Integration of the Comprehensive Microbial Resource into the BioSPICE Data Warehouse

Jessie Tenenbaum
Department of Biomedical Informatics
Stanford University

BioMedIn 231: Computational Molecular Biology
Final Project
Professor Doug Brutlag
December 2002

Introduction

As part of the doctoral training program in Biomedical Informatics, I did a quarter-long research rotation with Dr. Peter Karp at SRI International, a nonprofit research institute "committed to discovery and to the application of science and technology." One focus of Dr. Karp's research group is the BioSpice Data Warehouse (BDW). It is maintained for the BioSpice[1] community as "an environment for constructing bioinformatics database warehouses that collect together a set of public and private bioinformatics databases into one physical relational database management system."[2]

The model for the BDW project is as follows: SRI provides a web-accessible database containing data from multiple existing data stores, including but not limited to CMR, KEGG, GO, Genbank, and Swissprot. SRI also makes available the loader tools used to get these data sources into the BDW. Any given user or research group may then access

the data in SRI's instance of the Warehouse, or create their own database, independently license the data from any desired data sources, and use our import tools to load the desired data. Some advantages to the latter option are: greater flexibility in information retrieval, free access to otherwise potentially limited data, and the ability to incorporate data from their own lab. Though the current version is implemented in an Oracle database, plans for the future include support for MySQL, a freely available relational database server.

My project goal was to import data from the Comprehensive Microbial Resource [3] (CMR) database, maintained by researchers at The Institute for Genomic Reseach, into the BioSpice Data Warehouse. The project involved matching concepts between the two ontologies, mapping fields between the two schemas, modifying the BDW ontology and database schema where necessary to accommodate CMR data, and developing and documenting the import tools used in the process. This report covers 1) the motivation for this research, 2) the computational issues and 3) the methods and results of my solution.

Motivation for the Warehouse Database

Biologists today are limited less by the amount of data available for analysis, and more by the ability of the human mind to synthesize, visualize, and analyze the data that is available. Researchers have attempted to address these relatively new limiting factors in a number of ways. One example is PubMed, an indexed text retrieval tool. This resource allows the researcher to access articles using string matching and search criteria both for article text and for key fields such as title, author, and date of publication. Another is the use of amalgamated databases that contain data gathered in labs around the world, for example SwissProt and GenBank. Each of these databases has its own set of semantics, level of abstraction of objects stored, file formats, and access mechanism.[4] This can make it difficult to perform research that needs requires access to data from multiple sources. Additionally, researchers often face substantial obstacles to integration of local data with existing information from public databases.

The stated goal of the BDW project is to "create an environment/toolkit for constructing bioinformatics database warehouses that collect together a set of bioinformatics databases into one physical relational DBMS."[5] It thus addresses the issues above first by gathering data from a number of sources and then providing access to this information through a uniform mechanism. Data types include information on genes, proteins, enzymes, reactions, gene expression, and organism taxonomy, among others. These heterogeneous data types are mapped into a single global ontology that also supports relationships between data, enabling researchers to make computationally derived inferences among data from different sources. Finally, the complete ontology for BDW is freely available to researchers, enabling them to create mappings from their own data for integration purposes. Once integrated, their data can be queried and analyzed within the context of the rest of the data already stored in BDW.

Need for Schema modifications

Consider the following research scenario: a scientist works in a lab with a focus on the human genome. She is attempting to study the functionality and regulation of certain human genes based on homology to known genes in other organisms. To do this work, she will need access to gene sequence data, both in humans and in other organisms. Additional relevant data may include gene function and transcription, homology, and pathway information. As mentioned above, much of this information already exists in public databases, and BDW already provides a single, unified retrieval mechanism for it. However, BDW does not currently include sequence or homology information. The CMR is a perfect source for this because it provides both of these data types, as well as a largely new domain of organisms not formerly represented in BDW. Inclusion of this new information in BDW necessitates two different types of changes to the existing ontology and consequently to the underlying database schema: (1) modification of existing objects, for example by adding the notion of sequence, and (2) creation of new object types, for example for alignment pairing information.

