then, the differentiation between minor and major updates is more suitable because of better understanding of a user that will immediately find out why the data, he was working with, has all of a sudden changed. The minor version can be changed every month, but major changes should be made only once a year.

For the case that user came back, and would be looking for the specific record, he had been working with a year ago, all records should have a unique identifier which will ensure that even though the record has changed beyond recognition, user will be sure that he is working with the same record, that only has been updated and not with a another record.

One has to be careful also at the deletion of obsolete records. Some of the database's users could have worked with the record in the time that it was not obsolete and once he would like to have a look at it again, he can be confused why the record is not in the database anymore. Thus, the records that are supposed to be deleted should be first marked as obsolete and the additional information, like why the data has been marked as obsolete, should be provided and what has happened with the data, and how should user continue.

One of these projects, where the periods between updating records are short, and where deletion of records from database happens regularly, is the eukaryotic genome sequencing projects [4]. Current experimental techniques of sequencing the genome works with dividing the genome into smaller parts, that are sequenced and then reassembled back into complete chromosome. The results of many of these experiments are made public immediately after they were recorded. But in the process, the smaller sequenced parts are merged together, and thus; these smaller parts soon become obsolete.

2.6 Availability

Before one starts to work with a database, he should do a research on an availability of a database he wants to work with. It is not only about whether one can download the whole database via Web or FTP. Into availability is also included whether there are some intellectual property restrictions. On the

web are also databases which have copyright restrictions which do not allow user to use the database's data other wisely than for commercial uses. But generally, these databases are released under such licences that they are basically available for free for everyone if one keeps given conditions. For example IntAct database was released under Apache License version 2, and all data under the Creative Commons Attribution License. On the IntAct web pages it is also explained: "This means that you are free to copy, distribute, display and make commercial use of all records from the IntAct database provided appropriate credit is given." [21]

Also the technical background has to be taken into an account when evaluating whether the database will be used. The different databases can be available in many formats and if one wants to work with the databases differently than the database's web interface allows to, then one has to do a research what language is used to build the database and on the web pages of the database usually is to be found also description of database architecture.

2.7 Summary

Bioinformatics databases are utilized for storing all the data that is gained form research made with modern high-throughput techniques. The data can be saved in database that can have many forms (such as in flat-files database, object-oriented database, relational database, etc.) and can be accessed either locally or can be available online.

Online databases can be primary or secondary. The secondary databases are filled with data from primary databases or derived data, which does not have to be so much reliable as the data from primary databases.

Online databases are accessed and filled in by data by many scientists. So it is important to use controlled vocabulary. Otherwise there could appear misunderstandings and misinterpretation of data stored in databases. For data concerning protein interaction was created Molecular Interaction Ontology (PSI-MI) which is basically vocabulary which explains meaning of every relevant term.

Online databases are more popular because of their availability. Each database contains a lot of data. Nevertheless, databases usually choose their topic of interest and thus if one is searching for answers in specific area, it is important to find out which of the available databases stores information that is suitable for him.

There can be a difference in data reliability in various databases. Data can be curated by experts, or just integrated from different sources, such as text-mining, where accurateness of the data sometimes reaches only up to 70%. In a record in database should be included information about origin of the data, and thus the user should be able to find more information about experiment where the data comes from and decide himself about data reliability. In addition, there are available online services, which can assess the data and tell to scientist its estimated reliability. Data reliability is also influenced by recency of the data in database and how often its content is updated. In addition it is important to find out what types of curation (manual, computer-based curation, etc.) databases perform over its data and whether the data is available for everyone without any charge.

3 International Molecular Exchange Consortium

3.1 Introduction

Protein-protein interaction data's importance has recently grown enormously. This data is used widely in biomedical research [20]. And with the risen utilisation of protein-protein interaction data, methods to gain this data are successively improved and also number of protein-protein interaction databases raises continuously. Many commercial bioinformatics databases which have private funding have been created, but besides that also many public databases, which are available for everybody without any charge.

Data stored in protein-protein interaction databases serves as a raw material, from which are built interaction networks. To build a network, scientists usually combine information from protein-protein interaction databases with other types of information. According to L. Bonetta [15] is one of the most popular tools, which can visualize networks, called Cytoscape. Cytoscape is also capable of enhancing the created network with other types of data. The raw material upon which can be built such a network needs to be taken from protein-protein interaction databases. L. Bonetta states that the most significant databases with protein-protein interactions are: DIP, BioGRID, IntAct and MINT.

The databases listed above are members of International Molecular Exchange consortium and in addition to that they also stated that their records will be available through a PSICQUIC. PSICQUIC is a web service and its employment is motivated by better access for all users to the data from all International Molecular Exchange consortium partner databases. Currently, on a PSICQUIC web site is stated that there are twenty-five databases (including the four databases already listed above). PSICQUIC will be described more in detail in Chapter 5.

This chapter describes the needs that led the most significant databases to create International Molecular Exchange Consortium (IMEx) and what are the biggest achievements of this consortium.

3.2 HUPO PSI-MI

While the number of protein-protein interaction databases grew, a necessity to create a standard that would help to interchange data between databases started to be more distinctly pronounced. Almost ten years ago [24], one of the biggest interaction databases realised that the effort they put into data curation could be smaller, if they joined their forces and shared data between them. To be able to share data, there had to be created a new standard for data exchange, because before that point in time, every database was developing their own structures and systems of storing all gained interaction data.

After years of development by Molecular Interaction (MI) group of the Proteomics Standards Initiative (PSI), a work of the Human Proteome Organization (HUPO) [25], new data model was introduced to public in 2004. This standard has been fortunately accepted by other online interaction databases. And these databases offer the whole data sets for downloading in PSI-MI extensible markup language (XML) interchange format. The PSI-MI was latterly supplemented with MITAB, which is simplified tabular format for "fast Perl parsing or loading into Microsoft Excel" [22].

3.3 MIMIx guidelines

As was mentioned before, the main goal of this consortium is to join efforts of the many recently created databases and faster aggregate available data. Some of online databases have private funding

and thus do not share their findings with others, but this approach is according to by IMEx considered to be inefficient. Different online databases should share their data and make searching for information as easy as possible. The current situation with online databases, where many of them can contain the same data, or worse, different data derived from the same source, or absolutely different set of data, and thus user is forced to write different queries and search many different databases, is not optimal. According to IMEx, the idea to facilitate data exchange became more significant because "accessing all publicly available molecular interaction data, even on a specific biological or biomedical topic, is a challenging, time-consuming task" [23]. Because of the number of available databases with similar areas of interest, one has to query many of these databases, and furthermore, each of them has a different interface. Each of the databases uses its own identifiers and the available data in one database can be also available in other databases. That can lead to misinterpretation of results of the experiments by scientist, who would like to utilize them in his work. Problems can arise because of incomplete information, which was stored into a database by irresponsible or forgetful scientist. Incomplete data in databases can among others lead to "time-consuming, error-prone attempt to derive the missing information" by scientists, which try to curate and aggregate the data from one database to another [24].

The minimum information required for reporting a molecular interaction experiment (MIMIx) guideline can be considered as the second step to facilitate the access to the information in various databases, after the development of a common file format for representation and exchange of protein interaction data.

As the unabbreviated title of this guideline suggests, MIMIx describes what information has to be provided when experiment is being reported. The MIMIx-compliant record was created as compromise after a discussion between scientists – it is not too thorough, so the scientists will not be bothered by filling-in too many unnecessary data while uploading results of their experiments, and it should have all required essentials for other scientists to avoid misinterpretation of the experiment's results. The MIMIx guidelines are not static, but they are supposed to dynamically evolve in the same way, as do bioinformatics environment itself.

To fulfil all MIMIx requirements and create MIMIx-compliant database, inserted information has to be understandable. To avoid misunderstandings or misspellings, vocabularies are under control and only those developed by HUPO-PSI should be used [22]. For instance, if one wants to find meaning of term "protein-protein", then he needs to go to Ontology Look-up Service browser web site at http://www.ebi.ac.uk/ontology-lookup. Utilizing controlled vocabularies eliminates misunderstanding and unifies terminology for all who wants to insert a new data into databases and also pertinent data curation should be less laborious and more effective.

A MIMIx-compliant database does not have to contain all necessary information to reproduce the experiment. It is rather intended to quickly provide information to scientist, who can assess it and decide if it is relevant to him. If needed, the record should contain a reference to original article with the experiment, where a user can find every necessary detail about the experiment. Nevertheless, all IMEx partners have adopted PSI-MI XML interchange formats [24] and by utilizing this format it is ensured that when every experiment is entered into a database, it is provided with more information that it is required by MIMIx.

3.4 Literature curation

IMEx tries to join efforts of all primary interaction databases and engage cooperation between them. After creation of common format for facilitating data exchange and defining the minimum information which has to be included in record, the next logical step was to unite also approach to the data curation process. If there is already ensured data exchange between primary databases, there also should be unified approach for getting data from literature, because that is the only way how to avoid collecting the same sets of data from the same publications by different databases. Furthermore, potential problems could arise because of the attitude of difference databases while curating data.

Without specifying how the data curation should be processed, two databases could have different interpretation of the same data from the same publication. If there appeared two different representation of the same publication in two different databases, then a user would be left confused and it would be up to him to choose which of the data he considers to be correct. To avoid such situations and quicken aggregation of information from available free interaction publications, five primary databases created IMEx consortium in September 2005 [23].

3.5 Development of the International Molecular Exchange consortium

International Molecular Exchange (IMEx) consortium was founded by five molecular interaction databases and currently the number of databases has grown to eight full members and one observer member. The full members are: DIP, IntAct, MatrixDB, MINT, Microbial Protein Interaction database (MPIDB), Interologous interaction Database (I2D), InnateDB and Molecular Connections (netPro). The only observer member is Biological General Repository for Interaction Datasets (BioGRID).

Full members agreed to work on curation of publications to IMEx consortium standard and to share their results with others. To each one of IMEx full member were assigned at least one journal. How the journals were divided among full member of IMEx consortium shows Figure 9. Thus should be ensured that the main source of data is covered by IMEx consortium and the work will be distributed fairly. Furthermore, the members could choose journals according to their experts' interests and specializations. Due to sharing of gained data in coordination with the rest of IMEx members, the progress was significantly accelerated. While full IMEx consortium members work on getting and presenting new curated data, observer member just cooperates with full members on creating curation rules and finding ways how to improve curation quality. In collective effort of full and observer members was created document, which describes the curation process. It is available online at IMEx consortium web pages [26] and it should solve all ambiguities which can appear while curating data. When some data from experiment is published, it usually takes up to three months for the curated data and to be made available in a database.[23]

Journal	Period of coverage	Database
Cancer Cell	January 2006–present	IntAct
Cell	January 2006–present	IntAct
FEBS Letters	January 2005–present	MINT
EMBO Journal	January 2006–present	MINT
EMBO Reports	January 2006–present	MINT
Journal of Bacteriology	August 2007–present	MPIDB
Journal of Molecular Signaling	November 2006–present	Molecular Connections
Matrix Biology	January 2009–present	MatrixDB
Molecular Cancer	September 2010–present	Molecular Connections
Molecular Microbiology	August 2007–August 2009	MPIDB
Nature Immunology	October 2010–present	InnateDB
Nature Structural & Molecular Biology	January 2006–present	DIP
Oncogene	September 2010–present	I2D
PLoS Biology	January 2003–present	DIP
Proteomics	January 2005–present	IntAct
Structure	January 2006–present	DIP

Figure 9 Journal coverage by IMEx consortium members in 2012 [23]

The experts curating the data from journals should not be targeting at any species and all articles in all journals should be treated in the same way. However, Figure 10 shows that the largest percentage (almost 88%) of all curated data belonged to the five well-studied model organism. These organisms are: Sacharomyces cerevisiae, Homo sapiens, Escherichia coli, Mus musculus or Caenorhabditis elegans.

Articles from the listed journals on Figure 9 are not the only source of data for IMEx consortium members. They can retrieve data also from additional publications. The choice of these additional publications is usually based on collaborating partners, specialisation of the database or expertise of curator. For instance, into IntAct database has been entered curated data set, which concern with protein interactions and their role in Alzheimer's disease [27]. In order to cope with new published articles more efficiently, in 2010 [23] was by IMEx consortium started a web service called IMExCentral. This service is for partner within IMEx consortium to eventually reserve article or publication for curation.

IMEx consortium partners decided to record extensive details of all experiments. That is important, because it has been proven that even small change in one of initial steps of experiment can affect the result considerably [28].

