OpenKnowledge

FP6-027253

Bioinformatics Scenarios

Dietlind Gerloff[1], Xueping Quan[1], Chris Walton[1],
David Robertson[1], Marco Schorlemmer[2], Joaquin Abian[2],
Carles Sierra[2], and Lorenzo Bernacchioni[2]

[1] School of Informatics, University of Edinburgh, UK
[2] Artificial Intelligence Research Institute, IIIA-CSIC, Spain

Report Version: final
Report Preparation Date:
Classification: deliverable D6.1
Contract Start Date: 1.1.2006 Duration: 36 months
Project Co-ordinator: University of Edinburgh (David Robertson)
Partners: IIIA(CSIC) Barcelona
Vrije Universiteit Amsterdam
University of Edinburgh
KMI, Open University
University of Southampton
University of Trento

Bioinformatics Scenarios Suitable for Peer-To-Peer Implementation and Experimentation with the OpenKnowledge System

Dietlind L. Gerloff, Xueping Quan, Chris Walton, David Robertson

University of Edinburgh

and

Marco Schorlemmer, Joaquín Abián, Carles Sierra, Lorenzo Bernacchioni

CSIC, Spanish National Research Council

Abstract: We have identified two problem areas within protein bioinformatics that provide opportunities for (a) experimenting with the OpenKnowledge (OK) peer-to-peer architecture in the context of active bioinformatics research areas; (b) initially targetting small, manageable components of larger OK network separately for implementation; and (c) producing original research results: protein structure prediction and proteomics. Two specific research problems have been found to constitute suitable scenarios. They are described here with respect to their interest to the biologist community and specific problem areas as well as implementation issues. The first scenario has already been implemented using current technology (MagentA) and yielded publishable results.

1 Introduction

Bioinformatics research is one of the two testbed domains that were selected for the OpenKnowledge (OK) project. To investigate the impact on scientific discovery of enabling peer-to-peer interactions via the OK system in the future, scenarios are to be constructed by experts in a part of this domain. These will be used to set up case studies and to demonstrate a method of applying the OK approach and technologies.

Proteins are the “molecular machines” of all living organisms. One of the most basic questions of biology is to understand what proteins do and what influences their functionality. Intensive efforts both in laboratory and computational biological research over the past decades have led to a variety of on-line resources that can be consulted for relevant information. Currently these resources consist primarily of three types: web implementations of analysis/prediction software, downloadable software, and (centralised) databases. The information of interest to researchers as they are aiming to characterise the functionality of one, or many, proteins is very varied. For example, knowledge of the specific three-dimensional structure a protein adopts is can be advantageous, as can knowledge of the amount of this protein found in the various tissues of an organism (e.g. human liver, or brain), or even the amount of other proteins that are known to be functionally linked. Accordingly bioinformatics research in the areas of protein structure prediction and facilitation of proteomics research (see Sections 2 and 3 for background information) is highly relevant to the cause, and new developments are highly visible.

This document aims to describe the two first two research problems we plan to implement using the new OK system. The first scenario is a well-defined, small task in which protein structure models generated by different programs are being checked for “consistency” amongst them; the models are available pre-computed in web-accessible databases. This type of

scenario is representative of a very common activity of bioinformatics researchers, consistency-checking between several web resources without necessarily involving the authors of these resources (knowingly) as peers in the network. A peer-to-peer implementation using MagentA [6] has already been carried out and this scenario is described by means of a paper accepted by the "International Workshop on Distributed, High Performance, and Grid Computing in Computational Biology (GCCB) 2006" for presentation in January 2007 (Section 2). The second scenario is more complex, both with respect to its architecture and the biological problem it seeks to tackle. Some experience gained from implementing the first scenario can be taken forward since consistency-checking can be viewed to be a sub-scenario here, too. Additionally, the benefits of distributed data sharing in proteomics are to be explored through this scenario and, once implemented as an OK system, this is likely to produce progress in proteomic analysis that can currently not be achieved by traditional (not peer-to-peer) means. The close collaboration between Dr. Joaquín Abián, the director of one of the well-established laboratory proteomics facilities in Spain, and the OK project should provide a vital link to the potential user community in this research area, for case studies. An overview description of the problem and thoughts towards its implementation are provided in Section 3.

2 Scenario I : Protein Structure Prediction

Extracted from GCCB-Paper 16 (accepted for publication and presentation), with minor adaptation:

Peer-to-Peer Experimentation in Protein Structure Prediction: an Architecture, Experiment and Initial Results (Xueping Quan[1], Chris Walton[2], Dietlind L. Gerloff[1], Joanna L. Sharman[1], and Dave Robertson[2] - [1]Institute of Structural and Molecular Biology, University of Edinburgh, UK; [2]School of Informatics, University of Edinburgh, UK)

Abstract. Peer-to-peer approaches offer some direct solutions to modularity and scaling properties in large scale distributed systems but their role in supporting precise experimental analysis in bioinformatics has not been explored closely in practical settings. We describe a method by which precision in experimental process can be maintained within a peerto-peer architecture and show how this can support experiments. As an example we show how our system is used to analyse real data of relevance to the structural bioinformatics community. Comparative models of yeast protein structures from three individual resources were analysed for consistency between them. We created a new resource containing only model fragments supported by agreement between the methods. Resources of this kind provide small sets of likely accurate predictions for non-expert users and are of interest in applied bioinformatics research.

2.1 Introduction

Peer-to-Peer Experimentation In Section 2.3 we describe novel results obtained for a specific experiment that concerns consistency in protein structure prediction. When read by itself, Section 2.3 is a novel piece of analysis with a result of interest to part of the bioinformatics community. The broader novelty of this paper, however, is the way in which this result is obtained and in particular the peer-to-peer architecture used to obtain it. A peer-to-peer architecture is one in which computation is distributed across processors and in which none of the processors has an overarching coordinating role - hence coordination for a given task must be achieved via communication between processors. Section 2.1.1 gives our perspective on scientific experimentation as a peer-to-peer activity. Section 2.1.2 summarises the way in which we tackle the crucial issue of maintaining the integrity of experiments in a peer-to-peer setting. With this general approach in place, we then describe (in Section 2.2) its use to implement the specific experiment of Section 2.3. Section 2.4 concludes by summarising the broader system of which this is the first part.

2.1.1 Scientists as Peers in a Web Community

When conducting experimental studies (or amassing information to support experimental studies) from Internet sources, each scientist (or group) may adopt a variety of roles as information providers, consumers or modifiers. Often these roles are narrowly specific, as for example the role one adopts when canvassing trusted sources for information about specific proteins and applying assessment metrics to these that are appropriate to a particular style of experimentation. In science, the roles we adopt and the specific ways in which we discharge the obligations of those roles are fundamental to establishing peer groups of “like minded” scientists in pursuit of related goals by compatible means.

The need to be precise about such obligations is strongly felt in traditional science - hence the use of rigid conventions for description of experimental method and monitoring of its execution via laboratory notebooks, enabling experiments to be monitored, replicated and re-used. Analogous structure is beginning to emerge in Internet based science. For example the structure of Web service composition in Taverna [1] provides a record of the associations between services when using these to manipulate scientific data. Like Taverna, we describe interactions as process models. Unlike Taverna, our process models are part of a system for peer-to-peer communication in which process models describing complementary roles in experimentation are shared between peers as a means of communicating and coordinating experiments.

2.1.2 Scientific Coordination as Peer-to-Peer Communication

Traditionally, peer-to-peer systems have not focused on the issue of maintaining the integrity of complex, flexible processes that span groups of interacting peers. Engineering solutions have polarised into those which are highly centralised (coordinating interactions through a server) versus those which rely entirely on the sophistication (and coordinated engineering) of peers to obtain reliable processes through emergent behaviours. It is, however, possible to have a distributed, de-centralised, interaction guided by a shared, mobile model of interaction. To support this we have developed a specification language, based on a process calculus, that can describe interactions between peers and (since the language is executable in the tradition of declarative programming) can be deployed to control interactions. The language is called the Lightweight Coordination Calculus (LCC) in recognition of our aim to produce the most easily applied formal language for this engineering task.

