Pacific Symposium on Biocomputing 14:403-414 (2009)

UNISON: AN INTEGRATED PLATFORM FOR COMPUTATIONAL BIOLOGY DISCOVERY

REECE K. HART, KIRAN MUKHYALA
Genentech, Inc.
1 DNA Way
South San Francisco, CA, 94080, USA
E-mail: rkh @ gene.com
Revision 186 (2008-09-16 23:01)

This paper describes the design and applications of Unison, a comprehensive and integrated warehouse of protein sequences, diverse precomputed predictions, and other biological data. Unison provides a practical solution to the burden of preparing data for computational discovery projects, enables holistic feature-based mining queries regarding protein composition and functions, and provides a foundation for the development of new tools. Unison is available for immediate use online via direct database connections and a web interface. In addition, the database schema, command line tools, web interface, and non-proprietary precomputed predictions are released under the Academic Free License and available for download at http://unison-db.org/. This project has resulted in a system that significantly reduces several practical impediments to the initiation of computational biology discovery projects.

1. Introduction

Computational biology projects frequently begin with the time-consuming process of downloading, formatting, characterizing, and integrating data of disparate types and sources [1]. These data preparation tasks require significant effort while contributing little to the distinguishing intellectual value of a project. Furthermore, these efforts are often duplicated for other projects, by other scientists, and even by the same scientist for the same project when new source data are released. The data preparation burden is a barrier to efficient and reproducible computational discovery efforts.

From a computational biologist's point of view, there are several important functional criteria for integrative databases: completeness and currency of the data, the breadth of source data and data types, query speed, technical accessibility, legal accessibility (*i.e.*, licensing restrictions), and the extent of semantic integration. Semantic integration means that data of the same type are modeled similarly and that appropriate relationships are established among data, regardless of source. Importantly, users experience semantically integrated databases as a representation of familiar concepts rather than a collection of proprietary data models (even if freely available). Reliable reasoning with data necessitates consistent, well-defined and well-understood definitions of the modeled data.

There are many techniques for data integration (for review, see [2][3]). Link integration, such as that provided by web pages, enables users to follow

prescribed links between data sources. Although link integration is useful for web browsing, it is insufficient for reliable and semantically precise querying. Similarly, full text indexing greatly facilitates searching, but it is insufficient for the reliable integration of concepts.

Database federation semantically integrates data that are stored remotely. The principle advantage of federation is that queries are always based on current data, but the drawbacks are the run-time dependencies on external resources, poor query performance compared to that for locally stored data, and the difficulty of devising interfaces that translate the semantics of the remote data models to those of the integrated schema.

Data warehouses, in which source data are aggregated within one database environment, eliminate external database dependencies and generally provide better query performance than other integration methods. The simplest data warehouses replicate source data locally but provide little or no semantic integration of the data. Because data are local, performance is improved relative to that of remote databases. More frequently, data warehouses provide a semantically integrated schema by creating database views to local replicas of source data ("view integration" [3]) or by materializing source data within an integrated schema during loading.

Despite the existence of several high-quality and well-known integrative databases (*e.g.*, ATLAS [4], BioMart [5], InterPro [6], RefSeq [7], STRING [8], UniProt [9], and others), data preparation for new computational discovery projects remains burdensome. There are technical, practical, and legal reasons that current databases do not meet the needs of computational biologists [2][3]. A few of these reasons are: limited content/project specificity, inaccurate or out-of-date data, limited access methods, necessity for local deployment, and licensing restrictions. Although no system will meet all needs of all users, there is a significant and unmet need for a standardized integration platform that lessens the data preparation burden of a broad audience.

Unison is a comprehensive data warehouse of a superset of nearly all available protein sequences (currently, 12M from 20 sources), extensive precomputed proteomic predictions (200M of 18 distinct types), and diverse auxiliary data. Unison includes predictions of protein domains and motifs, signal and transmembrane domains, secondary and tertiary structure, disorder, cellular localization, phosphorylation, and genomic alignment and clustering. The motivation for Unison is to lessen the data preparation burden of computational biologists by providing a standardized integration platform of commonly-used source databases and computationally expensive proteomic predictions. The integration enables the same resource to be used for traditional per-sequence domain analysis and for complex, holistic data mining queries regarding protein domain composition, structure, and function. Unison's schema is designed for incremental updating with respect to source sequences, models, and computational methods. The entire update process is fully automated. The

complete Unison package – schema, tools, web site, underlying database, and precomputed data – are freely available online for immediate use and for download.

