# A Workflow for the Prediction of the Effects of Residue Substitution on Protein Stability

Ruben Acuña[1], Zoé Lacroix,[1], and Jacques Chomilier[2,3]

[1] Arizona State University, Tempe, AZ 85287, USA
[2] Protein Structure Prediction, IMPMC, Université Pierre et Marie Curie, CNRS UMR 7590, Paris, France
[3] RPBS, Université Paris Diderot, 35 Rue Hélène Brion, Paris, France

**Abstract.** The effects of residue substitution in protein can be dramatic and predicting its impact may benefit scientists greatly. Like in many scientific domains there are various methods and tools available to address the potential impact of a mutation on the structure of a protein. The identification of these methods, their availability, the time needed to gain enough familiarity with them and their interface, and the difficulty of integrating their results in a global view where all view points can be visualized often limit their use. In this paper, we present the Structural Prediction for pRotein fOlding UTility System (SPROUTS) workflow and describe our method for designing, documenting, and maintaining the workflow. The focus of the workflow is the thermodynamic contribution to stability, which can be considered as acceptable for small proteins. It compiles the predictions from various sources calculating the $\Delta\Delta$G upon point mutation, together with a consensus from eight distinct algorithms, with a prediction of the mean number of interacting residues during the process of folding, and a sub domain structural analysis into fragments that may potentially be considered as autonomous folding units, i.e., with similar conformations alone and in the protein body. The workflow is implemented and available online. We illustrate its use with the analysis of the engrailed homeodomain (PDB code 1enh).

## 1 Introduction

As it has been reviewed by Tokuriki and Stawfik [42], amino acid substitution is now considered as a major constraint on protein evolvability, while it was previously admitted that most positions can tolerate drastic sequence changes, provided the fold is conserved. Actually, mutations affect stability and stability affects evolution. The level of deleterious mutations can be as high as one third [42]. Therefore, the prediction of the effects of residue substitution can be of great help in wet labs. In this paper, we focus only on the thermodynamic contribution to stability, which can be considered as acceptable for small proteins.

Potapov et al. [37] compared six different methods to predict the change in protein stability on a set of mutations taken from the FOLDEF paper [13] and a

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 253–264, 2013.

second set from ProTherm [17]. The tested tools are: CC/PBSA [3], EGAD [35], FoldX [41], Hunter [36], Imutant2 [4] and Rosetta [39]. The authors notice that Rosetta is not trained for $\Delta\Delta$G calculations, thus resulting in a low correlation coefficient compared to EGAD, the best in their study. One of the drawbacks of EGAD is the fact that they do not predict special mutations, namely Cys, Gly and Pro, because the perturbation to the backbone is too large with these residues. One can nevertheless notice that none of the methods is able to correctly predict all the $\Delta\Delta$Gs for all mutations, but the general trend is correct on average. The average error is 1.72 kcal/mol, thus one can reasonably put a threshold at 2 kcal/mol for the decision of hot spot positions. Khan and Vihinen [15] completed another study with: Automute [28], Cupsat [34], Dmutant [50], FoldX, Multimutate [8], Mupro [5], Imutant versions 2 and 3 [4], and the set SCide [9], SCpred [18] and SRide [27]. The latter three programs identify stability centers rather than provide a general prediction of $\Delta\Delta$G and so were excluded from our selection. Khan and Vihinen also examined Automute [28] but could not produce enough test data for statistical analysis.

Scientists interested in the prediction of the effects of residue mutations have therefore to select a tool among many tools available to compute such prediction [3,35,41,36,4,39,28,34,50,8,5,50,8,4,9,18,27], get familiar with its interface and various built-in specifications and limitations (not always documented), run the execution of the selected tool, and compare, often manually, the results with results obtained with a similar tool or a tool implementing a complementary concept. A benchmark study such as [37] may guide the selection of a tool, in contrast we demonstrate the benefits of a workflow that orchestrates the best tools, integrates the results and compiles a consensus into a single interface.

Workflows are used in business applications to assess, analyze, model, define and implement business processes. A workflow automates the business procedures where documents, information or tasks are passed between participants according to a defined set of rules to support an overall goal. In the context of scientific applications, a workflow approach may promote collaboration among scientists, as well as the integration of scientific data and tools. Scientific workflows focus on the support of scientific experiments replay, design and data retrieval whereas Laboratory Information Management Systems (LIMS) support the integration of different functionalities in a laboratory, such as sample tracking (invoicing/quoting), integrated bar-coding, instrument integration, personnel and equipment management, etc.