Space limitations prohibit detailed discussion of LCC, its semantics or of the mechanisms used to deploy it. For these, the reader is referred to [2]. In this paper we explain enough of LCC to take us through the bioinformatics experiment that we detail in subsequent sections. Our experiment relies on the collation of predicted structures for yeast proteins across a number of peers and comparison of the collated data across the peers to produce a tentative assessment of the predictions. The data is filtered based on these comparisons leaving behind only predictions deemed to be reliable on these grounds. Figure 1 defines a LCC specification for our example. Notice that it is specific about the message sequencing and essential constraints on this type of interaction but it leaves flexible the choice and number of peers supplying data and the forms of data lookup and filtering - so the LCC specification is a model of a class of interactions, and we can ground it in specific peers and constraints at deployment time (as we show in Section 2).

An interaction model (or, for scientists, an experimental protocol) in LCC is a set of clauses, each of which defines how a role in the interaction must be performed. Roles are described by the type of role and an identifier for the individual peer undertaking that role. The definition of performance of a role is constructed using combinations of the sequence operator (*'then'*) or choice operator (*'or'*) to connect messages and changes of role. Messages are either outgoing to another peer in a given role ('⇒') or incoming from another peer in a given role ('⇐'). Message input/output or change of role can be governed by a constraint defined using the

normal logical operators for conjunction, disjunction and negation. Notice that there is no commitment to the system of logic through which constraints are solved - on the contrary we expect different peers to operate different constraint solvers.

A data collator, X, accepts a request for data on a yeast protein specified by its yeast-id, I, from an experimenter, E; then X changes role to being a data collector over the set, Sp, of data sources about which it has knowledge; then it reports back the set, S, of filtered data from those sources.
A data collector takes each peer identifier, P, in turn from the given set, Sp, of data sources and changes to the role of data retriever to obtain the data set, D, from P. The set Sd contains each instance of D obtained by data retrieval.
A data retriever requests data on yeast-id, I, from data source, P, and then awaits the report of the data set, D from that source.
A data source accepts a data request and replies with the the data set obtained via lookup of its data base.

$$
\begin{array}{l}
a(data\_collator, X) :: \\
\quad data\_request(I) \Leftarrow a(experimenter, E)\ then \\
\quad a(data\_collector(I, Sp, Sd), X) \leftarrow yeast\_id(I) \wedge sources(Sp)\ then \\
\quad data\_report(I, S) \Rightarrow a(experimenter, E) \leftarrow filter(Sd, S) \\
\\
a(data\_collector(I, Sp, Sd), X) :: \\
\quad (\ null \leftarrow Sp = [] \wedge Sd = []\ )\ or \\
\quad \begin{pmatrix} a(data\_retriever(I, P, D), X) \leftarrow Sp = [P|Rp] \wedge Sd = [D|Rd]\ then \\ a(data\_collector(I, Rp, Rd), X) \end{pmatrix} \\
\\
a(data\_retriever(I, P, D), X) :: \\
\quad data\_request(I) \Rightarrow a(data\_source, P)\ then \\
\quad data\_report(I, D) \Leftarrow a(data\_source, P) \\
\\
a(data\_source, P) :: \\
\quad data\_request(I) \Leftarrow a(data\_retriever(I, P, D), X)\ then \\
\quad data\_report(I, D) \Rightarrow a(data\_retriever(I, P, D), X) \leftarrow lookup(I, D)
\end{array}
$$

Figure 1. Example peer-to-peer architecture in LCC.

2.2 Experiment Implementation

The LCC specification, presented in the previous section, defines how the various peers will interact during the experimental process to provide the desired outcome. This specification describes how the experiment will be performed without directly identifying the peers that will be involved. To perform the actual experiment, it is necessary to supply a set of peers that match this specification, and thereby enable us to instantiate the LCC protocol. In this section, we describe how the actual experiment was performed, and outline the computational services that we constructed to accomplish this task.

From a computational point of view, our experiment is essentially a service composition task. That is, we will construct our experiment by identifying a collection of independent services, and then compose these services together dynamically to enact our experiment. In doing so, we adhere to the popular Service Oriented Architecture (SOA) paradigm, which is commonly used in Grid computing. We use our MagentA tool to perform this dynamic service composition as it was designed specifically for this purpose, as part of the OpenKnowledge project. MagentA is effectively an interpreter for LCC specifications, where the peers are defined by Web Services. We have previously demonstrated the use of MagentA to compose services in the astronomy domain [3, 4]. Nonetheless, there are a number of important differences between the experiment that we perform here, and our previous astronomy experiments. Three of the key differences are summarised below:

1. Previously, we were using web services that had already been constructed for the AstroGrid project. In this case, while the necessary data is freely available on the web, there are no web services constructed to access this data. In other words, the data is accessible using

HyperText Markup Language (HTML) pages intended for humans, but there are no Web Services Description Language (WSDL) interfaces and procedures to make this data available to computation entities, e.g. peers. Therefore, it was necessary to construct our own services to query and retrieve the data through form posting and screen-scraping techniques.

2. The astronomy data was all obtained from the same source and was uniform, i.e. all data was the same format and quality. Here, we are attempting to reconcile data from three independent sources. Each of these sources has derived their data using different methods, and have classified their data in different ways. To overcome these issues, it was necessary to build our services so that they can cope with missing and incomplete data, and can present the data in a uniform way. It was also necessary to design our services so that they could place quality thresholds on the data, and exclude results which did not meet these thresholds.
3. The final difference concerns the control of the underlying services and data. In the astronomy scenario, we were closely associated with the individuals who constructed the services and gathered the data. This meant that we could ask questions about the quality and distribution of the data, and obtain advice on using the services effectively. This time, we have no such close link to the data providers, and we are simply using the data that they have made publicly available. As a result, meta-information such as the quality and coverage of the data sets had to be derived experimentally.

A diagram that illustrates the main components and services in our experiment is given in Figure 2. At the left of this diagram are the three data providers for our experiment, namely: SWISS, SAM, and ModBase. These providers all make their databases available through a standard web page (i.e. HTML) interface. We note that the there is an extra pre-filtering step required for the ModBase database, and we do not operate on the ModBase dataset directly. This step is described later in this document.

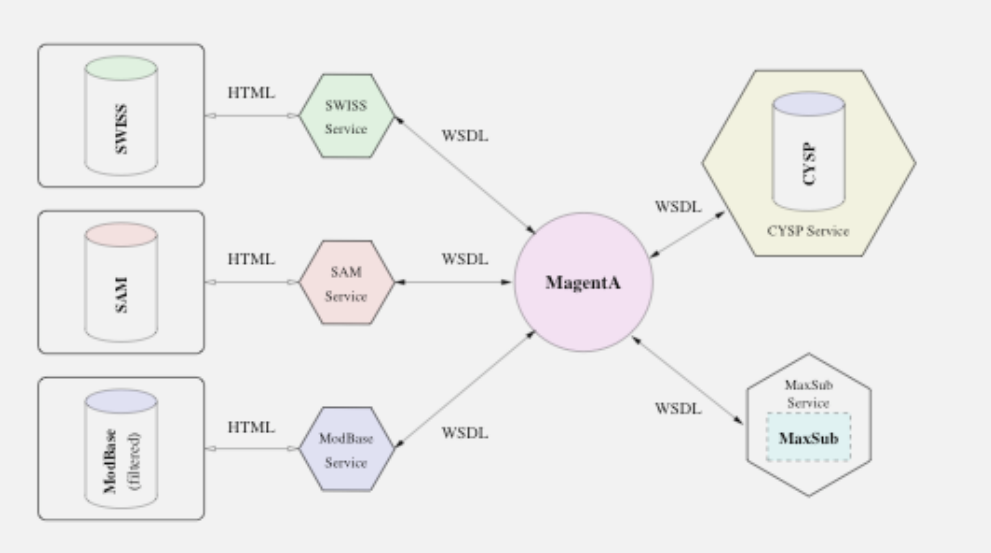

Figure 2. Experiment Architecture

To enable the various data sources to be used in our experiment, we have constructed a web service companion for each of the providers: a SWISS service, a SAM service, and a ModBase service. These companion services enable us to access the data through a standard web service (i.e. WSDL) interface. These services also provide the same abstract interface to the data sources, so that we can query them in exactly the same way. This interface corresponds to *lookup(I, D)* in Figure 1.

