# A Method for Extracting Pathways from Scansite-Predicted Protein-Protein Interactions

**Tiberiu Simu**

**Master's dissertation**

**University of Skövde**

**19 June 2006**

# A Method for Extracting Pathways from Scansite-Predicted Protein-Protein Interactions

**Tiberiu Simu**

Submitted by Tiberiu Simu to the University of Skövde as dissertation towards the degree of Master by examination and dissertation in the School of Humanities and Informatics.

19 June 2006

I certify that all material in this thesis which is not my own work has been identified and that no material is included for which a degree has previously been conferred on me.

_____

Tiberiu Simu

# A Method for Extracting Pathways from Scansite-Predicted Protein-Protein Interactions

Tiberiu Simu[1]

[1] University of Skövde,
S-541 28 Skövde, Sweden
b04tibsi@student.his.se

**Abstract.** Protein interaction is an important mechanism for cellular functionality. Predicting protein interactions is available in many cases as computational methods in publicly available resources (for example Scansite). These predictions can be further combined with other information sources to generate hypothetical pathways. However, when using computational methods for building pathways, the process may become time consuming, as it requires multiple iterations and consolidating data from different sources. We have tested whether it is possible to generate graphs of protein-protein interaction by using only domain-motif interaction data and the degree to which it is possible to automate this process by developing a program that is able to aggregate, under user guidance, query results from different information sources. The data sources used are Scansite and SwissProt. Visualisation of the graphs is done with an external program freely available for academic purposes, Osprey. The graphs obtained by running the software show that although it is possible to combine publicly available data and theoretical protein-protein interaction predictions from Scansite, further efforts are needed to increase the biological plausibility of these collections of data. It is possible, however, to reduce the dimensionality of the obtained graphs by focusing the searches on a certain tissue of interest.

## 1 Introduction

In this work, an information fusion approach is applied to build aggregated knowledge about interacting proteins. The analysis of protein interactions is important to deepen our understanding about cell functioning, by revealing regulatory mechanisms, in which proteins take part. Furthermore, the study of interactions between proteins can help in understanding the functionality of unknown proteins, i.e. proteins that have not yet been annotated.

Integrative approaches in protein interaction studies have to deal with the large amounts of data that can be generated currently. Several techniques are presented and analysed and then a new method is proposed in the report.

The main aim of this work was to design and implement an algorithm capable to automate the construction of protein interaction maps using interaction data available from Scansite and additionally implementing methods of filtering out surplus information by enforcing tissue specificity constraints. Several objectives of this work have also been defined:

- literature survey of current approaches in building tools for aggregation or fusion of information;
- defining an information aggregation approach and scenario;
- deciding which technology of data aggregation to use with respect to the programming environment and with respect to the data acquisition and storage approach;
- implementation of the algorithm;
- assessment whether the proposed method of aggregating interaction data available from Scansite to construct protein interaction maps and the representation paradigm used are a feasible approach

The results obtained in this work show that it is possible to aggregate publicly available domain-motif interaction predictions to obtain protein interaction maps. It is possible to reduce the dimensionality of the data collection obtained by the aggregation process by retaining only the data concerning a specific tissue. However, describing protein-protein interactions using only the domain-motif interaction paradigm is not sufficient to obtain biologically plausible protein interaction maps. Several explanations of this finding are explored.

The rest of the report is structured in the following chapters:

- Background, which introduces the important concepts used in the report and in the study of protein interactions
- Methods, which introduces the chosen representation and algorithms used
- Design and Implementation, which covers the details related to the software construction and introduces parts of the program
- Results and discussion, which lists some of the outcomes of the project and reviews the limitations and the possible future developments
- References, the list of referred publications

# 2 Background

In this section, the main concepts used in data integration and protein interaction research are introduced. Relevant related work in these fields is also briefly reviewed.

## 2.1 Data Integration

Data generation is an important part of the bioinformatics focuses. As a general trend, the availability of biological data continues to grow and to become more difficult to handle and understand by simple manual inspection. Thus, integrative approaches

that are able to make the wealth of data more understandable and to direct further research are needed. As a consequence, integration of heterogeneous data in life sciences receives continuous interest. Integrative approaches constantly appear, both as efforts of integrating separate data sources, as well as essential steps of experimental techniques. Generally speaking, data integration strives to provide a better picture about the enclosed body of knowledge of today's data abundance, by offering new representation paradigms and easier access across domains.

Given the great diversity of data sources, representation paradigms, area of interest, storage modality and addressing, a complete classification of data integration methods is virtually impossible. However, general systematic divisions can be made, and a description of those follows in sections 2.1.1-2.1.3.

### 2.1.1 Type of data source

When data sources are of the same type, or concern the same area of research, data integration refers primarily to summing of the sources, and mainly focuses on format compatibility, dealing with inconsistencies and redundancies. One of the basic approaches in bioinformatics concerns building databases to support researches or to store a knowledge base.

When data sources are of diverse nature, concerning different areas of a biological phenomenon, the focus of data integration shifts to the systematic understanding and harmonisation of the data to describe whole systems in biology more generally or more completely.

In both cases, the usual result is a quality improvement of the processed data. Data can also originate from different experiments and be integrated into a unique resource, as, for example, Scansite (Obenauer et al., 2003), where several peptide library screenings and phage display experiments were used to derive the scoring algorithms.

### 2.1.2 Data storage and retrieval -- data acquisition

There are two main approaches concerning data storage and retrieval (Vdovjak and Houben, 2001): data warehousing and on-demand approaches. In data warehousing, the data is stored locally, typically in relational databases, which are regularly updated, following the updates in the primary data sources. This is also called the "eager" approach. In contrast, the on-demand approach is a lightweight solution that chooses to retrieve the information from the primary sources (typically databases available over the Internet) whenever that is needed. This is also called the "lazy" approach.

This separation is mainly made according to organisational constraints. Davidson et al. (1997) highlight that various integrative solutions respond to different needs. Warehousing approaches, on the one hand, are characterised by higher efficiency in operation (having the data stored locally and accessed in effective ways). Warehousing requires however more efforts and expenses to set up and design databases and to maintain equipment. It is thus targeted at bigger and more stable projects. A drawback would be that backtracking information generally becomes more difficult. On demand approaches, on the other hand, come with quicker availability, less time spent

in implementation, greater flexibility in operation and lower prices. Such approaches preserve the autonomy of the sources, and thus make backtracking easy and eliminate the constant concern of updating. A drawback is that they are slower in operation, as the data needs to be retrieved from sources at the moment of query. It is common practice for successful projects to move from one type of integration to the other, as their operation requirements change.

### 2.1.3 Transparency of the sources

Depending on how "visible" the data sources are to the user, the integrative approaches are classified as *integrated systems*, that offer unified access to heterogeneous sources, and *mediation architectures*, which allow automatic processing of complex queries.

In the integrative approach, the users are fully aware of which information sources they use. They can choose or exclude sources using a meta-search front-end. Some examples of services using the integrative approach and cross-linking between various resources are Swiss-Prot[1] (Bairoch et al., 2005, Boeckmann et al. 2005), National Center for Biotechnology Information[2], SRS, at European Bioinformatics Institute[3] (Etzold and Argos, 1993; Etzold et al., 1996), HUSAR Bioinformatics[4] (Ernst et al., 2003), MyGrid[5] (Wroe et al., 2003) etc.

Some of them, like HUSAR, go beyond simple merging of diverse data sources by introducing concepts such as executing chained steps over an integrative platform (Devignes and Smaïl, 2004).

In the mediation architectures, various sources of information appear transparent to the user, who will use a form of query language, addressing them as to a unique resource. Several mediation architectures that have been released are: K2/Kleisli (Davidson et al., 1997; Chung and Wong, 1999), TINet, P/FDM, DiscoveryLink, TAMBIS (Goble et al., 2001), Xmap and Xcollect project (Devignes et al., 2002), and BioMoby[6] which integrates bioinformatics resources as web services and allows users to define their own workflows (Wilkinson and Links, 2002; Wilkinson et al., 2005).

Other examples of mediation architectures:

- systems that focus on representing the knowledge to make it accessible for mining and visualisation; these represent relationships between biological entities as (complex) networks along with implementing certain metrics and claim to produce testable hypotheses (Gopalacharyulu et al., 2005);
- pipeline approaches, that act on experimental data sources (cDNA sequencing projects), and search public web-based databases extensively in order to systematically identify and characterise novel genes; these are automatic annotation tools (del Val et al., 2004); for example, ProtSweep, one of the workflows, uses the sequence to identify a protein.

---

[1] http://us.expasy.org/sprot
[2] http://www.ncbi.nlm.nih.gov
[3] http://srs.ebi.ac.uk
[4] http://genome.dkfz-heidelberg.de/
[5] http://www.mygrid.org.uk
[6] http://biomoby.open-bio.org

These systems usually have to deal with multiple answers (sometimes contradictory, complementary or in different qualities of precision) coming from different resources. One of the approaches in dealing with multiple answers relies on sorting the results according to user-defined criteria and integrating these according to consistency, discrepancy, or precision (Devignes and Smaïl, 2004).

## 2.2 Protein-Protein Interactions and Protein Networks

A particular case of data integration refers to aggregating data describing protein interactions with other molecules. Proteins are known to be versatile molecules, undergoing structure modifications throughout their life, able to perform a wide variety of actions and playing a central role in the biology of living organisms. Most of their functionality relies on their ability to interact with other molecules.

In systems biology, the study of protein interactions with other molecules (nucleic acids, proteins, various other messenger components) is of prime interest in understanding and describing the functionality of the organisms.

The main focus of this report is towards protein-protein interactions (PPI). Currently, there is a wealth of available data describing PPI (Tucker et al., 2001, Droit et al., 2005, Steffen et al., 2002). This, combined with other data sources (systematic localisation of proteins, mutant screens, and functional tests) can give a network of interactions between proteins. These can be refined in potential signalling pathways and interactive complexes, or used in functional annotations of proteins (Tucker et al., 2001). Tucker et al. (2001) describe a roadmap in PPI studies: from simple pairs of interaction, to protein interaction maps, protein networks organisation, towards regulatory networks and cellular modelling.

### 2.2.1 Protein-protein interactions (PPI)

There are currently various models trying to describe the interactions between proteins. Some of them refer to binding and forming protein complexes, in which case the focus is on docking study and descriptions, or on experimental techniques able to detect protein complexes. Other models concentrate on the temporal aspects to determine a possible interaction, and thus focus on assessing the expression patterns, both by theoretic approaches and experimentally. Some of the models focus on modular signalling domains, and study the interaction through the domain-motif binding paradigm (Obenauer et al., 2003). A possible development of gathering data about binary PPI is functional annotation (identifying proteins with known function as interaction partners for an unknown protein) and experiment guidance (by performing only certain confirming experiments and on only certain proteins, thus directing the search), as highlighted by Tucker et al. (2001).