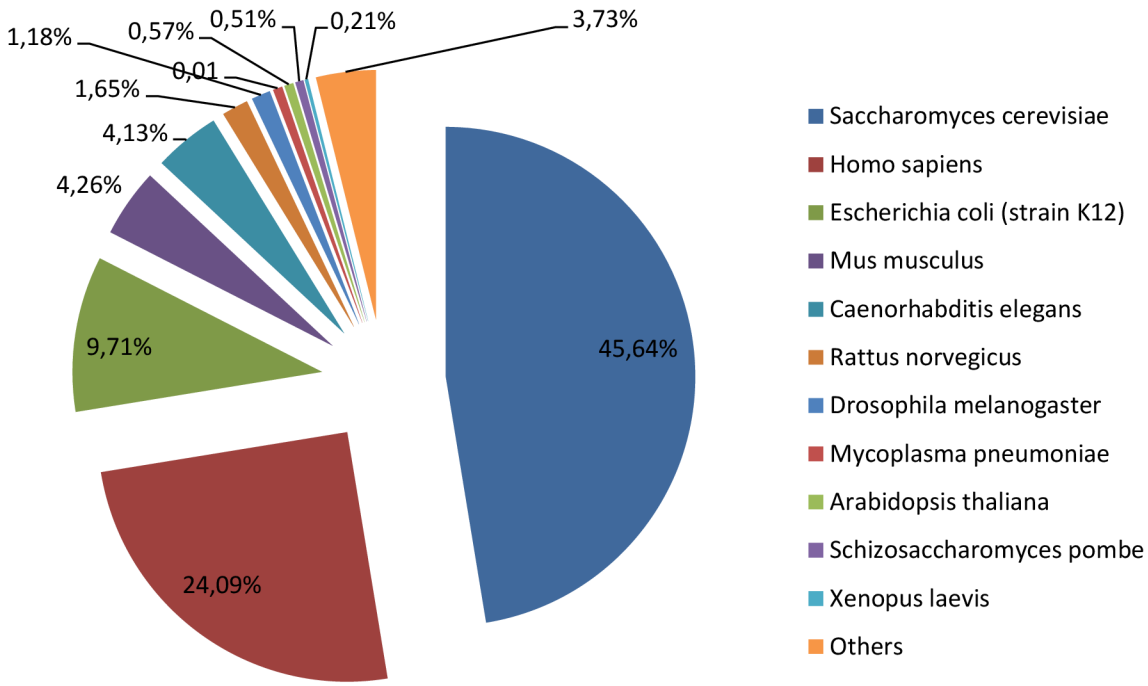


Figure 10 Species for which data were available in the IMEx dataset in December 2011 [23]

Interactions between two molecules are called binary interactions by IMEx consortium members. These binary interactions can be divided according to their type. On Figure 11 are displayed eight most significant types of binary interactions. The two most significant types of binary interactions – *Physical association* and *Association* – indicate possibility that not all interacting members were identified. This is due to methods that were utilised. The *direct interaction* group consists of molecules about which it is known that they have actual physical contact with each other. As *direct interaction* are considered only interactions that where identified only in vitro methodologies even though it is possible to create a strong evidence of direct interaction even with properly performed yeast two-hybrid assays 57[23].

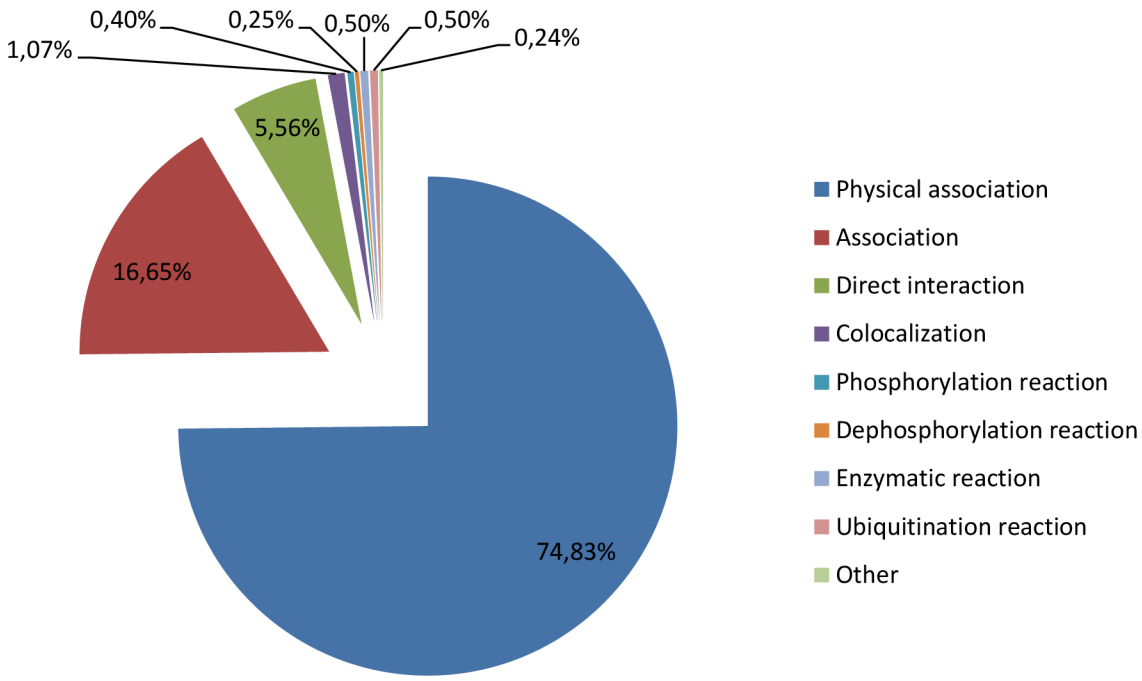


Figure 11 Types of interaction data represented in IMEx dataset [23]

3.6 Quality control

To sustain the level of quality across all databases, IMEx consortium worked on tools which should facilitate mutual quality control. One of these tools is PSI validator [29]. PSI validator provides syntactic and semantic checking of XML files based and furthermore executes rules on PSI-MI ontology. Also exercises are performed to compare the curation level in the individual databases. This exercise is formed by a paper which is given to each of partner databases to curate it. The curated results are then compared and possible differences are discussed to ensure that all curation rules are still applied consistently.

3.7 Data exchange

Data exchange between collaborative databases is highly resource-consuming. This applies to both copying complex data from one partner to another and managing records update (or deletion).

IMEx consortium developed "a standard interface for direct computational access to standards-compliant molecular interaction data resources" [23]. This interface is called PSI Common Query Interface (PSICQUIC). It allows user to query multiple databases and IMEX consortium members agreed to use it to minimize the data-exchange overhead while they distribute their data to partner databases. Every IMEx partner has its own active PSICQUIC web service, which allows other partners to query all other partners' databases and search in their most current data.

3.8 Summary

The International Molecular Exchange (IMEx) consortium was founded after a success of the PSI-MI, so it could take care of new standards for regular information exchange between primary databases and cooperation in curation, in order to cover all accessible interaction data from the literature.

Main goal of this consortium is to provide user with a comprehensive non-redundant set of curated data of consistent quality, in which user can search for information he needs. IMEx stands behind MIMIx guidelines and also set rules how to take care of coordination of curation tasks between partner databases. To accomplish that, common literature curation rules needed to be released and the literature curation started to be central controlled.

The information between them should be facilitated by utilizing standard exchange language called the Human Proteome Organization's Proteomics Standards Initiative Molecular Interaction format (HUPO PSI-MSI) [22]. Members by joining IMEx agreed to share and provide all available data sets concerning molecular interactions.

When full members of IMEx consortium agreed on common approach in literature curation, they released manual, which describes how the curation needs to be performed and how to react in different situations. Thanks to this document the results of curation process are consistent.

The consortium has many members and it is necessary to keep the same level of quality across all members. IMEx has developed tools how to keep consistent quality of output (such as PSI validator) and methods how to compare results of all members work by randomly giving to members the same publications for curation and then comparison.

IMEx consortium has in plan to continue in aggregation of new data in IMEx records. The data set will continue to grow bigger as new IMEx consortium partners will be accepted and their data sets integrated and also when archive data, which was created before curation rules were fully defined, will be curated again. The goal of IMEx consortium partners is to leave the scenario when curation is done after publication and move to more efficient curation before publication. In this way will be ensured that the curated data will be described and precisely represented in the way author intended to, without any lack of misunderstanding or space for ambiguities and factually correct.

PSICQUIC is REST-compliant web service developed by IMEx and it is also accessible via SOAP. Therefore, in the next chapter will be described REST architectural style and SOAP protocol in order to make decision in chapter 5, which of these two approaches will be utilized while accessing PAICQUIC.

4 Web services and REST

4.1 Introduction

Recently, number of information systems, which migrated from desktop systems to online environment has significantly increased. In online environment the systems are usually distributed, and thus there arises new questions on how to make the system functional, how to deal with high demands on throughput of these systems. There also appears requirements to connect together heterogeneous systems. In this case, the web services are the solution.

Web services will be introduced and their meaning in heterogeneous environment explained. Also will be described two main groups of Web services. Each one will be described and then those two groups will be compared in order to choose one of them for the implementation of application for retrieval of protein interactions.

4.2 Web services

Web services facilitate communication between machines connected over a network. In other words, it is software interacting exchanging messages in a heterogeneous environment. The messages can be sent in many formats, such as in XML (or other industry recognized platform independent standards). Web services have their standard developed and defined by W3C consortium. And its official definition from W3C web pages is: "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards" [37].

According to Booth, et al [37], Web services can be divided into two main groups

- REST-compliant Web services
- Arbitrary Web services

The difference between those groups is apparently that the services from the first group have to have REST architectural style implemented. The second group of Web services is created by services which infrastructure is created by three basic technologies:

- Web Service Description Language (WSDL)
- Universal Description, Discovery and Integration (UDDI)
- SOAP (originally stood for Simple Object Access Protocol)

Web services reveal set of operations which are available and described using WSDL. It is standard format for description of web service interface. For every web service there should be description of its interface. WSDL in version 1.1 was not suitable for describing communication with HTTP and XML, and thus it did not have means how to describe access to applications in REST-compliant Web services. However, since version 2.0, WSDL was improved and now it can be used also for description of REST Web services [38]. In larger systems, it is possible to register web service into UDDI registry, which utilizes searching for certain service with certain parameters. If client wants to use some web service, then he needs to acquire its description directly from service or over UDDI. SOAP is protocol used for communication and will be described in following chapters.

4.3 SOAP

It serves for exchanging messages in decentralized and distributed environments. SOAP was created as a light-weight protocol to be a foundation for web services. The other standards - WSDL and UDDI - were created after SOAP was released and their purpose is just making usage of SOAP easier. First version of SOAP protocol was released in 1999 and it was created by DevelopMentor, Microsoft and userLand. It was developed because these companies wanted to have a RPC protocol based on XML [40].

SOAP enables sending messages between two peer clients, but upon sending message can be built common communication scenarios. SOAP is protocol from application layer and defines the XML format in which the messages have to be sent over another protocol in application layer (typically HTTP).

If one application wants to request some information from another application in SOAP, then it sends message in XML format to that application, which servers that request and then sends back result to the first application. SOAP can be utilised instead of Remote Procedure Call (RPC).

4.3.1 SOAP message

XML format for SOAP messages was chosen because it is easy to transform file from XML into another format needed by specific application. XML is also readable by human and another advantage of XML is that it can be easily validated and thus it can prevent errors from occurring during file processing. On the other hand the XML format is quite verbose and that is why it has impact on the amount of data sent over network; thus it has negative influence on speed of communication.

Structure of SOAP message is created by root element e*nvelope*. It contains namespaces definitions to distinguish SOAP structure defining elements from the content of a message. Envelope encapsulates two other elements: optional *header* and compulsory *body*. In the header additional information for message processing can be found, such as information about transaction, user permissions etc. In the body element is stored the information about which one of the methods provided by server should be launched. In the body can be stored either method name with its parameters, or result data (if it is a body element from response message).

```
<soap:Envelope xmlns:soap="http://www…soap-envelope">
   <soap:Header>
      <!-- Header information (optional) -->
   </soap:Header>
   <soap:Body>
      <!-- Only child element here would be Fault if error happens -->
      <m:GetSomething xmlns:m="http://www…someghing">
         <m:SomethingProperty>blue</m:SomethingProperty>
      </m:GetSomething>
   </soap:Body>
</soap:Envelope>
```

Figure 12 Example of simple SOAP Message. Header is empty, because it is optional and body contains request. If it was a response, then the GetSomething element would be renamed to GetSomethingResponse and it would have new child elements with found "Something" that is blue.

4.4 REST architectural style

REST architectural style was for the first time introduced by Roy Fielding in 2000 in his PhD thesis. Simplified explanation of the REST architectural style is that it describes principles how to use Web standards (e.g. URI, HTTP) and if creator of a web service will stick to these principles, then the created system will utilize the best of web's architecture in creator's advantage.

In Fielding's thesis is an architectural style defined as "a coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style." [35] For instance, one of architecture styles is called Null Style - without any style at all.

In a REST web service, communication between client and server are stateless; thus, information about the state is sent in requests and responses. REST architectural style differentiates between *resource* and *representation*. Client access uniquely identified resources on a server utilizing resource identifiers (i.e. URL), but server will send him representation.

If web service is REST-compliant, then it has to fulfil some constraints, which basically facilitate programing new applications. Programmer does not have to implement any remotely accessible services in RESTful application, but instead of it uses standard interface to a set of resources.

4.4.1 REST constraints

Fielding in his work writes that the REST is a hybrid style. That means that it was creating by utilizing constraints from other architectural styles merged together with some additional constraints. In following subchapters are listed and described six basic constraints that have to be fulfilled by RESTful system.