2. Methods

2.1. *Schema overview*

Unison is principally a data warehouse of sequences, annotations, and precomputed predictions in an integrated schema. Familiar concepts that transcend multiple data sources, such as protein sequences and Hidden Markov Model (HMM) alignments, are modeled as abstract types in the semantically integrated core schema. Specialized data, such as NCBI GeneRIF and SCOP, are incorporated as auxiliary data without remodeling.

One of the tenets of Unison's core schema is to represent only the salient features of an entity or concept rather than to fully represent the character of each source database. As a result, most tables in the Unison schema are concise representations of the essential features of a biological entity or piece of information. This design decision is consistent with the goal of including data only when it is likely to inform queries or analysis. Links to source data enable users to pursue source-specific content. Unison makes extensive use of table inheritance to model abstract data types and their concrete descendants.

Normalization is essential to enabling efficient incremental updates of source data, which is one of Unison's primary goals. Nearly all tables in Unison are third normal form [10]. In rare cases when normalization leads to unacceptable performance loss, materialized views are provided. In practice, this level of normalization is readily understood by Unison's users and does not impose a barrier to use. Nonetheless, Unison provides a functional layer of views that provide a simplified, pragmatic, and stable query interface for users.

2.2. *Essential schema objects*

The essential objects modeled within the Unison schema are shown in Figure 1 and described below. The full schema and schema documentation are available online.

The `origin` table stores the provenance of all data, the version of the source database, a Uniform Resource Identifier (URI) to information about the source (*e.g.,* a project's "home page"), a URI of the file or directory from which data were loaded, and a URI template that is used to construct web page links to source records. A flag also indicates whether the data are publicly distributable.

Protein sequences are stored non-redundantly in the `pseq` table and referred to by a the primary key, `pseq_id`. A database trigger on this table computes the MD5 checksum on the protein sequence during loading, and a uniqueness constraint on the checksum ensures that sequences are distinct. A second trigger prohibits updating or deleting any sequence so that predictions are

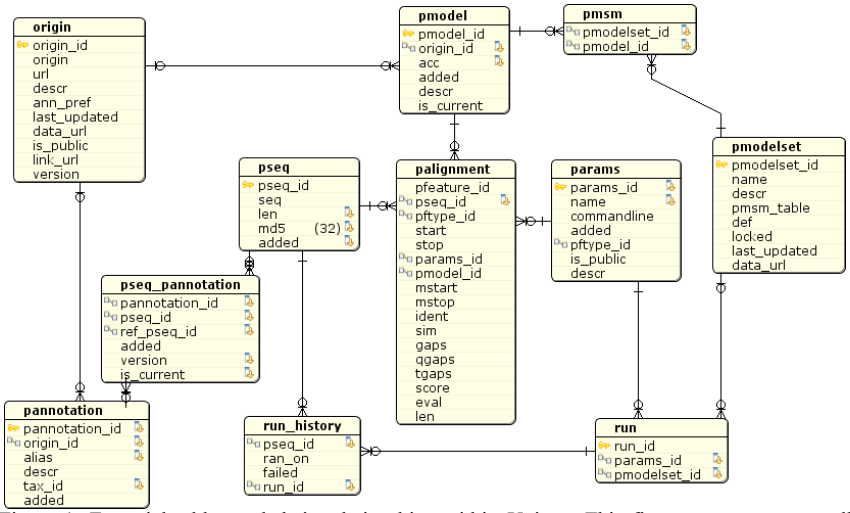

Figure 1: Essential tables and their relationships within Unison. This figure represents a small subset of the schema and a majority of the content of Unison.

guaranteed to reflect the stored sequence; sequence changes require inserting a new sequence.

The `pannotation` table represents the "alias" (accession or identifier from a source database), description, and taxonomic annotation of a sequence. A uniqueness constraint prohibits duplicate aliases within the same origin. The `pseq_pannotation` table maps and versions annotations in the `pannotation` table to sequences in the `pseq` table. Annotation versioning enables Unison to properly track obsolete aliases and changes to a sequence changes that is associated with a single alias. The current annotations for sequences are available through a view.

