Protein Classification Workflow

Judith Reso

Potsdam University, Potsdam, D-14482, Germany
reso@uni-potsdam.de

Abstract. The protein classification workflow described in this report enables users to get information about a novel protein sequence automatically. The information is derived by different bioinformatic analysis tools which calculate or predict features of a protein sequence. Also, databases are used to compare the novel sequence with known proteins.

Keywords: bioinformatics, sequence analysis, amino acid properties, homology modeling, protein structures, protein structure prediction, clustering.

1 Introduction: Workflow Scenario

Imagine you have a cell culture with some features you did not observe before. You want to know which proteins the cells contain because there might be something special. After cleaning up the cells and extracting proteins you need to get their sequences. This will be done by an external company and you get back the protein sequences as amino acid sequences. You query them against a protein database, for example PDB [4], and recognize that there is no entry. Congratulations! You might have discovered a new protein. This is the point where the workflow starts.

One would create pairwise alignments which means that the sequence is queried against known sequences stored in databases to find similar structures by comparing the sequence of amino acids. Structure similarity could give hints about proteins sharing a common ancestor (homologs) or having similar functions due to divergent evolution. Another possibility is to calculate amino acid propensities. Amino acid propensities are for example hydropathy, charge and polarity. These features are essential for protein folding. A protein is not only a linear strand of continuous amino acids. Depending on their features they interact to each other which means that bridges could be build between residues by e.g. hydrogen and sulfide bonds, Van-der-Waals forces and electrostatic interactions. There are several tools available which use those properties to predict the fold of a protein structure. This is called the secondary structure.

The secondary structures could be also aligned to known structures to determine the family the protein might belong to. Aligning structures means that it is checked whether there are the same motifs (structural elements with a specific pattern) in the same order and of the same length. There exist databases like SCOP [5] and CATH [1] which separate proteins depending on their composition

A.-L. Lamprecht et al. (Eds.): Process Design for Natural Scientists, CCIS 500, pp. 65–72, 2014.
DOI: 10.1007/978-3-662-45006-2_5

of motifs and domains (helices, β-sheets, loops) into families which could also help to get a functional annotation.

Interactions also occur between secondary structure elements which builds the tertiary and quaternary structure. These structures are important for the function of the protein and determine its catalytic effects and binding to other molecules. Multiple sequence alignments could be generated to cluster a set of more similar structures corresponding to the protein. This results in a phylogenetic tree which is similar to a dendrogram. Because we know the species where the protein is derived from this phylogeny could give hints about the evolutionary context of the protein.

The jABC workflow created in this project takes this novel amino acid sequence as FASTA format (which is a typical text like format in bioinformatics to describe structures) and will search and generate some of the introduced features automatically using web services from EBI [2] and PFAM [3]. The process includes calculation of amino acid propensities, browsing databases for homologous sequences by calculation of alignments and prediction of secondary structure. Domains which are important for protein function will be extracted by comparison with the PFAM-database and a phylogenetic tree will be computed also. The task of the workflow is to give a coarse overview about the novel sequence by automatically calling the preselected tools. This overview can give suggestions for further research on the protein itself and also the workflow could be extended for other functions.

2 Service Analysis

To realize the workflow services were needed which are able to do the introduced tasks. Because this workflow is about the analysis of a novel protein sequence, such a sequence has to be provided initially. Therefore one could create amino acids manually by using the basic SIB 'PutString' and converting it into a FASTA file. Another way is to use 'Makeprotseq' which is a web service provided by the EBI [2]. These platform interacts with several bioinformatic databases and gives access to web tools without needing any licenses. To make these web tools usable within jABC, they have been implemented as SIBs [15]. To work with those SIBs a connection to the internet is required but nothing else needs to be installed.

After getting the random sequence by 'Makeprotseq', it has to be analyzed. 'Pepinfo' is a service which enables to calculate and plot amino acid propensities. The services 'WuBlast' and 'PsiSearch' use different alignment algorithms to search for similar sequences in a specified database, e.g. PDB (Protein Data Base). 'WuBlast' is optimized for queries with novel structures as input on databases which are specially formatted. In addition 'PsiSearch' is using an iterative alignment algorithm called Psi-Blast which computes multiple local alignments between proteins. The secondary structure of the novel protein is predicted using 'Garnier' which uses the GOR-Algorithm (for further information see [8]). All of the previous services are derived from the platform EMBL-EBI.

To get an idea about the functional regions of the sequence the service 'SequenceSearch' from the protein family database PFAM [3] is used. This is also

a tool which uses multiple alignments to assign a function to a sequence by comparing it to other known sequences and detecting similar motifs and domains between the structures. The results are evaluated using Hidden Markov Models (HMMs).