4.4.1.1 Client-server

This constraint describes server as waiting component and offering a set of services. When client wants the server to perform some action, then he needs to connect to connector and requests an action. Server can reject or perform the requested action and then send result to the client.

Utilizing this principle led to separation of concerts and thus to better portability. User interface is not dependent on data storage and furthermore, there can be more user-interfaces, more data storages and all of them can be developed independently.

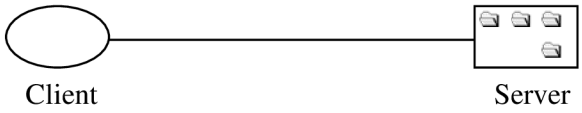


Figure 13 Client-server [35]

4.4.1.2 Stateless

This is constraint that is added to the client-server constraint. The most significant property of this constraint is that context is kept on client side. This means that when client requires server to perform some action, the client is needed to send in his request everything necessary for understanding that request on server's side.

Server not having to store any information about active session brings several useful properties. These are visibility, reliability and scalability. Better visibility means that server understands what he is requested to do from client's request and does not have to reach for the context of the request on his side. Visibility brings up better reliability [39] and also improved scalability, because when server is not forced to store context and some other additional information about all clients' active sessions, then it is easier for server to free all resources after a client's request has been served and thus it facilitates implementation.

Drawbacks of this constraint are higher demands on network throughput and the fact that server loses control over implementation of logic on client's side. The network load is higher because server is not allowed to store information about client's session even because of repetitive requests; thus, all the information that could be stored on server needs to be sent over the network with every request.

4.4.1.3 Cache

If to the client-server constraint is added also a cache, then it should lead to more efficient work with network. With some data already kept in his cache, client can avoid asking server the same question several times in a row. By storing some possible useful data at client's side, it is possible to improve scalability and user's experience. For many requests in a row the cache can significantly reduce average latency. Utilizing cache can, however, also lead to worse reliability in case when the data stored at client is different than the data stored on server.

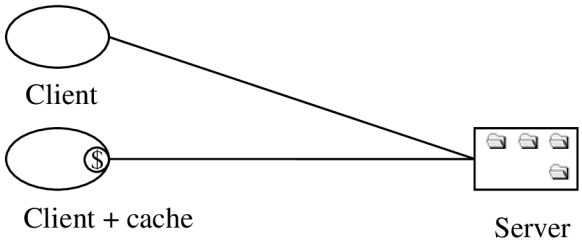


Figure 14 Client-server + Cache [35]

4.4.1.4 Uniform interface

Fielding considers uniform interface to be the most distinguishable constraint from all the other architectural styles.

Any information, which can have a name, can be also called *resource*. For example documents and images can be resources. And every single resource needs to have identifier. In HTTP it is URI which unambiguously identifies resources. There is a difference between resource and its *representation*, which server sends back to client in requested format. If user accessed a resource, then he has his representation and if he has sufficient permissions, then the resource can be deleted. In the message server sends back to client can be included links to relevant information that could user utilize.

Although according to Fielding REST is not bound to any explicit protocol, the uniform interface constrain and its advantages can be shown on an example with HTTP protocol. If there was protein-interaction database, then very simple way how to add new protein into database would be to connect to running Web service and call AddProtein method, if one wanted to delete protein, then one would have to call DeleteProtein method. One needs to know names of the methods. However, utilizing REST architecture style facilitates access to data and it is significant easement when REST architectural style is utilised with HTTP protocol and thus there standard HTTP methods can serve as a uniform interface.

Normal Method names	HTTP methods	REST Uniform URL
AddProtein	PUT	Protein/Insulin
DeleteProtein	DELETE	Protein/Insulin

Figure 15 REST uniform interface constraint demonstration utilizing HTTP protocol

Simplifying architecture leads to better interactions visibility and uniform interface also shields individual applications from problem of arguing about communication protocol. The disadvantage is that efficiency of communication is not as good as if for every application there was optimal interface and it would be possible to call methods with exact number of parameters specific for each application.

4.4.1.5 Layered system

Layered system is supposed to simplify complex system by dividing it hierarchically. In such a hierarchical system communication is allowed only from lower layer to immediate higher layer. It is possible to put between server and client intermediary and client will not recognize whether is

communicating directly with server or not. Intermediaries can serve for better load balancing across the network or applications that are not used often can be moved from server to intermediary to make the server simpler.

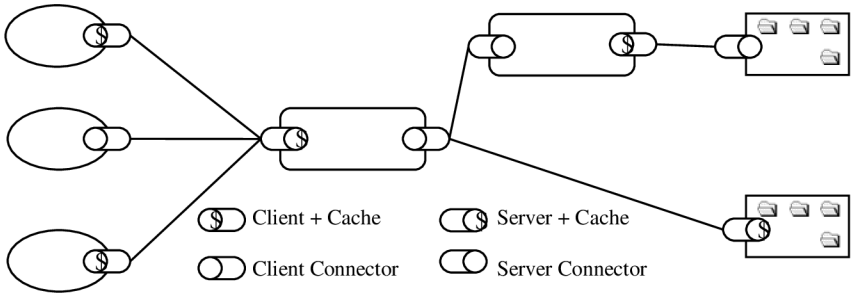


Figure 16 Layered system

Layered system can be both advantage and disadvantage. As was mentioned earlier, it can ensure load-balancing, enlighten server by moving its components to intermediaries. Number of layers can have an influence on latency; however, if cache is utilized properly, the latency can be even improved in some cases.

4.4.1.6 Code on demand

This constraint ensures that not only data, but also code that can be executed on client can be sent from server. Client does not have to be always implemented with the all functionality and some of its functions can be stored on server and sent to the client only when requested. Because of this constraint the client can be centrally managed to perform certain actions from server. Nevertheless, it can bring security issues when server cannot be trusted. Because client cannot see what he is supposed to do by executing code that has been sent to him, visibility is reduced. Thus, code on demand is the only optional constraint in REST.

4.4.2 RESTful web services data access

Above were described all constraints that have to be fulfilled so the web service can be pronounced a RESTful web service. Furthermore, there has to be pre-defined set of operations. If HTTP is utilised, then the set of operations is taken from it. There are five standard methods in HTTP and each of them has effect on REST web service resource that is identified by URL.

Method	Description
GET	Get resource representation
POST	Create a new resource
PUT	Modify an existing resource
DELETE	Delete an existing resource

Figure 17 HTTP methods and their meaning in RESTful web service

If the meaning of above mentioned methods will not be implemented exactly, then the web service is not pure RESTful web service.

4.5 REST and SOAP comparison

Both REST and SOAP are language, platform and transport agnostic. Whereas SOAP can work only with XML, one of great advantages of REST is that it permits many different formats. The most significant features of REST architectural style are scalability of communication between components and possibility to use intermediaries, which brings up robustness and higher efficiency due to usage of cache. REST facilitates independent and easier development of individual components and also

facilitates communication between individual components, because it defines unified interface (set of methods) and thus programmer is not bothered by creating new communication protocol or solving communication in another way.

When one needs to discover new functionalities on REST-compliant web service, it could be a difficulty, because there are no standards available so far, whereas for SOAP there is available UDDI registry. Furthermore SOAP allows more flexibility while designing user interface and number of methods, because REST is limited by HTTP methods. On the other hand, SOAP does not allow application to utilize all HTTP properties as well as REST.

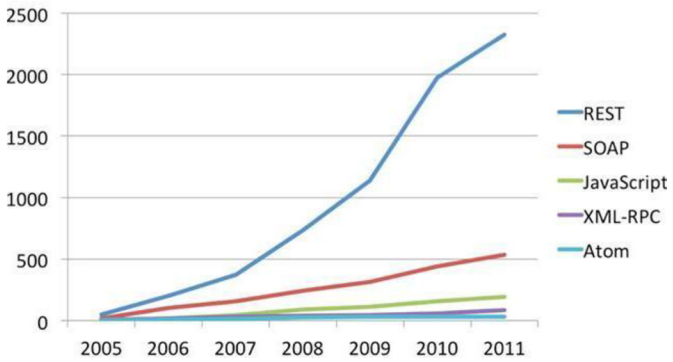
SOAP uses its own message format and all communication goes through HTTP POST method. That is why all network members should understand the SOAP messages otherwise it is far more complicated to use proxy. REST do not have this problem while uses only HTTP protocol and that is also why it is easier to identify what client wants to do. In REST it is sufficient to use only URL to identify resource, whereas SOAP needs to send whole XML message in proper SOAP format with proper request in body.

While writing application which uses SOAP, programmer can utilize many tools which can generate plenty of code automatically. Also client of web service can be generated directly from definition in WSDL.

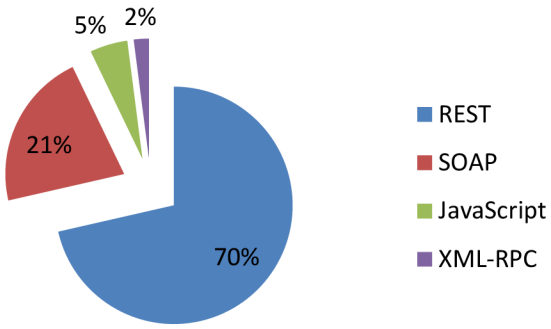
Despite of many good properties of SOAP, Paul Prescot in his article discusses the importance of easy and unified interface and compares development in web services to Unix file system [42]. Even in Unix file system there is unified way how to create file, edit it and delete it from hard drive. Even though user can improve usage of file system by some extensions, it is still based on the same principles. User can in SOAP think of many methods how to access stored data and give to these methods different names. Nevertheless the basic operations over every data are still the same as they are in Unix file system, or in relation databases: SELECT, INSERT, UPDATE and DELETE. The system utilizing such a general interface can lose some of its performance because this approach does not have to be optimal for storing all kinds of data. However, interoperability benefits from this. Applications are able to access all kinds of data in the same uniform fashion by using SEELCT and they do not have to know that if they want to access data from customer table, they have to use SelectCustomerTableRecords and if they want to see data from car table, they have to use SelectCarRecords. Paul Prescot claims that although file systems and databases differ from web systems, the interoperability, reliability and scalability are for all of them central goals. Even web services have to work with various types of data information similarly to file systems and databases. And the REST is the way how to deal with all the data diversity on individual web services.

Furthermore, SOAP was created as “Simple Object Access Protocol”. However, the changes that it has gone through changed it so much that the former name is not true anymore and thus the SOAP is meaningless nowadays. According to Paul Prescot, SOAP has formed into a protocol framework. And furthermore, it does not integrate into Web architecture. The complexity of SOAP causes that many developers start to discover benefits of REST API (Figure 12).

Jon Flanders in his article, in which he tries to compare REST and SOAP [43], states that REST and SOAP have their advantages and disadvantages. Nevertheless, utilizing REST architectural style is in majority of all cases more advantageous and SOAP should be used when one of its particular features is wanted.



(1)



(2)

Figure 18 Development of API distribution since 2005 (1) and API distribution state in March 2012 (2) from statistics on programmableweb.com [41]

4.6 Summary

There are two main approaches how to create Web service – utilizing SOAP protocol and creating REST-compliant web service. The SOAP protocol was created as a RPC utilizing messages in XML format.

SOAP has gone through rapid development and therefore it was renamed from Simple Object Access Protocol to meaningless SOAP. The SOAP is complex protocol, which can cooperate with WSDL, which is language to describe web service interface and it also cooperates with UDDI, which is able to find applications according to inserted parameters.

The main goals of RESTful architectural style are general interface, independent development and deployment of components, hierarchical structure to reduce latency, improve security and scalability and encapsulate legacy systems.

Between users the utilizing REST while building web services is more popular because it is easier to implement and also because of its better usage of HTTP protocol and therefore better integrating into networks with proxy. REST has clear goal because it tries to give to each of its resources unified interface, which is in HTTP protocol created by four basic HTTP methods: PUT, DELETE, POST and GET.

For purposes of the implementation of the protein interaction retrieval application will be utilized communication with REST-compliant PSICQUIC web service.

5 PSICQUIC

5.1 Introduction

To better understand protein and its position in cellular processes, it is necessary to identify all molecules with which this protein can interact. There are many databases dealing with protein interactions. All of them have their own sources of data, which comes from experiments and observations of various laboratories all over the world. Because of new technologies and modern high-throughput experiments, it is challenge to maintain certain a level of quality and accessibility of the data.

Until 2004, before Human Proteome Organization Proteomics Standards Initiative (HUPO-PSI) created standard for molecular-interaction data called PSI molecular interaction XML format, there were not any unified means how to exchange data between different interaction databases or how to integrate disparate data sets. And after it was released, its simplified standardized version was created - Molecular Interaction Tabular format (MITAB). These community standards were widely accepted and implemented in more than thirty databases and now they are supported by many software tools. [30]

This PSI-MI format aids user with data integration from many different sources. But if user wants interaction data, he still needs to query multiple databases or download data from various servers. In addition, if one wants to have up-to-date data, he needs to download the data from these servers regularly. This was the motivation for developing community standard for computational access to molecular-interaction data resources – PSI common query interface (PSICQUIC).