The `params` table represents prediction methods and their invocation. Each row includes a primary key, `params_id`, the command line that specified how a program is invoked, and a flag that indicates whether the results of the method are publicly distributable. The command line provides important traceability of all predictions in Unison.

Some prediction methods depend on additional external input, generically called models and represented using an abstract base table, `pmodel`. For example, Position Specific Scoring Matrices (PSSMs), HMMs, and Prospect Pro (http://bioinformaticssolutions.com/) protein structure templates are represented by `pmpssm`, `pmhmm`, and `pmprospect` respectively, each of which inherits from the abstract table, `pmodel`. The `pmodel` table and its derived tables contain a primary key, `pmodel_id`. Sets of models are represented by `pmodelset`.

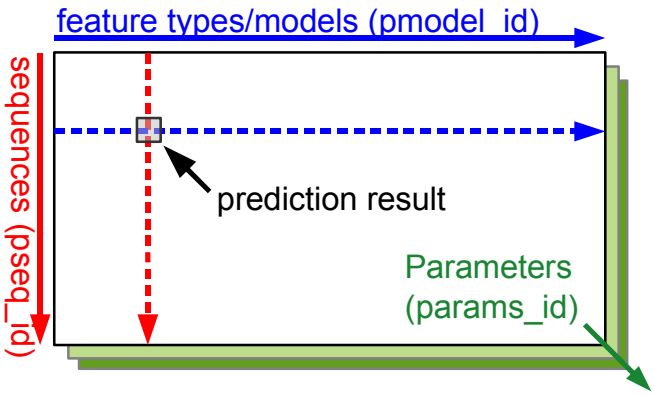

Figure 2: Visualization of the prediction results volume. The cube dimensions are unique sequences, feature types/models, and parameters. An analysis of protein features corresponds to horizontal slices of data. Feature-based mining corresponds to a vertical slices of data.

The `run` table is used to associate parameters and appropriate model sets, referenced by `params_id` and `pmodelset_id` respectively. The `params`, `pmodelset`, and `run` tables together permit Unison to distinguish how a method is invoked on a sequence and which models a sequence is run against.

Prediction results in Unison are represented using an abstract base table, `pfeature`, that represents a localizable feature on a protein sequence, specified by the foreign key `pseq_id`, using a specific prediction method, specified by the foreign key `params_id`. By including `params_id` in this table, Unison explicitly models alternative invocations of a predictive method and this, in turn, enables specialized or exploratory work using multiple prediction parameters with a single method. The `palignment` table is a subclass of `pfeature` that represents an alignment of sequence to a model, as specified by the foreign key `pmodel_id`. Results for each prediction method are modeled as distinct types by subclassing `pfeature` or `palignment` as appropriate and adding columns that are specific to the prediction method. Readers may benefit from visualizing the arrangement of prediction results as a volume of distinct predictions, each of which depends on sequence (`pseq_id`), method (`params_id`), and an optional model (`pmodel_id`), as shown in Figure 2.

Unison's schema includes a `run_history` table to track which sequence analyses – that is, runs from the `run` table – were performed on which sequences and the date of the execution. Run histories enable Unison to provide incremental updates with respect to new sequence releases, new versions of predictive methods, and new models/modelsets.

Hidden Markov Models and sequence alignments to them permit a concrete and representative discussion of the schema in practice. HMMs for HMMer [11]

are available from many sources, such as PANTHER [12], Pfam [13], Superfamily [14], and in-house efforts. Such sources are stored in the `origin` table. HMMs from those sources may be loaded into `pmhmm` with an obligatory reference to the model origin. Details specific to HMMer-built HMMs, such as score cutoffs, are included in the model table to enable advanced queries using these criteria. HMM alignments are represented using `pahmm` (Protein Alignment HMM), which is a subclass of `palignment` and therefore contains foreign keys to the query sequence (`pseq_id`), the parameters (`params_id`), and the HMM (`pmodel_id`), in addition to the positions of the alignment on the sequence and the model, the score, the E-value for the individual alignment, and other alignment details. When the alignments are loaded, the `run_history` table is updated to reflect this. The `pahmm_v` view provides a denormalized relation that simplifies the querying of HMM data.

2.3. *Loading and Updating Unison*

Most of the Unison's initial build process and all of the update process, including data downloads and submission of prediction jobs to a compute cluster, are automated by a series of makefiles and Perl scripts. The loading process consists of several phases that must proceed in order and without error before continuing to the next phase.