Computational Issues

The data in BDW and CMR are stored in different database formats. CMR uses a Sybase database, while BDW uses Oracle. This difference created the need to parse and translate the CMR data. My collaborator at TIGR used existing stored procedures in the Sybase DB to export table contents into delimited text files. The tools I wrote, which will be made available to interested parties who wish to incorporate CMR data into their own instance of BDW, were then used to parse the files in order to populate tables in our Oracle database.

Integration of disparate ontologies presents several challenges, specifically with respect to resolving mismatches between the two, both at the "language level" and at the "ontology level," or what we might think of as the semantic level.[6] At the language level, one may run into issues such as difficulty in one language to express disjointedness, or the inability of one to express negation. At the ontological level, disparities are observed in scope, coverage, and conceptual paradigm. Because CMR data in this case was reduced to text file format, the issues faced were primarily at the ontology level.

The first real challenge was in simply understanding the CMR schema. TIGR provides a graphical representation of their schema. Tables names include *asmbl_data* and *egad*; column names include *feat_method* and *ed_pri*. To someone familiar with the CMR schema, these names are logical, but on preliminary investigation, they provide limited information regarding the associated data. This obstacle was resolved through a request for more thorough documentation, as well as multiple long distance phone call tutorials.[*] In addition, the CMR schema has undergone the somewhat inevitable evolution process common to large scale software projects. A large number of fields are no longer used,

[*] An interesting commentary, I believe, on how many gigabytes of data a little human interaction can be worth.

often not even populated. Both fields and relationships have evolved to the point where the same information may have a slightly or completely different name in different tables and parent/child relationships are overloaded with peer-to-peer relations.** Having finally deciphered the puzzle that is the CMR schema, I came to the hard part: mapping CMR objects and data fields to those in BDW.

The integration process had a few distinct issues to be resolved. The first involved concepts from CMR that did not correspond to any existing elements in BDW. This situation was relatively easy to fix by simply expanding the BDW ontology to account for these new objects. When a table did not exist, I was able to examine the data that needed to be stored, think about future generalization, and design the best possible schema in which to store it. For example, a table in CMR contains the results of a pre-computed BLAST protein alignment that TIGR runs on all proteins of all organisms in their database. Any two molecules with greater than 40% similarity are stored as a pair in this table, along with the percent similarity, percent identity, match length, pairing rank, and P-value of the match. To date, the BDW ontology had no notion of such alignment pairings. In creating such an object, I generalized field names and definitions so that they would be applicable moving forward not just for proteins but nucleic acid sequences as well, and for other sequence alignment algorithms.

In other cases, a data object from CMR already had a corresponding concept in BDW, but there was a mismatch in what attributes were stored, how they were stored, or the level of abstraction of the data types to which they applied. For example, the Warehouse already had a table called *Feature* that is used to store features associated with protein molecules. CMR contains a table called *asm_feature* which also stores feature

** With no disrespect meant to my main contact Tanja Davidsen, who has been nothing but competent, professional, and helpful throughout the process.

information, but here feature applies to genes instead. Also, CMR has a broad definition of what constitutes a DNA feature, including the gene itself. I had the option to expand the scope of the BDW concept of Feature to include gene features as well as those that apply to proteins, or create a new table for gene features. Additionally, CMR stores their sequence data as an attribute of two different data objects: gene feature and assembly, neither of which has a direct counterpart in BDW. A decision needed to be made as to what level should we store sequence data so as to maximize efficiency of common queries while minimizing space needed for storage.

Once it was established how data objects in the CMR ontology map to those in the BDW, and how individual CMR data fields map to the BDW schema, I needed to decide exactly what data to import. At one end of the spectrum would be importing the entire database. That is, for every column and row in CMR, copy this information into the appropriate table, existing or new, in BDW. This would seem like the wrong thing to do, given that even the TIGR folks don't believe that all of the data is actually useful. At the other extreme would be to import only the *all_vs_all* table. However, to import only the data in this table is impractical because we lose all context (i.e. NCBI source organism information) of what these sequences actually represent.