There are two additional services that we have constructed for our experiment. These services are illustrated on the right of Figure 2. The first of these is the MaxSub service, which provides

a web service wrapper and WSDL interface for the MaxSub application. This application is used to perform comparisons between sequences. However, it was previously only available as a stand-alone application and could not be run over the web. The second service that we have constructed is the CYSP (Comparison of Yeast 3D Structure Predictions) service. This service is the core of our experimental process. It is responsible for querying the three data providers, invoking MaxSub to perform comparisons, and storing the results in our CYSP database. We provide our own database so that the experimental results can be reused without the need for recalculation, and for future experiment validation purposes. Our CYSP service effectively acts as a filter over the three data sources, and the interface to this service corresponds to *filter(Sd, S)* in Figure 1.

Our experiment is enacted by MagentA, which is at the centre of Figure 2. As previously noted, MagentA is essentially an interpreter for LCC. This interpreter executes LCC specifications directly. An LCC specification is defined in terms of peers, and in our scenario we have five peers: one for each of the web services. During execution, the various peers interact, and the services that they represent are dynamically composed. The details of how LCC protocols are actually executed in MagentA is beyond the scope of this paper. However, further details on the MagentA system can be found in [3,5,6]. We have constructed the five key services for our experiment, namely the SWISS service, the SAM service, the ModBase service, the MaxSub service, and the CYSP service. These services provide us with a uniform way to access the data sources, and to perform computation over the data. We have also used the MagentA tools to compose these services, based around the LCC protocol that we previously presented. The results of the experiment are detailed in the remainder of this paper.

2.3 Example experiment: Consistency-checking in Protein Structure Prediction

Knowledge of a protein molecules three-dimensional structure is vital for understanding its function, targeting it for drug design, etc. The two prevalent techniques for determining the 3-D coordinates of all protein atoms with high precision are X-ray crystallography and nuclear magnetic resonance spectroscopy (NMR). However, compared with the ease with which the amino acid sequences of proteins (their “1-D structures”) are deduced through the sequencing of genomes and cDNA, the effort and cost required for determining a protein 3-D structure remains tremendously high. Accordingly, a number of computational biology research groups specialise in producing structural models for proteins based on their amino acid sequences alone. Where they are accurate, predicted protein structures can provide valuable clues for biological research relating to these proteins.

Thanks to regular rounds of independent assessment using newly emerging atomic protein structures as “blind” tests at the so-called CASP [7] and CAFASP [8] experiments, protein structure prediction techniques have improved noticeably during the past decade. This is particularly true for template-directed protein structure prediction, in which the knowledge of a previously determined protein 3-D structure is used to generate a model for a different protein. If evolutionary relatedness between the two proteins can be established based on sequence similarity between a protein of interest and another protein whose 3-D structure is already known, then a model can be generated through comparative modelling. The known structure serves as a modelling template in this approach. The target protein sequence (i.e. the protein of interest) is aligned optimally onto the structural scaffold presented by the coordinate structure of this template. This typically includes the atoms making up the protein backbone, and the directionality of the side-chains (see [9] for an overview). Comparative modelling is generally considered “safe” to apply when the similarity between the sequences of target and template is sufficient to establish their alignment confidently over the whole length of the two proteins, or at least over relevant portions. However, the confidence in each individual prediction is not easily estimated at the time of the prediction. As a consequence, a substantial proportion of the models submitted for CASP/CAFASP comparative modelling targets are wrong and the degree of accuracy of their atomic coordinates varies substantially [7, 10].

In practice, biologist users of the model databases providing access to the structure predictions are often interested in a single target protein. Such users often apply a "consistency-checking" strategy to assess whether or not to trust the predicted coordinate structures for their protein of interest by comparing the models proposed by different groups/databases to each other. Where the models agree, over the whole or a part of the protein molecule, they are deemed an approximately correct representation of the actual 3-D molecular structure of the target protein.

2.3.1 3-D Structural Models for Yeast Proteins

As an implemented example in which a consistency-checking experiment is undertaken we have investigated the consistency between pre-computed comparative models for the proteins encoded by the genome of the budding yeast *Saccharomyces cerevisiae*. The yeast genome sequence has been known since 1996 [11] and it is currently predicted to encode 6604 proteins. For 330 of these proteins (or fragments of them) 3-D structures for have been determined through X-ray crystallography or NMR to date. For this experiment we selected three public-domain repositories offering access to pre-computed coordinate models for yeast proteins generated by different automated methods: SWISS-MODEL [12], MODELLER (ModBase) [13], and SAM-T02/Undertaker [14]. We systematically retrieved and compared the models for all yeast proteins with to-date undetermined structures and extracted a sub-set of protein models that were "validated" by agreement between all three methods.

2.3.2 Data Sources

The systematic open-reading frame (ORF) names of all predicted protein-encoding genes in the yeast genome, commonly referred to as YIDs, were extracted from the *Saccharomyces* Genome Database (SGD) [15]. The 6604 ORFs listed in SGD on 7 June 2006 were used to query the three model databases.

SWISS : The SWISS-MODEL Repository [16] is a database of annotated protein structure models generated by the SWISS-MODEL [12] comparative modelling pipeline. SWISS draws the target sequences for its entries from UNIPROT (the successor of SWISS-PROT/TrEMBL). Yeast proteins are also annotated with their YIDs. Only models for proteins of unsolved structures are accessible. If the structure of a protein was already determined experimentally (through X-ray crystallography or NMR) SWISS links directly to this structure in the PDB [17].

ModBase : This database [18] contains comparative models generated by the program ModPipe (an integration of PSI-BLAST [19] and MODELLER [13] based on protein sequences extracted from SWISS-PROT/TrEMBL. ModBase typically contains a large number of models for the same target protein that can be considered redundant and imposes only minimal quality standards. In order to streamline the procedure for this experiment and only work with models that have chances of being correct we downloaded all ModBase entries for yeast proteins and eliminated redundancy and extremely low quality models from the set locally (see below). Note that, by contrast to SWISS, ModBase also contains models for proteins with crystallographically or spectroscopically determined structures (re-modelled onto themselves as templates and/or closely homologous proteins).

SAM : The Karplus group at UC Santa Cruz provides WWW-access to provisional models for all predicted yeast proteins using their combination of local-structure, hidden Markov model (HMM)-based fold recognition and ab initio prediction [14]. The methodology underlying fold recognition is similar to that of the comparative modelling in that a template of known structure is used. However, besides various technical differences, the application range targeted by fold recognition methods differs from that of the programs used by SWISS and ModBase in that the former specialize in predicting structures where target-template sequence similarity is too remote to detect by standard methods. Accordingly, this data source offers protein structure predictions for all ORFs of yeast (including some not listed as genes by the SGD database) and a confidence estimate for the corresponding target-template matches together with a collection of provisional (i.e. Unrefined), often fragmented, coordinate models for each.