In phase 1, auxiliary data are loaded. These data include cytogenetic bands from UCSC [15], Gene Ontology [16], Human Genome Organization official gene names [17], NCBI Entrez Gene, GeneRIF, HomoloGene and taxonomy files [7], PDB [18], and SCOP [19]. The PDB schema includes structure summary data, ligands with canonical names and various descriptors, and an explicit sequence-to-residue mapping to facilitate the localization of primary sequence features on structures.

In phase 2, sequences and annotations are loaded from approximately 30 public sequence sources; the list of origins and currency is available at http://unison-db.org/contents.pl. Sequence loading from the commercial GENESEQ patent database [20] is also supported, but these data are not in the public release. Within Genentech, in-house sequences and annotations are also loaded.

In phase 3, three nested sets of sequences – runA, runB, and runC – prioritize sequences by origin, species, size and other criteria. Because predictions take widely varying time, expensive computations such as protein structure predictions are maintained for reliable human sequences of moderate size (runA); methods of intermediate computational cost are maintained for certain eukaryotic sequences from reliable sources (runB); and inexpensive computations are maintained for sequences from a wider selection of sources and species (runC). Ad hoc analyses of any type may be arranged for any sequence.

In phase 4, sequences are submitted to a compute cluster queue for analysis and the resulting data are loaded. The sequences to be run are determined by subtracting the set of sequences for which data are already available, as determined by the `run_history` table, from the run set that is appropriate for the analysis type. The 18 methods and command lines are listed at http://unison-db.org/contents.pl. Unison itself does not infer any features by sequence similarity, although many methods use sequence similarity internally.

In phase 5, materialized views that depend on phase 4 data are built. In phase 6, database summary statistics, maintenance, and cleanup are performed. The public version of Unison is built by specialized scripts that extract only the publicly distributable sequences, annotations, and predictions.

2.4. *Hardware and software requirements*

Unison is implemented in the PostgreSQL relational database running in a Linux/GNU environment. Deploying the database server and Unison on other operating systems should be straightforward. Installation requires approximately 200 GB of disk space. Although the extensive use of table inheritance and server-side functions in C and Perl would make porting to other database systems difficult, this is under consideration.

The Unison Perl API facilitates access to the Unison database and provides many convenience functions. The API makes extensive use of BioPerl [21] and other modules readily available from the Comprehensive Perl Archive Network.

The Unison web interface runs on Apache in a Linux/GNU environment. All web pages are currently implemented as Perl CGI scripts to enable Kerberos authentication. This configuration causes user authentication and authorization to be delegated to the database and security policies to be enforced there rather than in the middle tier. Neither registration nor authentication is required for the public version.

2.5. *Accessing Unison*

Direct database access to Unison is available at host `unison-db.org`, port `5432`, database `unison`, and username `PUBLIC` via the PostgreSQL native protocol, ODBC, JDBC (for Java), and SDBC (for OpenOffice). ODBC access within R has been tested. These connection methods are not specific to the client or server operating system. The number of concurrent connections and statement query times are limited to promote equitable access.

Web access is available at http://unison-db.org/. Unison web pages use URLs with simple "GET"-method query arguments, thereby facilitating programmatic linking to content. For example, all sequence analysis pages allow specification of the sequence by any recognized alias, protein sequence, MD5 checksum, or Unison's internal `pseq_id`. MD5 checksums provide an intrinsic key that obviates inter-database coordination of protein sequence accessions.

```
CREATE OR REPLACE VIEW itim_search_v AS
SELECT IG.pseq_id,
       IG.start as ig_start,IG.stop as ig_stop,
       TM.start as tm_start,TM.stop as tm_stop,
       ITIM.start as itim_start,ITIM.stop as itim_stop
  FROM pahmm_current_pfam_v IG
  JOIN pftmhmm_tms_v T ON IG.pseq_id=TM.pseq_id   AND IG.stop<TM.start
  JOIN pfregexp_v ITIM ON TM.pseq_id=ITIM.pseq_id AND TM.stop<ITIM.start
 WHERE IG.name='ig'
         AND IG.eval<1e-2
         AND IG.params_id=params_id('HMMer')
         AND ITIM.acc='MOD_TYR_ITIM';
```

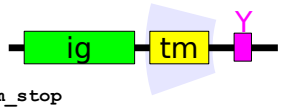

Figure 3. A canonical ITIM-containing protein and a query which mines Unison for such proteins. The query identifies sequences that contain an extracellular ig domain, a TMHMM transmembrane domain, and an intracellular Immunoreceptor Tyrosine Inhibitory Motif using the regular expression from ELM.