Solution Methods and Results

Process

Obtaining data from CMR involves the following steps: A stored procedure is used to dump the contents of each table to a delimited text file. These files, some with as many as 30 million rows, are zipped and made available on TIGR's ftp site. Zipped files are ftp'ed to a server in the domain of the interested party. Thus far, this party has been only myself at SRI. In the future it may include anyone with an interest in adding CMR data to an instance of the Warehouse. These files are then unzipped and parsed via tools written primarily in C using embedded SQL, and parsed data is written to the BWD Oracle database.

The parsing tools read in the delimited text files, interpreting each row of the file as a row in the database. Information from the file is stored in a temporary data structure, which is then inserted into the database. Additional columns in the new entry may be filled in with information from other table parsed later in the process. In addition to writing the tool itself, I authored the CMR Loader Semantics document to describe the CMR ontology in detail, and to document the final field mapping and schema modification.

Semantics

Following are visual representations of the conceptual schemas to date.

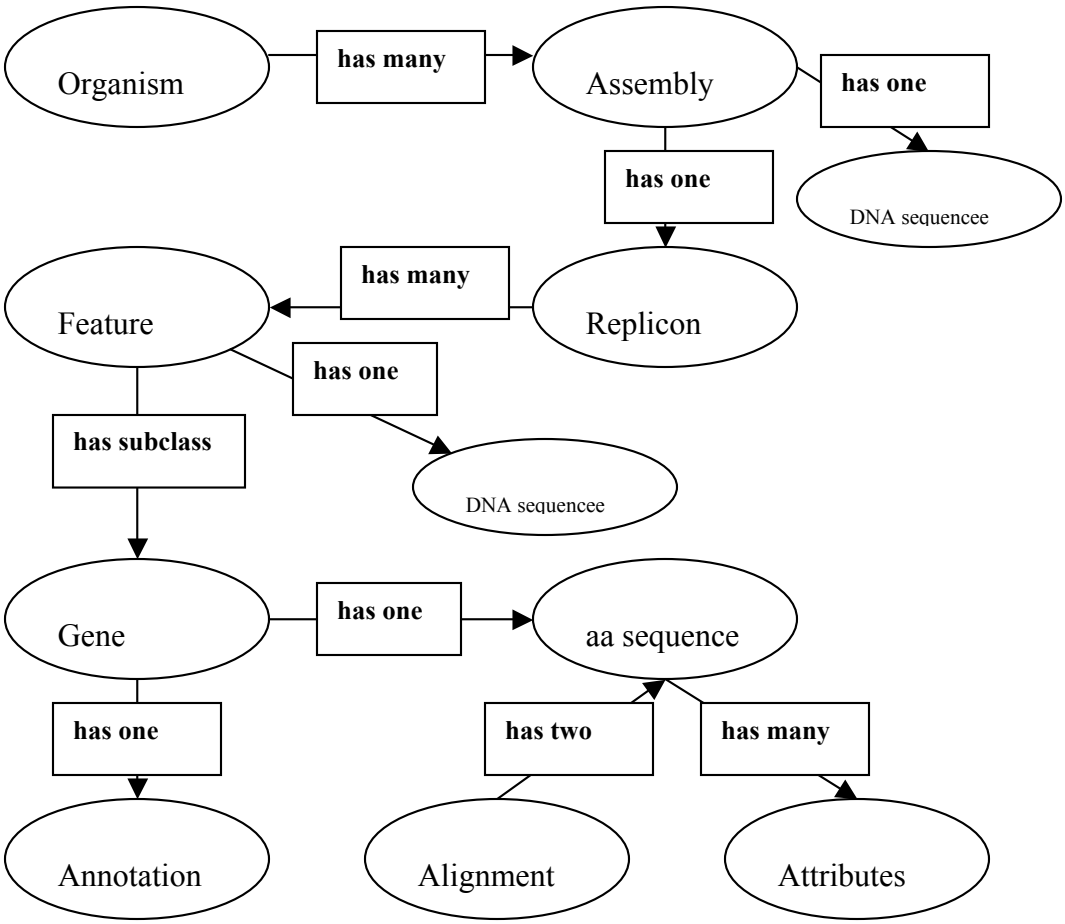


Figure 1.The world according to TIGR

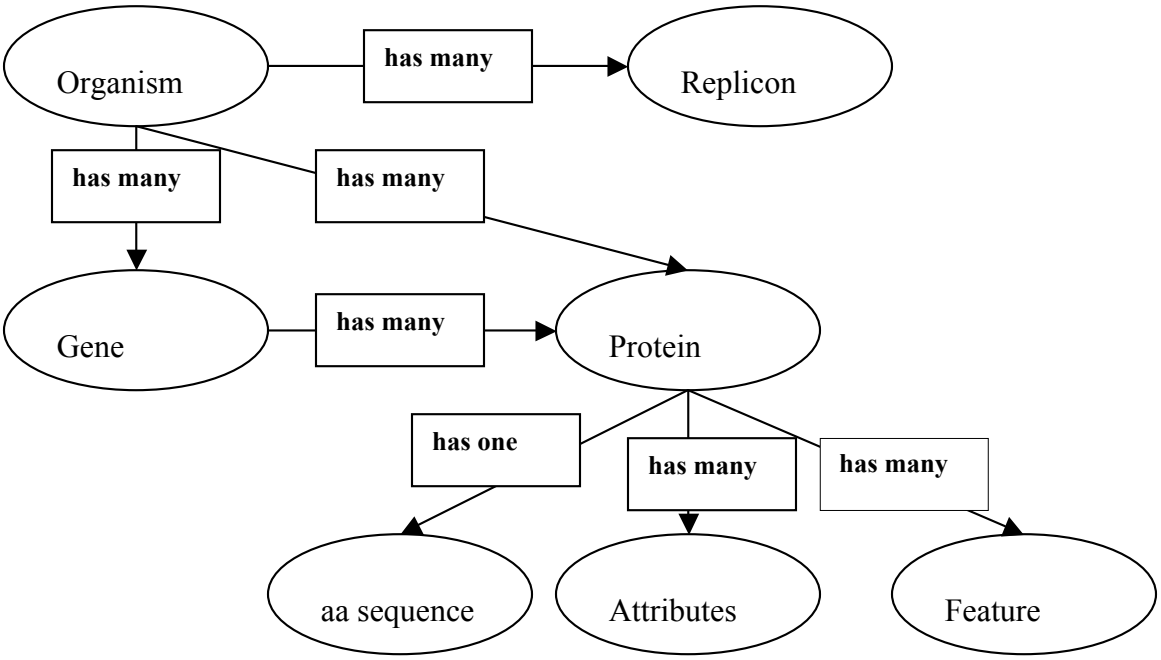


Figure 2. The world according to SRI *before* CMR assimilation

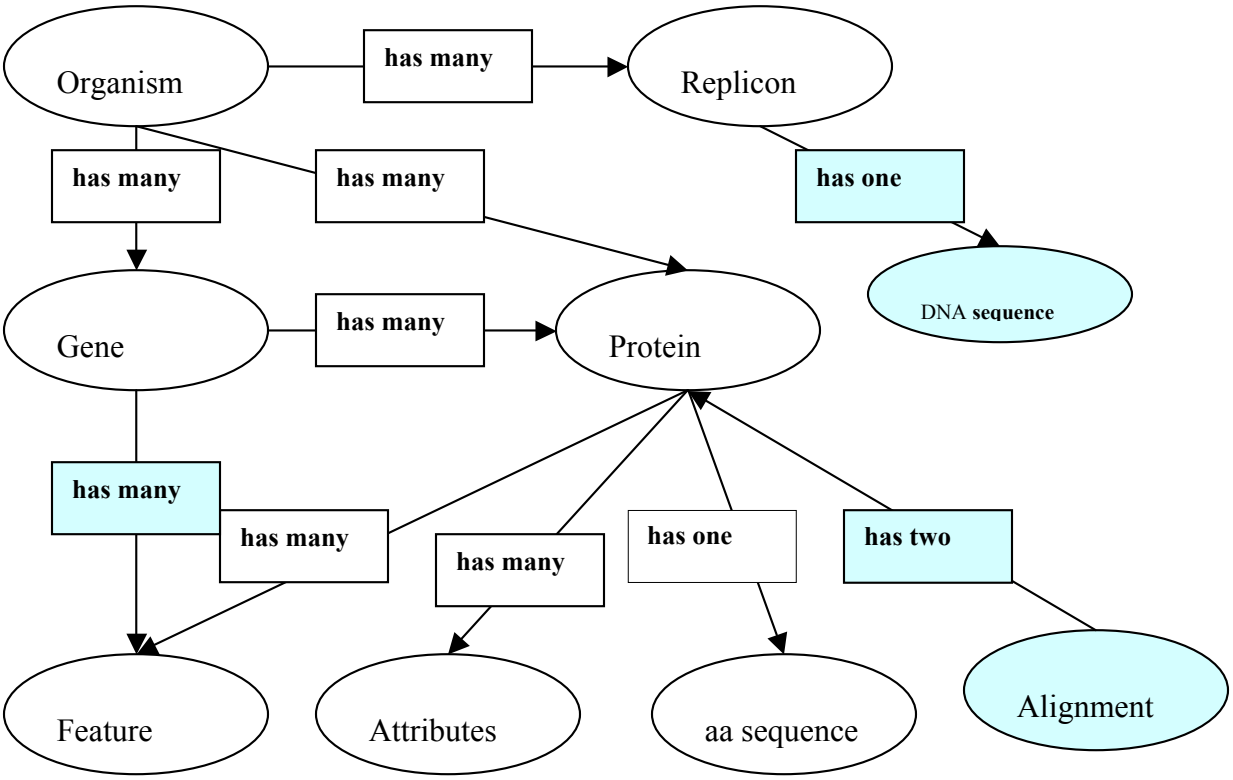


Figure 3. The world according to SRI *after* CMR assimilation

Our primary reason for inclusion of CMR is their model of protein BLAST alignment data, as described above. Our goal in obtaining this data is two-fold: first, the data in itself is interesting. Second, accommodation of this new data motivated modifications to our ontology not only to handle this specific data, but also to allow for additional imports in the future. That is, I deigned new schema not just for protein BLAST alignments, but for any such alignment algorithm, against any amino acid or nucleic acid sequence.

As mentioned above, new data also included gene feature information. Having determined that the fields in the existing [protein] *Feature* table were similar enough to those that would be required for gene features, instead of creating a new table I simply added a new MoleculeType column to the *Feature* table to indicate this distinction. I also

changed the column formerly called “Type” to “FeatureType” to reduce confusion with this new field.

The newly acquired sequence data necessitated a more drastic change to another previously existing table. The BDW *Replicon* table contained information on chromosomes and plasmids such as name, topology, and sequence length. I expanded the scope of this table, renaming it *NucleicAcidSequence*, and added a column for Sequence. This table is now used for any stretch of nucleic acid base pairs for which we might want to store sequence information. For any given DNA sequence we import from CMR (from the *asm_feature* table), that sequence is added to the *NucleicAcidSequence* table, in addition to an entry being created in protein for the amino acid sequence associated with it, and a corresponding entry is created either in our *Gene* table or in the newly expanded *Feature* table depending on sequence type (indicated by their feat_type field). Sequences representing an open reading frame are considered genes, while sequences representing, for example, ribosomal binding sites or terminator sequences are considered features.

