PhyloSource: An Interactive Phylogenetic Relationship Repository

Dong-Guk Shin[1], John Blius[1], Ravi Nori[1], Hsin-Wei Wang[1], Pinglei Zhou[1], and J. Peter Gogarten[2]

[1]Computer Science and Engineering
University of Connecticut
Storrs, CT 06269-3155

[2]Molecular and Cell Biology
University of Connecticut
Storrs, CT 06269-3044

1. Problem

Use of phylogenetic relationship computation tools such as ClustalW (Thompson et al., 1994) can be very effective in computationally analyzing gene functions. ClustalW produces the phylogenetic distances between the target gene and its homologs. The outcome of the program can be presented in a tree-like style. This graphical display allows scientists to visualize evolutionary relationship between similar genes from multiple species. Knowing such evolutionary relationships can be very useful in understanding functional variations between homologous genes, identifying subfamily-specific sequence features, and conducting advanced study of evolution such as *mosaic* genomes (e.g., Koonin et al., 1997; Olendzenski et al., 1998) and horizontal gene transfer (e.g., Doolittle, 1999; Olendzenski et al., 2000). One problem in utilizing the evolutionary relationships in the gene function study is the sheer amount of computational work that a scientist needs to undertake. One would have to gather completed species sequence data from the primary sequence data source(s) such as GenBank. She then would have to run programs like BLAST and ClustalW in tandem repeatedly. Conducting such computationally extensive processes may take days or weeks depending on the computational power the scientist is equipped with. In addition, maintaining intermediate and final analysis data would require use of scalable repository system such a database management system. Another problem is the data dependency and extension. Whenever new sequence data is deposited to the primary sequence source, new computation should be done to derive the most up-to-date phylogenetic relationships. Yet another problem is that once phylogenetic relationships are derived, there should be a *systematic* way of associating this data set with other related data.

Considering all these computational issues, taking advantage of phylogenetic relationships for gene functional studies is not something that an ordinary biology laboratory would easily be able to undertake. We believe that it is desirable for one group with sufficient computing resources to develop a comprehensive repository containing phylogenetic relationships and become responsible for maintaining and sharing it with the rest of the genome analysis community. We made an effort to develop such a repository and it resulted in, what we call, PhyloSource. Currently it contains phylogenetic relationships of 21 species whose sequencing data has been reported complete or near complete. The goal of this work is to present the computation issues that we have dealt with in developing this repository, the methodology that we were able to successfully adopt, and the challenges that we are currently facing.

2. Approach

In developing PhyloSource, we emphasized two issues: (i) there should be an automated way of dealing with the data dependency and extension, and (ii) users should be able to access this repository through an intuitive query mechanism and yet this data access mechanism should also permit seamless interoperability with third party data resources.

2.1. Overview of System Architecture

Figure 1 shows the overview of the system that aims at systematically computing phylogenetic relationships using the publicly available sequence data. The overall system is divided into three major subsystems: the data assembly module (Figure 1.a), the data service module (Figure 1.b) and the user interface module (Figure 1.c).

The data assembly module is responsible for producing phylogenetic relationships from the raw sequence data through a sequence of analysis processes and finally depositing the relationships in the Oracle database. For the sake of comparison, we developed two data assembly methods, one is to use the

NCBI's archive of pre-computed BLAST hits mostly through web page parsing, and the other is to conduct BLAST comparisons locally using raw sequence data available from the NCBI's ftp site. Figures 1.a.1 and 1.a.2 summarize, respectively, the two methods. These two methods produce two alternative versions for the phylogenetic relationships. This data assembly pipeline is described in more detail in Section 2.2.

The data service module undertakes the middle layer service functions. Two such functions are running a ClustalW server and running a Java RMI server. The RMI server is responsible for funnelling SQL queries against the Oracle database and passing the query results back to the users. The ClustalW server computes, in real-time, distances between multiple gene sequences after it retrieves the sequences from the Oracle database.

The user interface module is a thin client program that is embedded in a web page. This module has been developed as a Java Applet. This module is responsible for providing intuitive ways of browsing and querying the phylogenetic relationships stored in the Oracle database. The user interaction is done mostly via graphical object manipulation. This module includes a third party Java Applet for displaying phylogenetic relationships in tree-style displays. This web-based graphical interface is described in more detail in Section 2.3.