**Experimental techniques for generating PPI.** Droit et al. (2005) mention a multitude of experimental techniques, roughly classified as molecular biology-based methods, mass spectrometry methods and protein microarray techniques. Molecular biology-based methods are categorised into traditional methods and high throughput methods (Uetz et al. 2000; Ito et al. 2001; Gavin et al. 2002; Ho et al. 2002). Examples of traditional methods are affinity chromatography, immunoprecipitation, and gel-filtration (Phizicky and Fields, 1995).

The most used and known method is the yeast two-hybrid system (Y2H) (Fields and Song, 1989; Ito et al. 2000). Y2H is established as a standard technique in molecular biology. Some advantages with using Y2H are independence of endogenous protein expression, high sensitivity, and making the method applicable to weak interactions. Y2H screens have however some disadvantages, like high rate of false positives (Uetz 2002). Further limitations concern their ability to only describe binary relations, and that the technique is not applicable to kinetics studies.

Other "classic" methods mentioned by Droit et al. (2005) are immunoprecipitation assays, ubiquitine-based split-protein sensor (Johnson and Warshavsky 1994), Fluorescence Resonance Energy Transfer (FRET) (Truong and Ikura 2001), Bioluminescence Resonance Energy Transfer (BRET), and a variant of FRET (Xu et al. 1999; Angers et al. 2000).

Mass Spectrometry-based methods are a second type of experimental method for determining protein-protein interactions. Large scale projects were conducted using mass spectrometry methods, showing that mass spectrometry can generate large amounts of protein-protein interaction data (Gavin et al. 2002; Ho et al. 2002).

Protein microarrays have also recently emerged as a high throughput, automated method for generating protein-protein interaction data (Droit et al., 2005).

**Bioinformatics methods for generating PPI data**. The bioinformatics methods complement experimental methods. They use well-known techniques of the field (like data mining, annotation by sequence similarity, phylogenetic profiling, gene neighbour[7] and domain name fusion analyses) to generate protein-protein interaction data. Bioinformatic methods also focus on creating protein-protein interaction databases, literature-based interaction repositories and computational methods.

**Bioinformatic repositories for PPI data.** PPI databases constitute usually a more elaborate step in gathering and processing information. Droit et al. (2005) mention the following repositories of experimentally determined interactions: BIND (Bader et al. 2003), DIP (Xenarios et al. 2002), GRID (Breitkreutz et al. 2003), SGD (Christie et al. 2004), HPRD (Peri et al. 2004).

**Literature based methods for generating PPI data**. Droit et al. (2005) and Tucker et al. (2001) describe literature-based methods to generate PPI data. These use text-

[7] If two genes are found to be neighbours in several different genomes, a functional linkage may be inferred between the proteins they encode. The method is most robust for microbial genomes but works to some extent even for human genes where open-like clusters are observed.

mining tools to transform journal-reported interactions into database entries. There is a wealth of methods using literature mining in various settings and with different reported efficiencies. Also, new such methods emerge constantly. One literature mining approach worth mentioning is PreBIND (Donaldson et al. 2003, cited by Droit et al. (2005)), which uses PubMed abstracts to extract interaction data. The resulting interactions are then manually reviewed and used to feed records in BIND, or have been used, for example, in the literature compilation study in yeast (Schwikowsky et al. 2000).

**Computational methods for generating PPI data**. While the bioinformatics methods use a mix of experimentally derived and theoretically predicted data to study functional associations between proteins, the methods that focus only on theoretical predictions are named computational methods. These computational methods for predicting functional associations (including direct binding) are generally based on the assumption that interacting proteins have to be regulated similarly and must be maintained in the genome together. Hence, as shown in Droit et al. (2005), computational methods have been using diverse approaches related to this assumption. Marcotte et al. (1999) used domain fusion analysis, based on the assumption that genes regulated together have a tendency to be fused into a single gene. Pellegrini et al. (1999) and Huynen and Bork (1998) used phylogeny conservation, based on the assumption that co-regulated genes tend to be either present or absent together. Dandekar et al. (1998), used conserved gene pairs, based on the assumption that co-regulation requires genes to be close neighbours. Computational statistical learning theory was used by Bock and Gough (2001) for expanding the range of predictions to whole proteomes. Aloy and Russel (2002) used three dimensional interaction modelling.

Finally, integrative approaches, like STRING (von Mering et al. 2003) and POINT (Tien et al. 2004) use combinations of various computational methods to predict protein-protein interactions.

### 2.2.2 Protein interaction maps (PIM)

When combined, the binary PPI relationships are able to generate protein interaction maps, which are sets of interacting proteins, represented as networks or circuits (Tucker et al., 2001). Nodes represent molecules, while the edges represent the interactions. Several large-scale projects using the Y2H method (Pandey and Mann, 2000; Bartel et al. 1996; Walhout et al. 2000; Flores et al. 1999; Ito et al., 2000), showed the way to the generation of large collections of protein-protein interactions in the form of protein interaction maps. A recent large scale Y2H analysis effort by Uetz et al. 2000 on *S. cerevisiae* was combined with other *S. cerevisiae* interaction data from the Yeast Proteome Database (YPD) and Munich Information Center for Protein Sequences (MIPS) repositories to generate a global yeast PIM (Schwikowsky et al., 2000). There are also commercial PIMs available (Pronet / Myriad, PathCalling / Curagen).

Visualisation tools for PIMs come to help in the effort of understanding the multitude of relations in such large collections of interactions. Examples of such tools are:

Cytoscape[8], Osprey[9] (Breitkreutz et al. 2003b), Biolayout[10] (Enright and Ozounis 2001), as cited in Droit et al. (2005).

Such PIM modelling tools typically allow the user to concentrate on certain regions of interest in the networks. This is done by allowing rearrangement of the layout, or selective viewing, like eliminating portions of the network. Other facilities include offering links to the relevant literature and offering contextual information about the relation between two connected proteins. For example, in Osprey, one can see characteristics of the selected links or nodes. Some viewers also allow performing extended searches using the protein's identifier, accession number, or using keywords (Suzuky et al. 2003).

### 2.2.3 Protein network organisation and regulatory networks

It is clear that PIMs contain a wealth of information, which is difficult to handle unless unlikely or insignificant relationships are filtered out. The next step, which usually includes integrating and improving the quality of information, generates organised protein networks. These are simply protein interaction maps that have been undergoing filtration and removal of the relationships considered meaningless, redundant, or uninteresting.

There is a multitude of approaches used for the purpose of generating protein networks, but there are no fixed standards yet. As an example, one filtering method, used in Steffen et al. (2002), is based on the assumption that interacting proteins have to share the same pattern of expression. Microarray expression data are used to rank paths generated by Y2H experiments and to gather the paths into a PIM. Known paths are used as a model for the program output, to fine tune the program's parameters, along with statistical tests to assess the reliability of the results (comparison of real data with randomised data generates far less meaningful pathways in the randomised sets at certain settings). To improve the utility of the representation, ranking data is used as a metric to calculate the lengths of the edges.

Large PIMs can also include functional category assignment or classification. Another possible approach towards simplification and generalisation is to depict connections between functional classes of proteins instead of individual representatives of those classes. These are called "regulatory networks" as they focus on understanding and explaining the functionality (Schwikowsky et al., 2000, as cited by Tucker et al., 2001). Different paradigms in graphical representation of PIMs can thus help generating regulatory networks. Nevertheless, there are still features missing from these representations, like the kinetics involved or the strength of interactions, or the ability to differentiate between individual interactions and complexes of proteins (Tucker et al., 2001).

---

[8] http://www.cytoscape.org/

[9] http://biodata.mshri.on.ca/osprey/servlet/Index

[10] http://www.ebi.ac.uk/research/cgg/services/layout
or http://cgg.ebi.ac.uk/old/cgg/services/layout

### 2.2.4 Cell modelling and automatic generation of pathways

This section describes the final target of all efforts described in 2.2.1-2.2.3, to deepen our understanding of the processes that take place in living cells. It is expected that the usage of other sources of information can reveal regulatory connections and generate regulatory networks and pathways. Such additional sources of information were proposed by Tucker et al., (2001), who used DNA microarrays, mass spectrometry, expression profiles in deletion mutants. Other additional sources were proposed by Steffen et al. (2002), who used more complex datasets, homology modelling for differential weighting of molecules towards the kinases, genetic interaction data biased towards co-regulated proteins, data from protein kinase chips and signalling motif identification.

## 2.3 Scansite: detecting domains and motifs in interacting proteins

Scansite 2.0 (Obenauer et al. 2003) represents an interesting paradigm for the characterisation of protein-protein interactions. The main concept of the approach is that eukaryotic proteins are often built with a modular architecture, combining domains that fold and function independently into larger polypeptides. These domains often occur in multiple unrelated proteins, where they fulfil similar targeting functions. The Scansite authors consider that identifying such a domain in a protein can help in including it on a cell-signalling pathway and thus help to indicate the protein's function. Domains bind to the corresponding ligands by forming direct interactions with small amino acid sequences, which are called motifs.

### 2.3.1 Detecting domains

Obenauer et al. (2003) mention several approaches to predict putative modular binding domains, like sequence comparison methods (Pfam, by Bateman et al. 2002) and Hidden Markov Models (SMAR, by Letunic et al., 2002). They also indicate that modular binding domains are fairly straightforward to predict, and give as an example the abundance of these motifs in public repositories like Pfam (Bateman et al. 2002).

### 2.3.2 Detecting bound motifs

Detecting motifs proves to be more difficult than detecting domains. Motifs are typically short amino acid sequences (under 10 amino acids) and cannot reliably be subjected to techniques like sequence alignment of Hidden Markov Models. Obenauer et al. (2003) use the data from oriented peptide library experiments designed to be recognised and bound by a certain domain, and thereafter isolate and sequence the peptides that were bound. The information gathered in this way is then used to create position specific scoring matrices of different amino acids in the bound motif. A scoring matrix indicates quantitatively the preference for each amino acid type at each position within a certain recognised motif.