The Unison schema, tools, makefiles, web pages and documentation are released under the Academic Free License and available for local deployment. Non-proprietary data (sequences, annotations, and methods) are available as a PostgreSQL database dump. The source code repository is also publicly accessible. All users are advised to join the mailing list. Links to all of these resources are provided at the Unison web site.

3. Application of Unison to Immunoreceptor Tyrosine Inhibitor Motif (ITIM) proteins

A demonstration of the utility of a comprehensive set of sequences, precomputed predictions, and auxiliary data is best served by example. Immunoreceptor Tyrosine Inhibitory Motif-containing proteins are an important class of immune system regulators (see [22] for review). ITIMs are short (6-8 amino acid) sequences in the intracellular domain of immune receptors. ITIM-containing receptors were initially identified on the surface of NK cells and macrophages where they are believed to prevent self reactivity. Phosphorylation of the tyrosine within an ITIM leads to the binding of, for example, SHP1 or SHP2 phosphatases and the attenuation of a corresponding activating receptor.

Canonical ITIM-containing proteins possess an extracellular immunoglobulin domain and a transmembrane domain in addition to the intracellular ITIM. This simple biological model may be translated into a Structured Query Language [10] command that searches for such proteins within Unison, as shown in Figure 3. This query may be modified to use other HMMs, structure prediction or PSSMs to identify immunoglobulin domains, or to identify related Immunoreceptor Tyrosine Activating or Switch Motifs. This query takes approximately 20 seconds on an otherwise idle system.

Unison's extensive precomputed data enables queries that develop new hypotheses. The first known ITIM-containing proteins possessed immunoglobulin extracellular domains. Suspecting that diverged ITIM protein homologs might contain alternative extracellular interaction domains, Unison was used to identify and count all extracellular Pfam domains in the context of a

transmembrane domain and intracellular ITIM. Models of immunoglobulin subfamilies, fibronectin III, and cadherin domains, all members of the same beta-sandwich structural superfamily, stood out as obvious alternative extracellular domains and informed subsequent searches that resulted in several candidates, at least one of which was later demonstrated to be an ITIM protein.

The HMM, TM, and ITIM criteria used in Figure 3 are individually non-specific. In such cases, evolutionary conservation provides compelling evidence for the functional significance of domains or motifs. The query in Figure 3 may be extended using the HomoloGene data within Unison to identify Human ITIM-containing proteins that have an ortholog of similar domain composition. The queries in this section are available online in the Unison tutorial.

4. The Unison web interface

The Unison web interface presents an easy-to-use portal to much of Unison's content. In addition to the sequence analysis pages, several special-purpose tools are likely to be useful to a broad audience. BabelFish is Unison's sequence accession translator. For a list of protein sequence accessions, identifiers, and/or MD5 checksums, BabelFish returns a table of identifiers and links to other databases. This tool is aimed at users who have protein accessions from gene expression analysis, mass spectrometry or other high-throughput method and wish to identify identical sequences in other databases.

Unison's AliAn tool provides tabulated protein annotations. For a set of protein accessions, identifiers, and/or MD5 checksums, AliAn will return a table of "best" (*i.e.,* most reliable) annotations, NCBI Gene information, human locus, Gene Ontology terms, Affymetrix probes, and other data. As with BabelFish, this tool is intended for users who have protein identifiers from high-throughput experimental methods.

The Jmol-based structure viewer, shown in Figure 4, enables users to explore the structural context of sequence features. Clicking on features or SNPs in the causes those features to be highlighted on the structure. The URL syntax enables users and programmers to define and display features not currently in Unison. The same interface can also be used to display protein structure prediction results, but these data are not included in the public release.

