RIBOWEB: Linking Structural Computations to a Knowledge Base of Published Experimental Data

Richard O. Chen, Ramon Felciano & Russ B. Altman
Section on Medical Informatics, MSOB X-215
Stanford University,
Stanford, CA, USA, 94305-5479
{rchen, felciano, altman}@smi.stanford.edu

Abstract

The world wide web (WWW) has become critical for storing and disseminating biological data. It offers an additional opportunity, however, to support distributed computation and sharing of results. Currently, computational analysis tools are often separated from the data in a manner that makes iterative hypothesis testing cumbersome. We hypothesize that the cycle of scientific reasoning (using data to build models, and evaluating models in light of data) can be facilitated with resources that link computations with semantic models of the data. RIBOWEB is an on-line knowledge-based resource that supports the creation of three-dimensional models of the 30S ribosomal subunit. It has three components: (I) a knowledge base containing representations of the essential physical components and published structural data, (II) computational modules that use the knowledge base to build or analyze structural models, and (III) a web-based user interface that supports multiple users, sessions and computations. We have built a prototype of RIBOWEB, and have used it to refine a rough model of the central domain of the 30S subunit from *E. coli*. procedure. Our results suggest that sophisticated and integrated computational capabilities can be delivered to biologists using this simple three-component architecture.

Introduction

The pace at which scientific data is being published, particularly on the WWW, threatens to overwhelm our ability to build coherent, self-consistent scientific models. Such models depend on 1) having all relevant information, 2) interpreting this information properly, and 3) integrating multiple sources of potentially contradictory data. Each of these tasks becomes increasingly difficult as the volume of relevant data increases. One of the major goals of computational molecular biology, therefore, is the creation of integrating technologies to support the process of developing models consistent with large, heterogeneous, distributed data sets. The WWW offers a vehicle for bringing these technologies to desktops, without the need for special hardware or constant software updates.

There are several ongoing efforts to provide automated (or semi-automated) support for scientific computation. The Socrates project emphasizes the management of heterogeneous hardware and software resources in solving computationally intensive problems in structural biology [1]. In work similar to our own, the *B. subtilis* sequencing project uses an object-oriented and knowledge-based system for genome sequence analysis [2]. Several groups have developed web-based scientific tools and workbenches including the NCSA's Biology Workbench [3]. Recent efforts to integrate multiple biological databases [4-6], are based on the premise that scientific hypothesis formation can be assisted by providing seamless access to these resources. While very general, these approaches are less useful for supporting advanced reasoning about specific biological systems, for which a deeper collection of information is required. There is, therefore, an increasing interest in creating special purpose web-based resources for specific areas of biology such as protein-kinase molecules [7] or cytokines [8].

Focusing on the structural computations for the computation of the 30S ribosomal subunit of *E. coli* , we have built a prototype system called RIBOWEB to demonstrate the benefits of a system that explicitly links computational methods with a principled knowledge base of biological data. In RIBOWEB, the scientific hypotheses are the three-dimensional models of the 30S subunit (which consists of 21 proteins and the 16S rRNA), while the data are the contents of published articles containing relevant structural information such as distances, angles, topology, and locations. The volume of data sources and interpretations for these data makes thorough testing and analysis of structural models difficult. In addition, the technologies used to compute structures are expensive and require independent validation and sensitivity analyses. Our initial goals are to develop 1) a knowledge base of ribosomal structural information, 2) a collection of structural computation and evaluation methods, and 3) a model for linking the computational methods with the knowledge base— all in a web based environment that allows multiple users to perform structural computations, access the knowledge base, and potentially collaborate with other users.

Methods

The basic architecture of RIBOWEB involves the interaction of three components. The knowledge base serves as the source for all information required by the system, including biological information about the 30s ribosomal subunit and declarative information about the computational modules themselves. The computational methods are collections of programs useful for calculating and analyzing structures and related results. These methods draw on the knowledge base and the user for

information it needs to perform the various calculations. The user interface consists of the session manager (to handle current and past computational results) and the interface generator which provides web based user access to the system. These modules work in concert to provide a web based environment for doing structural computations (Figure 1).