The Structural Prediction for pRotein fOlding UTility System (SPROUTS) workflow was developed to provide a global view of the potential impact of mutations on proteins. It aims at integrating several concepts and implements each of them with various methods and tools. In this paper we focus on the predictions from eight resources calculating the $\Delta\Delta$G upon point mutation and a consensus method.

The paper is organized as follows. We first discuss work related to scientific workflows in Section 2. The development of a scientific workflow requires addressing many challenges including design, implementation, maintenance, and

performance. They are discussed in the context of the SPROUTS workflow in Section 3 whereas our approach is described in Section 4. We present a use case in Section 5. Future work is presented in Section 6.

## 2 Scientific Workflows

Scientific workflows often are executed manually. The reasons for manual executions include, among others, the need to validate the results of intermediate steps, the benefit of graphical interfaces of the tools they integrate, the better knowledge of the resource functionalities by experiencing them manually, the changes and updates made on resources that are more easily traceable when the user is using them. These processes are very often poorly documented and scientists experience difficulties in reproducing their datasets as the resources they use may change over time (new database entries, data curation, new version of a tool, etc.). This lack of documentation also affects the ability of integrating and comparing datasets and analyses produced over time. Moreover, the manual execution of a workflow is typically time and manpower consuming. Scripting programming environments such as Perl and Python have also been proven incredibly successful to support the rapid development of workflows. Although this automation saves time and manpower they typically fail to support the proper design and documentation of the process. Lack of documentation not only affects data integration and comparison but also workflow re-use and revision. Various Web-based work benches offer an alternative solution to the problem of automation of orchestrated bioinformatics resources by providing unified access with a simplified interface to multiple resources running on their servers. They include PISE [23], wEMBOSS [40], and Mobyle [33] among many others.

Workflow systems are very successful among the biological community as they provide scientists with the ability to express their scientific protocols as a sequence of connected steps [22]. They describe the scientific process from experiment design, data capture, integration, processing, and analysis that leads to scientific discovery. They typically express *digital* workflows and execute them on platforms such as grids. The procedural support of a workflow resembles the query-driven design of scientific problems and facilitates the expression of scientific pipelines (as opposed to a database query). Kepler [25], which extends the Ptolemy II system [29,30], supports modular workflow design and task scheduling. WOODSS [31] emphasize the support of several abstraction levels of workflow design and facilitates workflow composition and reuse. Many scientific workflow systems focus on execution in general [29,1] or in the Grid computing environment. For example, the GriPhyN Project [12] is developing Grid technologies to collect and analyze distributed scientific and engineering datasets. The Pegasus framework [7,26] uses the Chimera system [10] to describe abstract workflows, and Condor DAGMan and schedulers [6] to generate concrete workflows for execution on the Grid. In Taverna [14] a workflow is composed of *processors* connected with data dependencies links. Its revised updated version is now extensible and scalable that can be used from a workbench, a command

line or remotely as a server [32]. One of the challenges not yet addressed by these approaches is the legacy of scientific workflows. Indeed while they offer support for the development of new workflows the automation, documentation, and revision of legacy workflows such as SPROUTS remains a challenge.

## 3 The SPROUTS Workflow

The initial process was designed to populate the SPROUTS database [24] with six tools: DFIRE version 2.0 [49], I-Mutant 2.04 and I-Mutant-DSSP 2.04 [4], MUpro version 1.1 [5], PoPMuSiC [11], and a stability consensus method. The development of the new revised workflow followed three successive revision steps: automation of the database population process, update of the workflow with more recent tools, and support of on-line submission of proteins.

To compose the revised SPROUTS workflow we concentrated on programs that could be run on our servers, therefore excluding the Web submission systems, such as Eris [48] (the standalone version is commercial), Cupsat, Automute, in order to avoid manipulation of various formats when new releases of the programs are proposed. We had intended to include CUPSAT, however, we were unable to contact the authors due to issues with their website and contact addresses. We performed trial use of MultiMutate but found it incompatible (unstable) with the existing (Ubuntu based) server that the workflow must execute on. In addition to the tools analyzed by Khan and Vihinen, we also examined SDM [46] and Pro-Maya [44] but they are not currently available as a local executable or a Web service and so cannot be integrated with the existing workflow.