2.3.3 Processing/Pre-Filtering of data sets

As described above the amount of data pertaining to each YID, and its organization, differs quite dramatically between the three data sources. To ensure that the structural comparisons between the models could be run efficiently for the set of all available yeast models, we undertook a minimum of local processing and pre-filtering of the entries after retrieving them from the three databases. Note that most of this would be unnecessary in the more common situation where a biologist user is interested in comparing only the entries pertaining to a smaller set of YIDs, usually even only a single protein. (In this case, a simple cross-check of the confidence value (which is often expressed as an E-value) associated with the model and/or a search for the best model available could be incorporated in the interaction and a larger number of pair-wise comparisons would be undertaken to extract the protein model region that is supported by the different modelling methods.)

ModBase: Our MagentA interface to ModBase retrieved 3448 files with model coordinates when queried with the list of YIDs. These files typically include more than one 3-D model for the same protein sequence (see above). We pre-filtered the ModBase set of models in two steps, one selecting only high-quality comparative models, and a second to eliminate redundancy. Our selection criteria for “high-quality” models were: percentage sequence identity between target and template > 20 %; model score > 0.7; E-value < 1*E*-06. The relevant values were directly accessible within the ``REMARK'' part of each 3-D coordinate file. Eliminating redundancy is important in cases where individual proteins are represented by multiple ``high-quality'' models in ModBase. As shown schematically in Figure 3, the sequence regions covered by the different models often overlap. This can occur because different template structures were chosen (correctly, or incorrectly) each giving rise to one modelled region. To allow efficient comparison of the complete set of yeast models, we made a choice as to which one model was retained if such a redundant set was encountered. This was based on clustering of the model protein sequences extracted from the 3-D coordinate files for each YID using the program BLASTclust which is part of the BLAST suite (`http://www.ncbi.nlm.nih.gov/BLAST/download.shtml`). Of multiple models with > 90% pair-wise sequence identity over at least 90% (of the length of the shorter model in each comparison), only the model covering the largest sequence region was retained (Model 1 in the example in Figure 3).

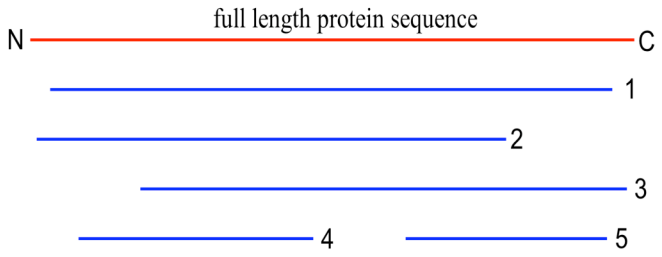

Figure 3. Schematic showing multiple redundant models for one protein. The red line represents the full-length sequence for this protein (running from the N-terminus to the C-terminus of the molecule). Blue lines represent different models (1, 2, 3, 4, and 5) covering different, overlapping and non-overlapping, regions.

By contrast to such instances of redundancy, other proteins may be represented by several models without substantial overlap. Protein structures are generally easier to determine experimentally if they only encompass a small number of structural domains. Accordingly, multi-domain protein sequences are often covered using different template structures for each domain, thus giving rise to several meaningful models. These models may not overlap at all (Figure4A), or partly (less than 90%) overlap with each other (Figure 4B). Multiple models of this kind were retained in our filtered ModBase model set, which contained 2546 models for 2280 yeast proteins.

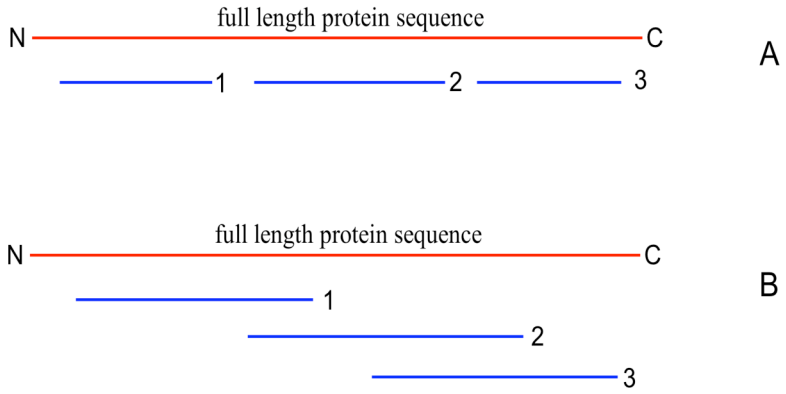

Figure 4. Schematic showing multiple non-redundant models for one multi-domain protein. The red lines represent the full-length sequence for two multi-domain proteins. Blue lines represent models covering different regions of these sequences.

SWISS: The majority of yeast proteins that can be found in the SWISS database only have one associated 3-D model. By contrast to ModBase multiple entries in SWISS can be considered non-redundant, i.e. relate to multi-domain proteins. Moreover, stringent quality standards are imposed by the authors of the database. Our MagentA interface to SWISS queried the "Advanced Search" WWW-interface to the database (`swissmodel.expasy.org/repository/smr.php?job=3`) with the complete list of YIDs and retrieved 769 3-D models for 717 proteins when queried. In the case of multi-domain proteins, all available 3-D models were extracted. An additional 330 returns were crystallographically or spectroscopically determined structures extracted from the PDB database; these were disregarded in the structural comparisons.

SAM: Our MagentA interface to the SAM database of yeast models (`www.soe.ucsc.edu/research/compbio/yeast-protein-predictions/lookup.html`) returned sets of 3-D models for all 6604 YIDs. The models delivered in each set are based on different templates and ordered according to SAM-T02s confidence in the underlying target-template match. Unfortunately this organization is not suited for extracting multiple non-redundant models from the set easily in the case of multi-domain models. For the purpose of this experiment we chose to select the top model from each set, which will usually also select the model covering the largest region of the sequence. In addition we imposed a maximum E-value cut-off of 1*E*-03 for the target-template match. This resulted in 2211 SAM models being considered in the structural comparisons described below.

2.3.4 Consistency Checking

Pair-wise comparisons between the retained 3-D models from the three data sources relating to the same YID were carried out with the program MaxSub [20]. MaxSub performs sequence-dependent pair-wise comparisons between different 3-D structures (predicted models or known structures) of the same proteins, aiming to find the largest substructure over which the two structures superimpose well upon each other. It only considers the base (C_α) atoms of the protein side-chain and also ignores the details of the other backbone atoms. As a metric of the similarity of the two structures that are being compared, MaxSub computes a single score (referred to as Mscore below) ranging from 0 for a completely unmatched pair, to 1 for a perfect match. Since the values for Mscore are asymmetrical, i.e. dependent on which of the two proteins is considered to be the reference protein, we carried out the pair-wise comparisons in forward and reverse order. The distance threshold parameter was set as 3.5 Å throughout the analysis. For proteins whose 3-D structures were previously determined in the laboratory, there

is no interest in a comparison between the X-ray/NMR structure and the 3-D models contained in ModBase and SAM, since the known structures may have been used as the modelling template. (This is different for newly determined structures which will be useful for evaluating the accuracy of the models, as is discussed below.) Pair-wise model comparisons were performed for all YIDs represented by at least one retained model in each of the three sets. In total 4556 pair-wise comparisons of the remaining yeast 3-D models returned non-zero results.

Based on the pair-wise comparisons we extracted three-way "MaxSub-supported substructures", *i.e.* the maximum overlap between all pair-wise matched regions for the same protein sequence. In the derivation of these substructures (illustrated schematically in Figure 5) we chose to ignore the gaps of up to 35 consecutive amino acids that were found sometimes within the regions matched in the MaxSub comparisons between two 3-D models. Such gaps were caused either by strong local deviation between the two models or missing residues in one of the models. MaxSub-supported substructures encompassing fragments of less than 45 amino acids in length were discarded to keep the number of structurally uninteresting matches (for example over only a single α-helix) as small as possible.