3 Workflow Realization

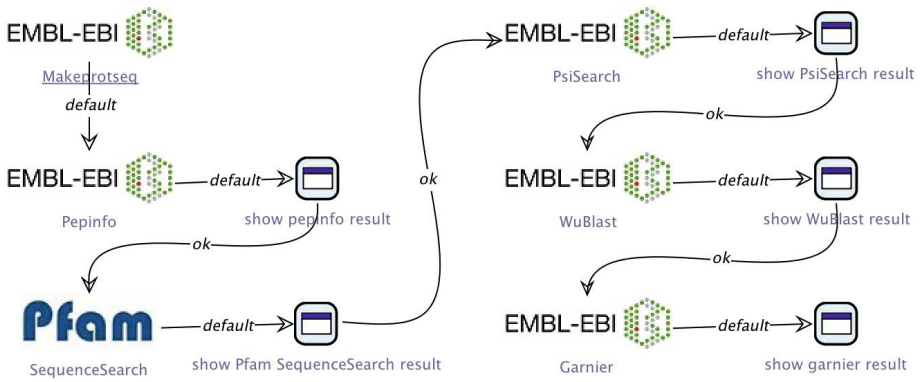

Fig. 1. Protein classification workflow

Figure 1 shows a possible workflow for protein classification based on the services described in the previous section. The SIB where the workflow begins is the 'Makeprotseq'-SIB. This one is declared as 'Start'-SIB. There one can specify how long the novel protein sequence should be. For this workflow, 150 amino acids is set with the parameter 'Length'. The parameter 'Amount' specifies how many sequences should be created and is set to 1. The output of this SIB, i.e. the created sequence, is called 'sequence' and can be used as input by the following services in their ${sequence} parameters.

To get information about amino acid propensities 'PepInfo' takes the created sequence as input and calculates the propensities. This tool requires an email address, the parameter 'Sequence' is used to choose the input sequence (${sequence}) a 'Title' can be specified and the parameter 'result' is used to specify the name of the local variable resulting as output. This result is then shown to the user in a pop-up text dialog using the 'ShowTextDialog' SIB.

The next step uses a Pfam service called 'SequenceSearch'. This SIB takes as input the sequence created by 'Makeprotseq'. The sequence is then queried against the entries in the protein family database 'PFAM' for motifs and domains having a specific structure or function. The return specified with the parameter 'results' consists of Pfam identifiers for domains which are similar to those included in the novel sequence. This result is again simply shown to the user by the 'ShowTextDialog' SIB.

Multiple structural alignments are then calculated with the services 'WuBlast' and 'PsiSearch'. As described before, both tools use different methods so the results of the alignments don't have to be identical. Both tools take the sequence derived from 'Makeprotseq' as input for the parameter 'Sequence'. A further parameter which needs to be specified is the 'Email'-address and the parameter 'Database' which is specifying the database to query. This parameter is set to 'pdb' to search trough the protein data base PDB. Both tools have the option to specify the number of alignments in the output. Here the default is used. 'WuBlast' also allows to specify the alignment output format with the parameter 'align'. This can be useful for further usage of MSAs. 'WuBlast' requires the specification of the alignment program with the parameter 'program'. It is set to blastp because a protein has to be aligned. The parameter 'Stype' must be set to protein and determines the type of sequence. All other parameters are optional and refer to advanced settings. They don't need to be specified. The results will be given as database(pdb)-identifiers, xml-file and output of the tool itself. They all get the prefix 'wublast_' to distinguish them in the local context from the other outputs. The tool 'PsiSearch' requires no further specification of parameters because all the others are optional. The name given to 'resultJobID' is specifying the name of the local variable for further usage and called 'Psi_res'. As for the previous steps, also the results returned by these services are simply displayed by the 'ShowTextDialog' SIB.

Finally, to predict the secondary structure of a protein the web service 'Garnier' is used. It takes as input parameter 'Sequence' the local variable of the sequence and the result which could be named with the parameter 'results' is called 'secondary'. The output of this tool is text like containing the amino acid structure itself and the predicted structural elements at the corresponding position as annotated letters which are repeated as long as a part of the sequence is assumed to build this structure, e.g. a helix will be annotated by 'H'. Also this result is simply shown to the user by the 'ShowTextDialog' SIB.

4 Conclusion

The previous section described a workflow where a sequence is randomly created and used to calculate amino acid propensities, multiple sequence alignments and predict protein secondary structures. Built from SIBs which access publicly available web services, it is fully executable and can be used "as is" as long as only a randomly created sequence is used to initialize the process. If one wants to read in a real novel sequence the SIB 'Makeprotseq' has to be replaced by a SIB which enables to choose a file or the SIB 'PutString' is used as Start-SIB which allows to Copy-Paste a sequence manually. Then the next SIB must be a SIB which converts this string into FASTA-format for further usage by the analysis tools. Similarly, if optional parameters, e.g. for advanced alignments should be set, the corresponding parameters of the SIBs needs to be changed accordingly.