The new revised SPROUTS workflow processes data for DFIRE 1.1 (Dmutant), FoldX 3.0 beta 5.1, I-Mutant 2.0 sequence/structure modes, I-Mutant 3.0 sequence/structure modes, and MUpro. Our database also contains legacy data from PoPMuSiC [19], these data were part of the original database. Because no local executable version of this tool was available, we were unable to include it in our workflow. These represent the most recent versions of the respective tools with one exception: DFIRE 2.1 [47]. This most recent version of DFIRE operates directly on a (possibly) mutated PDB structure. Because our current workflow does not support dependencies between tools, we were unable to produce the necessary mutant PDBs to use DFIRE 2.1. MuD [45] an interactive Web server for the prediction of mutations from a structure based on a machine learning algorithm was published recently. We have not integrated this tool yet because it does not provide a $\Delta\Delta$G calculation but rather an estimate of function conservation. The revision of the SPROUTS workflow with MuD would require changing the consensus method in such a way the $\Delta\Delta$G calculation of the other tools can be combined with an estimate of the function conservation.

The SPROUTS workflow[1] populates the SPROUTS database [24]. Submitting a new protein to the SPROUTS system executes the whole SPROUTS workflow and uploads the results into the database. Because the execution of the workflow

[1] The SPROUTS workflow is available online at `http://bioinformatics.engineering.asu.edu/sprouts`.

takes time, a link to the the database entry is provided to the user to retrieve the results from the database after completion of the workflow. The user retrieves the information pertaining to one protein at a time through its PDB ID. The user may then select a single tool or access the results of all the tools. A specific residue and mutation may also be selected (by default every residue and every mutation will be returned). The residue number may be specified (note that SPROUTS numbering does not follow PDB numbering: in case the user specifies an amino acid and a number, SPROUTS will check if this is the right amino acid at this position). By default, no residue is specified and so all residues will be considered. Another parameter offers the possibility to visualize only the mutations which increase the stability or at the opposite which decrease the stability (the default mode is to return all results. The last parameter offers the possibility to limit the number of results displayed on the result page. By default, the value is set to 190 lines which correspond to the results of all the 19 possible mutations for 2 residues and for all the tools. Even if the option is available, it is strongly advised not to select the "all" option especially for long proteins.

## 4 Developing the SPROUTS Workflow

Our method for workflow development involves the characterization of the workflow at four levels: semantic, implementation, execution, and data. To document the workflow we follow the approach developed with ProtocolDB where workflows are first expressed in terms of a domain ontology where each task expresses a specific aim [16]. Domain ontologies[2] can be used to describe the concepts and relationships of a discipline as well as to document the tools and methods [20]. A *design protocol* (or workflow) is defined top-down from a conceptual design task that describes the workflow as a whole. The conceptual design is defined in terms of input and output parameters which are expressed as complex conceptual types (collections of concept variables). Each design task may be split either sequentially (with the $\otimes$ operator) or in parallel (with the $\oplus$ operator) into two design tasks. The semantic characterization of the workflow enables reasoning on workflows at a conceptual level. Semantic equivalence of workflow implementations (mapped to the same semantic representation) can be used to validate data integration, compare implementations performances and support workflow optimization [21].

The concepts involved in the SPROUTS workflow include `Protein`, specified with its name and PDB code, sequence, structure, and secondary structure, `Residue`, specified by its name and location on the sequence, and the value of Gibbs free energy as an approximation to characterize the stability of a given structure. See [43] for an ontology devoted to structural bioinformatics. We consider the difference of energy for the wild type of the protein $\Delta G_{wild}$ and for the mutant $\Delta G_{mutant}$. We define[3] the difference as $\Delta\Delta G = \Delta G_{mutant} - \Delta G_{wild}$ in

[2] See The Open Biological and Biomedical Ontologies at `http://www.obofoundry.org/` for a list of ontologies for various scientific domains.

[3] Note that different stability prediction methods use different definitions.