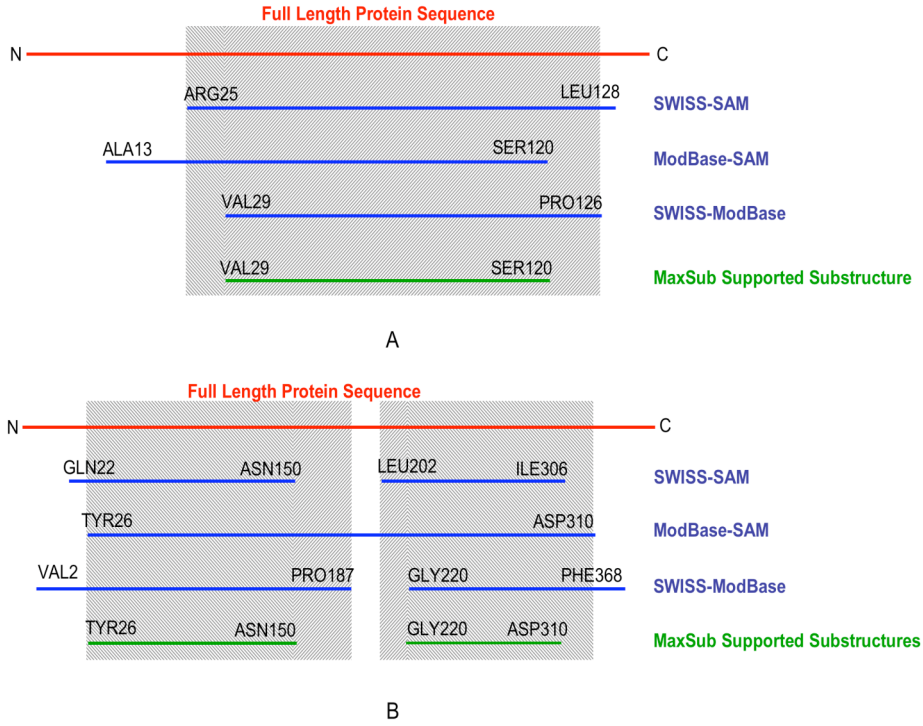

Figure 5. Schematic illustrating the derivation of three-way MaxSub-supported substructures. Two examples are shown, a single-domain protein (A) and a multi-domain protein (B). Red lines represent full-length protein sequences. Blue lines represent the pair-wise matched regions between 3-D models for these proteins. Green lines represent the resulting MaxSub-supported substructures.

2.3.5 Results

The detailed results of this experiment are publicly accessible via our WWW database CYSP (Comparison of yeast 3-D structure predictions, linked from `www.openk.org`). The records currently relate to the yeast proteins for which at least one model was retained in each of our sets of SWISS, ModBase, and SAM models after pre-filtering. Information is given regarding their associated 3-D model coordinates as they can be obtained from the three repositories, which regions match pair-wise between 3-D models of the same protein by different methods according to MaxSub comparison, and the Mscores attained by the matches. For proteins where three-way agreement between the methods was found, 3-D coordinates are also provided for the model fragment spanning their Max-Sub supported substructures. To the non-

computational biologist looking to find an approximate 3-D structural representation of his/her yeast protein of interest, the model fragments in this new, filtered, resource are likely to be the most relevant. While there is of course no guarantee (since there always is a chance that all three methods could have erred in the same way) they would be deemed "likely correct by consensus". This philosophy is applied widely in other areas of protein structure prediction as well, for example in secondary structure prediction, and its viability is generally supported by independently derived experimental structural information [21–23]. Attributing greater confidence to consensus predictions is certainly considered appropriate where the methods consulted are different as this was the case here.

A previous similar study by the Baker group at Washington University St. Louis compared fold predictions for yeast proteins between different fold prediction methods [24]. By contrast to our comparisons Dolinsky et al. did not carry out model superpositions but designed their SPrCY database (`agave.wustl.edu/yeast/`) for consistency checking at the template structure/fold level. Given the lower structural accuracy in general that is attained by models based on fold prediction methods (which aim primarily to identify fold resemblance to known structures in cases where no sequence similarity is detectable, and where producing a detailed model is often too difficult a problem to tackle) this is certainly justified, although it makes it impossible to directly compare our results with theirs.

We obtained 578 MaxSub-supported substructures for 545 yeast protein sequences with non-identical YIDs in this experiment. Fragments of 3-D models are most informative to the users if they span entire structural domains, or at least 3-D structurally separable parts. While some few domains are known to include less than 45 amino acids, and domain lengths spread widely, short fragments should be considered more at risk of being structurally uninformative than long fragments. As the length distribution of MaxSub-supported substructures shows (Figure 6) we would retain 136 (23%) of the corresponding model fragments even if a minimum length of 90 amino acids were imposed, rather than the 45 amino acid cut-off we chose. Thus it seems likely that the majority of the model fragments in CYSP would be useful for investigating the local structure of the yeast proteins they represent. This notion was confirmed through visual inspection of the models and is illustrated by the three examples presented in more detail below.

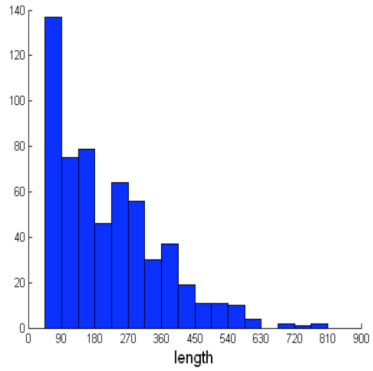

Figure 6. Length distribution of MaxSub-supported substructures.

	SWISS	ModBase	SAM
SWISS	769 (717)	649 (594)	585 (559)
ModBase		2546 (2280)	620 (594)
SAM			2211 (2211)

Table 1. Number of pair-wise matched regions between models from the three data sources (ignoring gaps). The numbers of represented yeast proteins is given in parentheses. The total number of models and proteins in each set that were considered is apparent in the diagonal. Note that comparisons were only performed for YIDs represented by at least one retained model in each filtered set.

The number of models yielding pair-wise matches is shown in Table 1. The number of matched models between SWISS and SAM is very similar to the number of three-way MaxSub-supported substructures. At first glance this coincidence may appear to reflect that only one of the SWISS-SAM matched regions is not also supported by ModBase. However, this is not necessarily true as multidomain proteins can give rise to different numbers of pair-wise matched regions depending on which methods are compared (as is illustrated in Figure 5B). Indeed inspection of the results reveals that the relation between the pair-wise and three-way supported regions is not straightforward. Comparing the numbers of represented yeast proteins should be more informative and we note that the 545 proteins represented by the three-way MaxSub-supported substructure set in CYSP make up make up 91.8% of those represented in SWISS-ModBase; 97.5% of those in SWISS-SAM; and 91.8% of those in ModBase-SAM. While these numbers do not differ dramatically for different method-pairs, the proportion is higher for SWISS-SAM is than for the others. If one assumed that all three-way supported substructures are correct, but that there are likely to be more correct predictions in the set than the ones supported by all three methods, this could be explained by SAM being a slightly weaker predictor than the other two in this experiment. If this were the case it could be useful to look at the SWISS-ModBase set for additional (possibly) correct models. Alternatively, if one assumed that the three-way supported predictions are the only ones that are correct then this difference would indicate that SAM is the most useful method for preventing false models (which would make most sense if ModBase and SWISS were very similar methods). Neither of these assumptions can be expected to be entirely accurate (no known method guarantees that three-way supported predictions are actually correct) and forthcoming laboratory-determined protein structures will only provide an evaluation of a small number of the predictions in the near future. Moreover it would not be appropriate to derive more than very cautious conclusions based on this survey data. However, given the fact that the models provided by the SAM data source are deemed to be at “unrefined” stage, it is plausible that the first possibility is closer to the truth than the second. Our implementation makes it straightforward to perform a repeat experiment at a later stage of SAM-model refinement and/or to consult additional data sources in the future.