In the end, what data gets imported will depend on resource constraints, specifically time remaining before the new quarter. The plan is as follows: import alignment data, protein and organism information for each protein, all features for which CMR has DNA sequences (which is not all of them), and a basic level of annotation. Notably absent from this recommendation: GO roles, paralogous family alignment information, functional roles, extended annotation, extended taxonomy information, and full assembly sequences. These latter data types may certainly be useful and should be considered in the future, perhaps in the course of another graduate student rotation.

In the course of importing the databases mentioned above, obviously a number of genes and proteins will be duplicated. We make no effort to resolve such duplicates. Another potential area for future work would be to cull the data for apparent duplicates and see if any new insights are to be gained, for example checking for when we might expect to find duplicates but don’t.

Discussion

Throughout this course, Biomedical Informatics 231, we have learned about literally dozens of bioinformatics databases available on the web, each with its own specific dataset, each with its own user interface (much to the chagrin of those of us in cyberland, squinting at the 2" x 2" demo). There is some, even a considerable amount of cross referencing between the databases, and yet the hassle remains- we need an entire course devoted to just learning what tools are out there. In an ideal world, a tool like the BioSpice Data Warehouse could solve this problem. The user can got to this one site and find data from any database we've discussed, all merged into one common ontology, all accessible through one uniform interface. In a decidedly sub-optimal world, BDW becomes just one more demo in future instances of this course.

What sets BDW apart is the ability, promoted from its inception, to customize it for whatever new conceptual data may be desired. This means both that SRI's instance of the Warehouse can continue to expand, and that researchers can customize individual instances for their own purposes. Whichever approach is chosen, and whether the Warehouse proves as useful as tools such as GenBank or SwissProt, science will benefit when users are enabled to spend less time learning how to use all the tools that are out there, and more time actually putting them to use.