### 2.3.3 Functionality of Scansite

Having solved the detection / prediction of both domains and motifs, Obenauer et al. (2003) put together the Scansite set of tools, currently at version 2.0. It consists of a set of two groups of programs, that can either search within a database of sequences and discover all proteins that can be predicted to bind to a specific motif or set of motifs (program group named Database Search), or detect on a given sequence / protein all the occurrences of the chosen motifs (program group named Motif Scan). Both sets of programs can be accessed over the Internet, as searching engines. There are currently 63 motifs with corresponding domains available in Scansite.

As search results can be quite large, Scansite offers the functionality to restrict searches to a certain organism class or species. Other search restricting criteria include proteins' molecular weight, isoelectric point range, number of possible phosphorylated sites, as well as sequence composition. Further restrictions of the searches can be imposed by specifying a keyword that is then matched against portions of the proteins' entry or annotation in databases.

### 2.3.4 Stringency levels in Scansite's Motif Scans

The Scansite authors implement a scoring and threshold system for scanning query proteins with the Motif Scan programs. This is done to decide which scores are likely to suggest real interactions. There are three stringency settings defined, labelled "high", "medium" and "low". To determine the scores, motif matrices of interest were applied to the vertebrate subset of SWISS-PROT domains. The stringency settings correspond to the following thresholds:

| **stringency** | **score falls in the top** |
|---|---|
| high | 0.2% **of all scores** |
| medium | 1% **of all scores** |
| low | 5% **of all scores** |

The values were chosen according to the authors' findings that they increase the reliability of prediction of true positive "hits" while minimising the number of predicted false negative interactions.

### 2.3.5 Scoring results in Scansite's Database Searches

Scoring in the Database Search programs will always return the same score at a certain site for a given protein. However, the relevance of the score depends on the protein database subset selected for the search. The Scansite authors give the following example: "a search among human proteins will yield sites whose percentiles are relative to all human proteins included in the search; the same site can thus have a different percentile for different database searches" (Obenauer et al., 2003, p 3636).

# 3 Methods

## 3.1 Representation paradigm

The program presented in this work relies on the following paradigm: proteins can interact with one another through binding domains (designated in short as domains) that can recognise and bind to binding motifs (designated in short as motifs). A certain protein can, at the same time act as the binding protein and be bound by another protein, and thus "characterised" by both motifs and domains. The set of interacting proteins connected by interactions can be called a Protein Interaction Map (see section 2.2.2).

The representation is done by assigning each protein in the Protein Interaction Map to a node and by displaying the interactions as edges. As proteins are connected through motifs, each edge is associated with a collection of motifs (one or more, depending on the findings).

## 3.2 Overview of the method

Obenauer et al. (2003) claim that "predicted domain-motif interactions from Scansite can be sequentially combined, allowing segments of biological pathways to be constructed in silico" (p 3636). However, until now, this has not been done in an automated fashion, but rather required manual work, involving limitations in terms of time consuming searches and further processing of matching results. This is due to the fact that all information retrieval must be done by inputting textual information in search forms in web pages, and the answers are provided as web pages from which the user must extract the part of interest and usually start another search in a different context. Furthermore, generating consolidated reports from the results is not directly (or at all) supported.

These observations generated the question "is it possible to automate such a process?" We defined the following minimal requirements for an automated system:

- the system should complement Scansite's capabilities of sending requests to Scansite's databases in the form of web page requests
- the system should output the results in a user-friendly way.

The second question was directed to reducing the dimensionality of the results, as Scansite does not provide a method to limit the searches to a certain tissue, only to certain species and organism classes (it is also true that the tissue restriction would logically apply only to organisms that have tissue differentiation). The question was "is it possible to restrict the searches to a certain tissue?" As the tissue information is not included in Scansite's reports, this revealed the need for additional sources of information.

## 3.3 Algorithm

### 3.3.1 General considerations

Generating a Protein Information Map from the type of data provided by Scansite can be implemented as a repetitive process of enumerating the possible interactions between the proteins in the data set. In our case, the first step is to start from a protein, and try to find out what motifs can be found in it. The second step is to find what other proteins can bind to those motifs. The proteins found in this step are added to the list of proteins of interest, the ones constituting the Protein Interaction Map, and this constitutes the third step in our method. When this process is completed, it can be repeatedly applied on another protein in the Protein Interaction Map that has not been processed by the algorithm.

When generating a Protein Interaction Map, new proteins are added to the Protein Interaction Map on the basis of their possible interaction with proteins already in the Protein Interaction Map. This takes place by enumerating the possible protein-protein interactions of a given protein at a time. We named this process "the expansion" of the node that represents the protein.

Deciding the expansion order of the nodes has been left to the user's choice under the consideration that the user ought to have the needed knowledge to "guide" the Protein Interaction Map expansion to the interesting areas.

### 3.3.2 Generating interaction information / expanding a node

As a first step, a protein has to be scanned for motifs, to determine whether it contains certain motifs, or not. The results vary according to the settings used in the search. This step is accomplished by running a Motif Scan on the chosen protein. The user can choose to use one or many motifs and the stringency level. As a result, the list of binding motifs is generated.

When scanning a protein for motifs, it is common to find a motif represented many times. Each motif is present at a different site. This occurs more often when the stringency is low. However, for the binding probability, the multitude of motifs of the same type on a protein is not considered important. If a motif is present, this creates the possibility to bind to it, and more than one motif should not make a difference to the possibility of binding. It could be argued that many motifs could increase the probability of binding, but this information is not taken into consideration when expanding a node. This is discussed in the chapter 5.3 “Further developments”. For the representation paradigm however, the assumption that one present motif has the same effect as many motifs of the same type allows us to characterise the edges (the links between proteins) by their motifs and not by their binding sites.

This little more abstract characterisation allows us to have a comprehensible representation in the presumably possible case that two proteins can connect to one another by several types of motifs. In this case, the edge can be associated with a list of motifs, which is by far more manageable and easier to understand than a list of sites.

Figure 1 shows a schematic representation of how motifs can appear on a protein. Several sites can contain the same motif.

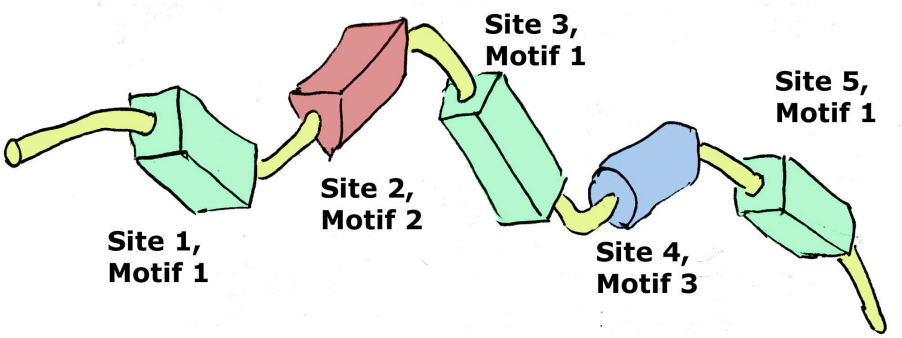



**Fig 1**. Sites containing various motifs on a protein chain

In this case, a motif list would look like this:

```
Motif 1 {Site 1, Site 3, Site 5}

Motif 2 {Site 2}

Motif 3 {Site 4}
```

The second step is to find out which proteins will bind a certain motif. For each motif in the list obtained for our expanded node / protein, we must search the proteins able to bind to that motif. For this, we use the Scansite's Database Search interface. Here, the user can specify the database from which proteins are to be searched, and to optionally restrict the search to an organism class, species, molecular weight range, isoelectric point range, and a number of supposed phosphorylated sites. The keyword search can limit the results to matches from within the "Protein name"[11] field in the UniProtKB/TrEMBL entry and there is an option to limit the search to proteins containing a certain amino acid sequence. Thus, for each motif, a list of proteins can be obtained.

The third step is to retain from this list only the proteins that are related to a certain tissue of interest. Determining information about the tissue a protein corresponds to is not a straightforward process. The detailed explanation of how this step is accomplished may be found in the section 3.3.3. At the end of this process, we have a list of possibly connecting proteins for each motif. The list is restricted to a certain tissue of interest.

[11] The “Protein Name” is a textual description of the protein

The fourth step is to check, for each protein in each list, whether the newly found proteins are not already present in the network. If this is the case, the existing link is updated with the new motif.

The fifth step is to check for every new protein, whether its interaction with the starting protein is not characterised by multiple motifs. This is revealed by the protein being found on lists generated by different motifs. In such a case, the new protein is to be registered only once, but with multiple motifs. This is actually done simultaneously with step four, by traversing the lists and, for each entry, checking if that entry (and respectively the link described by it) is already present in the list of links for the Protein Interaction Map. In such a case, the link description is updated with the new motif (multiple motifs can characterise the interaction between the proteins). Otherwise, a new link is created and the protein is added as a new unexpanded node of the network. At the end of these steps the node that started the expansion process is added to the list of expanded nodes and a number of new links and new nodes are registered in the Protein Interaction Map.

From this place, the user chooses a new node to expand, or interrupts the process of generating the Protein Interaction Map.

### 3.3.3 Getting the tissue information

The simplest and most direct way to get tissue information was found to be using the UniProtKB/TrEMBL entry. The format of these records is standardised in what concerns the general form, while the content formulation can vary quite a bit. Several places in a UniProtKB/TrEMBL entry can give information about the tissue:

- the DE lines hold a description of the protein; this field is not standardised, which results in varying formulation, probably decided by the annotator.
- the RC line holds some standardised fields (the "STRAIN" annotation and the "TISSUE" annotation) in a fixed order (STRAIN would always appear before TISSUE) but contains as well some non-standardised elements, as STRAIN or TISSUE can miss either one or both, and their formulation is not standard. For example, TISSUE can be formulated either with an organ name (like "Aorta" or "Testis") which designates the dominant tissue of the organ, with an enumeration (like "Eye and skin"), or with a generic description (like "whole body" or "embryo"). Furthermore, the data in the TISSUE line comes from research published about that tissue, and there is no guarantee against false negatives in annotation (if a UniProtKB/TrEMBL entry has not listed a tissue, we cannot be sure the protein is not expressed in that tissue at all).
- the CC line holds various comments, where the comment "TISSUE SPECIFICITY" may be found. This is a highly irregularly formulated note, usually containing a detailed description in natural language which is difficult to use for automatic text extraction purposes (an example, "Widely expressed. Low levels found in liver with slightly higher levels present in thymus and testis" gives a snapshot of the type of information).