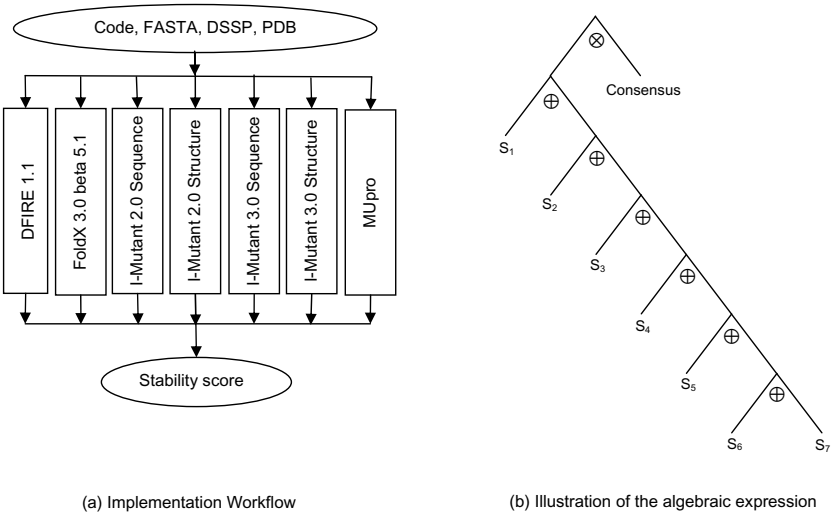


**Fig. 1.** SPROUTS Implementation Workflow

kcal/mol. At this level of definition, the workflow consists of a single task that links the concept `Protein` to a score that expresses the impact of a mutation on its stability for each residue. The revisions of the workflow discussed in Section 3 do not impact the semantics of SPROUTS. The phase of automating the legacy population workflow does not change its semantics nor does updating the tools the workflow is composed of. Indeed, the *design workflow* captures the semantic aim of the workflow which is not affected by the proposed revision.

The second development phase consists of the specification of the resources that are implementing each of the design tasks. Each design task is mapped to a *implementation protocol* (or workflow) defined as follows. An *implementation protocol* is a graph composed of connected scientific resources (database queries or tools) whose inputs and outputs are data types. A single bioinformatics service is an implementation protocol. Complex implementation protocols are composed of scientific resources connected with the same two binary operators $\oplus$ and $\otimes$ used to express design protocols. Here, the design task can be implemented by many existing resources as discussed in Section 1. Because we chose to exploit multiple stability prediction methods, and integrate their results in a consensus step, the single design task will be first mapped to two successive implementation steps connected with the $\otimes$ operator. The second implementation step will be specified with the consensus method. The first implementation step will be split with the parallel operator $\oplus$. The first of the two steps will be specified with the first stability prediction tool DFIRE 1.1 when the second one will be split into two parallel steps. The first of the two will be assigned to FoldX 3.0 beta 5.1 whereas the the second one will be, again, split into two parallel steps, and so on.

The resulting implementation workflow is expressed by

$$(S_1 \oplus (S_2 \oplus (S_3 \oplus (S_4 \oplus (S_5 \oplus (S_6 \oplus S_7)))))) \otimes Consensus$$

where $S_1, \ldots S_7$ denote respectively DFIRE 1.1, FoldX 3.0 beta 5.1, I-Mutant 2.0 sequence, I-Mutant 2.0 structure, I-Mutant 3.0 sequence, I-Mutant 3.0 structure, and MUpro.

The input (resp. output) of the implementation workflow consists of the input (resp. output) datatype. The input of the implementation workflow describes the concept `Protein` as follows. It consists of a 4-character code (that may be a PDB ID), the protein primary structure or sequence in FASTA format, the description of the secondary structure in DSSP format, and the 3-D structure in PDB format. The output consists of the protein sequence (list of residues) and the stability scores (for each residue, 8 scores are computed: one for each tool and the consensus score). The SPROUTS implementation workflow illustrated in Figure 1 can be represented with a binary tree.