Utilizing PSICQUIC to search in the whole set of data can be relatively difficult, because currently an origin of displayed data is not obvious. It is hard to say whether the data came from functional associations, text-mining, or experimentally proven binary pairs. Also, search will be done over redundant data, because part of searched data can come from data aggregators such as iRefIndex or Search Tool for the Retrieval of Interacting Genes/Proteins (STRING). While integrating data from other databases, it is possible that some data loss occurred. It means that some references and links can be missing. In an article about protein interaction data curation [23] is stated that on February 27th 2012 was PSICQUIC searched for data associated with publication with PubMed identifier (PMID):17923092. After querying PSICQUIC, there appeared six resources and in many of them was missing detailed information to explain that the data is derived from genetic interference assays. It is recommended to rather search only IMEx datasets. In IMEx datasets are not redundant data, and record details clearly defined. The data available through PSICQUIC is also available to be downloaded in MITAB and PSI-MI formats for free.

The biggest advantage of utilizing PSICQUIC is that queries do not have to be written all over again. For all data sources, which are implementing PSICQUIC, can be utilized the same query. And into the response message will be put all relevant answers from all queried independent sources. This message can consist of protein identifier or it can be built utilizing molecular interaction query language (MILQL) [30]. An example of utilizing PSICQUIC can be seen on Cytoscape. Cytoscape is an open source platform for complex network analysis and visualisations. It uses PSICQUIC queries to get all relevant information before final network is rendered. [31]

In total more than 16 million interactions are available to be queried through PSICQUIC services. List of all available PSCQUIC services is available online in the PSCICQUIC registry. Tags are assigned to every service to identify all accessible data. All these tags are from controlled vocabulary and their meaning can be easily found in Ontology Lookup Service in Molecular Interaction (PSI MI 2.5) [MI] ontology. Example of utilizing this lookup service is on Figure 19.

definition	**Interaction between a protein or peptide and a corresponding protein or peptide.**
subset_PSI-MI_slim	Subset of PSI-MI
xref_definition	PMID:14755292

Figure 19 OLS – Ontology Lookup Service example: result for searching for term "protein-protein"

5.2 PSCICQUIC registry

On PSCICQUIC registry web page is information that there is 151,781,435 Interactions from twenty-five PSICQUIC Services available. From these twenty-five databases, only three databases do not contain any data concerning protein-protein interaction (without protein-protein tag): ChEMBL, BindingDB and DrugBank. These twenty-two databases are displayed on Figure 20. In the most right column are listed tags for each of databases.

Name	Interactions	Tags
DIP	107,619	protein-protein, internally-curated, imex curation, mimix curation, spoke expansion, evidence
InnateDB	17,89	protein-protein, internally-curated, spoke expansion, mimix curation, evidence, nucleicacid-protein
BioGrid	337,957	protein-protein, internally-curated, rapid curation, spoke expansion, evidence
IntAct	294,273	protein-protein, smallmolecule-protein, nucleicacid-protein, internally-curated, imported, imex curation, mimix curation, spoke expansion, evidence
iRefIndex	1,374,549	protein-protein, imported, bipartite expansion, evidence
MatrixDB	845	protein-protein, smallmolecule-protein, internally-curated, imex curation, mimix curation, spoke expansion, evidence
APID	416,124	protein-protein, imported, spoke expansion, clustered
Interoporc	208,558	protein-protein, predicted, evidence
BIND	192,961	protein-protein, smallmolecule-protein, nucleicacid-protein, spoke expansion, clustered
Reactome-FIs	209,988	protein-protein, predicted, imported, clustered
MINT	137,403	protein-protein, internally-curated, imex curation, mimix curation, spoke expansion, evidence
STRING	26,045,661	protein-protein, predicted, imported, spoke expansion, clustered
Reactome	113,204	protein-protein, predicted, evidence
MPIDB	24,268	protein-protein, internally-curated, predicted, predicted, imported, rapid curation, mimix curation, imex curation, spoke expansion, evidence
Spike	36,248	protein-protein, evidence, internally-curated
GeneMANIA	120,644,180	protein-protein, predicted, imported
VirHostNet	13,808	protein-protein, evidence, internally-curated, rapid curation
InnateDB-IMEx	390	protein-protein, nucleicacid-protein, smallmolecule-protein, internally-curated, imex curation, evidence
I2D	817,915	protein-protein, internally-curated, evidence
MolCon	291	protein-protein, nucleicacid-protein, smallmolecule-protein, internally-curated, imex curation, evidence
TopFind	9,542	protein-protein, evidence, internally-curated, predicted
I2D-IMEx	904	protein-protein, nucleicacid-protein, smallmolecule-protein, internally-curated, imex curation, evidence

Figure 20 PSICQUIC Registry displays only services with tag "protein-protein". Definitions of individual tags are listed in Figure 5

The tags on the right describe available data in the database displayed in the most left column. The meaning of the tags can be found by utilizing Ontology Lookup Service (OLS). In the Figure 5 are listed all the tags from Figure 20 and their definition from OLS.

5.3 PSICQUIC data availability

PSICQUIC as an effort from the HUPO-PSI to standardise programmatic access to the data stored in molecular interaction databases, makes data available in many different ways and formats. HUPO-PSI decided that the best way how to reach the goal to satisfy needs as many potential users as possible and facilitate data access will be by specifying standard web service and by specifying common query language Molecular Interactions Query Language (MIQL). PSICQUIC is specified to be RESTful service and besides that it is also accessible via SOAP [34].

5.4 PSICQUIC REST access

The simplest way how to access data stored in databases is just utilizing web browser without any need of special tools how to access data. That is why some of users prefer REST over SOAP [34]. Nevertheless, using REST, there is possible to query databases by writing queries in MIQL, by interactor, participant or interaction identifiers. The URL syntax how to query databases is presented on Figure 21. There can be seen that the first part of the URL is the address of the web service. The list of addresses is available at PSICQUIC Registry page. Then, there is version of the web service. Currently there are three versions available and the newest one is version 1.2. After The specified version one has to choose a method how she wants to retrieve the data. As was mentioned before, one chooses between interactor, interaction and query. According to the chosen method the query itself has to be written by user. In the case that interactor or interaction methods were selected then the query is built up with identifiers, which are separated by AND or OR operands. As the last part of the URL are optional parameters. In these parameters one can choose for example format. In current specification 1.2 there are available nine different formats. In addition to the format, user can choose which one of the found results is supposed to be retrieved is the first one and what is the maximum of retrieved records. The newest specification offers along with format, first result and max result, also the choice whether the bandwidth should be saved and the data should be sent back compressed.

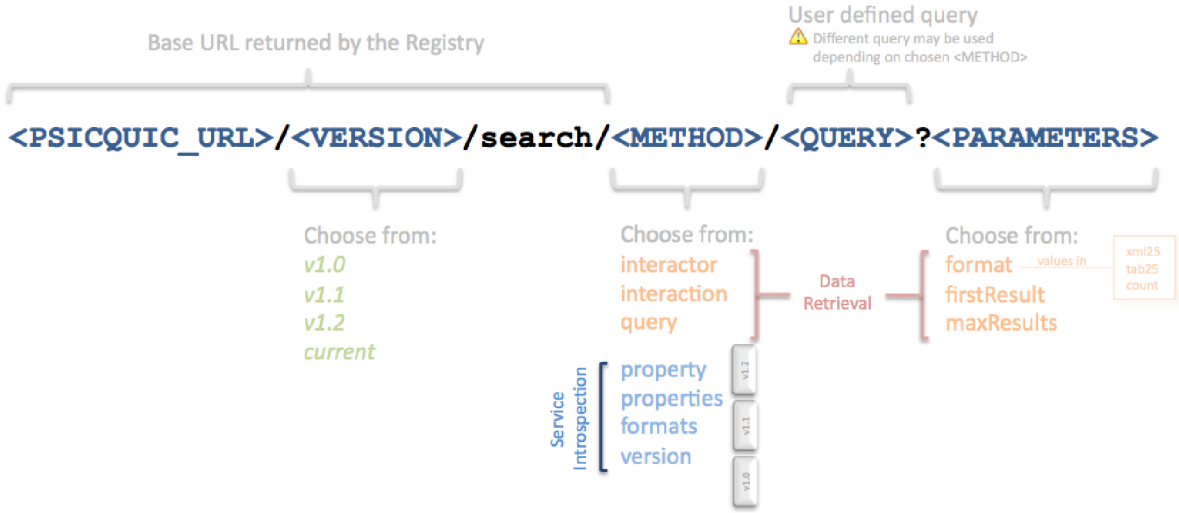


Figure 21 URL structure to fetch PSICQUIC data [34]

The status of the web service is the status code that is sent back to web browser. The status code can be found in header of the request.

Status code	Description
200	Not an error. Request is OK
406	Format not supported
500	Internal server error

Figure 22 Status codes supported by PSICQUIC REST

5.5 MIQL flexible query

One of options how to access data in PSICQUIC is writing own queries utilizing MIQL. Once there is PSICQUIC web service, which allows user to access different databases, it is necessary to also have a language, which will allow user to query them all by writing single query.

MIQL is based on MITAB, which is part of the PSI-MI 2.5 standard, and it allows querying data in certain columns.

MIQL is composed of:

- **Terms** – more words have to be surrounded by double quotes “
- **Fields** – in the most left column on Figure 23 are names of fields. If one wants to query specific field, then the syntax is: `field:”term”`
- **Term modifiers** – all wildcard searches, fuzzy searches, proximity searches, etc. For example: `field:”t??m”` will find both term and team. Many other modifiers are also available, such as asterisk ‘*’ with common meaning,
  inclusive search: `mod_date:[20120101 TO 20130101]`,

  exclusive search: `title:{Alza TO Canteen}`, etc.
- **Operands**: AND, OR, NOT, +(must contain word after this character), -(must not contain word after this character, synonym of NOT)
- **Grouping and field grouping** – searching for field with two conditions: `field:(+”must contain” –”must not contain”)`

Field Name	Searches on	MITAB 2.5 Columns	Example
idA	Identifier A	1, 2	idA:P74565
idB	Identifier B	3, 4	idB:P74565
id	Identifiers (A or B)	1..4	id:P74565
alias	Aliases (A or B)	5, 6	alias:(KHDRBS1 OR HCK)
identifier	Identifiers (A or B) or Aliases (A or B)	1..6	identifier:P74565
pubauth	Publication 1st author(s)	8	pubauth:scott
pubid	Publication Identifier(s)	9	pubid:(10837477 OR 12029088)
taxidA	Tax ID interactor A: be it the tax ID or the species name	10	taxidA:mouse
taxidB	Tax ID interactor B: be it the tax ID or species name	11	taxidB:9606
species	Species. Tax ID A and Tax ID B	10, 11	species:human
type	Interaction type(s)	12	type:"physical association"
detmethod	Interaction Detection method(s)	7	detmethod:"two hybrid*"
interaction_id	Interaction identifier(s)	14	interaction_id:EBI-761050

Figure 23 Standard fields utilizable in PSICQUIC searches [34]

5.6 Summary

PSICQUIC is web service, which can be accessed utilizing SOAP protocol or via REST-compliant interface. Because in the previous chapter in the comparison of SOAP and REST approaches the latter appeared more beneficial, the PSICQUIC was studied with regard to its REST-compliant PSICQUIC web service.

The path to resources in PSICQUIC consists of four basic parts. It is the base URL which specifies service that is going to be queried. The second part is specifying method. One can choose from three possibilities. There can be entered interactor id, interaction id or query. The last part of the URL is consisted of available parameters

If one wants write his own queries, then he has to utilize MIQL flexible query. MIQL specifies how the queries should be built. IMIQL has sufficient means for querying protein databases. Its specification describes which columns in MITAB 2.5 format are queried by which statement.

6 Application for protein interactions searching

6.1 Introduction

The number of available online interaction databases increases rapidly. Every year are created new databases, which could be filled in by commercial data, which is not accessible for everybody, or by data which is gathered from all the other databases and then changed into current form.

Because there is already many database databases with aggregation functions, the new application should be in several ways different than already available solutions. During the conducted research was found information about IMEx consortium, which connects together many various databases. Few of them are filled by primary data gained from experiments, and also databases with derived data are part of IMEx consortium. This chapter will suggest how application utilizing IMEx consortium member's sources could look like.

6.2 Simplicity

Already was mentioned that there is already many available online databases with many functions. If the application should be accepted, then it has to be simple. If it was difficult to operate on the web site, then it would not get many new users after it was released. The new service needs to be simple so it will earn new users who will find advantage in the simplicity.

Users usually do not trust new services and do not want to go through process of registration on every page. It is usually time consuming operation and furthermore it usually requires creating new user name and password. So the application should be simply enough, but powerful as can be even without the necessity for user to log in.

6.3 Novel interface

Along with simplicity goes also unprecedented interface. When user comes to a web site, then he is drove away because he does not like the web he visited, or he can be attracted by its appearance and then the site has few more minutes to convince the user to stay a little longer. The competition is big and therefore the first impression of the user should be that the page he has visited is not only simple and capable, but also interesting.