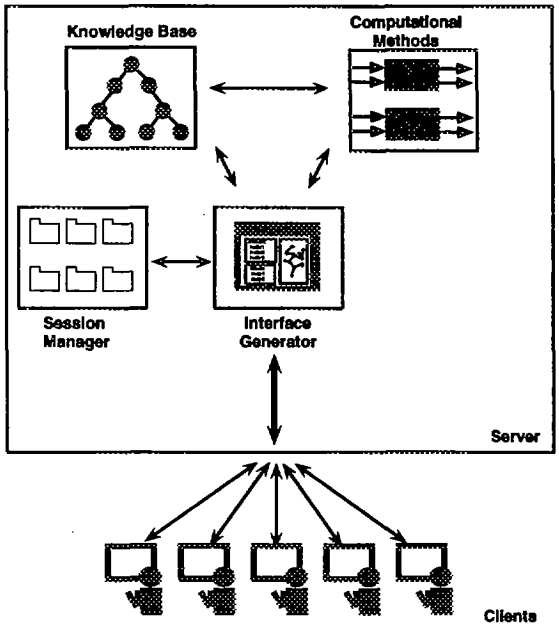

Figure 1. The overall architecture of RIBOWEB. The knowledge base, computational modules and session-manager/interface generator are on the server side (though each may be on a different computer). Multiple clients can communicate with the interface generator to perform computations or browse the knowledge base. In the current implementation, the interface generator runs in PERL. Computational modules run on a separate UNIX workstation in C, C++, PERL or LISP. The knowledge base is written in PERL, and most communication is done via HTTP.

The Knowledge Base

The knowledge base is a repository for information about the structural elements of the ribosome, as well as experimental observations about structure. We use a knowledge representation schema that captures the hierarchical nature of biological concepts. For our initial prototype, we built a knowledge representation tool called VKB (Virtual Knowledge Base) using the PERL language. VKB is a frame-based knowledge representation system similar to CLIPS, CLASSIC or ONTOLINGUA [9] that allows us to define a schema of concepts and their attribute names and types. Most entries in the knowledge base are instances of these objects associated with specific attribute values. VKB can be accessed in read/write mode both through direct procedure calls and through a web-accessible CGI interface.

The prototype RIBOWEB knowledge base contains structural data from eighteen published papers, chosen for their relevance to our central domain calculations. These papers report experimental observations about proximity between RNA bases or between the RNA bases and the S-proteins. The knowledge base also contains the Cartesian coordinates for previously reported models of the 30S subunit, the published relative positions of the 21 proteins in the 30S subunit, and the primary and secondary structure information for the 16S RNA. The key advantage of frame-based representations is the ability to provide multiple links between concepts to create a network of useful information.

The RIBOWEB knowledge base also contains declarative information about the computational methods. Each method is represented as an object containing a description of the input and output data types of the method. These descriptions resembles a primitive distributed object representation, such as CORBA [10], and have similar advantages. First, we can incorporate computational modules written in different languages. Second, the task of using these computational methods is simplified since we need only know how to get the necessary input to the method, how to store the output from the method, and how to run the method.

The Computational Methods

A computational method can be single program or a combination of several programs working to produce a result. The RIBOWEB knowledge base contains an explicit representation of the data types required for input and produced as output for each method. The invocation of a method with a set of particular input parameters causes the session manager to issue a unique "computation ID" that identifies this particular computation and allocates file space for the computation. All the resulting output is stored in this file space, including the input parameters. Thus, an audit trail is kept of all computations, and results from previous computations can be accessed at any time and used for future computation.

When beginning a calculation, the session manager queries the knowledge base for information about the method being invoked. The method description contains parameters describing the nature of the inputs including the variable name, variable type (for example: knowledge base class, enumerated type, or text), and default values. Using this method description, the interface generator dynamically generates an HTML page that lists each required variable, and an appropriate widget for acquiring the value. Once the input values are obtained, the session manager spawns a process that runs the computational module. At the same time, the system creates a URL where the results of the computational module can be

accessed upon completion. Some of these results can be used as starting points for new computations.

Our prototype has ten methods for computing structure and analyzing results. The main computational method is a probabilistic constraint satisfaction algorithm that computes the "most likely" position for each atom given a set of uncertain distance constraints, starting map coordinates, and the uncertainties associated with those values [11]. The analysis modules range from those that find the constraints that are least well satisfied by a model to those that generate fully hyperlinked 3D graphical models in VRML.