2.2. Data Gathering and Assembly Pipeline

Figure 2 shows the overview of our data gathering and analysis pipeline. This figure illustrates the method of using NCBI's precomputed BLAST neighbors. The local BLAST approach is in principle similar to this web page-based method. The entire process can be decomposed into four major steps.

Step 1 (Identifying and filtering species of interests URLs from NCBI Web site): We start with NCBI's Bacteria Genome web site (http://www.ncbi.nlm.nih.gov/PMGifs/Genomes/eub.html) that contains a list of all the complete bacteria sequences. This web page is locally transferred and stored into an html file. This html file is then matched against the list of speciesOfinterests.txt that contains the list of 21 species we are interested in computing the phylogenetic relationships for. Output of this process is the list of web-page URL addresses for the 37 species and the genome identifiers (GI numbers).

Step 2 (Getting protein table URL for the species of interest): Each of the URLs produced from Step 1 is visited and its content is downloaded into local files. The URLs of protein table pages for each species will be extracted for further link.

Step 3 (Getting BLAST hits URL for the species of interest): Each of the protein URLs produced from Step 2 is visited and its content is downloaded into local files. A parser will extract the useful data and store as local files for loading into the database. The data set includes Location, Length, Strand, PID, Gene, Synonym and Product.

Step 4 (Parsing BLAST hits URL and storing the result into a database): A parser will parse the BLAST hits web page of each gene for all species of interest to get the BLAST hits. Note that during this process only the top BLAST hits that belong to the species of interest are extracted. Finally, all the information that was stored in local files during the process will be pumped into the Oracle database.

Of the four steps mentioned above, Step 4 is the bottleneck. When one PC machine is used for this step, it takes about 6 hours on average to process about 2,500 genes per species. If we use multiple BLAST servers locally to obtain BLAST hits instead of downloading and using NCBI's precomputed BLAST neighbors, the overall process time should be dramatically reduced.

2.3. Maintenance and Extension

One key issue we need to address in developing PhyloSource is the transient nature of its content. As soon as new sequence data is entered into GenBank, PhyloSource could be considered obsolete. An economic model would be recomputing its content periodically, for example, weekly or monthly. Establishing the analysis pipeline allows us to automate the revision process, which must be done entirely. Another issue is to add a new species into the list of species of interest. Again, the established analysis pipeline makes this extension a minor chore. All we need to do is to modify the master list of species of interests by adding a new entry into the file speciesOfInterest.txt in Figure 2. The rest of the computation can proceed without a single line of code change.

Our earlier version of PhyloSource started with 11 species in September 2000. Currently, the number of species has grown to 21 as more completed genome species are available from NCBI. Our current plan is to run the data assembly pipeline over night periodically every month so that the content of PhyloSource is not behind more than one month from the GenBank sequence release. In addition, we plan to add additional species in the master list of species of interests as soon as its complete sequence information becomes available. As a goal, the addition of a new species should take less than one day of computation. We are currently exploring ways of running computation intensive processes concurrently on a cluster of low-powered workstations.

2.4. Web-based Graphical Query Interface

The graphical user interface for PhyloSource is web-based and aims at providing users with two functionalities: browsing and querying. Browsing means that the interface enables the user to freely move around from one portion of the data to another. Two methods are generally used in browsing, point-and-click or text-based search. Querying means that the interface permits the user to issue a descriptively formulated question with proper qualifications to find certain facts or relationships from the data repository. Figure 3 includes a number of screen shots illustrating the browsing and querying. We defer the discussion of the querying to Section 3. Browsing issue is discussed below.

(1) Choosing Species: Once the user enters into the PhyloSource web site, the main page (Fig3.1) is displayed that includes the list of species that PhyloSource deals with. The user can choose a set of species of her interests. For example, the user may choose only 10 species among the 21 species list if she is interested in the relationships among the 10 species. This choice interaction works like filtering or setting up a context for the future interaction with the system. Any subsequent queries will be limited only to these 10 species even if the database contains data for 21 species.

(2) Choosing a region and a gene: Once the global view (Figure 3.2) of the chosen species is presented, the user can choose a specific gene by clicking on the area view. This will guide the user to a bar diagram representing the overview of this species. The gene name and its location will be shown just below the cursor and such help options make it easy for the user to choose a gene of her interest. Clicking on any region will bring the user to the object viewer (Fig 3.3). This viewer consists of three components: control bar, information panel and query viewer. Moving the cursor along the X-axis will update the information panel. Each rectangular region on the information panel represents one gene. The rectangles colored in blue means that phylogenetic information is available for this gene.