If user starts to use one application just because he likes it and he is curious what it can do, then he may realise that the application is exactly what he ever needed.

6.4 Customizable interface

It can be expected that there will be vast amount of results found in databases as a result of written queries. The simple interface should be able to handle displaying of these results in a way that the outcome will be easy to read. That means that there user will be able to change appearance of the outcome so it will be possible to change the layout if necessary.

6.5 Intelligence

The application will not have the advantage of remembering current user name. However, it should facilitate the user's job. Thus, the interface should provide simple access to all available functions even without user's headache.

6.6 Maintenance

The application should not require any administration, because its source of data will be PSICQUIC web service, and therefore the application will update its sources itself. Also was mentioned earlier that application for protein interaction searching does not require user registration. Thus, in this application will not be anything to manage or to maintain and therefore maintenance should be needed only in cases when PSICQUIC service will change current communication protocol.

6.7 Summary

The new application that will be implemented will be one of many applications that access online databases. Thus, it should be distinguishable from the other applications. In this section were listed main features that the new application should possess.

7 Components of the application

7.1 Introduction

The new application is required to be different than other application with similar purpose. Therefore, the most distinguishable part of the application is its interface. With regards to the new generation of machines which can access Web, new requirements arose on the available applications. The applications should be able to view also a user from his phone, table or other mobile device.

Designer of application cannot know from which kind of device will come his users more often or the way how they will use the implemented system. Creators of Windows 8 invented new interface Metro from which can user switch back to common interface. The implemented application will take similar step and will a provide user with two main interfaces.

7.2 Simple interface

The first interface that this application should provide is simple interface. This should be the main interface. When user comes starts the application, this is the first site he will see.

The simplicity, however, will not be redeemed by lack of functions. The simple interface will be colourful, simple as possible, but it will allow user to enter an arbitrary MIQL query. If user doubts and will not be aware of MIQL syntax, then it should not be hard to get to specification of MIQL with several examples.

With the query input field should be on the simple interface page also list of available databases. This list will get updated in reasonable amounts of time. It will not be updated with every single page refresh in web browser. So the simple scenario, when this simple interface can be used is scenario like this one:

1. User knows syntax of MIQL query language and he knows exactly what he is looking for.
2. User writes a query into the query windows, selects database he wants to search in and then he hits "Search for interactions" button

Example of such a MIQL query is following

```
brca2 AND species:human –mouse
```

7.3 Easy interface

The second interface after simple interface is easy interface. Even though there are more fields to be filled in with necessary data, from the fields is then built final MIQL query. The invisible logical operator between visible fields in this interface will be AND, and thus if user will fill into a species field '`human`' and into publication author field '`scott`', then the resulting query will be

```
species:human AND pubath:scott
```

Into the fields in an easy interface will be possible also write more difficult statements using well known operators such as AND, OR, etc. If user who wants to search for protein that does not belong

to neither a human nor mouse and was published by a man called Scott can put into species field ‘NOT(human OR mouse)’ and into the publication author field ‘scott’ and the resulting query will be

```
Species: NOT(human OR mouse) AND pubauth:scott
```

7.4 Results page

The web service is capable of sending to user data in PSI-MI XML format or in PSI-MI TAB 2.5. One of these results will be processed and made more readable for the purposes of researchers. Furthermore there should be available statistics about the data from PSICQUIC web service. For instance amount of retrieved records from the PSICQUIC web service or number of displayed records.

The results will be displayed in a customizable table. In this table it will be possible to hide unnecessary columns or to change colours of the table colours. Also, there will be option to get link on the displayed results and that will be how the user can store his findings in case he will need them in future (and this web will be no longer operational) or send them to someone else.

On the Results page will be also displayed the query in case that user utilized easy interface for entering his query conditions. This will lead to better understanding of MIQL queries and thus simple displaying the query that was generated can teach user how to create his queries without usage the easy interface. When user wants to query database more often, it can be easier to utilize simple interface, because he has the control over the written queries and the queries can be constructed in a manner that easy interface will never be able to achieve.

7.5 Help

The application should be so simple that user will be able to use it even without reading long articles. However, there will be available help in the places, where could user get lost with suggestions how to use this application or how to proceed when there are too many results.

In the help can be also accessible useful information that does not have to be related to the application, but can explain to user where the information comes from and where to search for new information that can be utilized in this application. However, if the application is simple enough, then the unnecessary omnipresent context help starts to be annoying and only distracts the impression of the application.

7.6 Summary

The application that will be implemented in order to retrieve data from interaction databases will consist of three main components. The purpose of the individual components was described in this chapter. In order to conserve the on the first look nice and simple interface, but to offer user to utilize more of the application functionality, there will be also available easy interface. Easy interface will basically offer similar functionality as the simple interface, but it will be more user-friendly and it will teach user how to write his queries.

The simple interface does not have so many input fields as the easy one, but entering valid MIQL queries in this easy interface will be easier and more fun for user, who will be able to learn from the generated queries in order to switch to simple interface and write his queries for higher efficiency.

8 Implementation

8.1 Introduction

In previous chapters were introduced possible ways how to access online databases and these methods were afterwards studied in order to get the knowledge that is necessary for implementing new protein interaction finder.

Implementation of this application for getting protein interactions is beneficial for better understanding of the problems related with retrieving protein interaction data.

In this chapter will be discussed the choice of Ruby on Rails framework for implementation of the new application. Its helpful built-in features will be pointed out and compared with other languages, which could be used for similar purposes. The process of implementing application that is capable of retrieving data from protein interaction databases will be described and the outcome will be compared with already available online solutions.

8.2 Choice of programming langauge

So far, I have not had experienec with writing information systems in Ruby on Rails. However, I have implemented information system in Yii PHP framework, which takes inspiration from Ruby on Rails and therefore I decided to study this new framework, which is built upon Ruby language. Furthermore, while studying PSICQUIC web service, I have noticed that bioinformatics uses many languages, but the most popular are: Java, Perl and Python (Figure 24). Though those listed languages are multi-platform programming languages, I think that Ruby is also very capable language and should be included in the supported languages. It is easy to use language providing many advantages. One of them is that the scripts written in Ruby can be easily integrated into application written in Ruby on Rails. Ruby on Rails can be installed on each of the main platforms nowadays. And the installation of Ruby on Rails is not a problem. For instance very simple Ruby installer exists for users of Windows. It will setup Rails on user's computer with only few clicks.

Figure 24 In PSICQUIC menu are available materials in three most popular bioinformatics programming languages [34]

8.3 Querying PSICQUIC

After decision to utilize PSICQUIC REST-compliant Web service, I started to look whether Ruby offers means how to access REST-compliant Web services. For the needs of *Protein interactions finder* will be serving Ruby's library called Net::HTTP. Example of Net::HTTP usage is shown on Figure 25. From example code one can also see that reading of Ruby language is not difficult in comparison with other languages.

```
uri = URI(UrlToRegistryList)
params = {
      :action =>      'ACTIVE',
      :format =>      'xml',
      :restricted => 'n',
      :tags =>        'protein-protein'
}
uri.query = URI.encode_www_form(params)
response = Net::HTTP.get_response(uri)
all_active_services_in_xml = response.body
```

Figure 25 Example of retrieving list of Active services in XML format

8.4 Working with simple initerface

When user arrives at the applicatoin's site for the first time, he will be redirected to the simple interface. The simple interface is displayed on Figure 26. Because the intention was to create user friendly and simple interface, the page contains only one link, one drop down list, one input field and one button which sends the request to the results page.

It has plentiful of colours and the layout was chosen so it was clear what the purpose of each of the elements on the page is. The layout is basically so easy that any context help would be useless.

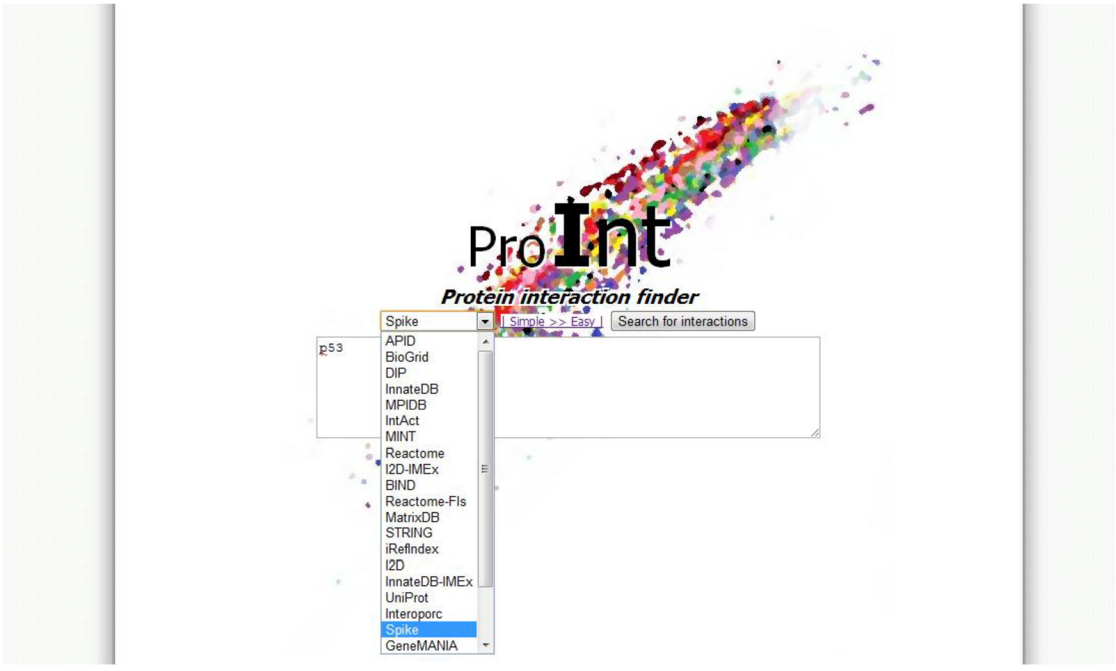


Figure 26 Simple interface

Nevertheless, even on the simple layout user can find link called ‚Simple >> Easy, which swithces from simple to easy interface.

8.5 Work with easy interface

The easy interface has more fields that the simple interface. But it is not because it would send the final query to the different search engine. The easy interface has simply built-in query generator, which from the text entered into the fields creates the same query as could write user by hand. User can see the resulting query on results page.

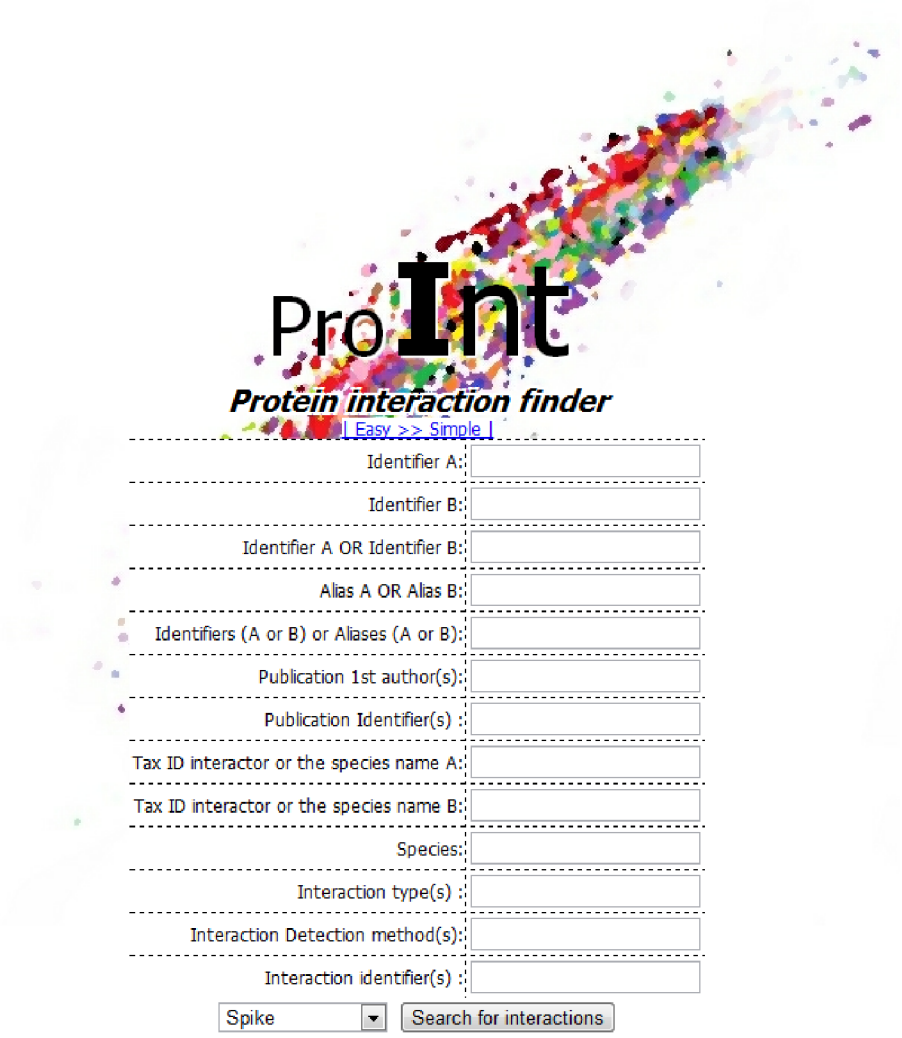


Figure 27 Easy interface

8.6 Work with results page

The reults page was implemented also in two versions. One version is basically big table, and the other one is supposed to be shelf for a user. Above the results link can be found that switches the application layout from table layout into the field layout.