The Interface Generator and Session Manager

The most of the functionality of the interface generator has been discussed in the context of the other modules. The interface generator is independent of the other modules in the sense that it is not affected by adding or changing information in the other components of the system.

To support a multi-user environment, we distinguish public and privately accessible information. The biological data described above comprise the core of the knowledge base, and are made "public" to all users. However, any new data created by a user through direct editing of knowledge base information or through new computations is automatically tagged with that user as the owner. Any given user will only have access to the "public" data and his or her own "private data" in the knowledge base. This allows users to impose their own interpretations on the data and pursue different lines of experimentation. Users can also share the results of their computations, and this shared information can be used by others to perform additional analysis and computations. RIBOWEB tracks each login session, so user data is available for performance monitoring and user interface validation.

Results: The Structure of the 30S Central Domain

The 30S central domain is composed of bases 567 to 883 in *E. coli*, and represents about 20% of the entire 30S subunit. It is of particular interest because it is evolutionarily conserved, and contains a number of antibiotic binding sites. We have published two low resolution structures of this domain [11, 12]. We used RIBOWEB to refine our most recent model by including additional RNA bases not previously modeled. Through a series of computations we improved the structure from an initial average constraint error of 10.9 SD (standard deviations from target value) to a final average error of 0.07 SD. The final structure has good convergence.

We were able to use RIBOWEB to externally validate our models. The computed RMS distance from other models in the knowledge base averaged 31.6Å which is similar to the 36.6Å reported between other models. [12, 13]. Furthermore, we found the structure to be consistent with data from seven published data sets *not used* in our refinement.. In these data sets, the 15 constraints relevant to the central domain were on average within 2.7 standard deviations error. On the other hand, the structure was relatively inconsistent (> 6.63SDs) with constraint data from three papers. It is clear that though this structure is reasonable, significant issues must be resolved before a refinement at this level of detail can be accepted as a consensus structure.

Discussion

RIBOWEB, despite being a prototype system, is capable of supporting real structural computations and detailed analysis of the results. In our experiments, we performed three cycles of structural model generation, refining the interpretation of the data after each cycle. The structural models were internally and externally evaluated using computational modules closely coupled with the knowledge base. RIBOWEB does not aim to completely automate the process of ribosomal model building; rather, it begins to provide a web-based computational environment that supports the user in formulating computations, visualizing results, and sharing results. The chief advantage of linking computational capabilities with a knowledge base of objects and associated data is the ability to define an explicit context for both the inputs and outputs of the computations. For example, we were able to quickly identify problematic structural regions by asking the system which constraints were not well satisfied, and then visualizing the structures involved in those constraints.

Our results further suggest that the web is an adequate environment for supporting complex computations. By designing the system to be web based, we are bound by some of the web's current limitations (such as its stateless nature), but we have demonstrated that these limitations are not enough to prevent the creation of a usable system over the web.

To accommodate future enhancements, RIBOWEB has been built in a modular fashion and can be incrementally upgraded. The VKB knowledge base is being redesigned using the ONTOLINGUA network-based frame representation system. The computational modules could be written in JAVA or as CORBA objects, with their associated languages for specifying the input/output interface. The session manager and interface generators can be replaced by database management systems and more biology-friendly user interfaces like SSTRUCTVIEW, a interactive java applet displaying RNA secondary structures that are hyperlinked to various information sources across the web, including the VKB [14]. Common graphical idioms like this may serve as more intuitive interfaces to parts of the RIBOWEB system. With these opportunities for improvement, we believe that these replacement technologies will only lead to a more robust and powerful integrated system.

Our results in this paper also expose some of the problems with the current RIBOWEB system. First, we

have not demonstrated the ability to link RIBOWEB to other web-based databases (like the Ribosomal Database Project, RDP) in order to support more diverse data analysis and combination. We have shown (Altman, Abernethy, Chen, in this volume) that the RIBOWEB knowledge base component can be linked to the RDP in order to generate useful structural inferences, but we must next demonstrate that distributed information sources can be fully integrated into the RIBOWEB system. The second major limitation in RIBOWEB is the lack of support for informal communication between users and a mechanism for annotating results. We have not designed this prototype system to be a sophisticated collaboratory environment. It may be necessary to allow concurrent visualizations of data and structures through interactive VRML windows and whiteboards to make RIBOWEB a robust collaboratory environment.