The third level of workflow characterization is the execution plan. This level requires the specification of the programmatic environment (e.g., Taverna, Kepler, or scripting language such as Python, Perl). The first step of the SPROUTS workflow revision (automation of the database population process) did not affect the first two layers of representation. The revision consisted in replacing the manual execution by a Python program. Although the orchestration of the steps that were initially used to populate the database into a single script was not likely to produce a well designed workflow with suitable performance and adaptability, it was the chosen path because it was also the one less likely to impact the availability of the SPROUTS database. The second step of the revision (workflow update with more recent tools) had an impact on both the implementation layer as new tools were used and the execution layer as the overall structure of the workflow had also changed. The main challenges of SPROUTS development have been importing applications and tools which lack documentation, including the specification of the limitations (often implicit) of their input, a description of their computational time (performance) and execution failures. Moreover, none of the tools exploited in the workflow offers a description of its interface expressed in a machine readable format such as Web Service which limits the ability of implementing and executing the workflow on a system such as Taverna.

## 5 Use Case

The SPROUTS workflow is implemented and available online. Once the protein has been submitted to the SPROUTS workflow and the execution has completed, the results are stored in the SPROUTS database and can be accessed with the query form. All stability prediction tools of the workflow are selected by default.

The results for 1enh are shown in the table (left of Figure 2). In the 2D mode, the results of all the tools but FoldX 3 are displayed (right). The consensus graph is currently created by taking the mean of the available data. Due to evolution,

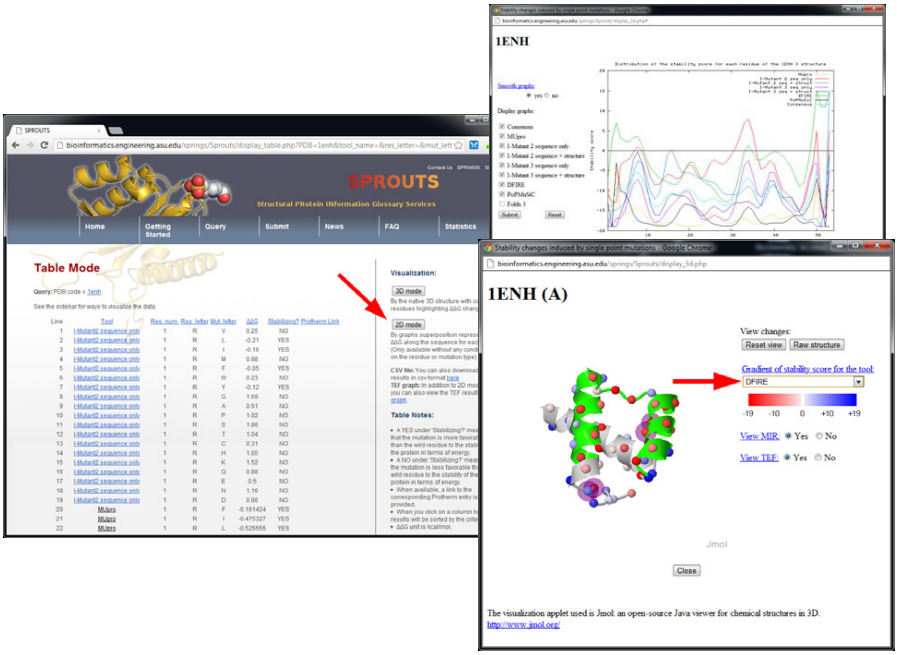


**Fig. 2.** Engrailed homeodomain (PDB code 1enh)

the number of stabilizing mutations is smaller than destabilizing ones. One must mention that a stabilizing mutation is not necessarily related to an improved efficiency of the mutated protein, as far as function is concerned. Sometimes, a more stable structure results in an increased rigidity, while the function requires a certain level of flexibility. This is the case for instance with enzyme catalysis [46]. Therefore, it seems reasonable to place a threshold of 2 kcal/mol in either way of $\Delta\Delta$G (stabilizing or destabilizing) in order to claim to a putative malfunction. Mutations in conserved positions usually cause large stability decreases. The 3D mode (bottom right) displays the protein structure retrieved from PDB.

The engrailed homeodomain (PDB code 1enh) is a small single domain (54 residues), monomeric, composed of three helices, and without any disulfide bridge. It is considered as a model for the hierarchic type of folding, and one Leucine, at position 14, is deeply buried in the core of the structure, stabilized by hydrophobic interactions with amino acids from the two other helices. This particular residue has been mutated by the group of Fersht [38] and the NMR structure determined (PDB code 1ztr). The mutated form is no more a globular protein, since the accessible surface area is increased by 50% due to mutation. Nevertheless, most of the local stability remains since the three helices are still present.