Given the information available in a UniProtKB/TrEMBL entry, it is clear that the tissue identification is partially unreliable. The easiest source to use is the RC line, which lists standardised names of tissues. It favours, however, false negatives. Using

other fields (like DE or CC) is difficult to implement (text extraction tools and automatic classification could be used) or requires user decision. Other supplemental information sources could be used (like the high quality Human Protein Atlas[12], Uhlén and Polen, 2005), but they are not complete and thus yield many false negatives.

As the limited time for implementation would not allow for intricate solutions, and what was really wanted was a proof of concept, the final decision was to use simple text matching against the TISSUE note in the RC field for filtering purposes and then extend the program to more nuanced behaviours in future development steps.

### 3.3.4 Additional comments on connections between proteins

As a protein can be scanned for several motifs, a set of other proteins able to bind to those motifs can be generated. These are then nodes in the Protein Interaction Map that connect to the protein of interest and an edge is to be drawn from each new found protein for a certain motif towards the starting protein. The edges are directional, in the sense that they express which protein binds to which partner. Building the network starts from the bound protein towards the binding proteins, even though the arrows, the directions of edges, point inversely. The protein displaying the motifs, that is the bound protein, is hence considered the destination of the edge and the binding protein is considered source of the edge.

There can be three types of connections or edges: unary, in which a protein connects to itself (source identical with destination), binary unidirectional, in which a protein is able to bind to another one, which is the general case, and binary bidirectional, in which each protein can bind the other one (source and destination interchangeable).

For each binding motif, a list of proteins able to bind to it can be generated. When a protein displays many different motifs, the equal number of lists of proteins able to bind the respective motifs can be generated. It might be the case that some of the entries in the separate lists (for different motifs) repeat. In this case we have found that a protein (that one that is to be found on many lists) can bind to the destination protein through more than one motif. Thus, the edge is characterised by more than one motif.

### 3.3.5 External representation of a Protein Interaction Map

A Protein Interaction Map is represented by generating a text file in the Osprey (Breitkreutz et al., 2003) format, variation 3. There is the option, as a further development choice, to use the variation 4. The Osprey file formats are documented in Osprey Operator's Manual, sections 3.12.3 and 3.12.4[13].

In such a file, a link is represented as a line with values separated by tab characters. The starting and destination nodes of the link are the first two values, followed by the information characterising the link that can be stored in the next three positions of the line, under the headings of: "Experimental System", "Source" and "PubMed ID",

[12] Human Protein Atlas, http://www.hpr.se

[13] http://biodata.mshri.on.ca/osprey/Documents/Osprey_1.2.0.pdf

respectively. In these fields, information about the motifs characterising the interaction could be stored, but this is still in course of being implemented.

It should be noted that the Osprey visualisation tool and environment is not directly targeted at representing Protein Interaction Maps, but rather at representing gene networks and that the software has many more capabilities than we exploit.

An excerpt from an OCF Osprey file, custom file variation 3, shows as an example of how a Protein Interaction Map is represented in a text file.

The first row contains the "root" node, which is provided for convenience, to make easier for the user the task of finding the starting node of the Protein Interaction Map. For each connection, the source nodes are listed in the column "GeneA" and the destination nodes are listed in the column "GeneB". The first five proteins connecting to the starting node are shown. The following two columns, "Experimental System", and "Source" can be used to store information that characterises the connection, as Osprey displays this information when a connection is selected. An example is provided in figures 2 and 3. The last column, "PubMed ID" is to be left blank, as it is required by the Osprey viewer to store the information needed to connect to literature references.

| GeneA | GeneB | Experimental System | Source | PubMed ID |
|---|---|---|---|---|
| P31749 | root | link characteristics | search | |
| Q86UW6 | P31749 | | params. | |
| O15040 | P31749 | motif one | stringency=low | |
| | | | species-human | |
| Q9P241 | P31749 | | | |
| O94916 | P31749 | | | |
| Q8TEU7 | P31749 | | | |

Fig 2. Tab-separated text file describing a Protein Interaction Map, according to the format required by the Osprey viewer; the third row, the connection between O15040 and P31749 depicts an example of usage of the contextual information

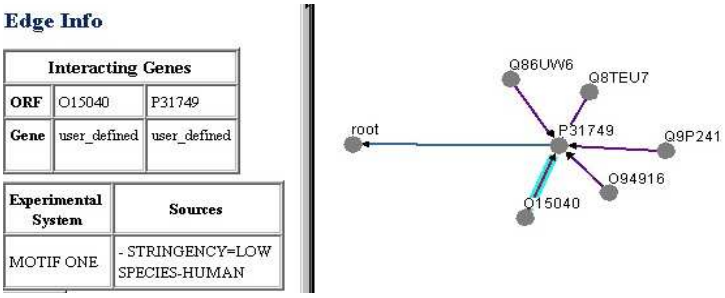



Fig 3. The corresponding apearance of the Osprey file showing the edge information displayed for the example edge between O15040 and P31749

# 4 Design and Implementation

## 4.1 Design overview

This chapter describes the software that was developed in this project. People interested in obtaining the software can find the archive containing the source code and the auxiliary files in the “Files” area at the address:

http://groups.yahoo.com/group/p_i_e_s .

The result of the software is a Protein Interaction Map (PIM). A PIM is represented as a set of three lists: one that holds the expanded nodes in the PIM, one that holds the non-expanded nodes in the PIM, and one that holds the links describing the interactions between the proteins of the PIM. For simplicity reasons, we call the PIM a network. All lists are implemented through the class java.util.Vector, an implementation of the java.util.List interface.

A Protein Interaction Map is here represented by a Network object (Network.java class). All the rest of the software is built to support the generation of the PIM, that is, it consists of helper classes, with various degrees of specialisation, organised in a number of packages, according generally to their specialisation.

The Network class is a member of the sandbox package, the main package developed in the project. An overview of the package sandbox, with the most important classes and relationships is presented in figure 4.

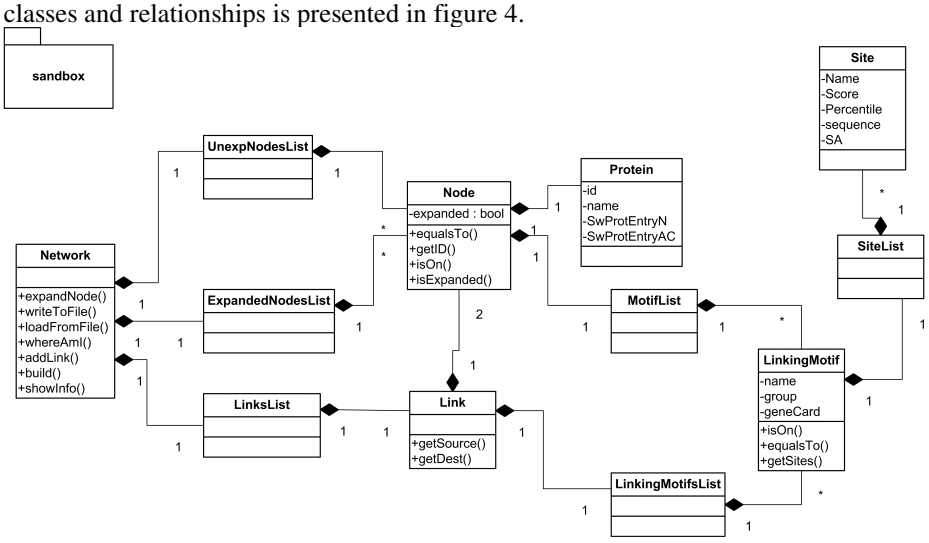



Fig. 4. An overview of the package sandbox, with the most important classes and relationships: note the central place of the Network class

Some of the support operations implemented in the other packages are listed below:

- interfacing with users
- communication over the Internet to retrieve web pages
- reading and writing local files
- text processing, text extraction from retrieved resources

### 4.2 Overview of the software organisation

The application developed in this project consists of an entry class (main class) named Interface.java and a number of packages:

- **package engines**, which contains classes used generally to build the queries for the external databases, handling their results, as well as configuring the connection over the Internet; the scenarios (see section 4.4) are implemented by classes in this package.
- **package html**, which contains classes used generally for parsing web pages and extraction of information.
- **package protein**, which contains classes used generally to handle and store information related to proteins: representation of protein entries, filtering lists of proteins according to some criteria, saving and reading temporary files that represent lists of proteins (used mainly for debugging purposes).
- **package sandbox**, which contains the classes used to build the Protein Interaction Map, that is the class representing the PIM (Network.java), the classes representing the other objects associated in a Network (Nodes, Links, LinkingMotifs, Sites) as well as some helper classes used to interface easier the package with the rest of the software (like Glue.java).
- **package uts**, which contains helper classes, that is the ones who perform various minor operations (presenting menus to users, reading from keyboard, clearing the screen), as well as the class that handles the actual communication over the internet (PageRequester).

Most of the important classes have as a general feature, a main method, which can be used to call the class separately, usually for testing purposes.

As a final note, an important component of the software is the **GNU wget** web page retriever and its configuration files. This complements in certain cases the class that is capable to request documents over the Internet, when requesting web pages requires a more complicated communication protocol than the basic request-response dialogue with the web server.

### 4.3 Interfacing with the user and program usage

The program is a command-line application. To run the program, the users have to start a console, direct it to the program location and start the program through the Interface class, which is the standard entry point. No command line parameters are required, as the program is intended to be used interactively.

The Interface class provides a menu for choosing various actions, like:

- configuring the connection to the Internet, the proxy configuration
- executing simple searches / queries on the Scansite and SwissProt
- executing a complete scenario of building a PIM
- getting explanations on the software usage, help system
- quitting the program

Not all the actions are implemented completely, the less covered area being the help system.

### 4.4 Scenarios

Running the software is done starting from what data is available provided by the user. In a first usage scenario, the user is assumed to know the name of the protein and to perform a Motif Scan on it and start building a Protein Interaction Map from that point. The scenario starts with interrogating the name of the protein in UniProtKB/TrEMBL repository to get an accurate accession number. As for textual names several hits can be returned, there is a choice to filter the results using a species, and a tissue of choice, as well as performing a manual selection on the filtered lists. The goal is to find an Accession Number. Using a name as a starting point for finding the protein identification is considered to be a stable option, as opposed to the Protein ID, which can change over successive updates in the UniProtKB/TrEMBL databases. After finding the protein's Accession Number, this is fed in a Motif Scan query, where the user can choose which motifs to scan for, specify what database the Accession number comes from and, finally, set a stringency level. The result is a list of motifs identified for the chosen protein. This process is actually similar to expanding a node, that is, to expand the starting protein node. The building of the Protein Interaction Map continues under the user's control, who is supposed to choose which node to continue the expansion of the Protein Interaction Map with. Because of the command line interface, there are two possible options for helping the user in orienting which nodes to be expanded. First, the user can list all unexpanded nodes of the Protein Interaction Map and choose which one to expand further. Secondly, the user might generate an Osprey file and inspect the image in order to decide which node to expand further. None of these solutions is optimal. For the first case, the listing showing the unexpanded nodes in a Protein Interaction Map can become quite large and thus be difficult to examine. Using an Osprey file requires on the other hand additional steps in operation and, Osprey being an external program, switching between applications. This limitation could be overcome if the application would be a graphical one.