To illustrate the results accessible at CYSP we have selected three examples: YPL132W, YBR024W, and YLR132C. The 3-D models of YPL132W in SWISS, ModBase, and SAM were all generated based on the same template structure, 1SO9A. By contrast, different template structures were used by the each of the three model sources to model YBR024W and YLR131C (Table 2).

YID	**Protein Name**	**Template (E-value)**		
		SWISS	ModBase	SAM
YPL132W	COX11_YEAST	1SO9A (4.5E-75)	1SO9A (3E-53)	1SO9A (4.0E-22)
YBR024W	SCO2_YEAST	2B7JB(7.9E-76)	1ON4A(4E-20)	1WP0A (2.8E-27)
YLR131C	ACE2_YEAST	1NCS (4.2E-17)	1UN6B (2E-15)	2GLIA (4.7E-27)

Table 2. Three examples of proteins included in CYSP, specified by their YIDs and SWISS-PROT identifiers, with the template structures used and the E-values the three data sources attributed to their structure predictions.

Pair-wise comparisons between 3-D models of YPL131W generated by SWISS, ModBase and SAM indicate that, with exception of some missing residues, the models are in perfect agreement throughout (Table 3 and Figure 9). By contrast, the three data sources only agree on the core regions of the structures predicted for YBR024W (the blue regions), and disagree otherwise (the green and red regions). Finally the three data sources disagree almost entirely

on YLR131C, except over a short a-helical region (the blue regions). In this example the MaxSub-supported substructure would not be considered informative. Since it is shorter than 45 amino it was discarded and is not included in the results accessible through CYSP.

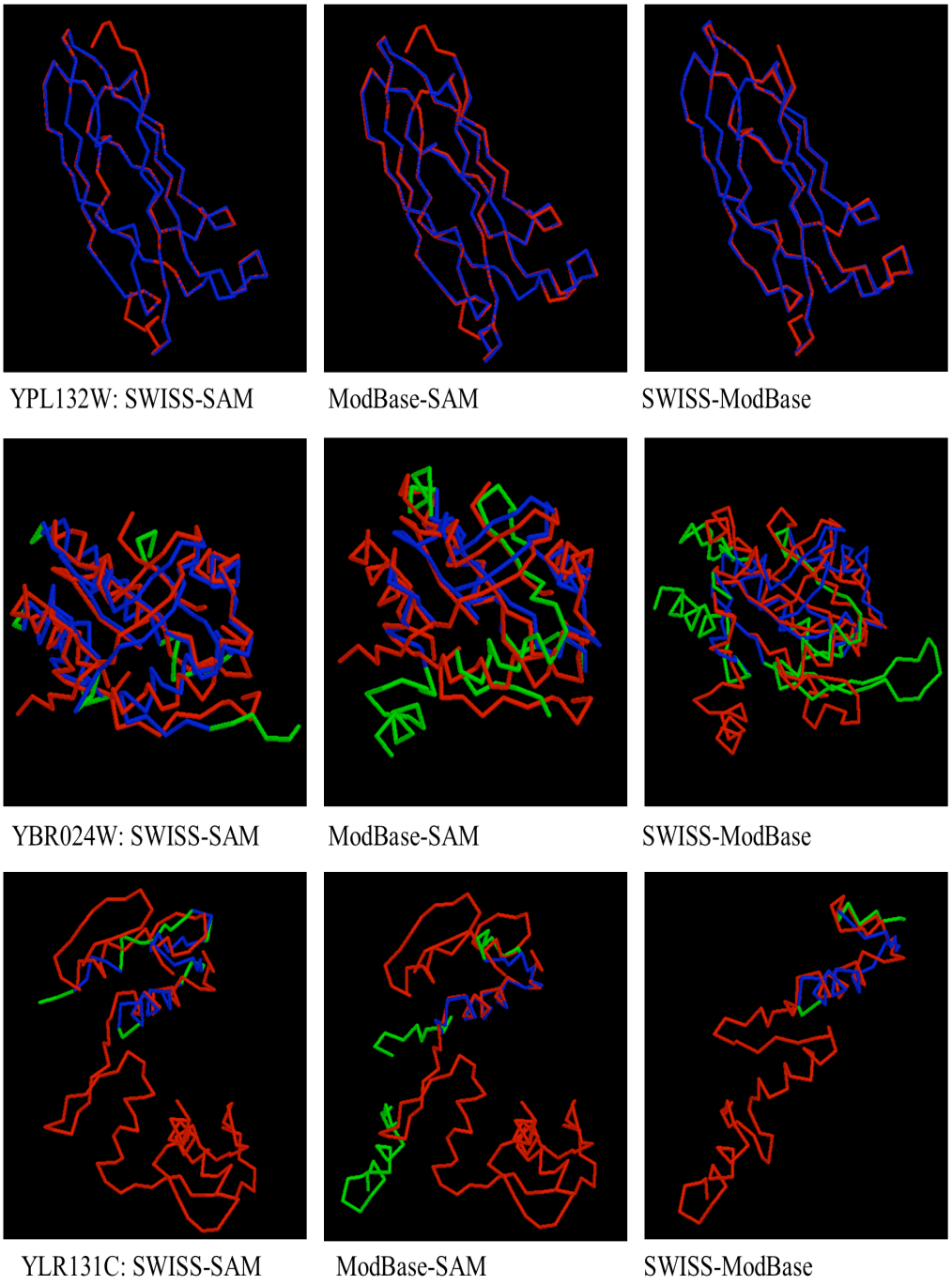

Figure 7. Backbone representations of the MaxSub results for three proteins: YPL135W, YBR024WC, and YLR131C. For each pair-wise comparison between the structures of Model 1 and Model 2 (in the order MaxSub read these models), substructures in blue are the regions in Model 1 that superimpose well onto Model 2, while the other parts of the models are shown in green (Model 1) and red (Model 2).

YID	Model 1	Model 2	Missing Residues	Pair-wise matched regions	MScore	MaxSub - supported region
YPL132W	SWISS	SAM	2	GLU138-GLU170 ALA172-GLY219 GLU221-PHE253	0.982	GLU138-PHE253
	ModBase	SAM	2	VAL135-GLU170, ALA172-GLY219, GLU221-ALA255	0.979	
	SWISS	ModBase	0	GLU138-PHE253	0.995	
YBR024W	SWISS	SAM	44	ALA120-ALA120, GLY122-PHE125, LEU127-LYS143, SER145-TYR148, SER152-HIS153, GLU160-GLU160, LEU162-ARG164, THR166-LYS175, HIS177-ILE178, ILE180-ASP203, ILE208-SER230, TYR250-GLY263, TYR266-ARG276, GLN278-ILE279.	0.660	PRO124-GLY263
	ModBase	SAM	26	PRO124-PHE125, LEU127-LYS143, SER145-TYR148, SER152-CYS154, GLU160-GLU160, LEU162-ARG164, THR166-ASP173, ILE180-ASP203, HIS205-HIS205, TYR250-GLY263.	0.459	
	SWISS	ModBase	22	PRO124-LYS141, LYS143-HIS153, PRO155-PRO155, GLU160-SER170, GLN181-PHE204, PRO206-PRO206, PHE248-PRO254, GLY256-LEU262, ARG264-ARG264.	0.400	
YLR131C	SWISS	SAM	0	PRO587-ILE591, VAL595-VAL595, PRO599-ASP600, LEU602-PHE614, ARG617-GLN629	0.555	(PRO606-GLN629)
	ModBase	SAM	25	LEU606-LEU606, CYS610-ASN611, PHE614-PHE614, ASN619-PHE637	0.000	
	SWISS	ModBase	10	LEU598-LEU598, CYS610-ARG616, TYR618-GLN629	0.000	

Table 3. This table lists information extracted from the pair-wise structural comparison results by MaxSub. The missing residue column reports how many residues in Model 1 are not found in Model 2. Reported as well-matched regions are all strictly continuous sequence fragments over which Model 1 coincided well with Model 2 after 3-D structural sequence-dependent superposition.