(3) Viewing the pre-computed data: When right clicking on a blue colored rectangle, the following select menu appears: Show Sequence, Show BLAST Hits, Show Tree and Search Tool. Clicking on “Show Sequence” will retrieve the amino acid sequence for this chosen gene. Clicking on “Show BLAST Hits” will bring out the corresponding BLAST search result (Fig 3.4). “Show Tree” will display the phylogenetic tree based on the results of BLAST hits. For this feature, the ClustalW server (Figure 1.b) is being called in real-time. “Show Search Tool” will pop up the query viewer as shown in Figure 3.3.

(4) Displaying the tree view graphically: Choosing the tree option in Step (3) (e.g., “Show Tree”) will start the tree view program after furnishing it with appropriate data (Fig. 3.5). One interesting aspect of our implementation is that we use a third party tree viewer for the display. There are many graphical tree-drawing programs and our architectural philosophy is not to duplicate the work. We include third party application as a component system. Among many available candidates, we have chosen ATV (Zmasek and Eddy, 2001) for the purpose because it is a Java Applet and it can be readily included in our interface without modification. The process behind driving the graphical viewer is basically the following. Once the user chooses a gene, the id for this gene is passed to the repository. The entire amino acid sequence data stored for this gene is passed to the ClustalW program server. The outcome of ClustalW is passed to ATV. The output data format of ClustalW and the input data format of ATV are both Newick format (http://evolution.genetics.washington.edu/phylip/newicktree.html).

(5) Comparing multiple tree views side by side: The user can choose multiple genes and review their phylogenetic relationships graphically side by side. An alternative approach is to use a short cut. The user can choose a branch of the tree display and can have ATV display the tree view that starts with the

newly chosen gene. If the genes are homologous, the two views will depict the same relationships. Otherwise, the two displays could show different phylogenetic relationships.

The PhyloSource graphical interface provides a number search features to let user bypass extensive point and clicking steps which otherwise would be necessary (Figure 3.6). The user can directly home into a specific gene or a group of genes in one-shot by using combination of four search conditions: specie name, gene name, protein ID, and gene function. When the search finds a gene or a group of genes satisfying the search condition, the user is presented with a gene-level view (e.g., Figure 3.3) or a list of these so that she can proceed to the steps (3) – (5) discussed in the above.

3. Query System

One salient feature of PhyloSource is its advanced query mechanism. As an alternative to running ClustalW in real-time, we save all the possible ClustalW outputs in the database. When the database stores the trees explicitly, the user is able to query for possible homologous or orthologous relationships. The fact that one can examine gene evolution relationships by merely issuing a query is the novel part of our proposal .Using this feature scientists should be able to do more sophisticated analyses beyond the conventional sequence alignment comparison.

The search feature previously discussed in the context of browsing relies on mostly simple database table lookups (e.g., SQL queries). Unlike this type of structured database queries, the evolutionary relationship querying involves an algorithmic way of discriminating trees given a set of qualification constraints. For example, consider the following scenario. Suppose a scientist observes that in most cases the evolutionary relationship among species A, B and C are ((A B) C), meaning that A and B share an ancestor and so do the ancestor of (A B) and C. In a rare occasion, suppose he observed that there is a gene that supports ((A C) B) relationship. Such a relationship demonstrates the possibility of horizontal gene transfer (Doolittle, 1999). The scientist might want to know whether there are other genes that support the ((A C) B) relationship. This user should be able to request all trees that match the given hypothesis. The matching trees can be seen as possible candidates for supporting the suspected horizontal gene transfers between the species A and C.

Our method permits users to specify hypotheses to let them specify constraints on a set of pre-established tree templates. In this way users are building queries directly over a tree. Each template is used for describing a specific hypothesis. The template tree can grow or shink depending on the user’s manipulation of the template tree and depending on the level of specificity the user would like to express on the query. The users fill in data values and choose parameters to complete the hypothesis creation. The types of constraints that users place to express hypotheses include the following.

- Find trees based on proximity of single/multiple species within a tree.
- Find trees based on confidence levels (e.g., bootstrap values); In this case, allow the user to specify a minimum acceptable bootstrap level.
- Find trees based on distance score (e.g., the distance score is the average number of mutations expected at each position of the sequence).