In a second usage scenario, the user might have a description of the characteristics of a protein (like molecular weight, isoelectric point, type of motif it is supposed to bind, the organism class and species it comes from) and might want to find out which proteins correspond to the given description. This is done by a Database Search in Scansite. Then, each of these proteins can be scanned for motifs of choice (which is done by a Motif Scan in Scansite), and then a PIM can be built repeating the two operations on the rest of the proteins, in a similar fashion as in the first scenario.

## 4.5 Implementation details

### 4.5.1 Class Network

Implemented in the file Network.java, part of the sandbox package.

It is the class that represents the Protein Interaction Map. The class is an aggregation of three lists (implemented as java.util.Vectors) keeping count of the members of the Protein Interaction Map: expanded Nodes, unexpanded Nodes and Links between Nodes. The classes describing a Node (Node.java) and a Link (Link.java) are detailed in further sections.

The Network class mainly implements methods for keeping track of Nodes and Links in a Protein Interaction Map, for expanding a Node, for generating an Osprey file, for saving the Network status to a file, for loading a Network from a file and for providing an interface to the user to direct the expansion of the Protein Interaction Map.

#### *4.5.1.1 The **build()** function*

It runs an infinite loop and lets the user decide where to drive the program execution: list the unexpanded nodes in the Network, expand a Node, show the location (paths to the root of the Network) of a Node, show statistics about the Network (number of Nodes, number of Links), generate an Osprey file, write or read the Network description to and from a file, or quit working with a Network. Each of these actions is implemented as a separate function which gets called, executes and returns the control back to the build() function.

#### *4.5.1.2 The **expandNode()** function*

It is the place where a Node is expanded to extend the Network. It is highly procedural, and can be described by this pseudo-code:

```
1. is node expanded?

    if yes, announce the node is already expanded and
return

    if no {

       2. mark the node as expanded and put it on the
       expanded Nodes List

       3. does the Node have a linking motifs list (has
       it been Motif Scanned)?

          if not, scan Node for Motifs

       4. is the node the root, the first node in the
       network?

          if yes, generate a mock node root and a link to
          it for later easier identification in agglomer-
          ated graphs
```

```
        5. for each motif in the motif list of the node,
        Database Search Motif and get a list of proteins
        {
           6. filter the list of proteins versus species
           7. filter the list of proteins versus tissue
           8. for each protein left in the list make a
           temporary link with the associated "source"
           node (the "destination" being the node we are
           expanding){
              9. check whether this link is already in the
              network (known relation); only source and
              destination nodes are checked;
              if yes, add the current Motif to the already
              existing Link
              if not, the link is completely new{
                 10. add link to the network
                 11. and add the source node of the link to
                 the unexpanded Nodes list of the network
              }
           }// end for each protein in the list
        }// end for each Motif in the node's list
  //end expansion
```

*4.5.1.3 The **whereAmI()** function*

It prints paths that connect a given Node to the root of the Protein Interaction Map. The function is recursive and does backtracking for visited nodes, so as not to get into infinite loops. It takes as parameters a Node and a Vector containing already visited Nodes, Nodes that are already on the description of the path to the root of the Protein Interaction Map.

The operation of the function is described by the following pseudo-code:

```
  1. is the current node in the visited Nodes list?
     if yes, it's a cycle and we don't need to further
     follow this path
     if not {
        2. add the node to the visited path
        3. check whether the search has hit the root
        if yes, print the path we found
        if not {
           4. enumerate all the links that leave the node
           (where Node is source)
           5. for each link{
```

```
            6. get the destination node
            7. apply the whereAmI() function to the des-
            tination Node with the visited Nodes list
          }// end for each Link
        } // end if Node not root
      } // end Node not visited already
  8. do some error checking for the current Node{
      9. does node exist in the Network?{
        10. if not print error
        11. if yes, check if it has connections to other
        nodes in the Network{
          12. if not, it's a stray node, print error, no
          path to root
        } // end node existing but stray (11 and 12)
      } // end 9., node existing in Network?
  } // end final error checking
```

*4.5.1.4 The **writeToFile()** function*

It generates the Osprey file. The function should be more complex, and should save as well data about which of the Nodes is expanded as well as elements that characterise Nodes and Links. The current implementation is rather simple:

```
for each link in the Links list{
    generate a string of the form: <source node> + <tab
    character> + <destination node>
}
```

### 4.5.2 Class Link

It is implemented in the file Link.java, part of the sandbox package. It is an aggregation of two Node objects (a source and a destination Node) and a list of Motifs that characterise the Link implemented as a java.util.Vector. The class is used mainly to describe the relationship between two nodes. Another design option would have been to store information about the neighbours in the Node class, but that proved to be a complicated approach, given the possible complexity of a Link description.

#### 4.5.3 Class Node

It is implemented in the file Node.java, part of the sandbox package. It is an aggregation of a Protein object that characterises the Node, a list of found expressed Motifs implemented as a java.util.Vector, an array holding the parameters used in the MotifScan search, an array holding the parameters used in the Database Search and a Boolean qualifier that signals if the Node is expanded. The search parameters are saved to leave the possibility in the future to increase the automation of the Network generation, by passing down these parameters to the newly generated nodes.

The functionality implemented in a Node object is:

- it can return a list with the Links leaving the Node;
- it can report the identification data of the Protein stored in the Node;
- it can return its status, if it's an expanded Node or not;
- it can check if it is on a list of Nodes;
- it can check whether it is equal to another Node (by checking if the contained Protein has the same Accession Number as the other Node's Protein).

#### 4.5.4 Class Protein

It is implemented in the file Protein.java, part of the protein package.

The class acts mainly as a holder for protein descriptions provided by Scansite's Database Search and MotifScan output lists and by the SwissProt entries (from which, by minor text processing, it extracts the information and constructs the object) and as a tool for finding the main accession number of a protein.

This second functionality appeared and increased in complexity during the development process to cope with the differences in identifying a protein between Scansite and SwissProt. It is implemented in the function getSwProtAccNumber() which can get from the SwissProt repository a primary accession number, given a secondary accession number or a protein ID. As the SwissProt servers' response can vary over a set of different web pages, several cases are taken into consideration.

#### 4.5.5 Class Interface

It is implemented in the file Interface.java. It is an infinite loop that presents the user the possibility to start sections of the program.

## 5 Results and Discussion

In this report we investigated the feasibility of using a domain-motif representation to generate Protein Interaction Maps by using domain-motif interaction data publicly available from Scansite, (Obenauer et al. 2003).

After reviewing the literature, to our best knowledge, an approach to generate Protein Interaction Maps starting primarily from domain-motif interaction data has not yet been reported. Instead, domain-motif interaction data have only been used to fine-tune existing protein-protein interaction sets.

Secondly, we investigated the possibility to reduce the dimensionality and increase the biological plausibility of the results by limiting the collection of data to a tissue-specific set of proteins. Such tissue specificity information is not readily available with the domain-motif interaction data, and had to be integrated in the system from publicly available protein annotation sources (Uniprot/Swissprot, Bairoch et al. 2005).

Thirdly, we investigated the feasibility of automating the procedure of the data collection.

The assumption was that the presented approach of using domain-motif interaction data together with focusing the data collection on a certain tissue would be a possible way to generate Protein Interaction Maps. An ideal form of a Protein Interaction Map is presented in figure 5.

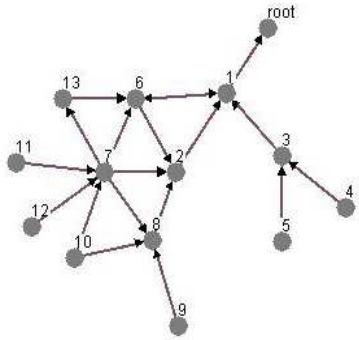



Fig. 5. An ideal example of a Protein Interaction Map; node 1 is the starting node, root being an identification pointer

### 5.1 Results

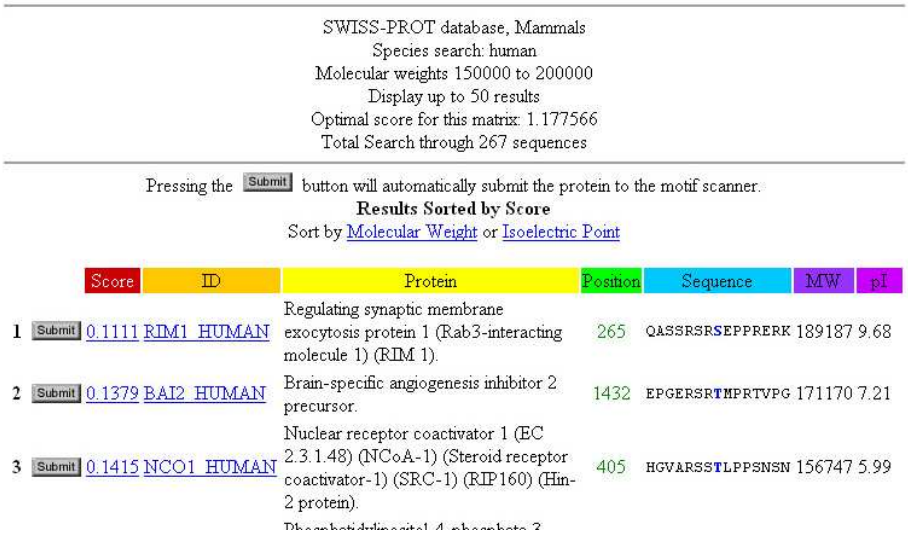
Running the software generates sets of pairs of possibly interacting proteins, showing that it is possible to use publicly available domain-motif interaction data to build such data sets. Furthermore, choosing to retain only the protein entries that are reported to have a connection with a certain tissue of choice, that is, applying the tissue specificity filter, reduces the number of total proteins in the data set. This behaviour of the presented software comes to complement and enhance the other specificity filters already available with the public resources, for example the organism class, the organism species, the molecular weight and the isoelectric point reported for a protein entry.