In the present version of the workflow all the web services are simply called one after another, and the (default) results are simply visualized by a pop-up text

message. To exploit the distributed nature of the remote services, the workflow could execute in parallel all SIBs which take the same initial protein sequence as input. Instead of just showing the different results in a message box once during execution, they could also written into files and stored on the hard disk for future use. Similarly, the workflow could create a log file where the tools could write their errors for easier traceback.

With regard to further analyses, this workflow could be extended for tertiary structure prediction and structure alignment to check whether there are functional similar proteins evolved by different species which might not be not or only far related to this species. This could illustrate the evolutional context. This tertiary structure could be also checked for interactions with other molecules. There are several tools to predict for example ligand binding.

The workflow could also be extended to automatically check the consistency of developed tools. In bioinformatics it is a big problem to evaluate results biologically correct. This is also shown with this first implementation: A randomly built sequence is used as input for all the web services and results are outputted. One should assume that a non-biological sequence should give no results. But at the moment most of the implemented algorithms are not able to distinguish between biological correct and 'only' mathematically correct. So those workflows could be used to easier optimize services by checking them more automatically.

Also one could implement a SIB or workflow which enables to provide a real novel protein sequence automatically to a database. But then one should ensure that this sequence is really a new one and not only a mutation of a known protein. Therefore one has to check how similar the results of a sequence alignment are.

Regarding the technical details of the workflow realization, using the EBI web services was often difficult for me, as often the documentation of the web services is not clear. Often I tried to find out the valid inputs for the parameters of the EBI SIBs, but the website with the web services of EBI-EMBL [2] is not documented very well. There only the services in SoapLab documented valid values for input (e.g. how to specify the name of the database to query, namely PDB in lower-case letters, pdb).

In the beginning it was also planned to query the input sequence against the SCOP database to classify the protein also by a database tool which is optimized to classify proteins depending on their structure within one step and to annotate the family where the protein sequence could belong to. This was not possible because the SIB 'Scopparse' requires a raw scop file for classification, which was not available in the present setting. A similar idea was to compare the predicted secondary structure to other secondary structures to detect the higher conserved/functional regions of the secondary structure which could also give hints to the protein function. Therefore the SIB 'Ssematch' had been implemented, but could not be used because it requires a DCF-file as input, which was not available.

To calculate a multiple sequence alignment using ClustalW which might be usable by the 'ClustalW2Phylogeny'-SIB the tool 'Emma' should be used in the beginning. But this web service only runs on a multiple input of sequences. It is

not querying a sequence against a database. So if there is only one sequence the return value of 'Emma' is empty. I assume that the 'ClustalW2Phylogeny' might need the result of 'Emma' or another multiple sequence alignment as an input. First I thought that a multiple alignments would be computed by PsiSearch, which could then be used to calculate a phylogenetic tree by clustering the sequences depending on their similarity with the service 'ClustalW2Phylogeny'. However, this is apparently not the case, and so the results of 'PsiSearch' can not be processed by this service.

This article is part of a larger evaluation [10], which aimed at illustrating the power of simplicity-oriented development [20] by validating the claim that process modeling can indeed be handed over to the domain experts by providing them with a graphical modeling framework [26] that covers low-level details in a service-oriented fashion [22], integrates high-level modeling in the overall development process in a way that user-level models become directly executable [21,18], and supports ad-hoc adaptations and evolution [17,19].

The project described in this article can be characterized as follows:

- Scientific domain: bioinformatics
- Number of models: 1
- Number of hierarchy levels: 1
- Total number of SIBs: 11
- SIB libraries used (cf. [15]): common-sibs (5), ebi-sibs (5), pfam-sibs (1)
- Service technologies used: SOAP web services, REST web services

The bioinformatics part of this volume contains five other articles on workflow applications in this domain [6,16,25,24,27]. Further examples of workflow projects with he bioinformatics-specific incarnation of the jABC framework, called Bio-jETI [12], have been described, for example, in [11,13,7]. As shown in [12,14,9], bioinformatics is also a suitable field for the application of semantics-based (semi-) automatic workflow composition techniques (as provided by, e.g., [23]) to support the workflow design process.