8.6.1 Table layout

On the results page user can see list of records that matched the selected criteria. On Figure 28 can be seen that the results page with table layout is in default state divided into three parts. The top one is consisted of statistics. One can see that application tells user how many records have been found, which MITAB version is found on the server.

On PSICQUIC pages is stated that via this service should be available three basic types of MITAB:

- MITAB 2.5
- MITAB 2.6
- MITAB 2.7

However, some of the databases do not support the correct MITAB formats and therefore the implemented application truncates the invalid format to MITAB 2.5. The only reason between all these formats is simply number of items. With every new version of MITAB new columns are added into the specification. If the application is implemented with MITAB 2.7 support, then it will not know names of all the new columns from newer version. The solution for this could be for example storing names of the columns in a file (thus the database would not be needed) and user could simply edit the stored file with new column names and if the application found new formats, then it could sent an email to administrator with information that new MITAB format was released and that the names of columns need to be added.

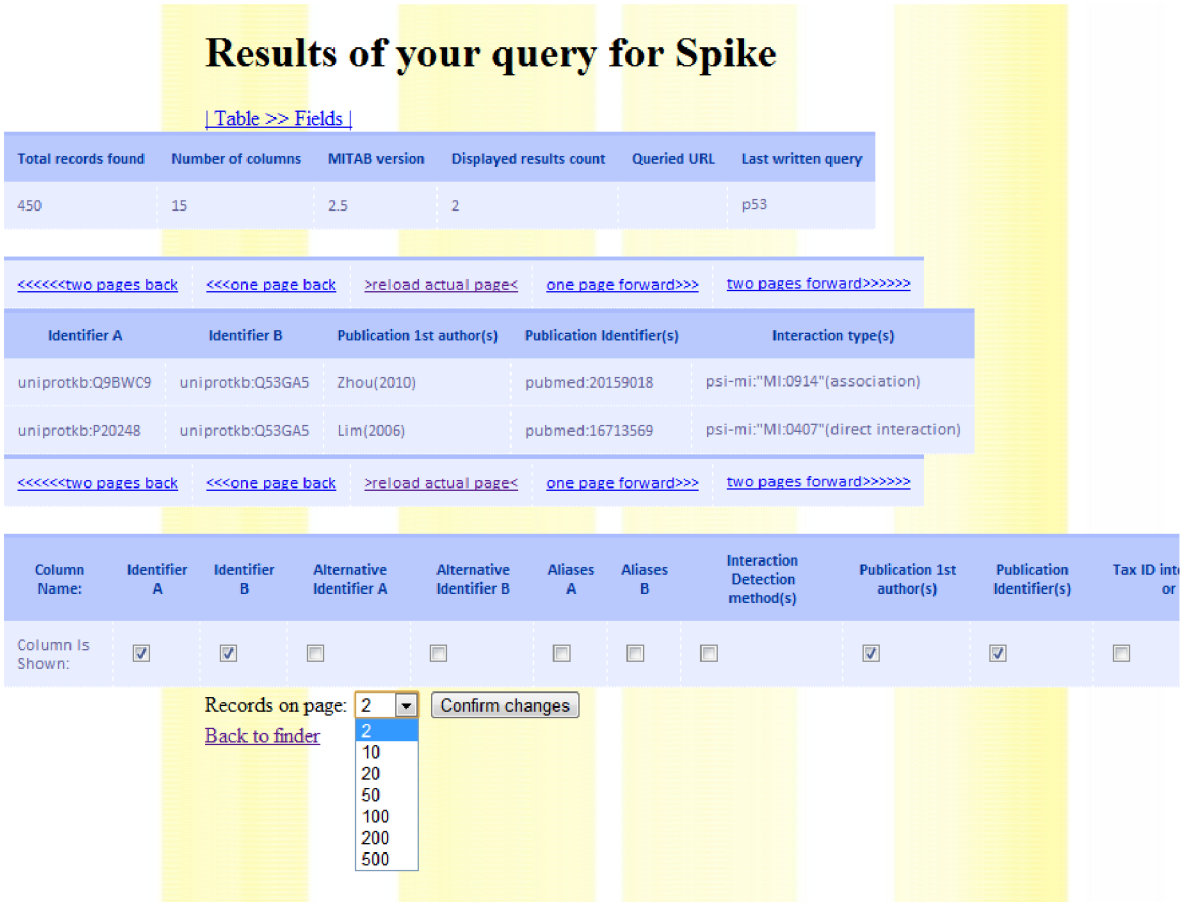


Figure 28 Table layout of the implemented application (only two records displayed)

After the first iteration I found out that when there are thirty-six fields in MITAB 2.6 and forty-two fields in MITAB 2.7, then the screen starts to be smaller. I suppose that not all of the fields such as 'publication identifiers' are always important. Therefore, the solution is available at the bottom part of Figure 28 and Figure 29. There is list of all columns and user can arbitrary turn off and then turn back on each of the columns.

8.6.2 Field layout

The main difference between table and field style is that the latter one is adjusting to the user's window. The secondary advantage of field layout is that when user has screen wide enough, then it will be very similar to the table layout. However, the field layout has more colours. The colours are randomly chosen for every new field in order to make the interface more interesting. If the screen is not wide enough, then the line with fields will break and user will be able to see all the fields without horizontal scrolling.

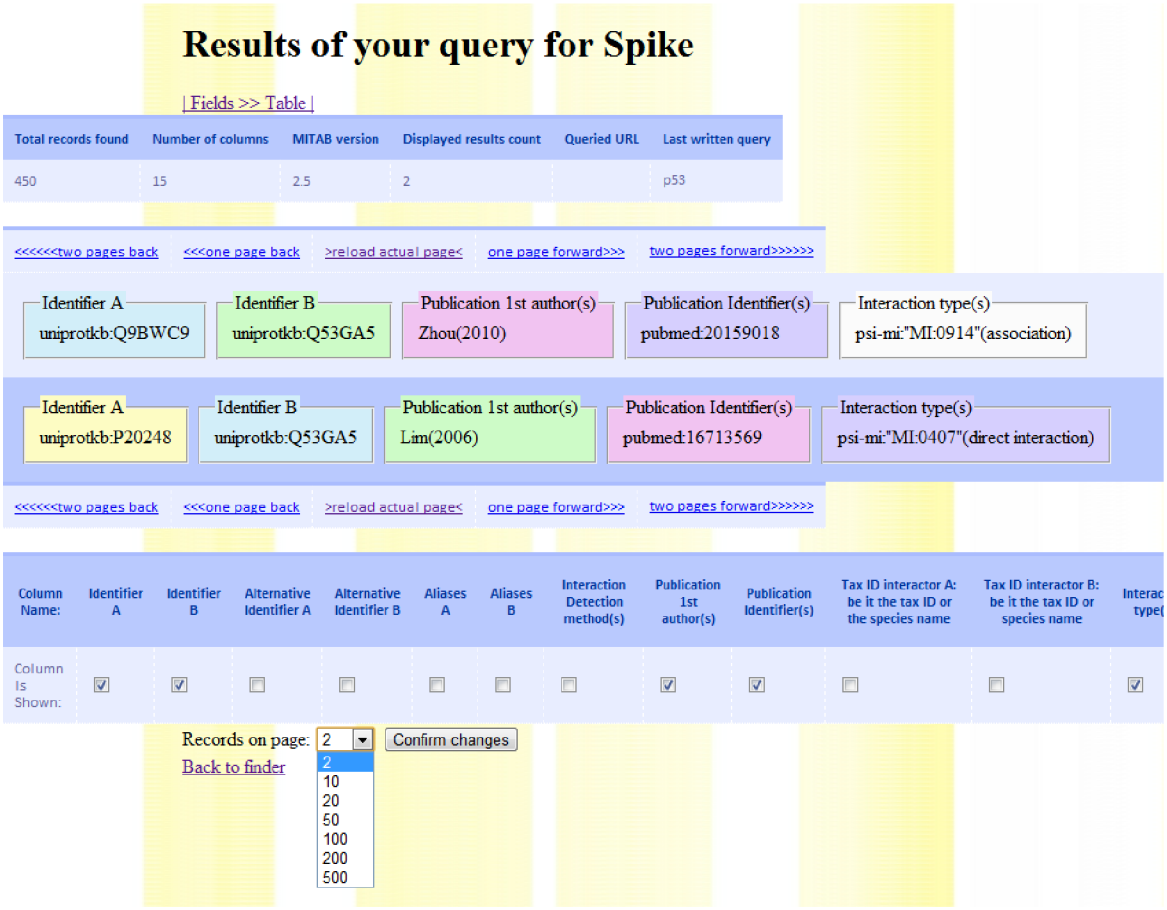


Figure 29 Implemented application in field format (only two records displayed)

The field layout is more flexible, because it there can be modified number of fields and furthermore it can adjust to the page. The design with yellow stripes in the behind pattern was not chosen accidentally. The application if given enough space gets wider. The ultimate background for such application is hard to create, so I went in opposite direction and instead of dimensioning the background to the all possible widths, I designed simple yellow-orange vertical stripes and now the application should look like it is hovering about the background stripes.

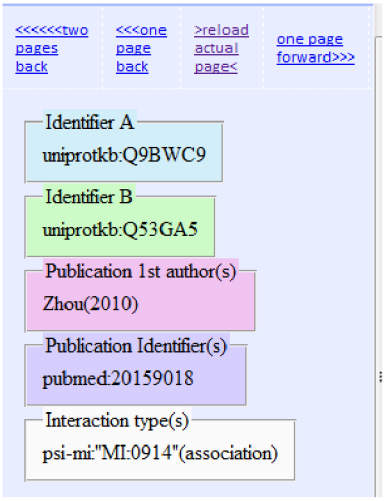


Figure 30 Example how storable is the field layout of implemented application. If normal table layout was used, user would see only two fields.

8.6.3 Intelligence

User cannot login into the application and therefore the application extensively utilizes sessions. In sessions is stored information about the last query, about last queried database, chosen number of displayed fields per page and even which one of the interfaces user used the last time. The application tries to facilitate user's job as much as possible. Also the PSICQUIC registry are not queried with every web browser access, but there is a time-limit during which the application uses only data stored in current session.

8.7 Introducing other applications for protein interaction searching

8.7.1 STRING 9.0

This application available at string-db.org was built upon data of STRING database. On Web pages of this application is stated that it can be used for many purposes, because it offers to user extensive user interface with possibility to enter detailed demands. It has well-arranged layout with nice whisperer that will appear when one wants to enter organism name. This is a useful feature, which because of performance reasons could be implemented only if application had direct access to database with properly created indices.

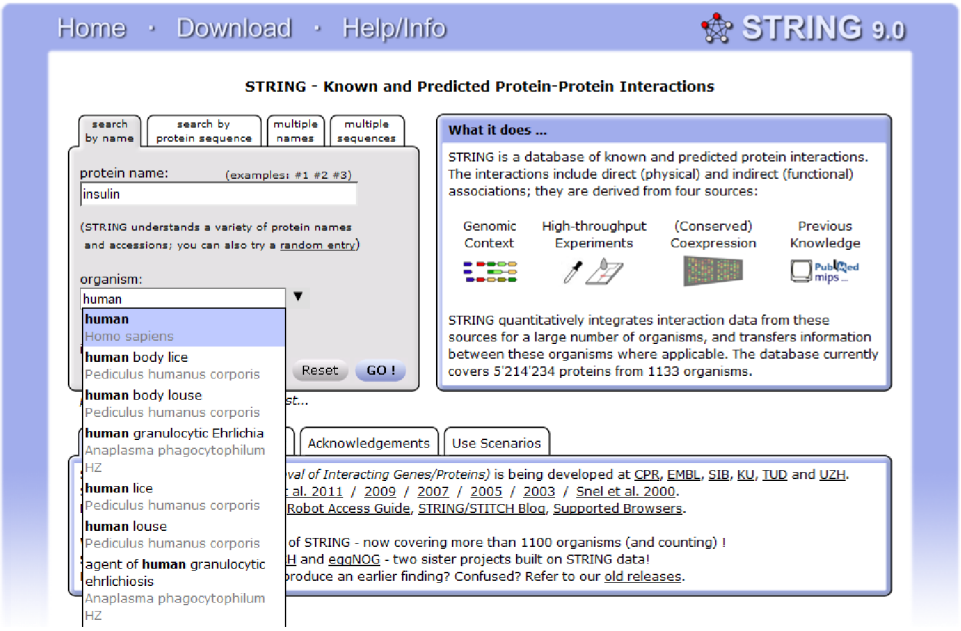


Figure 31 Application for searching for protein interactions in STRING database on string-db.org

Application at string-db.org displays all predicted functional partners (Figure 32), however this feature is connected with the origin of STRING database – it contains only predicted partners for the searched protein. The implemented application only searches in all available databases, where can be mixed curated records with incorrect records. Even only suggesting possible functional partner from all the available databases would be highly time-consuming.