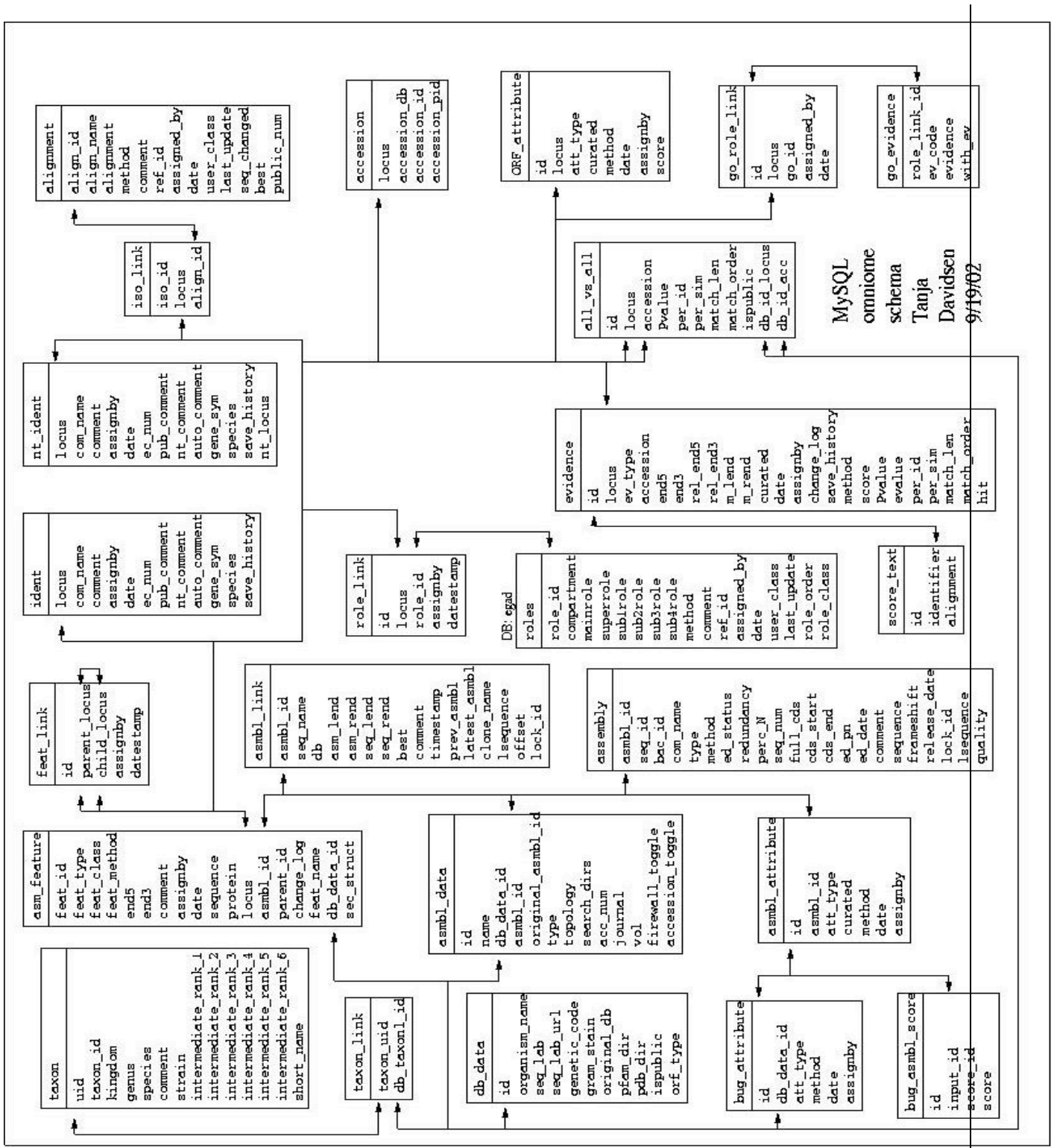
Appendix A: Field Mappings

The following table shows the columns in BWD that were populated with CMR data and where in CMR that data was stored. Newly populated columns with repetitive or auto generated data is not included. For example, every table in BWD has a WID column (Warehouse ID) which serves as a primary key and unique identifier for the entity, and many tables have a DataSetID field to indicate from where the data was obtained.

BWD Table	BWD Column	CMR Table	CMR Column
Organism	Name	db_data	Organism_name
	NCBI	taxon_link	taxon_uid

NucleiAcidSequence	Name	asm_feature	Locus
	Circular	asmbl_data	topology
	GeneticCodeNumber	db_data	genetic_code
Gene	Name	ident/nt_ident	com_name
	GenomeID	asm_feaure	locus
	StartPosition	asm_feature	end5
	EndPosition	asm_feature	end3
Feature	Name	asm_feature	feat_name
	FeatureType	asm_feature	feat_type
	StartPosition	asm_feature	end5
	EndPosition	asm_feature	end3
	MoleculeType	Always Nucleic Acid for CMR data	
Protein	Name	asm_feature	locus
	AASequence	asm_feature	protein
	MolecularWeightCalc	orf_attribute	score
	PICalc	orf_attribute	score
Alignment	Molecule1	all_vs_all	locus
	Molecul2	all_vs_all	accession
	PerSim	all_vs_all	per_sim
	PerID	all_vs_all	per_id
	PValue	all_vs_all	Pvalue

Appendix B: CMR schema



[1] BioSpice is an organization devoted to the development and sharing of tools for biological research, especially in the area of cellular network modeling.. See http://www.biospice.org/

[2] BioSpice Data Warehouse announcement, http://www.biospice.org, October 22, 2002

[3] J.D. Peterson, L.A. Umayam, T.M. Dickinson, E.K. Hickey and O. White. The Comprehensive Microbial Resource. Nucleic Acids Research, 29:1 (2001), 123-125

[4] BioSpice Project Team, SRI International. In A Bioinformatics Data Warehouse: Conceptual Requirements. Unpublished Draft Version 0.1. February 22, 2002.

[5] BioSpice Project Team, SRI International. In A Bioinformatics Data Warehouse: Conceptual Requirements. Unpublished Draft Version 0.1. February 22, 2002.

[6] Michael Klein. Combining and Relating Ontologies. In Proceedings of the Workshop on Ontologies and Information Sharing, IJCAI '01, Seattle, USA, August 4-5, 2001.