However, the data sets obtained after performing some experimental runs proved to rapidly increase in size and to display a very low level of organisation, that is, a large ratio of the connections number compared to the number of the proteins in the interaction map.

The automation of the data collection procedure is at the moment only partial, in the sense that the user's intervention and knowledge are required to decide which nodes in the Protein Interaction Map are to be expanded and which search criteria and parameters must be used. However, obtaining the tissue information is greatly simplified and takes place transparently to the user, in an automated way.

Without using the software, obtaining the annotation information for a protein using its "Protein ID", a common identifier in the reports provided by Scansite, can require performing several additional steps of following the links provided by the reports and requesting additional web pages. In the worst case, the user may need to require three additional web pages to get access to the annotation of a protein.

The reason for this is that the "Protein ID" identifier can change in the consecutive revisions of the UniProtKB/TrEMBL repository and hence there might be differences between the identifiers used by Scansite and the ones used by the UniProtKB/TrEMBL. An example of a situation when additional steps are needed to have access to the annotation of a protein is provided in figures 6 to 9, featuring the protein with the ID **RIM1_HUMAN**.



SWISS-PROT database, Mammals
Species search: human
Molecular weights 150000 to 200000
Display up to 50 results
Optimal score for this matrix: 1.177566
Total Search through 267 sequences

Pressing the Submit button will automatically submit the protein to the motif scanner.

**Results Sorted by Score**

Sort by Molecular Weight or Isoelectric Point

| | | Score | ID | Protein | Position | Sequence | MW | pI |
|---|---|---|---|---|---|---|---|---|
| 1 | Submit | 0.1111 | RIM1_HUMAN | Regulating synaptic membrane exocytosis protein 1 (Rab3-interacting molecule 1) (RIM 1). | 265 | QASSRSR**S**EPPRERK | 189187 | 9.68 |
| 2 | Submit | 0.1379 | BAI2_HUMAN | Brain-specific angiogenesis inhibitor 2 precursor. | 1432 | EPGERSR**T**MPRTVPG | 171170 | 7.21 |
| 3 | Submit | 0.1415 | NCO1_HUMAN | Nuclear receptor coactivator 1 (EC 2.3.1.48) (NCoA-1) (Steroid receptor coactivator-1) (SRC-1) (RIP160) (Hin-2 protein). | 405 | HGVARSS**T**LPPSNSN | 156747 | 5.99 |

Fig. 6. Typical Database Search report provided by Scansite; following the link provided for obtaining the annotation for protein RIM1_HUMAN results in an error


Fig. 7. Error message provided by UniProtKB/TrEMBL when using a protein ID that has changed: the user is pointed to another page and has to follow a link in order to get there

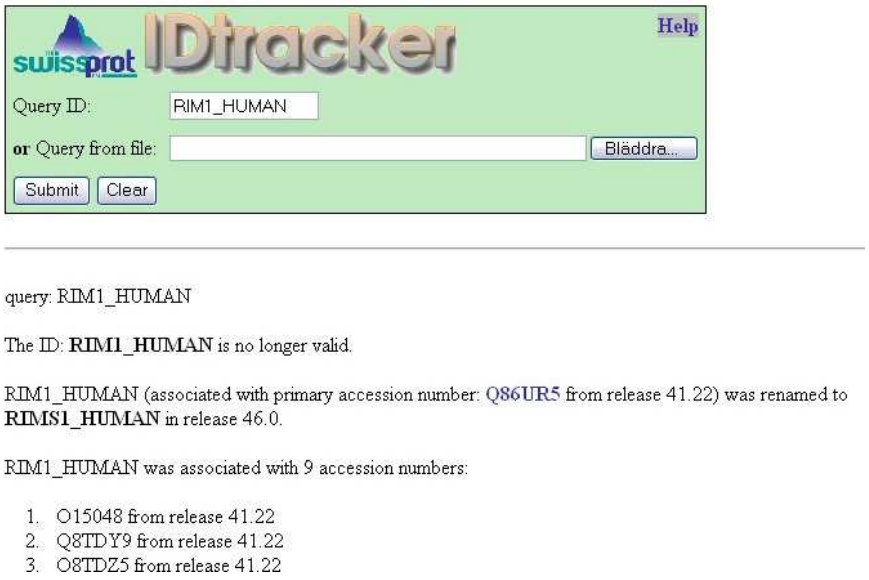



Fig. 8. The next step in tracing a renamed protein, the message showing the newly assigned name, in this case RIMS1_HUMAN; the accession number, an unchangeable identifier is provided as well

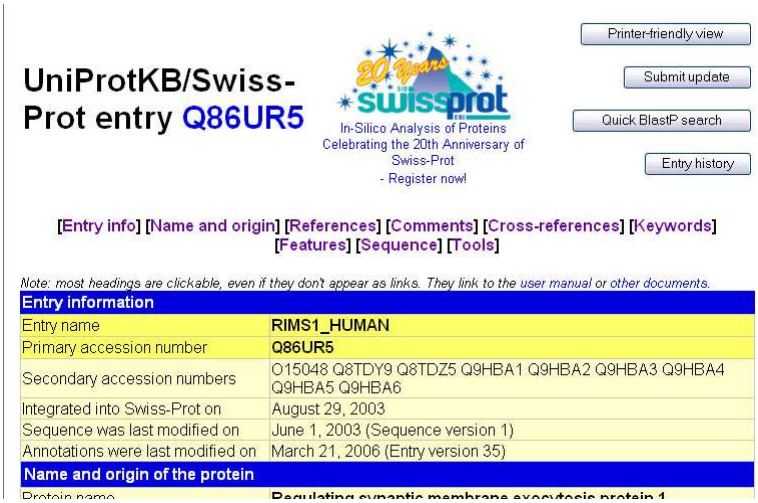



Fig. 9. The third step, getting access to the annotation of the protein through its accession number

The software presented in this report is capable to shortcut all these steps and automatically get the web page containing the annotation for a protein with a changed identifier.

Following, the results obtained in a real run are successively shown in figures 10 to 13, which are obtained by expanding four nodes in an Interaction Map.

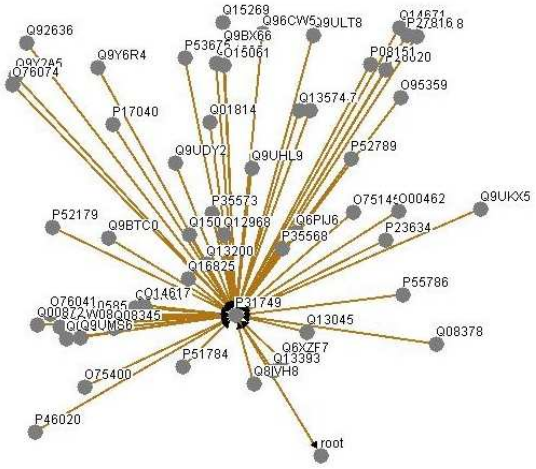



Fig. 10. First step of expanding a Protein Interaction Map, node P31749 expanded

The generation of the PIM starts from the protein with the accession number P31749. This is obtained by searching in the UniProtKB/Swiss-Prot repository all the proteins with the name "AKT1 HUMAN" that have in their annotation the tissue "muscle". There is only one protein conforming these criteria, it is the one with accession number P31749.

In a first step, the protein is scanned for the "14-3-3" motif, with the stringency set to "low". There are three possible sites where the motif can be found. The proteins able to bind this motif are retrieved, with the search being limited by the following constraints:

- molecular weight between 100 and 200 KDa
- organism class: mammalian
- species: human
- unspecified isoelectric point
- unspecified number of phosphorylated sites
- tissue: muscle

The resulting interaction map is shown in figure 10. At this point, the network contains 56 proteins.

In a second step, a similar procedure is applied to the protein having the accession number Q13574. This is scanned for the motifs: "Abl SH2", "Crk SH2" and "Fgr SH2", with the stringency set to "low". The proteins able to bind to these motif are retrieved, using the same settings and restrictions as in the first step.

The resulting interaction map is shown in figure 11. At this point, the network contains 121 proteins.

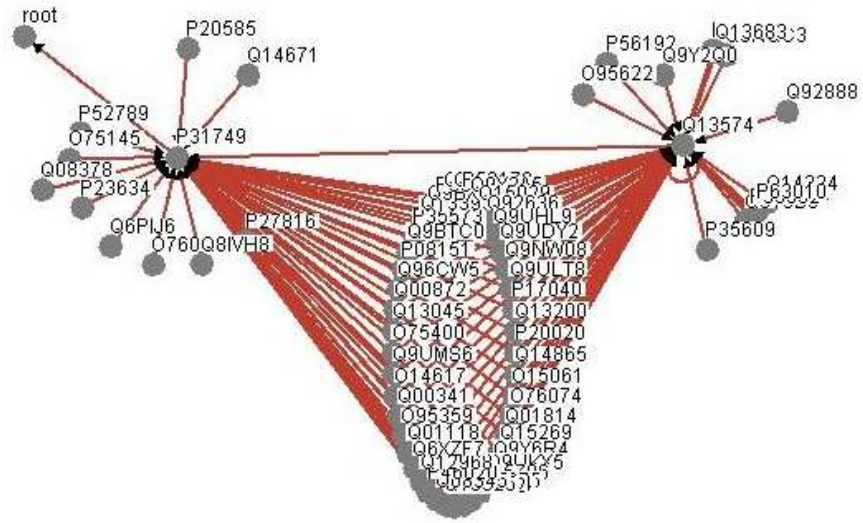



Fig. 11. The second step of expanding a Protein Interaction Map, node Q13574 expanded

In a third step, a similar procedure is applied to the protein having the accession number Q9BX66. This is scanned for the motifs: "Grb2 SH3" and "Itk SH2" with the stringency set to "low". The proteins able to bind to these motif are retrieved, using the same settings and restrictions as in the first and second step.

The resulting interaction map is shown in figure 12. At this point, the network contains 149 proteins.