When comparing the 1enh and 1ztr 2D plots, the differences are not significant, unless some N and Cter effects due to the non symmetrical process of smoothing. But introducing the structure in the algorithm has an effect in

I-Mutant. Although the general shape is similar between 1enh and 1ztr for I-Mutant 2.0 with structure, the highest divergence occurs around position 14. Such a peak does not appear in the two algorithms considering only the sequence. It pleads in favor of the proof of a better prediction with structures included, specially when single mutations are concerned. When comparing now the two versions of I-Mutant (2.0 vs 3.0) the high peak of instability is conserved for 1enh around position 8. But the peak previously discussed around 15 in I-Mutant 2.0 almost vanishes with I-Mutant 3.0. Nevertheless, although the peak decreases in the middle of the first helix, the global gross features of the shape of the curves are looking like for the wild type structure. This is not the case for the mutated structure, and one may argue that the underlying principles ruling I-Mutant 3.0 are scaled on compact globular proteins, and do not apply to proteins looking like NUP (Natively Unfolded Protein).

## 6 Conclusion and Future Work

The workflow is under significant revision and extension with new functionalities and improved interface to come. Once the revision is completed, a mirror of SPROUTS 2.0 will be deployed in the Ressource Parisienne en Bioinformatique Structurale (RPBS) [2]. The current version of the SPROUTS workflow is available at http://bioinformatics.engineering.asu.edu/springs/Sprouts/.

**Acknowledgment.** We acknowledge and thank Pierre Tufféry, Dirk Stratmann, Elodie Duprat, Mathieu Lonquety, Christophe Legendre, Nikolaos Papandreou, Fayez Hadji, as well as the authors of the different tools used in SPROUTS. We also wish to acknowledge our collaborators at ASU: Rida Bazzi, Antonia Papandreou-Suppappola, Anna Malin, and Banu Ozkan. This research was partially supported by the National Science Foundation[4] (grant CNS 0849980) and an invitation by the Université Pierre et Marie Curie.

## References

1. Aeschlimann, M., Dinda, P., Lopez, J., Lowekamp, B., Kallivokas, L., O'Hallaron, D.: Preliminary report on the design of a framework for distributed visualization. In: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 1833–1839 (1999)
2. Alland, C., Moreews, F., Boens, D., Carpentier, M., Chiusa, S., Lonquety, M., Renault, N., Wong, Y., Cantalloube, H., Chomilier, J., Hochez, J., Pothier, J., Villoutreix, B.O., Zagury, J.-F., Tufféry, P.: RPBS: a web resource for structural bioinformatics. Nucleic Acids Res. 33(web Server issue), W44–W49 (2005)
3. Benedix, A., Becker, C.M., de Groot, B.L., Caflisch, A., Böckmann, R.A.: Predicting free energy changes using structural ensembles. Nat. Methods 6(1), 3–4 (2009)

[4] Any opinion, finding, and conclusion or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