5. Discussion

Unison is a data warehouse of 12M protein sequences from 20 sequence sources, 200M precomputed predictions of 18 distinct types, and an extensive collection of auxiliary data. The integration of these data facilitates computational discovery projects that require mining based on protein domain composition. Upon this foundation, Unison also provides a variety of web and command-line tools that address "real world" problems such as the annotation of large sets of protein accessions from high-throughput data. A Firefox browser "search plugin"

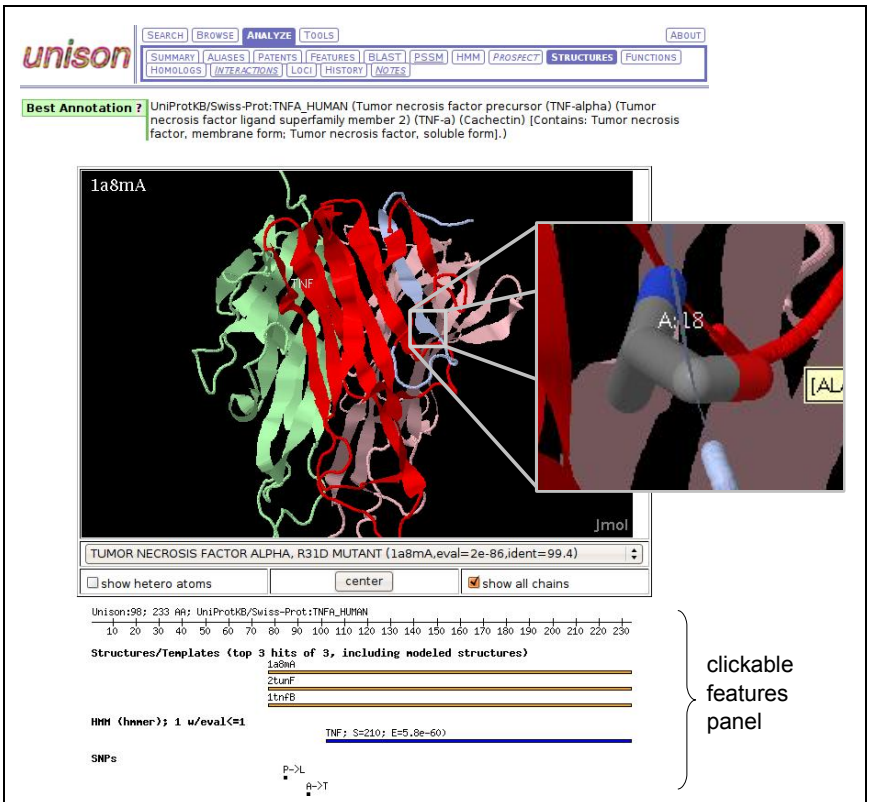

Figure 4: Unison structure tab for Tumor Necrosis Factor alpha (TNFA). Clicking features in the lower panel highlight the corresponding structural region, as shown for the Pfam domain (red) and for a SNP (inset).

is available at the Unison web site. Within Genentech, Unison has improved the consistency and currency of data used in discovery and sequence analysis projects for five years. It has become a foundation upon which many in-house tools and projects are based, including a recent discovery of four novel Bcl-2 family members in zebrafish [23], a systematic study of the evolutionary constraints of single nucleotide polymorphisms in terms of protein structure and function [24], and a comprehensive search for novel immune system regulators (in preparation).

Unison uses two approaches for data integration. Familiar concepts are abstracted into well-defined types within the semantically integrated core schema. An important advantage of Unison's integrated schema is that the table definitions are transparent to users, which facilitates semantically correct queries. Unison incorporates other useful data into auxiliary schema. The entire Unison database is updated incrementally and automatically by build scripts; at Genentech, Unison is updated biweekly by a Unix cron job.

Unison excels at identifying sequences that match combinations of features. Feature-based mining necessitates a comprehensive set of sequences, including speculative sequences beyond those in curated databases, and precomputed features for those sequences. The depth of Unison's content enables expert-level queries that are not possible in other systems, such as filtering HMM alignments based on the "trusted cutoff" score.

Unison is a practical solution to a significant problem and it addresses many common shortcomings of integrative databases [2][3]. Unison data models are concise and readily understood. The content and currency of source data, and the status of sequence analyses, are stored in the database and easily discerned by users. Unison is automatically updated. Documentation, a bug report system, and an announcement list are available to facilitate corrections and communication. Unison is available for immediate use online through a direct database connection and a web interface. Unison is released under an approved Open Source™ license and is available for local installation. Sites may load additional sequences and predictions into a local deployment using the tools included with the source package.