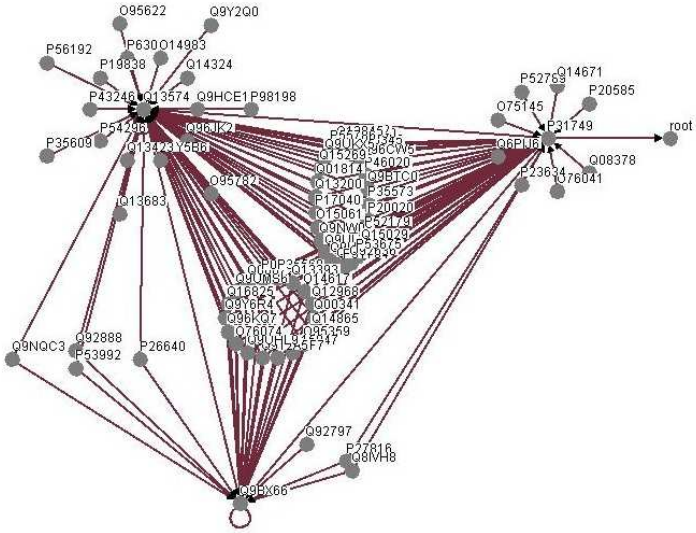



Fig 12. The third step of expanding a Protein Interaction Map, node Q9BX66 expanded

In a fourth step, a similar procedure is applied to the protein having the accession number P56192. This is scanned for the motif: "Shc PTB" with the stringency set to "low". The proteins able to bind to this motif are retrieved, using the same settings and restrictions as in the first, second and third step. At this point, the network contains 195 proteins.

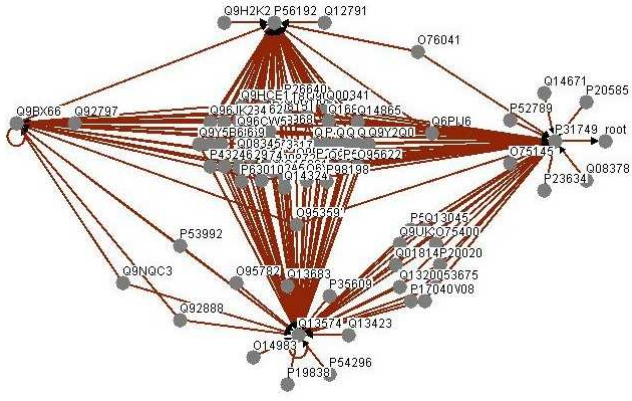



Fig 13. The fourth step of expanding a Protein Interaction Map, node P26640 expanded

### 5.2 Discussion

The wealth of the connections between the proteins that populate the generated Protein Interaction Maps and the large number of proteins included, even after using restrictions of limiting the search to a certain tissue, show that the representation paradigm is probably not powerful enough to generate comprehensible results as of yet.

For example, one possible weakness is that when dealing with an interaction path where all proteins are connected through similar or identical motifs, reconstructing the real path is difficult to achieve correctly. This is because the representation paradigm provides no way to order interactions that are characterised by identical motifs and to eliminate the presumptive interactions that have no biological relevance. Figure 14 shows what happens when reconstructing such a path.

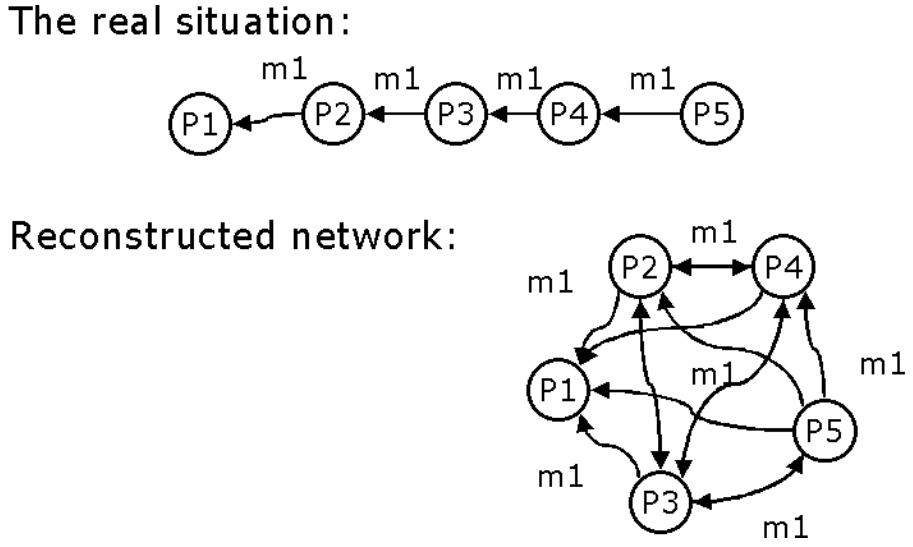



Fig 14. Reconstructing an interaction path described by identical motifs

Thus, the chosen representation paradigm of describing chains of protein interactions by the domain-motif interaction, while being straightforward, intuitive and being supported by the large availability of public data, has some drawbacks that make it not yet ready for completely building Protein Interaction Maps. These drawbacks mainly come from the inherent simplicity of the representation paradigm.

In the following, the possible drawbacks of the presented approach are summarised.

Firstly, the intervention of the user is required to a large degree. The user has to use domain-specific knowledge in choosing which parameters to use in searching possible connections between the proteins and has to decide on the levels of strin-

gency to be used. This might lead to variable results and might require further interpretation of data.

Secondly, there is no qualitative measure of the presumptive interactions described by the domain-motif interaction data. Using such a measure could help in eliminating possibly insignificant connections.

Thirdly, the representation paradigm uses the implicit assumption that a protein is able to interact with any other protein, and does not take into account that the proteins might be located in different cellular compartments. Another implicit assumption is that proteins can be treated as linear chains of amino acids. The tertiary and quaternary structure of proteins are not taken into account and the surface accessibility of the motifs is not taken into consideration.

### 5.3 Further developments

Further efforts are needed to improve the performance of the software, both in increasing the relevance of the results and in enhancing the general operation. There are two main areas in which efforts should be made.

#### 5.3.1 Improving the representation paradigm

As noted in the Discussion section, the representation paradigm has certain weaknesses. We expect that the improvement which would bring the most impact would be to order the sequence in which proteins can interact, especially in cases where the same motifs are driving the interaction on several different levels of interaction (see figure 14). Some types of information that might be added in future work to increase the plausibility of the interaction maps are:

- the cell location of the protein for ordering the elements in the interaction chain and thus removing some connections that might not be biologically relevant or meaningful,
- the accessibility of motifs at the surface of the protein; motifs in the core region should not be used as binding opportunities, thus eliminating some improbable links,
- using further annotation resources to increase the quality of filtering, for example text mining the highly non-standardised "Tissue specificity" field in the protein entry in SwissProt, under user control, or adding the possibility to access the information provided by other public resources, one example being the high quality Human Protein Atlas (Uhlén and Polen, 2005), but many other resources exist.

Another possible continuation of this work would be to develop a weighting and ranking algorithm for protein-protein interactions to take into consideration that:

- if many motifs of the same type are present for binding on a protein, this might increase the probability of binding,
- if an interaction between two proteins is possible through many different motifs, this might increase the probability of binding.

### 5.3.2 Improving the usability and the quality of the software

It is well known that achieving software quality is a time and resource consuming activity. More efforts are required to improve the quality of the presented program. For example, some of the implemented features need to be extended and require further testing.

Adding a graphical interface and the possibility to inspect the generated Protein Interaction Maps within the software without the use of external programs could also greatly improve the usability of the software by simplifying the process of choosing subsequent nodes to expand.

## 5.4 Conclusion

In this report, we have investigated the possibility of generating Protein Interaction Maps by primarily using publicly available domain-motif interaction data. A representation model of the interactions in a Protein Interaction Map has been proposed. It consists of representing the interaction between two proteins by the motif or motifs characterising it. In order to increase the biological plausibility of the generated data sets, using tissue specificity as a method of reducing the dimensionality has been tested. A program that partially automates the process of gathering the data is presented.

The results obtained by running the software show that it is possible to generate collections of presumably interacting proteins by using the domain-motif interaction data. Several limitations exist, the main one being that the intervention of the user in guiding the addition of new interactions to the data set is needed. The obtained data sets are large and further efforts must be done to improve their biological plausibility. Some possible approaches to achieve this have been presented.

As a conclusion, our approach of using primarily domain-motif interaction data for building Protein Interaction Maps has elements of novelty compared to the references in the literature, but at the same time it is difficult to assess its utility and meaningfulness.

During the running and testing of the software, obtaining the annotation data of a protein starting from its "Protein ID" identifier has been recognised as a frequent operation, that sometimes requires the user to perform additional searching steps, when the protein identifier has changed. Consequently, a method to automate this procedure in a transparent way for the user has been developed. While our focus was on obtaining information regarding the tissue specificity of a protein, the procedure can be easily adapted for obtaining any other information contained in the annotation data. This section of the software can easily be reused for obtaining and processing protein annotation data from UniProtKB/TrEMBL in other, broader, contexts.

# 6 References


Aloy P. and Russell R.B. (2002) Interrogating protein interaction networks through structural biology. PNAS 99 5896–5901.

Angers S., Salahpour A., Joly E., Hilairet S., Chelsky D., Dennis M. and Bouvier M. (2000) Detection of beta 2-adrenergic receptor dimerization in living cells using bioluminescence resonance energy transfer (BRET). PNAS 97 3684–3689.

Apweiler R., Bairoch A., Wu C.H. (2004) Protein sequence databases Curr. Opin. Chem. Biol. 8:76-80.

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

Bader G.D., Betel D. and Hogue C.W. (2003) BIND: the Biomolecular Interaction Network Database. Nucleic Acids Research 31 248–250.

Bairoch A., Apweiler R., Wu C.H., Barker W.C., Boeckmann B., Ferro S., Gasteiger E., Huang H., Lopez R., Magrane M., Martin M.J., Natale D.A., O'Donovan C., Redaschi N., Yeh L.S. (2005) The Universal Protein Resource (UniProt). Nucleic Acids Res. 33:D154-159.

Bartel P.L., Roecklein J. A., SenGupta D. and Fields S. (1996) A protein linkage map of Escherichia coli bacteriophage T7. Nature Genetics 12 72–77.

Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S.R., Griffiths-Jones, S., Howe, K.L., Marshall, M. and Sonnhammer, E.L.L. (2002) The Pfam Protein Families Database. Nucleic Acids Res., 30, 276–280.

Blom, N., Gammeltoft, S. and Brunak, S. (1999) Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. J. Mol. Biol., 294, 1351–1362.

Bock J.R. and Gough D.A. (2001) Predicting protein–protein interactions from primary structure. Bioinformatics 17 455–460.

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

Boeckmann B., Blatter M.-C., Famiglietti L., Hinz U., Lane L., Roechert B., Bairoch A. Protein variety and functional diversity: Swiss-Prot annotation in its biological context (2005) Comptes Rendus Biologies 328:882-99.