4. Capriotti, E., Fariselli, P., Casadio, R.: I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. Nucleic Acids Res. 33, 306–310 (2005)
5. Cheng, J., Randall, A., Baldi, P.: Prediction of protein stability changes for single site mutations using support vector machines. Proteins 62, 1125–1132 (2006)
6. Condor. Manual (7.0.1) (2008), http://www.cs.wisc.edu/condor/manual/v7.0/
7. Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Patil, S., Su, M.-H., Vahi, K., Livny, M.: Pegasus: Mapping Scientific Workows onto the Grid. In: European Across Grids Conference, pp. 11–20 (2004)
8. Deutsch, C., Krishnaoorthy, B.: Four body scoring function for mutagenesis. Bioinformatics 23(22), 2009–3015 (2007)
9. Dosztányi, Z., Magyar, C., Tusnády, G., Simon, I.: SCide: identification of stabilization centers in proteins. Bioinformatics 19(7), 899–900 (2003)
10. Foster, I., Voeckler, J., Wilde, M., Zhao, Y.: Chimera: a virtual data system for representing, querying and automating data derivation. In: 14th International Conference on Scientific and Statistical Database Management, pp. 37–46 (2002)
11. Gilis, D., Rooman, M.: PoPMuSiC, an algorithm for predicting protein mutant stability changes: application to prion proteins. Protein Eng. 13(12), 849–856 (2000)
12. GriPhyN. Grid Physics Network in ATLAS, http://www.usatlas.bnl.gov/computing/grid/griphyn/
13. Guerois, R., Nielsen, J., Serrano, L.: Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. J. Mol. Biol. 320(2), 369–387 (2002)
14. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic Acids Research 34(web server issue), 729–732 (2006)
15. Khan, S., Vihinen, M.: Performance of protein stability predictors. Hum. Mutat. 31(6), 675–684 (2010)
16. Kinsy, M., Lacroix, Z., Legendre, C., Wlodarczyk, P., Yacoubi, N.: ProtocolDB: Storing Scientific Protocols with a Domain Ontology. In: Weske, M., Hacid, M.-S., Godart, C. (eds.) WISE Workshops 2007. LNCS, vol. 4832, pp. 17–28. Springer, Heidelberg (2007)
17. Kumar, M., Bava, K., Gromiha, M., Prabakaran, P., Kitajima, K., Uedaira, H., Sarai, A.: ProTherm and ProNIT: thermodynamic databases for proteins and protein-nucleic acid interactions. Nucleic Acids Res 34, D204–D220 (2006)
18. Kurgan, L., Cios, L., Chen, K.: SCPRED: accurate prediction of protein structural class for sequences of twilight zone similarity with predicting sequences. BMC Bioinformatics 9, 226 (2008)
19. Kwasigroch, J.M., Gilis, D., Dehouck, Y., Rooman, M.: PoPMuSiC, rationally designing point mutations in protein structures. Bioinformatics 18, 1701–1702 (2002)
20. Lacroix, Z., Aziz, M.: Resource descriptions, ontology, and resource discovery. International Journal of Metadata, Semantics and Ontologies 5(3), 194–207 (2010)
21. Lacroix, Z., Legendre, C., Tuzmen, S.: Reasoning on Scientific Workflows. In: Proceedings of the IEEE International Workshop on Scientific Workflows, vol. World Conference on Services - I, pp. 306–313. IEEE Computer Society (2009)
22. Lacroix, Z., Ludaescher, B., Stevens, R.: Integrating Biological Databases. In: Bioinformatics - From Genomes to Therapies, vol. III, pp. 1525–1572. Wiley-VCH Publisher (2007)
23. Letondal, C.: A web interface generator for molecular biology programs in Unix. Bioinformatics 17, 73–82 (2001)