Some validation of this experiment will become possible in the future as additional structures of yeast proteins are determined in the laboratory. These newly emerging structures will allow an assessment of the accuracy of our models at least in a few cases. It will be interesting to verify, at least qualitatively, the assumption that the approach we applied extracted more accurate substructures than the models that were less well supported. To this end we are keeping track of new yeast protein structures in the PDB, and the accuracy of the corresponding models in CYSP, on www.openk.org.

2.4 Conclusions and Future Work

In the experiment of Section 2.3 we grounded the experimental protocol of Figure 1 in a

specific set of services, using the MagentA system as an interpreter for the protocol. It was necessary to route all the data and analysis services through MagentA (in the way described in Figure 2) because none of the original services was equipped to interpret the protocol. We therefore incurred a small one-off cost in enabling (via WSDL and HTML) the original services to communicate with MagentA. Having done this, however, we are able to use MagentA as a proxy for the original services for any LCC protocol, so that experimenters with different ideas about how best to coordinate these (and other suitably enabled) services can implement those by altering only the protocol. Notice also that the LCC protocol is separable from the mechanisms used to interpret it, and is shareable between peers during an interaction, so we can choose whether we want a single MagentA proxy for a group of services or a separate proxy for each service (giving a more or less finely grained peer-to-peer structure). Since MagentA is capable of interpreting any LCC protocol, we can in future add to the repertoire of protocols and thus extend the capabilities of the peer-to-peer system. For example, it is straightforward to write a LCC protocol for sharing filtered data between peers. It is also straightforward to write a LCC protocol that allows queries about specific types of protein structure to be routed between peers, thus allowing networks of peers to collate, filter and propagate results. The aim of this, ultimately, is to provide a peer network that, through sharing, can produce more confident predictions faster by sharing the analyses performed earlier by others. Opportunities for applying similar consistency-checking, and data sharing, strategies are found in many areas of bioinformatics. The single experiment in this paper shows the immediate benefit of this on a small scale for a specific form of analysis but to make this effective for large peer groups, where trust and provenance are (among other issues) important to the coherence of peer groups. This, although outside the scope of the current paper, is one of the central themes of the OpenKnowledge project (`www.openk.org`).

3 Scenario II: Peer-validated protein identification in proteomics research

3.1. Biological Background

Proteomics studies the quantitative changes occurring in a proteome (which is the protein equivalent of a genome) and its application for disease diagnostics and therapy and for drug development. We shall focus here on the initial step of protein analysis, called expression proteomics. During this step proteins are extracted from the cells and tissues, are separated either by two dimensional gel electrophoresis (2DE) or liquid chromatography (LC) techniques, and further digested, identified and sequenced by mass spectrometry (MS) methods. These techniques take advantage of the current knowledge of the genome from humans and other species, which is available in public databases and can be accessed through data-mining software that relates MS spectrometric information with database sequences. Protein sequence data held in databases, however, is mostly produced from the direct translation of gene sequences. But protein activity is determined by maturation events that include so called pre- and post-translational modifications of their structure. The importance of these modifications is so high that gene and protein expression in eukaryotes show no correlation in many cases.

Currently, however, technology allows the high throughput sequencing of proteomes using techniques such as multidimensional liquid chromatography coupled with tandem mass spectrometry (MDLC-MS/MS), which not only offer information on the proteins present in the proteome but also on their sequence (that can differ from the one in the translated databases), and type and position of their modifications. Unfortunately, MDLC-MS/MS proteomic analysis is currently an impossible task for humans to achieve. It produces a huge amount of spectra, each yielding several peptide or peptide tags candidates that can belong to the same or different proteins. Each step produces an identification score whose final evaluation (of hundreds of spectra) is performed manually or by taking high probability data.

The speed of production of this type of information is increasing very fast as a good number of proteomic laboratories being involved in the characterization of proteomes, protein complexes and networks using these strategies. In the first instance sequence tags are compared to

sequence information in centralised databases storing predicted protein and "expressed sequence tags" (EST) information. This is generally helpful when exactly, or nearly-exactly, identical protein sequences are found to be part of previously identified proteins. However, many factors complicate these searches, for instance unknown degrees of post-translational modification. In addition, success of peptide and protein identification depends on database and file quality, database errors in sequence annotations, post-translational modifications, protein mixtures, etc.

Currently, sequencing information from other laboratories, especially those archives that do not produce clear identifications with the tools available to the source laboratory at a given moment, is rarely accessible to other groups involved in similar tasks and most of it will never be reflected in protein database annotation. The most probable scenario is that this information is eventually trashed. This information, however, could be of high importance for other groups analysing the sequence/function of this or other homologous proteins. Modification information and sequence tags generated in one laboratory could be used by other laboratories, to evaluate the confidence of experimental (or predicted) sequences derived from their work (in the same or other species). Trying to get as much good quality precision and recall of hits in public databases *and* the data from other laboratories as possible would be of great benefit because such hits could increase the confidence in the identification of the proteins of the analysed sample.

3.2 Peer-to-Peer Proteomics

We envision, therefore, a scenario in which various proteomic laboratories join a peer-to-peer network into which they feed the sequencing information generated locally at their respective labs so that other proteomic laboratories of the network can look for sequencing information in those files that proteomics laboratories deemed "useless", because they did not yield the information they required for their own particular proteomic analysis. This is particularly so with the mass spectra themselves, as no mass spectrum database is currently available, and spectra whose sequences do not give hits in a database search are trashed. The Open Proteomics Database (bioinformatics.icmb.utexas.edu/OPD) attempts to favour the open sharing of mass spectra, but it still relies on centralised repositories [25].

Figure 8 defines an interaction model for our envisioned scenario. It follows initially a similar kind of protocol as that for the consistency checking in yeast protein structure prediction shown in Figure 1. In this scenario a protein identifier searches several search engines it is aware of for hits with a significant score. The particularity in this scenario is that, for the validation, the validator checks the sequences it considers of low score with additional data of several peer proteomic labs it knows, and which have put their experimental data files with MS and sequence tags available to other peers for cross-checking.

3.3 Semantic Heterogeneity

The level of semantic heterogeneity in peer-to-peer information sharing in expression proteomics is currently not at the level of sequence tags or mass spectra. However, there are a high number of proteomics technologies, each of which has developed several approaches and analytical conditions. Although proteomics facilities in Spain, for instance, are using up to seven different types of mass spectrometers producing their own file formats and using different procedures for raw data management, standard procedures for mass spectra interchange have already been proposed, such as mzXML [26] or HUP-ML[27]. Semantic heterogeneity arises at the annotation level of mass spectra and sequence tags.

```
a(protein_identifier,I) ::
  identify(MS) <= a(researcher,R) then
  a(searcher(MS,Es,Ps,Ds),I) <- mass_spectrum(MS) and
                                search_engines_and_parameters(Es,Ps) then
  identification(D) => a(researcher,R) <- combine_data(Ds,D)

a(searcher(MS,Es,Ps,Ds),I) ::
  ( null <- Es = [] and Ps = [] and Ds = [] or
    ( a(inquirer(MS,E,P,D),I) <- Es = [E|REs] and Ps = [P|RPs] and Ds = [D|RDs] then
      a(searcher(MS,REs,RPs,RDs),I) ) )

a(inquirer(MS,E,P,D),I) ::
  query(MS,P) => a(search_engine,E) then
  answer(Da) <= a(search_engine,E) then
  a(validator(MS,E,P,Da,D),I)

a(search_egine,E) ::
  query(MS,P) <= a(inquirer,I) then
  answer(D) => a(inquirer,I) <- search(MS,P,D)

a(validator(MS,E,P,Da,D),I) ::
  ( null <- significant_scores(MS,E,P,Da,D) and D=Da or
    ( a(peer_validator(MS,Ls,Da,Ds),I) <- not significant_scores(MS,E,P,Da,D) and
                                          peer_labs(Ls) then
      null <- combine_validations(Da,Ds,D) ) )

a(peer_validator(MS,Ls,Da,Ds),I) ::
  ( null <- Ls = [] and Ds = [] or
    ( a(checker(MS,L,Da,D),I) <- Ls = [L|RLs] and Ds = [D|RDs] then
      a(peer_validator(MS,RLs,Da,RDs),I) ) )

a(checker(MS,L,Da,D),I) ::
  check_sequence(MS,Da) => a(proteomics_lab,L) then
  answer(D) <= a(proteomics_lab,L)

a(proteomics_lab,L) ::
  check_sequence(Ms,Da) <= a(checker(MS,L,Da,D),V)
  answer(D) => a(checker(MS,L,Da,D),V) <- find_hit(MS,Da,D)
```

Figure 8. LCC interaction model for peer-validated protein identification.