Conclusions

We have described a prototype system for supporting integrated hypothesis formation and evaluation that demonstrates 1) the feasibility of coupling computational methods with an underlying knowledge base of information, 2) the suitability of the web to support this computing environment and the potential for sharing of results, and 3) the utility of such a system in computing ribosomal structure. RIBOWEB is a prototype of what may become a useful central repository for new ribosomal structural data, and a tool for checking the consistency of new data with published structural data and models. It also provides a place where new computational methods could be integrated and made widely available. Eventually, our architecture may be abstracted to provide general purpose computational support in other areas of computational biology. A demonstration version of RIBOWEB can be accessed at http://www-smi.stanford.edu/projects/helix/pubs/ismb97-cfa/.

Acknowledgments

We thank Lawrence Hon and Neil Abernethy for useful feedback, Harry Noller for access to data, and Cheng Che Chen for efficient code for structure calculation. This work is supported in part by NIH-LM05652, LM06442, and LM-07033, NSF DBI-9600637, and a grant from IBM.

References

1. Costian, C. & Marinescu, D., Socrates: An Environment for High Performance Computing. (1995). *IEEE*, : p. 199-206.
2. Medigue, C., Vermat, T., Bisson, G., Viari, A., & Danchin, A. *Cooperative computer system for genome sequence analysis.* in *Third International Conference on Intelligent Systems for Molecular Biology.* 1995. Cambridge, England: AAAI Press.
3. Fischman, J., Working the Web With a Virtual Lab and Some Java. (1996). *Science*, **273**(2 August 1996): p. 591-593.
4. Etzold, T., Ulyanov, A., & Argos, P., SRS: Information Retrieval System for Molecular Biology Data Banks. (1996). *Methods in Enzymology*, **266**: p. 114.
5. Buneman, P., Davidson, S.B., Hart, K., Overton, C., & Wong, L., eds. *A Data Transformation System for Biological Data Sources.* Proceedings of 21st International Conference on Very Large Data Bases. 1995: Zurich, Switzerland.
6. Buneman, P., Naqvi, S., Tannen, V., & Wong, L., Principles of Programming with Complex Objects and Collection Types. (1995). *Theoretical Computer Science*, **149**(1): p. 3-48.
7. Bourne, P., Gribskov, M., Ten Eyck, L., & Taylor, S., PKDB: The Protein Kinase Database Project. (1996). *Protein Data Bank Quarterly Newsletter*, **77**: p. http://www.pdb.bnl.gov/newslttr/txt/newslttr.txt_jul96.
8. Allen, G., Patrick, T., & Murtaugh, M. *World Wide Web-based access to heterogenous information resources for cytokine research.* in *20th Annual AMIA Fall Symposium.* 1996. Washington, D.C.: American Medical Informatics Association.
9. Karp, P.D., *The design space of frame knowledge representation systems*, 1992, SRI International Artificial Intelligence Center.
10. Achard, F. & Barillot, E., *Ubiquitous Distributed Objects with CORBA*, in *Pacific Symposium on Biocomputing, 1997*, R.B. Altman, *et al.*, Editors. 1997, World Scientific: Singapore. p. 39-50.
11. Altman, R.B., A probabilistic approach to determining biological structure: integrating uncertain data sources. (1995). *Int. J. Human-Computer Studies*, **42**: p. 593-616.
12. Fink, D.L., Chen, R.O., Noller, H.F., & Altman, R.B., Computational methods for defining the allowed conformational space of 16S rRNA based on chemical footprinting data. (1996). *RNA*, **2**: p. 851-866.
13. Malhotra, A. & Harvey, S.C., A Quantitative model of the E. coli 16S RNA in the 30S Ribosomal Subunit. (1994). *J Mol Bio*, **240**: p. 308-340.
14. Felciano, R.F., Chen, R.O., & Altman, R.B., RNA secondary structure as a reusable interface to biological information resources. (1996).*Gene-COMBIS*, **in press**: http://www1.elsevier.nl/journals/genecombis/jnl/articles/S0378111996008554/.