4. Data Interoperability Issues

Our plan is to make PhyloSource as a model data interoperable system. PhyloSource has a potential for promoting a new way of conducting gene function analyses. Considering the ever-growing number of specialty databases in the genome community, there is a genuine need to make this phylogeny resource database interoperable with other biology databases. We first illustrate a few data interoperation scenarios.

Scenario 1 (Interoperation between PhyloSource and GeneBank): After choosing one of more genes displayed on the tree view, the user might be interested in retrieving the genomic structure or other DNA sequence related information from GenBank. For example, the scientist might be in the middle of designing an experiment and she might want to know where the optimal primer sequences are located that can be used for PCR, and which research lab might own the cDNA clone. Accessing directly to GenBank from PhyloSource could make this search task from GenBank easier.

Scenario 2 (Interoperation between PhyloSource and PDB): Again after choosing one of more genes displayed on the tree view, the user might be interested in knowing and comparing the 3D protein structures of these chosen homologs. If there is a flexible and seamless way of interlinking PhyloSource with PDB, the user might be able to easily find special motifs that belong to this particularly chosen gene group.

Scenario 3 (Interoperation between PhyloSource and Pathway Databases): The user might be interested in finding pathway information directly from a gene selected from the PhyloSource interface display. For example, she might be interested in knowing how the expression of mur1 gene is regulated in Borelia by being able to query a pathway database (e.g., KEGG (Kanehisa, 1996)) directly from the PhyloSource interface.

We are currently developing two alternatives for data interoperability. One is using the first approach. Since PhyloSource is maintained in Oracle, providing SQL access will be straightforward. In addition, we are developing limited Java APIs to make execution of algorithmic queries possible. The second approach is to combine both the API approach and the XML approach. In this approach, instead of porting the entire content of the PhyloSource, we make only a limited set of XML subschemas public along with a set of associated API access functions. Further discussion of this topic will require lengthy discussion and doing so is beyond the scope of this work.

5. Conclusion

We label PhyloSource a "derivative" database in the sense that the content of PhyloSource is derived from primary data sources such as GenBank. A derivative database is constructed by gathering, assembling and reorganizing a subset of data from the primary data sources to provide "valued-added" information for a specific class of users. From "objects" view point, PhyloSource contains information-content-wise complex objects each of which has been constructed by "processing" a large group of information-content-wise simpler objects that are available in the primary data sources. We described how such complex objects, i.e., phylogenetic relationships, can be constructed from the primary data sources. We demonstrated how these complex objects can be queried and displayed graphically. We demonstrate that we can successfully utilize a third party java applet program as a component of our interface. We also discussed how these complex objects can be associated with other objects of the domain for more powerful uses of the data.

References

Deutsch A, Fernandez M, Florescu D, Levy A, Suciu D. 1998. "A Query Language for XML", http://www.research.att.com/~mff/files/final.html.

Doolittle WF. 1999. "Phylogenetic classification and the universal tree". *Science*. 19;286(5444):1443.

Kanehisa, M. 1996. "Toward pathway engineering: a new database of genetic and molecular pathways". In *Science & Technology Japan*, 59:34-38.

Koonin EV, Mushegian AR, Galperin MY, Walker DR. 2000. "Comparison of archaeal and bacterial genomes: computer analysis of protein sequences predicts novel functions and suggests a chimeric origin for the archaea". *Mol Microbiol.* 35(3):697-8.

Olendzenski L, Hilario E, Gogarten JP. 1998. "Horizontal Gene Transfer and Fusing Lines of Descent: the Archaebacteria - a Chimera?" In *Horizontal Gene Transfer,* M. Syvanen and C. Kado (eds.), Chapman and Hall, London. pp 349-362

Olendzenski L, Liu L, Zhaxybayeva O, Murphey R, Shin D-G, Gogarten JP. 2000. "Horizontal Transfer of Archaeal Genes into the Deinococcaceae: Detection by molecular and computer based approaches.*" J of Molecular Evolution*, 51(6): 587-599.

Thompson JD, Higgins DG, Gibson TJ. 1994. "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice". *Nucleic Acids Research,* 22:4673-4680.

Zmasek CM and Eddy SR, 2001. "ATV: display and manipulation of annotated phylogenetic trees". *Bioinformatics,* in press.