Your Input:

		Neighborhood	Gene Fusion	Cooccurrence	Coexpression	Experiments	Databases	Textmining	[Homology]	Score
TP53	tumor protein p53; Acts as a tumor suppressor in many tumor types; induces growth arrest or apoptosis depending on the physiological circumstances and cell type. Involved in cell cycle regulation as a trans-activator that acts to negatively regulate cell division by controlling a set of genes required for this process. One of the activated genes is an inhibitor of cyclin-dependent kinases. Apoptosis induction seems to be mediated either by stimulation of BAX and FAS antigen expression, or by repression of Bcl-2 expression. Implicated in Notch signaling cross-over (393 aa) *(Homo sapiens)*									

Predicted Functional Partners:

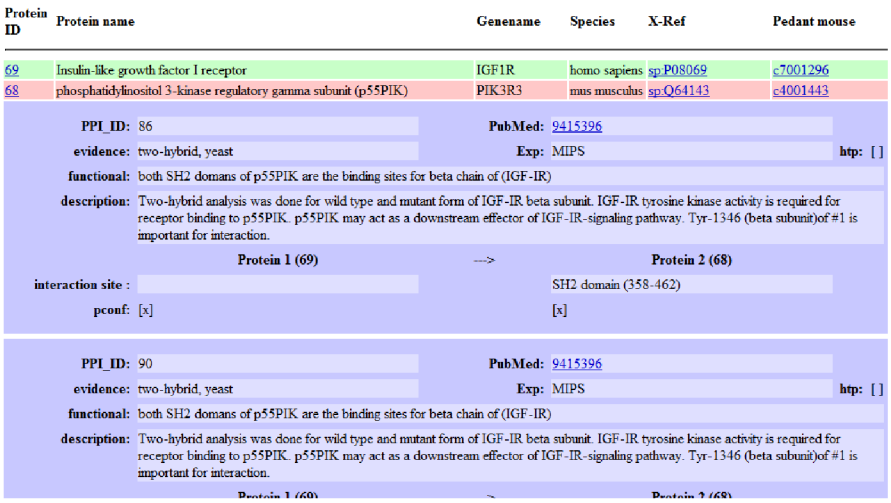
MDM2	Mdm2 p53 binding protein homolog (mouse); Inhibits TP53/p53- and TP73/p73-mediated cell cycle a [...] (497 aa)					•	•	•		0.999
EP300	microRNA 1281; Functions as histone acetyltransferase and regulates transcription via chromatin [...] (2414 aa)					•	•	•		0.999
ATM	ataxia telangiectasia mutated; Serine/threonine protein kinase which activates checkpoint signa [...] (3056 aa)					•	•	•		0.999
MDM4	Mdm4 p53 binding protein homolog (mouse); Inhibits p53- and p73-mediated cell cycle arrest and [...] (490 aa)					•	•	•		0.999
SP1	Sp1 transcription factor; Transcription factor that can activate or repress transcription in re [...] (785 aa)					•	•	•		0.999
USP7	ubiquitin specific peptidase 7 (herpes virus-associated); Cleaves ubiquitin fusion protein subs [...] (1102 aa)					•	•	•		0.999
BRCA1	breast cancer 1, early onset; The BRCA1-BARD1 heterodimer coordinates a diverse range of cellul [...] (1863 aa)					•	•	•		0.999
CDKN1A	cyclin-dependent kinase inhibitor 1A (p21, Cip1); May be the important intermediate by which p5 [...] (164 aa)					•	•	•		0.999
HIF1A	hypoxia inducible factor 1, alpha subunit (basic helix-loop-helix transcription factor); Functi [...] (826 aa)					•	•	•		0.999
TP53BP1	tumor protein p53 binding protein 1; May have a role in checkpoint signaling during mitosis (By [...] (1977 aa)					•		•		0.999

Figure 32 List of predicted partners on string-db.org

8.7.2 PPIoogle

Both the name PPIoogle and the main page were obviously derived from google.com (Figure 33). Even the implementation of this application is very similar to google. Available is only one input field, two radio buttons and search button. The long radio button makes the resulting list more extensive (Figure 34), if the radio button was switched back to short, then in the result would be displayed only names of interacting partners.

Figure 33 PPioogle main page in comparison with google.com



Protein ID	Protein name	Genename	Species	X-Ref	Pedant mouse
69	Insulin-like growth factor I receptor	IGF1R	homo sapiens	sp:P08069	c7001296
68	phosphatidylinositol 3-kinase regulatory gamma subunit (p55PIK)	PIK3R3	mus musculus	sp:Q64143	c4001443

PPI_ID:	86	PubMed:	9415396	
evidence:	two-hybrid, yeast	Exp:	MIPS	htp: [ ]
functional:	both SH2 domans of p55PIK are the binding sites for beta chain of (IGF-IR)			
description:	Two-hybrid analysis was done for wild type and mutant form of IGF-IR beta subunit. IGF-IR tyrosine kinase activity is required for receptor binding to p55PIK. p55PIK may act as a downstream effector of IGF-IR-signaling pathway. Tyr-1346 (beta subunit)of #1 is important for interaction.			
	Protein 1 (69)	--->	Protein 2 (68)	
interaction site :			SH2 domain (358-462)	
pconf:	[x]		[x]	

PPI_ID:	90	PubMed:	9415396	
evidence:	two-hybrid, yeast	Exp:	MIPS	htp: [ ]
functional:	both SH2 domans of p55PIK are the binding sites for beta chain of (IGF-IR)			
description:	Two-hybrid analysis was done for wild type and mutant form of IGF-IR beta subunit. IGF-IR tyrosine kinase activity is required for receptor binding to p55PIK. p55PIK may act as a downstream effector of IGF-IR-signaling pathway. Tyr-1346 (beta subunit)of #1 is important for interaction.			

Figure 34 PIIoogle searching results

8.7.3 MPPI search

MPPI search is after PPIoogle another interface how to search for interactions in MIPS database. The PPIoogle is similarly simple as the simple interface and MPPI search is more detailed as the easy interface of implemented application. MIPS database is not part of IMEx consortium, and that is why it is not included in the list of available databases in the implemented application.

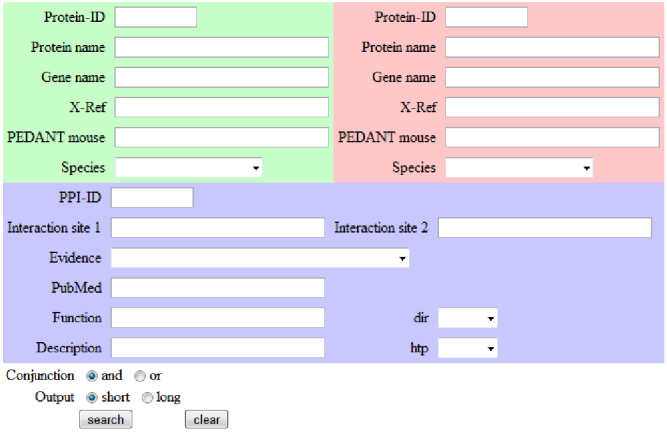


Figure 35 MPPI search

Nevertheless, the MPPI search engine works similarly as the PSICQUIC, and therefore it would be possible to include the results of the MPPI search into the results of the implemented application. The difficulty during the conversion of MPPI search results would be different data formats or different curation techniques.

8.7.4 BioXGEM

BioXGEM is representative of extensive searching engine. Its search has many settings and the result offers many statistics that can be utilized by scientists with very specific needs. However, there is utilized a good idea that user is allowed to choose from one of the five the most common species in databases, or he can search all species, or he can write his own species.

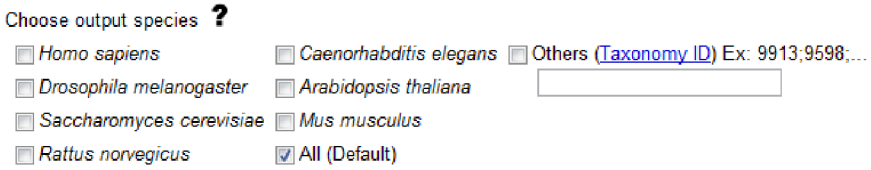


Figure 36 While setting up the search, it is possible to choose only from seven species with the highest number of records in database available

Next pleasant detail is question mark next to every option, which offers context help.

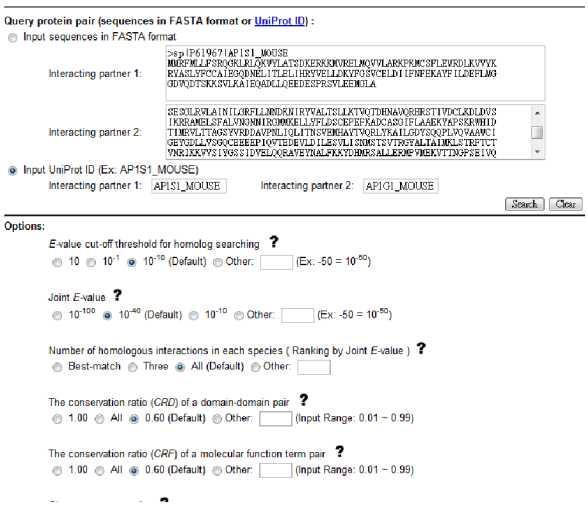


Figure 37 Many settings on BioXGEM.Protein-Protein InteractionSearch

8.8 Possible improvements

From the beginning of the project it was decided that user will not be allowed to log into this application. From this decision were derived all other steps. The whole environment of such an application could be customizable and the system could remember the customized environment for the registered user and change it for him every time he would log into the system. Signed user could also select his most favourite databases, change order of offered databases and also choose databases whose results would be displayed first. Also, the interface could be more connected to other services. Links can lead from exact field to exact service. For example, when publication identifier is '209590018', then it after clicking on the hyper link could be user redirected to the web page with the publication.

The implemented application retrieves data only from IMEx consortium partners. Therefore there are some limitations. If one wanted to add new database into the implemented application, then he would need to basically write the whole application from scratch, because none of application's parts would be utilizable in environment of databases that are not IMEx members. Also, it is not possible that the searcher could offer whisperer. There are no means how to achieve this with PSICQUIC web service so far. However, the application could be the beginning for a project of generating protein interaction networks. The main tools of bioinformatics are data assimilation, data storage and data visualization.

The application for building mentioned networks could be interesting, although one can expect that also very challenging.

I excluded from the final application some of the dynamic components, such as hiding unused components, because I found the functionality that is too much hidden from user as useless. That is why I rather implemented two easily switchable interfaces. It is easy to click on one link rather than search for something hidden in floating menu.

8.9 Summary

Implemented application was introduced. Its name is Protein interaction finder and utilizing PSICQUIC web service served as an illustration of operation of the REST-compliant web service. Final application was implemented in Web framework Ruby on Rails, which was chosen for the implementation because author has experience with Yii PHP Framework, which is similar to the Ruby on Rails. Furthermore, the Rails framework is built upon Ruby language in which it is easy to program and even read the written programs. Ruby is capable language and it is competition for language like Perl, Python or Java, which are very popular among Bioinformatists.

The introduced application was compared with other online solutions. Some of the services are even simpler (e.g. PPIoogle) than the implemented application. But it is not a problem, because on the other example BioXGEN can be shown that when is application too complicated, it can discourage users. Therefor the best way is to implement two interfaces. In the same way MIPS database took appropriate steps and also implemented two interfaces – already mentioned PPIoogle and MPPI search. PPIoogle is similar to the implemented simple interface and MPPI search is similar to the implemented easy interface. Nevertheless the MIPS application has advantage, because it is not IMEx member and it has direct access to the MIPS database and therefore the implemented solution could be better. In comparison with STRING the MIPS offers only limited possibilities and do not offer modern whisperer like STRING. On the other hand each of the databases was created probably for different sets of people and therefore, because STRING offers only predicted protein interactions and therefore also the output is a little bit clearer.

At the end of this chapter were discussed possible improvements. They could be made in user interface or in the details such as the STRING's whisperer or three colours based user interface like MPPI. These changes would be only minor, and therefore the suggested continuation of this project would be developing it into something more – protein interaction network generator.

9 Conclusion

The main goal of this thesis was to get reader acquainted with possible ways of retrieving data from protein interaction databases. To put reader into context, I decided to introduce to him problematics of bioinformatics, reasons of its existence and why it is important to develop new tools that will facilitate biologists' work. In this thesis was explained the position of protein interactions in the biology and also was emphasized that protein interactions are integral part of interaction networks, from which can be derived new hypothesis for testing; thus bioinformatics speeds up discovery cycles in modern biology.