References

1. CATH: Protein Structure Classification Database at UCL, `http://www.cathdb.info` (Online; last accessed December 10, 2012)
2. EBI Web Services, `http://www.ebi.ac.uk/Tools/webservices/` (Online; last accessed December 10, 2012)
3. Pfam: Home page, `http://pfam.sanger.ac.uk/` (Online; last accessed December 06, 2012)
4. RCSB Protein Data Bank - RCSB PDB, `http://www.rcsb.org/pdb/home/home.do` (Online; last accessed December 06, 2012)

5. SCOP: Structural Classification of Proteins, http://scop.mrc-lmb.cam.ac.uk/scop/ (Online; last accessed December 10, 2012)
6. Blaese, L.: Data Mining for Unidentified Protein Sequences. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 73–87. Springer, Heidelberg (2014)
7. Ebert, B.E., Lamprecht, A.-L., Steffen, B., Blank, L.M.: Flux-P: Automating Metabolic Flux Analysis. Metabolites 2(4), 872–890 (2012)
8. Garnier, J., Gibrat, J.-F., Robson, B.: GOR method for predicting protein secondary structure from amino acid sequence. In: Doolittle, R.F. (ed.) Computer Methods for Macromolecular Sequence Analysis. Methods in Enzymology, vol. 266, pp. 540–553. Academic Press (1996)
9. Lamprecht, A.-L. (ed.): User-Level Workflow Design. LNCS, vol. 8311. Springer, Heidelberg (2013)
10. Lamprecht, A.-L., Margaria, T.: Scientific Workflows and XMDD. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 1–13. Springer, Heidelberg (2014)
11. Lamprecht, A.-L., Margaria, T., Steffen, B.: Seven Variations of an Alignment Workflow - An Illustration of Agile Process Design and Management in Bio-jETI. In: Măndoiu, I., Wang, S.-L., Zelikovsky, A. (eds.) ISBRA 2008. LNCS (LNBI), vol. 4983, pp. 445–456. Springer, Heidelberg (2008)
12. Lamprecht, A.-L., Margaria, T., Steffen, B.: Bio-jETI: A framework for semantics-based service composition. BMC Bioinformatics 10(suppl. 10), S8 (2009)
13. Lamprecht, A.-L., Margaria, T., Steffen, B., Sczyrba, A., Hartmeier, S., Giegerich, R.: GeneFisher-P: variations of GeneFisher as processes in Bio-jETI. BMC Bioinformatics 9 (suppl. 4), S13 (2008)
14. Lamprecht, A.-L., Naujokat, S., Margaria, T., Steffen, B.: Semantics-based composition of EMBOSS services. Journal of Biomedical Semantics 2(suppl. 1), S5 (2011)
15. Lamprecht, A.-L., Wickert, A.: The Course's SIB Libraries. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 30–44. Springer, Heidelberg (2014)
16. Lis, M.: Constructing a Phylogenetic Tree. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 101–109. Springer, Heidelberg (2014)
17. Margaria, T., Steffen, B.: Agile IT: Thinking in User-Centric Models. In: Margaria, T., Steffen, B. (eds.) ISoLA 2008. CCIS, vol. 17, pp. 490–502. Springer, Heidelberg (2009)
18. Margaria, T., Steffen, B.: Business Process Modelling in the jABC: The One-Thing-Approach. In: Cardoso, J., van der Aalst, W. (eds.) Handbook of Research on Business Process Modeling. IGI Global (2009)
19. Margaria, T., Steffen, B.: Continuous Model-Driven Engineering. IEEE Computer 42(10), 106–109 (2009)
20. Margaria, T., Steffen, B.: Simplicity as a Driver for Agile Innovation. Computer 43(6), 90–92 (2010)
21. Margaria, T., Steffen, B.: Service-Orientation: Conquering Complexity with XMDD. In: Hinchey, M., Coyle, L. (eds.) Conquering Complexity, pp. 217–236. Springer, London (2012)
22. Margaria, T., Steffen, B., Reitenspieß, M.: Service-Oriented Design: The Roots. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 450–464. Springer, Heidelberg (2005)

23. Naujokat, S., Lamprecht, A.-L., Steffen, B.: Loose Programming with PROPHETS. In: de Lara, J., Zisman, A. (eds.) Fundamental Approaches to Software Engineering. LNCS, vol. 7212, pp. 94–98. Springer, Heidelberg (2012)
24. Schütt, C.: Identification of Differentially Expressed Genes. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 127–139. Springer, Heidelberg (2014)
25. Schulze, G.: Workflow for Rapid Metagenome Analysis. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 88–100. Springer, Heidelberg (2014)
26. Steffen, B., Margaria, T., Nagel, R., Jörges, S., Kubczak, C.: Model-Driven Development with the jABC. In: Bin, E., Ziv, A., Ur, S. (eds.) HVC 2006. LNCS, vol. 4383, pp. 92–108. Springer, Heidelberg (2007)
27. Vierheller, J.: Exploratory Data Analysis. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 110–126. Springer, Heidelberg (2014)