Several new features are being considered for forthcoming releases of Unison, including free text searching, pathways and gene expression data and new prediction types from InterPro. Several web interface improvements and a distributed annotation system [25] server are in progress. Protein family classification is being investigated using machine learning and the diverse predictions that are available in Unison.

Acknowledgments

Unison was made possible by numerous Open Source projects and by the freely available contributions of data and methods by scientific colleagues. RKH thanks Matthew Brauer and Josh Kaminker for comments on the manuscript, the Bioinformatics and Protein Engineering department for feedback on Unison, and Simran Hansrai and Dave Windgassen for exceptional computing support.

References

1. L. D. Burgoon and T. R. Zacharewski, *Methods Mol. Biol.* **460**, 145 (2008).
2. S. Philippi and J. Köhler, *Nat. Rev. Genet.* **7**, 482 (2006).
3. L. D. Stein, *Nat. Rev. Genet.* **4**, 337 (2003).
4. S. P. Shah, Y. Huang, T. Xu, M. M. S. Yuen, J. Ling and B. F. F. Ouellette, *BMC Bioinformatics* **6**, 34 (2005).
5. S. Durinck, Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma and W. Huber, *Bioinformatics* **21**, 3439 (2005).
6. N. J. Mulder and R. Apweiler, *Curr Protoc Bioinformatics* **Chapter 2**, Unit 2.7 (2008).
7. D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, L. Y. Geer,

Y. Kapustin, O. Khovayko, D. Landsman, D. J. Lipman, T. L. Madden, D. R. Maglott, J. Ostell, V. Miller, K. D. Pruitt, G. D. Schuler, E. Sequeira, S. T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, R. L. Tatusov, T. A. Tatusova, L. Wagner and E. Yaschenko, *Nucleic Acids Res.* **35**, D5 (2007).
8. C. von Mering, L. J. Jensen, M. Kuhn, S. Chaffron, T. Doerks, B. Krüger, B. Snel and P. Bork, *Nucleic Acids Res.* **35**, D358 (2007).
9. The UniProt Consortium, *Nucleic Acids Res.* **35**, D193 (2007).
10. C. J. Date. An introduction to database systems. Addison-Wesley (1999).
11. R. Durbin, S. Eddy, A. Krogh and G. Mitchison. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press (1998).
12. H. Mi, N. Guo, A. Kejariwal and P. D. Thomas, *Nucleic Acids Res.* **35**, D247 (2007).
13. R. D. Finn, J. Mistry, B. Schuster-Böckler, S. Griffiths-Jones, V. Hollich, T. Lassmann, S. Moxon, M. Marshall, A. Khanna, R. Durbin, S. R. Eddy, E. L. L. Sonnhammer and A. Bateman, *Nucleic Acids Res.* **34**, D247 (2006).
14. D. Wilson, M. Madera, C. Vogel, C. Chothia and J. Gough, *Nucleic Acids Res.* **35**, D308 (2007).
15. http://hgdownload.cse.ucsc.edu/goldenPath/.
16. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin and G. Sherlock, *Nat. Genet.* **25**, 25 (2000).
17. E. A. Bruford, M. J. Lush, M. W. Wright, T. P. Sneddon, S. Povey and E. Birney, *Nucleic Acids Res.* **36**, D445 (2008).
18. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne, *Nucleic Acids Res.* **28**, 235 (2000).
19. A. Andreeva, D. Howorth, J. Chandonia, S. E. Brenner, T. J. P. Hubbard, C. Chothia and A. G. Murzin, *Nucleic Acids Res.* **36**, D419 (2008).
20. http://scientific.thomsonreuters.com/pharma/geneseq/.
21. J. E. Stajich, *Methods Mol. Biol.* **406**, 535 (2007).
22. E. Vivier, E. Tomasello, M. Baratin, T. Walzer and S. Ugolini, *Nat. Immunol.* **9**, 503 (2008).
23. E. Kratz, P. M. Eimon, K. Mukhyala, H. Stern, J. Zha, A. Strasser, R. Hart and A. Ashkenazi, *Cell Death Differ.* **13**, 1631 (2006).
24. J. Liu, Y. Zhang, X. Lei and Z. Zhang, *Genome Biol.* **9**, R69 (2008).
25. R. D. Dowell, R. M. Jokerst, A. Day, S. R. Eddy and L. Stein, *BMC Bioinformatics* **2**, 7 (2001).