Breitkreutz B.J., Stark. C. and Tyers M. (2003) The GRID: the General Repository for Interaction Datasets. Genome Biology 4 R23.

Breitkreutz, B.J., Stark, C., Tyers M. (2003b) "Osprey: A Network Visualization System." Genome Biology 4(3):R22

Christie K.R., Weng S., Balakrishnan R., Costanzo M.C., Dolinski K., Dwight S.S., Engel S.R., Feierbach B., Fisk D.G., Hirschman J.E. et al. (2004) Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms. Nucleic Acids Research 32 (Database issue) D311–D314.

Chung, S.Y. and Wong, L. (1999) Kleisli: a new tool for data integration in biology. Trends Biotechnol., 17, 351–355.

Dandekar T., Snel B., Huynen M. and Bork P. (1998) Conservation of gene order: a fingerprint of proteins that physically interact. Trends in Biochemical Sciences 23 324–328.

Davidson, S.B., Overton, C.G., Tannen, V. and Wong, L. (1997) BioKleisli: a digital library for biomedical researchers. Int. J. on Digital Libraries, 1, 36–53.

del Val, C., Mehrle, A., Falkenhahn, M., Seiler, M., Glatting, K., Poustka, A., Suhai, S. and Wiemann, S. (2004) High-throughput protein analysis integrating bioinformatics and experimental assays. Nucleic Acids Res. 32(2): 742–748.

Devignes, M.D. and Smaïl, M. (2004) Integration of biological data from web resources : management of multiple answers through metadata retrieval, In 12th International Conference on Intelligent Systems for Molecular Biology - 3rd European Conference on Computational Biology - ISMB-ECCB (Glasgow, Scotland, United-Kingdom). 3 p.

Devignes, M.D., Schaaff, A., and Smaïl, M., (2002). Collecte et intégration de données biologiques hétérogènes sur le Web – Xmap : application dans le domaine de la cartographie du génome humain. Revue des sciences et technologies de l'information (RSTI) – Série Ingéniérie des systèmes d'information (ISI) 7 : 45-61.

Diella, F., Cameron, S., Gemünd, C., Linding, R., Via, A., Kuster, B., Sicheritz-Pontén, T., Blom, N., and Gibson, T. J., (2004) Phospho.ELM: A database of experimentally verified phosphorylation sites in eukaryotic proteins, BMC Bioinformatics

Donaldson I., Martin J., De Bruijn B., Wolting C., Lay V., Tuekam B., Zhang S., Baskin B., Bader G.D., Michalickova K. et al. (2003) PreBIND and Textomy-mining the biomedical literature for protein–protein interactions using a support vector machine. BMC Bioinformatics 4 11.

Droit A., Poirier G. G. and Hunter, J. M., (2005) Experimental and bioinformatic approaches for interrogating protein-protein interactions to determine protein function, Journal of Molecular Endocrinology 34, 263-280

Enright A.J. and Ouzounis C.A. (2001) BioLayout - an automatic graph layout algorithm for similarity visualization. Bioinformatics 17 853–854.

Etzold, T. and Argos, P. (1993) SRS—an indexing and retrieval tool for flat file data libraries. CABIOS, 9, 49–57.

Etzold, T., Ulyanov, A. and Argos, P. (1996) SRS: information retrieval system for molecular biology data banks. Methods enzymol., 114–128.

Farriol-Mathis N., Garavelli J.S., Boeckmann B., Duvaud S., Gasteiger E., Gateau A., Veuthey A.-L., Bairoch A. (2004) Annotation of post-translational modifications in the Swiss-Prot knowledge base Proteomics 4:1537-1550.

Fields S., Song O. (1989) A novel genetic system to detect protein-protein interactions. Nature 340:245-246.

Flores A. et al. (1999) A protein-protein interaction map of yeast RNA polymerase III. Proc. Natl. Acad. Sci. U.S.A. 96:7815-7820

Gagneur J., Krause R., Bouwmeester T. and Casari G. (2004) Modular decomposition of protein-protein interaction networks. Genome Biology 5:R57

Gavin A.C., Bosche M., Krause R., Grandi P., Marzioch M., Bauer A., Schultz J., Rick J.M., Michon A.M., Cruciat C.M. et al. (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature 415 141–147.

Gopalacharyulu, P. V., Lindfors, E., Bounsaythip, C., Kivioja, T., Yetukuri, L., Hollmén, J.and Oreši, M.(2005) Data integration and visualization system for enabling conceptual biology, Bioinformatics, Vol. 21 Suppl. 1, i177–i185

Hall, M. and Brown, L. (2001) Core Web Programming, second edition, Sun Microsystems Press, Prentice Hall.

Hass, L.M., Schwartz, P.M. and Kodali, P. (2001) DiscoveryLink: a system for integrated access to life science data sources. IBM Systems Journal, 40, 489–511.

Ho Y., Gruhler A., Heilbut A., Bader G.D., Moore L., Adams S.L., Millar A., Taylor P., Bennett K., Boutilier K. et al. (2002) Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. Nature 415 180–183.

Huynen M.A. and Bork P. (1998) Measuring genome evolution. PNAS 95 5849–5856.

Ito T. et al. (2000) Toward a protein interaction map of the buddibg yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. PNAS U.S.A. 97:1143-1147

Ito T., Chiba T., Ozawa R., Yoshida M., Hattori M. and Sakaki Y. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. PNAS U.S.A. 98:4569-4574

Johnsson N. &Varshavsky A. (1994) Split ubiquitin as a sensor of protein interactions in vivo. PNAS 91 10340–10344.

Kim, J. H., Lee, J., Oh, B., Koh, K. K. I. (2004) Prediction of phosphorylation sites using SVM, Bioinformatics, 20(17): 3179-3184.

Letunic, I., Goodstadt, L., Dickens, N.J., Doerks, T., Schultz, J., Mott, R., Ciccarelli, F., Copley, R.R., Ponting, C.P. and Bork, P. (2002) Recent improvements to the SMART domain-based sequence annotation resource. Nucleic Acids Res., 30, 242–244.

Manning, G., Whyte, D.B., Martinez, R., Hunter, T. and Sudarsanam, S. (2002) The protein kinase complement of the human genome. Science, 298, 1912–1934.

Marcotte E.M., Pellegrini M., Ng H.L., Rice D.W., Yeates T.O. and Eisenberg D. (1999) Detecting protein function and protein–protein interactions from genome sequences. Science 285 751–753.

von Mering C., Huynen M., Jaeggi D., Schmidt S., Bork P. and Snel B. (2003) STRING: a database of predicted functional associations between proteins. Nucleic Acids Research 31 258–261.

Mulder N.J., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., Binns, D., Biswas, M., Bradley, P., Bork, P., Bucher, P. et al. (2002) InterPro: an integrated documentation resource for protein families, domains and functional sites. Brief. Bioinform., 3, 225–235

Obenauer, John C., Cantley, Lewis C. and Yaffe, Michael B (2003) Scansite 2.0: proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucleic Acids Research, Vol. 31, No. 13 3635–3641

Pandey A. and Mann M. (2000) Proteomics to study genes and genomes. Nature 405 837–846.

Pellegrini M., Marcotte E.M., Thompson M.J., Eisenberg D. and Yeates T.O. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. PNAS 96 4285–4288.

Peri S., Navarro J.D., Kristiansen T.Z., Amanchy R., Surendranath V., Muthusamy B., Gandhi T.K., Chandrika K.N., Deshpande N., Suresh S. et al. (2004) Human protein reference database as a discovery resource for proteomics.

Schwikowski B., Uetz P. and Fields S. (2000) A network of protein–protein interactions in yeast. Nature Biotechnology 18 1257–1261.

Steffen M., Petti A., Aach J., D'haeseleer P. and Church G. (2002) Automated modelling of signal transduction networks. BMC Bioinformatics 2002, 3:34

Suzuki H., Saito R., Kanamori M., Kai C., Schonbach C., Nagashima T., Hosaka J. and Hayashizaki Y. (2003) The mammalian protein–protein interaction database and its viewing system that is linked to the main FANTOM2 viewer. Genome Research 13 1534–1541.

take from www.biomedcentral.com/1471-2105/3/34

Tien A.C., Lin M.H., Su L.J., Hong Y.R., Cheng T.S., Lee Y.C., Lin W.J., Still I.H. and Huang C.Y. (2004) Identification of the substrates and interaction proteins of aurora kinases from a protein–protein interaction model. Molecular Cell Proteomics 3 93–104.

Truong K. and Ikura M. (2001) The use of FRET imaging microscopy to detect protein–protein interactions and protein conformational changes in vivo. Current Opinion in Structural Biology 11 573–578.

Tucker C. L., Gera J.. F. and Uetz P. (2001) Towards an understanding of complex protein networks. TRENDS in Cell Biology Vol. 11 No. 3 pp. 102-106

Uetz p., Giot L., Cagney G., Mansfield T. A., Judson R. S., Knight J. R., Lockshon D., Narayan V., Srinivasan M., Pochart P. (2000) A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature 403:623-627

Uhlén M., Ponten F. (2005) Antibody-based Proteomics for Human Tissue Profiling. Mol Cell Proteomics 4 (4):384-393

Vdovjak, R. and Houben, G.-J. (2001) RDF based architecture for semantic integration of heterogeneous information sources, in: International Workshop on Information Integration on the Web, ed. E. Simon, A. Tanaka, Proceedings of the WIIW'2001, Rio de Janeiro, Brazil, April 9-11, p. 51-57.

Walhout A.J., Sordella R., Lu X., Hartley J.L., Temple G.F., Brasch M.A., Thierry-Mieg N. and Vidal M. (2000) Protein interaction mapping in C. elegans using proteins involved in vulval development. Science 287 116–122.

Wilkinson M, Schoof H, Ernst R, Haase D (2005). BioMOBY successfully integrates distributed heterogenous bioinformatics web services. The PlaNet exemplar case. Plant Physol 138, p1-13.

Wilkinson, MD, Links, M. (2002). BioMOBY: an open-source biological web services proposal, Briefings In Bioinformatics 3:4. 331-341.

Wroe, C., Stevens, R., Goble, C.A., Roberts, A., Greenwood, M.(2003) A suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data in International Journal of Cooperative Information Systems special issue 12(2):197-224.

Xenarios I., Salwinski L., Duan X.J., Higney P., Kim S.M. and Eisenberg D. (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Research 30 303–305.

Xu Y., Piston D.W. and Johnson C.H. (1999) A bioluminescence resonance energy transfer (BRET) system: application to interacting circadian clock proteins. PNAS 96 151–156.