24. Lonquety, M., Lacroix, Z., Papandreou, N., Chomilier, J.: SPROUTS: a database for the evaluation of protein stability upon point mutation. Nucleic Acids Res. 37, 374–379 (2009)
25. Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y.: Scientific Workflow Management and the KEPLER System. Concurrency and Computation: Practice and Experience, Special Issue on Scientific Workflows 18(10), 1039–1065 (2005)
26. Maechling, P., Chalupsky, H., Dougherty, M., Deelman, E., Gil, Y., Gullapalli, S., Gupta, V., Kesselman, C., Kim, J., Mehta, G., Mendenhall, B., Russ, T., Singh, G., Spraragen, M., Staples, G., Vahi, K.: Simplifying construction of complex workflows for non-expert users of the southern california earthquake center community modeling environment. ACM SIGMOD Record 34(3), 24–30 (2005)
27. Magyar, C., Gromiha, M., Pujadas, G., Tusnády, G., Simon, I.: SRide: a server for identifying stabilizing residues in proteins. Nucleic Acids Res. 33, W303–W305 (2005)
28. Masso, M., Vaisman, I.: Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis. Bioinformatics 24, 2002–2009 (2008)
29. McPhillips, T.M., Bowers, S.: An approach for pipelining nested collections in scientific workflows. ACM SIGMOD Record 34(3), 12–17 (2005)
30. McPhillips, T., Bowers, S., Ludäscher, B.: Collection-Oriented Scientific Workflows for Integrating and Analyzing Biological Data. In: Leser, U., Naumann, F., Eckman, B. (eds.) DILS 2006. LNCS (LNBI), vol. 4075, pp. 248–263. Springer, Heidelberg (2006)
31. Medeiros, C.B., Perez-Alcazar, J., Digiampietri, L., Pastorello, J.G.Z., Santanche, A., Torres, R.S., Madeira, E., Bacarin, E.: WOODSS and the Web: annotating and reusing scientific workflows. ACM SIGMOD Record 34(3), 18–23 (2005)
32. Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., Williams, A., Oinn, T., Goble, C.: Taverna, reloaded. In: Gertz, M., Ludäscher, B. (eds.) SSDBM 2010. LNCS, vol. 6187, pp. 471–481. Springer, Heidelberg (2010)
33. Néron, B., Ménager, H., Maufrais, C., Joly, N., Maupetit, J., Letort, S., Carrere, S., Tuffery, P., Letondal, C.: Mobyle: a new full web bioinformatics framework. Bioinformatics 22, 3005–3011 (2009)
34. Parthiban, V., Gromiha, M., Schomburg, D.: CUPSAT: prediction of protein stability upon point mutation. Nucleic Acids Res. 34, W239–W242 (2006)
35. Pokala, N., Handel, T.: Energy Functions for Protein Design: Adjustment with Protein–Protein Complex Affinities, Models for the Unfolded State, and Negative Design of Solubility and Specificity. J. Mol. Biol. 347(1), 203–227 (2005)
36. Potapov, V., Cohen, M., Inbar, Y., Schreiber, G.: Accurate structure modelling based on precise description of inter-residue interactions. BMC Bioinformatics 11, 374 (2010)
37. Potapov, V., Cohen, M., Schreiber, G.: Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. Protein Eng. Des. Sel. 22(9), 553–560 (2009)
38. Religa, T.L., Markson, J.S., Mayor, U., Freund, S.M.V., Fersht, A.R.: Solution structure of a protein denatured state and folding intermediate. Nature 437, 1053–1056 (2005)
39. Rohl, C., Strauss, C., Misura, K., Baker, D.: Protein structure prediction using Rosetta. Methods Enzym. 383, 66–93 (2004)
40. Sarachu, M., Colet, M.: wEMBOSS: a web interface for EMBOSS. Bioinformatics 21, 540–541 (2005)

41. Schymkowitz, J., Borg, J., Stricher, F., Nys, R.F., Serrano, L.: The FoldX web server: an online force field. Nucleic Acids Res. 33, W382–W388 (2005)
42. Tokuriki, T.D., Stability, N.: effects of mutations and protein evolvability. Curr. Opin. Struct. Biol. 19(5), 596–604 (2009)
43. Tufféry, P., Lacroix, Z., Ménager, H.: Semantic Map of Services for Structural Bioinformatics. In: Proc. 18th International Conference on Scientific and Statistical Database Management, pp. 217–224. IEEE, Vienna (2006)
44. Wainreb, G., Ashkenazy, H., Bromberg, Y., Starovolsky-Shitrit, A., Haliloglu, T., Ruppin, E., Avraham, K., Rost, B., Ben-Tal, N.: Protein stability: a single recorded mutation aids in predicting the effects of other mutations in the same amino acid site. Bioinformatics 27, 3286–3292 (2011)
45. Wainreb, G., Ashkenazy, H., Bromberg, Y., Starovolsky-Shitrit, A., Haliloglu, T., Ruppin, E., Avraham, K., Rost, B., Ben-Tal, N.: MuD: an interactive web server for the prediction of non neutral substitutions using protein structural data. Nucleic Acids Res. 38, W523–W528 (2010)
46. Worth, C., Preissner, R., Blundell, T.: SDM - a server for predicting effects of mutations on protein stability and malfunction. Nucleic Acids Res. 39, W215–W222 (2011)
47. Yang, Y., Zhou, Y.: Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely related all atom statistical energy functions. Prot. Sci. 17, 1212–1219 (2008)
48. Yin, S., Ding, F., Dokholyan, N.: Eris: an automated estimator of protein stability. Nature Meth. 4, 466–467 (2007)
49. Zhou, H., Zhou, Y.: Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Sci. 11, 2714–2726 (2002)
50. Zhou, H., Zhou, Y.: Fold recognition by combining sequence profiles derived from evolution and from depth dependent structural alignment of fragments. Proteins 58, 321–328 (2005)