Annotation information ranges from the identification of the organism, cell, organelle or body fluid from which the analysed sample was extracted, to the name of the identified peptide/protein of a mass spectrum or sequence tag. Although most of this annotation will usually not yield semantic mismatches between proteomics laboratories, it may nevertheless be the case that such semantic mismatches need to be addressed. This is particularly the case with protein names, due to the variability of terms used for identical sequences. For example, the protein lymphocyte associated receptor of death has several synonyms including LARD, Apo3, DR3, TRAMP, wsl, and TnfRSF12. Researchers often use different names to refer to the same protein across sub-domains. Semantic matching techniques will need to dynamically resort to external sources, such as scientific publications in which protein equivalences have been identified, in order to overcome this sort of semantic mismatches. They will also need to be capable of disambiguating homonyms, i.e., two or more protein names spelled alike but different in meaning [28].

Interestingly, in protein identification matching of sequence annotation may be approximate and partial and still be valuable for the task: a sequence tag taken from a human tissue, for instance, matching that of a protein coming form the sample of a rat's kidney still may provide high confidence measure to the identification task as both organism are mammals.

Acknowledgments. This work is supported under the OpenKnowledge Specific Targeted Research Project (STREP), which is funded by the European Commission under contract number FP6-027253. The OpenKnowledge STREP comprises the Universities of Edinburgh, Southampton, and Trento, the Open University, the Free University of Amsterdam, and the Spanish National Research Council (CSIC).

References

1. T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.
2. D. Robertson. Multi-agent coordination as distributed logic programming. In *International Conference on Logic Programming*, Sant-Malo, France, 2004.
3. C. Walton and A. Barker. An Agent-based e-Science Experiment Builder. In *Proceedings of the 1st International Workshop on Semantic Intelligent Middleware for the Web and the Grid*, Valencia, Spain, August 2004.
4. A. Barker and R. Mann. Integration of MultiAgent Systems to AstroGrid. In *Proceedings of the Astronomical Data Analysis Software and Systems XV, European Space Astronomy Centre*, San Lorenzo de El Escorial, Spain, October 2005.
5. C. Walton. Typed Protocols for Peer-to-Peer Service Composition. In *Proceedings of the 2nd International Workshop on Peer to Peer Knowledge Management (P2PKM 2005)*, San-Diego, USA, July 2005.
6. C. Walton. Protocols for Web Service Invocation. In *Proceedings of the AAAI Fall Symposium on Agents and the Semantic Web (ASW05)*, Virginia, USA, November 2005.
7. J. Moult. A Decade of CASP: Progress, Bottlenecks and Prognosis in Protein Structure Prediction. *Curr Opin Struct Biol,* 15(3):285–289, 2005.
8. D. Fischer, L. Rychlewski, R.L. Dunbrack, A.R. Ortiz, and A. Elofsson. CAFASP3: the Third Critical Assessment of Fully Automated Structure Prediction Methods. *Proteins*, 53(6):503–516, 2003.
9. E. Krieger, S.B. Nabuurs, and G. Vriend. Homology Modeling. *Methods Biochem Anal*, 44:509–523, 2003.
10. M. Tress, I. Ezkurdia, O. Grana, G. Lopez, and A. Valencia. Assessment of Predictions Submitted for the CASP6 Comparative Modeling Category. *Proteins*, 61(7):27–45, 2005.
11. A. Goffeau, B.G. Barrell, H. Bussey, R.W. Davis, B. Dujon, H. Feldmann, F. Galibert, J.D. Hoheisel, C. Jacq, and M. Johnston. Life with 6000 Genes. *Science*, 274(5287):563–547, 1996.
12. T. Schwede, J. Kopp, N. Guex, and M.C. Peitsch. SWISS-MODEL: An Automated Protein Homology-modeling Server. *Nucleic Acids Res*, 31(13):3381–3385, 2003.
13. A. Sali and T.L. Blundell. Comparative Protein Modelling by Satisfaction of Spatial Restraints. *J Mol Biol*, 234(3):779–815, 1993.
14. K. Karplus, R. Karchin, J. Draper, J. Casper, Y. Mandel-Gutfreund, M. Diekhans, and R. Hughey. Combining Local-structure, Fold-recognition, and New Fold Methods for Protein Structure Prediction. *Proteins*, 53(6):491–496, 2003.
15. S. Weng, Q. Dong, R. Balakrishnan, K. Christie, M. Costanzo, K. Dolinski, S.S. Dwight, S. Engel, D.G. Fisk DG, and E. Hong. Saccharomyces Genome Database (SGD) Provides Biochemical and Structural Information for Budding Yeast Proteins. *Nucleic Acids Res*, 31(1):216–218, 2003.
16. J. Kopp and T. Schwede. The SWISS-MODEL Repository: New Features and Functionalities. *Nucleic Acids Res*, 34:315–318, 2006.
17. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1):235–242, 2000.
18. U. Pieper, N. Eswar, H. Braberg, M.S. Madhusudhan, F.P. Davis, A.C. Stuart, N. Mirkovic, A. Rossi, M.A. Marti-Renom, and A. Fiser. MODBASE, a Database of Annotated Comparative Protein Structure Models, and Associated Resources. *Nucleic Acids Res*, 32:217–222, 2004.
19. S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a New Generation of Protein Database Search Programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.
20. N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer. MaxSub: an Automated Measure for the Assessment of Protein Structure Prediction Quality. *Bioinformatics*, 16(9):776–785, 2000.
21. D. Fischer. Servers for Protein Structure Prediction. *Curr Opin Struct Biol*, 16(2):178–182, 2006.
22. R.L. Dunbrack. Sequence Comparison and Protein Structure Prediction. *Curr Opin Struct Biol*, 16(3):374–384, 2006.

23. J. Heringa. Computational Methods for Protein Secondary Structure Prediction Using Multiple Sequence Alignments. *Curr Protein Pept Sci*, 1(3):271–301, 2000.
24. T.J. Dolinsky, P.M. Burgers, K. Karplus, and N.A. Baker. SPrCY: Comparison of Structural Predictions in the Saccharomyces cerevisiae Genome. *Bioinformatics*, 20(14):2312–2314, 2004.
25. John T. Prince, Mark W. Carlson, Rong Wang, Peng Lu, and Edward M Marcotte. The need for a public proteomics repository. *Nature Biotechnology*, 22(4):471–472, 2004.
26. P. G. Pedrioli et al. A common open representation of mass spectrometry data and its application to proteomics research. *Nature Biotechnology*, 22:1459–1466, 2004.
27. K. Kamijo et al. HUP-ML: Human proteome markup language for proteomics database. *Journal of the Mass Spectrometry Society of Japan*, 51(5):542–549, 2003.
28. H. Yu and E. Agichtein. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19:340–349, 2003.