Modern high-throughput methods generate vast amount of data that needs to be processed and stored id databases and it is important that the data stored in them will be accessible for all interested scientists. Because of the amount of recently accessible data there is a space for many new databases to be created. However, this thesis pointed out that until 2004 (when PSI-MI XML format was released) there were no standards how to exchange data between databases and each of available databases curated own data and that was why databases contained the redundant data. Furthermore, because of lack of any standards, there was no guarantee which data was correct if one found two records in two different databases with different conclusion from the same experiment. The process of data curation needed to be unified. And because PSI-MI XML from 2004 was widely accepted by community, the same databases which worked on the format decided to join their forces in IMEx consortium.

IMEx is consortium was created in order to share the effort of curating literature, facilitate data exchange and create standards that would make the access to the data simpler for all scientists. Its members agreed on strategy of most important publications curation and also released curation manual and invented procedures to keep data quality at the same level across all IMEx members. IMEX consortium members have in plan to cover more publications and curate new available data faster and maybe take a part in curation of the data before it is published. Number of records in IMEx members' databases grows rapidly and one can assume that the cooperation between online databases in data curation will increase data quality and in close future will bring more standards and improved web services, which will be available to every scientist.

IMEx members agreed on improved method of sharing data between them. The new way how to share data started to be called PSICQUIC web service. Because the PSICQUIC can be accessed with two different approaches (SOAP and REST), this thesis introduces both these terms and describes their properties and contributions. Generally, the main purpose of Web services is facilitating communication between various systems over a network. Individual systems can offer to other systems own functions and the access to the offered functions is not restricted by any platform or programming language. From the conducted comparison of these two categories was concluded that REST architectural style is more general and its development is more understandable and therefore more popular than SOAP protocol. SOAP was created as Simple Object Access Protocol, however because rapid changes it has gone through its name is no longer abbreviation for Simple Object Access Protocol. SOAP was transformed into protocol framework and the word SOAP is now meaningless. Furthermore REST can fully utilize HTTP protocol features whereas SOAP abuses HTTP protocol and in current state do not fully respect Web architecture.

PSICQUIC is REST-compliant web service also accessible via SOAP protocol. However, because of the conducted research I decided to study the PSICQUIC web service with regards to REST. The motivation for creation of PSICQUIC is that when user wants to query more than one of twenty-five databases available via PSICQUIC, it is no longer necessary to write more than one query. All available databases are listed in PSICQUIC registry and tags are assigned to each of them with information about data they contain.

After the studying the necessary basis the application for protein interaction data retrieving was implemented. First, the features were introduced in order to prepare reader and show him the direction in which the implementation will be led.

Then, there were introduced less general facts about future application. Essentially, block foundation of future applications was built while its basic components were listed and described. The main application consists of few main parts. To make the application more accessible, two interfaces were implemented for every screen, and from the comparison with other online application emerged that it is not so original idea and really similar idea already had tem of MIPS database, which implemented PPIoogle and MPPI, which are alternatives to the implemented *simple* and *easy* interface.

This thesis covers the basics how to deal with PSICQUIC on the level of retrieving protein interaction data. It would be possible to broaden this thesis with utilizing the data from PSICQUIC for various bioinformatics projects, such as creating interaction networks and deriving new hypothesis from them or to continuously improve user interface of implemented application in one or more of the suggested ways in the last chapter.

References

[1] Zydowsky, T., M. *Frederick Sanger*. [Internet] Available at: http://www.chemistryexplained.com/Ru-Sp/Sanger-Frederick.html [Accessed on 15th January 2012]

[2] Baxevanis, A., D., Ouellette, B., F., F. *Bioinformatics: a practical guide to the analysis of genes and proteins*. Third edition. Hoboken (New Jersey): John Wiley & Sons, Inc. 2005. 540s. ISBN: 0-471-47878-4.

[3] Lesk, A., M. *Introduction to bioinformatics*. Second edition. New York: Oxford University Press. 2005. 360s. ISBN-13: 978-0-19-927787-2, ISBN-10: 0-19-927787-7.

[4] Zvelebil, M., Baum, J., O. *Understanding bioinformatics*. New York: Garland Science. 2008. 772p. ISBN-13: 978-0-8153-4024-9. ISBN-10: 0-8153-4024-9.

[5] Cohen, J. *Bioinformatics – An introduction for computer Scientists*. ACM Computing Surveys, Vol. 36, No.2, June 2004.

[6] Wikipedia. *List of Intel microprocessors* [Internet] Available at: en.wikipedia.org [Accessed on December 4th 2011].

[7] Wikipedia. *TOP500*. [Internet] Available at: en.wikipedia.org [Accessed on December 4th 2011].

[8] J. Zendulka, I. Rudolfová. *Studijní opora k předmětu Databázové systémy*. 2006.

[9] G. Fowler. *Cql - A flat file database query language*. [Internet] Available at: http://www2.research.att.com/~gsf/publications/cql-1994.pdf [Accessed on January 15th 2012]

[10] *Data Integration Glossary*. U.S.Department of Transportation. 2001. [Internet] Available at: http://knowledge.fhwa.dot.gov/tam/aashto.nsf/All+Documents/4825476B2B5C687285256B1F00544258/$FILE/DIGloss.pdf [Accessed on January 15th 2012]

[11] *Tools for Analysis of Data Sets*. Available at: http://www.geneontology.org/GO.tools.microarray.shtml [Accessed on January 15th 2012]

[12] *Top 100 Journals in Biology and Medicine*. Special Libraries Association. 2009 [Internet] Available at: http://units.sla.org/division/dbio/publications/resources/dbio100.html [Accessed on January 15th 2012]

[13] *2012 NAR Database Summary Paper Category List*. Oxford Journals. 2011. Available at: http://www.oxfordjournals.org/nar/database/c/ [Accessed on January 15th 2012]

[14] Searls, D., B. *Grand challenges in computational Biology*. Computational Methods in Molecular Biology. Netherlands. 1998.

[15] Bonetta, L. *Protein-protein interactions: Interactome under construction*. Nature, Vol. 468, p. 851-854. December 2010.

[16] Vankatesan, K. *An empirical framework for binary interactome mapping*. Nature Methods, Vol. 6, 83-90, 2009.

[17] *BioGRID Database Statistics*. Available at: http://wiki.thebiogrid.org/doku.php/statistics [Accesed on April 19th 2012]

[18] Fields, S., Song, O. *A novel genetic system to detect protein-protein interactions*. Nature, Vol. 340, p. 245-246, 1989.

[19] Paz, A.et al. *SPIKE: a database of highly curated human signalling pathways*. [Internet] Available online at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3014840/?tool=pmcentrez [Accessed on April 26th 2012]

[20] Razick, S., Magklaras, G. and Donaldson, I., M. *iRefIndex: A consolidated protein interaction database with provenance*. Available online at: http://www.biomedcentral.com/1471-2105/9/405

[21] *IntAct Home page*. [Internet] Available at: http://www.ebi.ac.uk/intact/ [Accessed on 22nd April]

[22] Kerrien, S. et al. *Broadening the horizon–level 2.5 of the HUPO-PSI format for molecular interactions*. BMC Biology 2007, 5:44.

[23] Orchard, S. et al. *Protein interaction data curation: the International Molecular Exchange (IMEx) consortium*. Nature Methods 9, p. 345-350, 2012.

[24] Orchard, S. et al. *The minimum information required for reporting a molecular interaction experiment (MIMIx).* Nat. Biotechnol. 25, p. 894–898, 2007.
[25] Orchard, S. et al. *Autumn 2005 Workshop of the Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI) Geneva*, 4–6 September 2005. Proteomics 6, p. 738–741, 2006.
[26] *Curation Rules*. IMEX [internet] Available at: http://www.imexconsortium.org/curation/ [Accessed on April 20th 2012]
[27] Perreau, V.M. et al. *A domain level interaction network of amyloid precursor protein and Abeta of Alzheimer's disease*. Proteomics 10, 2377–2395, 2010.
[28] Chen, Y.C., Rajagopala, S.V., Stellberger, T. & Uetz, P. *Exhaustive benchmarking of the yeast two-hybrid system*. Nat. Methods 7, 667–668, 2010.
[29] Montecchi-Palazzi, L. et al. *The PSI semantic validator: a framework to check MIAPE compliance of proteomics data*. Proteomics 9, 5112–5119, 2009.
[30] Aranda, B. et al. *PSICQUIC and PSISCORE: accessing and scoring molecular interactions.* Nat. Methods 8(7), 528–529, 2011.
[31] Shannon, P. et al. *Cytoscape: a software environment for integrated models of biomolecular interaction networks*. Genome Res. 2003 Nov; 13(11):2498-504.
[32] Saveanu, C. *Protein-protein Interactions*. [Internet] Available at: http://www.functionalgenomics.org.uk/sections/resources/protein-protein.htm [Accessed on 29th April 2012]
[33] *Biological Pathways*. National Human Genome Research Institute. [Internet] Available at: http://www.genome.gov/27530687 [Accessed on 28th April 2012]
[34] PSICQUIC. Available at: http://code.google.com/p/psicquic/ [Accessed on 30th April 2012]
[35] Fielding, R., T. Architectural Styles and the Design of Network-based Software Architectures. PhD dissertation, University of California, Irvine, 2000.
[36] Cabrera, L., F., Kurt, Ch. And Box, D. *An Introduction to the Web Services Architecture and Its Specifications*. [Internet] Available at: http://msdn.microsoft.com/en-us/library/ms996441.aspx#wsawhitepapermsdn040825_def_web_service [Accessed on 13th April 2012]
[37] Booth, D., et al. *Web Services Architecture*. W3C, 2004. [Internet] Available at: http://www.w3.org/TR/ws-arch [Accessed on 1st May 2012]
[38] Mander, L. *Describe REST Web services with WSDL 2.0*.[Internet] Available at: http://www.ibm.com/developerworks/webservices/library/ws-restwsdl/ [Accessed on May 1st 2012]
[39] J. Waldo, G. Wyant, A. Wollrath, and S. Kendall. *A note on distributed computing*. Technical Report SMLI TR-94-29, Sun Microsystems Laboratories, Inc., Nov. 1994.
[40] Box, D. *A Brief History of SOAP*. O'Reilly XML.com. 2001. [Internet] Available at: http://www.xml.com/pub/a/2001/04/04/soap.html. [Accessed on May 2nd 2012]
[41] *API Protocols*. Programmable Web. [Internet] Available at: http://www.programmableweb.com/apis [Accessed on May 3rd 2012]
[42] Prescod, P. *Roots of the REST/SOAP Debate*. [Internet] Available at: http://www.prescod.net/rest/rest_vs_soap_overview/#fromN1019 [Accessed on May 10th 2012]
[43] Flanders, J. *More on REST*. MSDN Magazine, July 2009. [Internet] http://msdn.microsoft.com/en-us/magazine/dd942839.aspx [Accessed on April 29th 2012]

Appendix A

2012 NAR Database Summary Paper – categories and subcategories

- Nucleotide Sequence Databases
  - International Nucleotide Sequence Database Collaboration
  - Coding and non-coding DNA
  - Gene structure, introns and exons, splice sites
  - Transcriptional regulator sites and transcription factors
- RNA sequence databases
- Protein sequence databases
  - General sequence databases
  - Protein properties
  - Protein localization and targeting
  - Protein sequence motifs and active sites
  - Protein domain databases; protein classification
  - Databases of individual protein families
- Structure Databases
  - Small molecules
  - Carbohydrates
  - Nucleic acid structure
  - Protein structure
- Genomics Databases (non-vertebrate)
  - Genome annotation terms, ontologies and nomenclature
  - Taxonomy and identification
  - General genomics databases
  - Viral genome databases
  - Prokaryotic genome databases
  - Unicellular eukaryotes genome databases
  - Fungal genome databases
  - Invertebrate genome databases
- Metabolic and Signaling Pathways
  - Enzymes and enzyme nomenclature
  - Metabolic pathways
  - Protein-protein interactions
  - Signalling pathways
- Human and other Vertebrate Genomes
  - Model organisms, comparative genomics
  - Human genome databases, maps and viewers
  - Human ORFs
- Human Genes and Diseases
  - General human genetics databases
  - General polymorphism databases
  - Cancer gene databases
  - Gene-, system- or disease-specific databases
- Microarray Data and other Gene Expression Databases
- Proteomics Resources
- Other Molecular Biology Databases
  - Drugs and drug design

    - Molecular probes and primers
- Organelle databases
    - Mitochondrial genes and proteins
- Plant databases
    - General plant databases
    - Arabidopsis thaliana
    - Rice
    - Other plants
- Immunological databases
- Cell biology

Appendix B

CD with the source code of implemented application and thesis text

Content of individual folders:

- `text` – text of the Diploma Thesis
- `protin` – sourcode and help to the implemented application

Appendix C

How to Install & Start the system on Windows 7

1. If you do not have installed Ruby on Rails on your system, then download from http://rubyonrails.org/download Ruby on Rails installer. It will install everything that will be necessary.
2. When the installation process finishes, start windows command prompt and go to the folder where you copied the Protein interaction finder and in that folder start this command:

   ```
   rails server
   ```
3. The server should have started. Start your web browser and type into address bar

   ```
   http://127.0.0.2:3000
   ```
4. You should be welcome by the welcome screen of the application