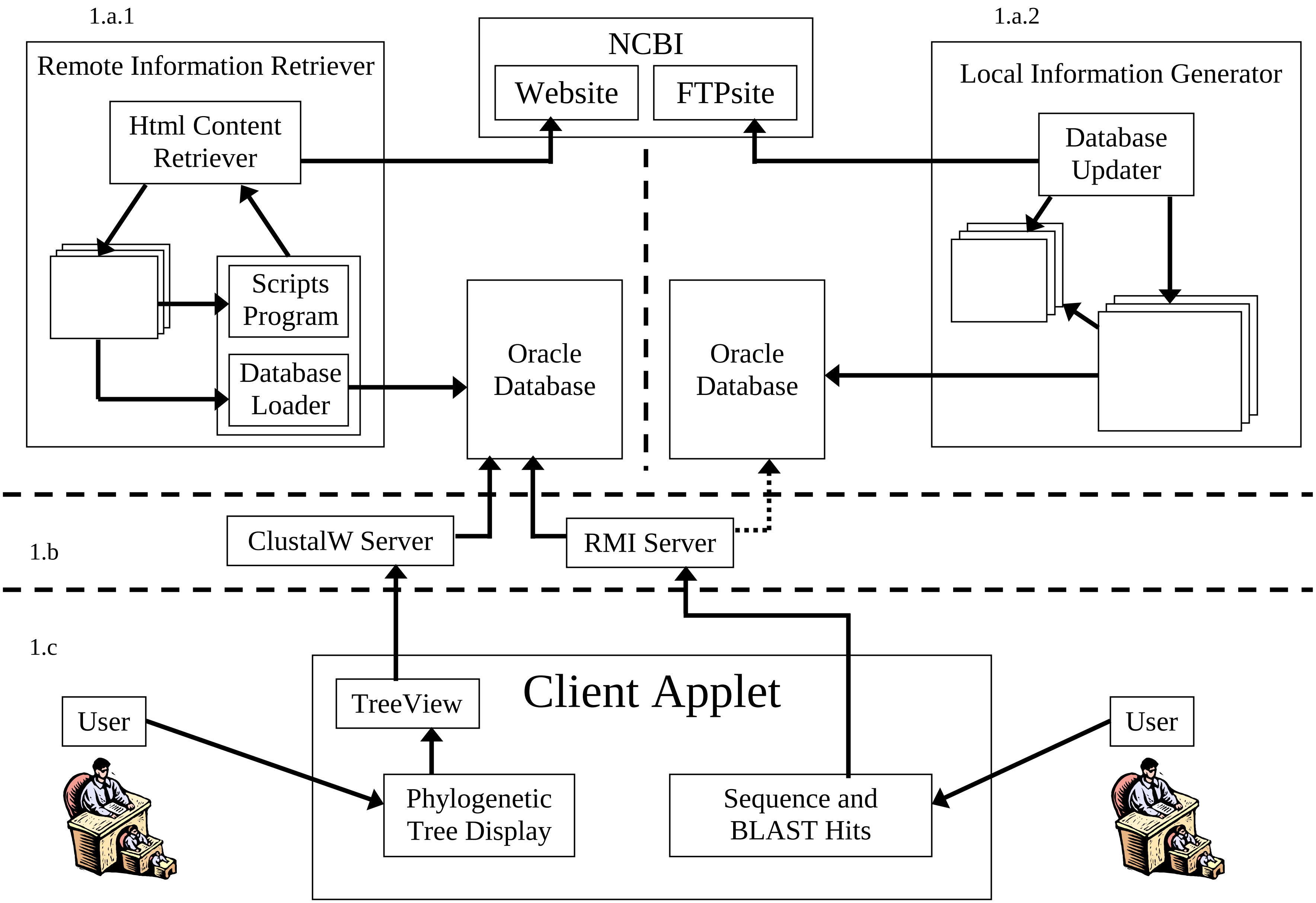
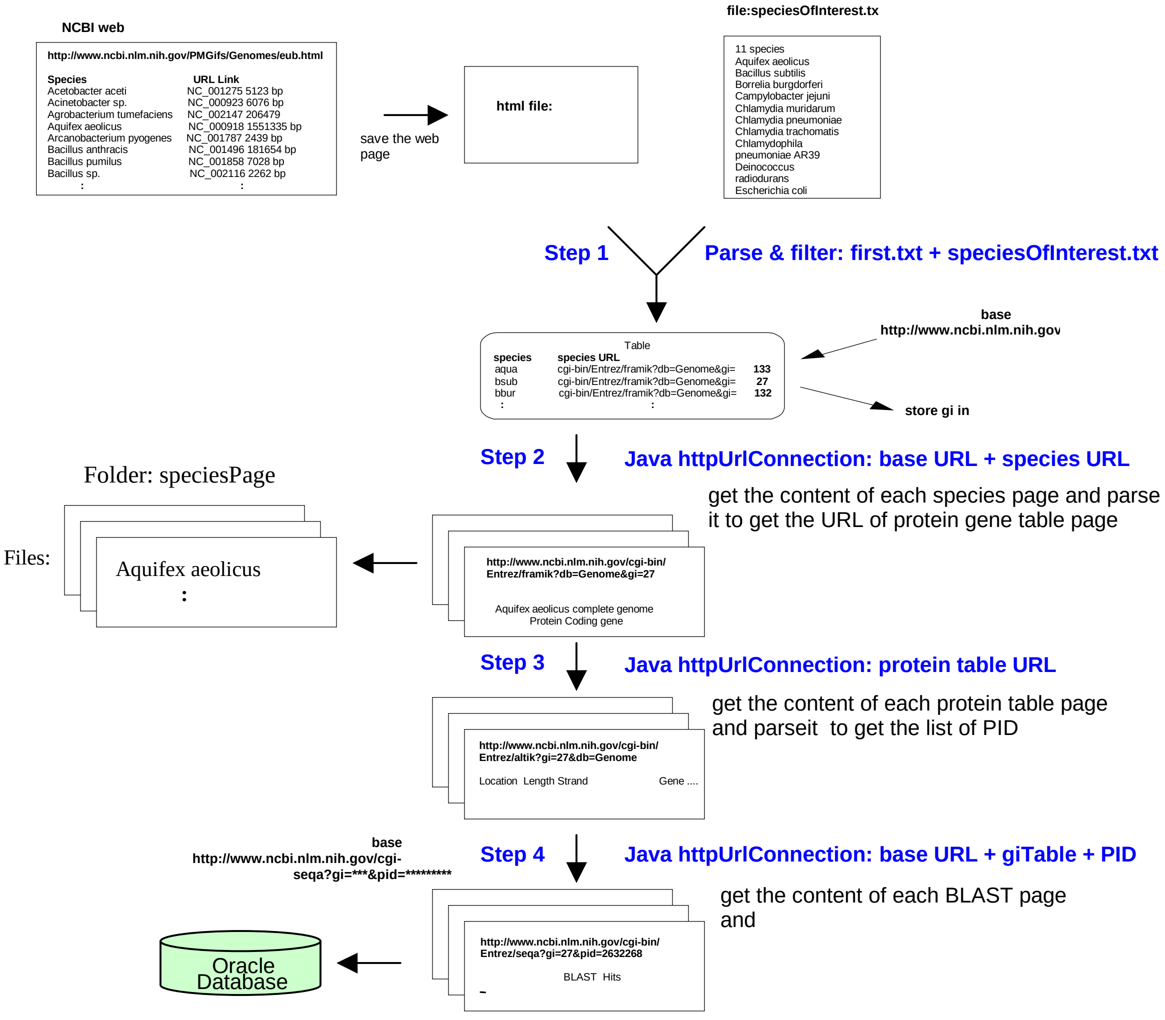


Figure 1. Overview of System Architecture


NCBI web
http://www.ncbi.nlm.nih.gov/PMGifs/Genomes/eub.html

Species	URL Link
Acetobacter aceti	NC_001275 5123 bp
Acinetobacter sp.	NC_000923 6076 bp
Agrobacterium tumefaciens	NC_002147 206479
Aquifex aeolicus	NC_000918 1551335 bp
Arcanobacterium pyogenes	NC_001787 2439 bp
Bacillus anthracis	NC_001496 181654 bp
Bacillus pumilus	NC_001858 7028 bp
Bacillus sp.	NC_002116 2262 bp
:	:

save the web page
html file:
file:speciesOfInterest.tx

11 species
Aquifex aeolicus
Bacillus subtilis
Borrelia burgdorferi
Campylobacter jejuni
Chlamydia muridarum
Chlamydia pneumoniae
Chlamydia trachomatis
Chlamydophila
pneumoniae AR39
Deinococcus
radiodurans
Escherichia coli

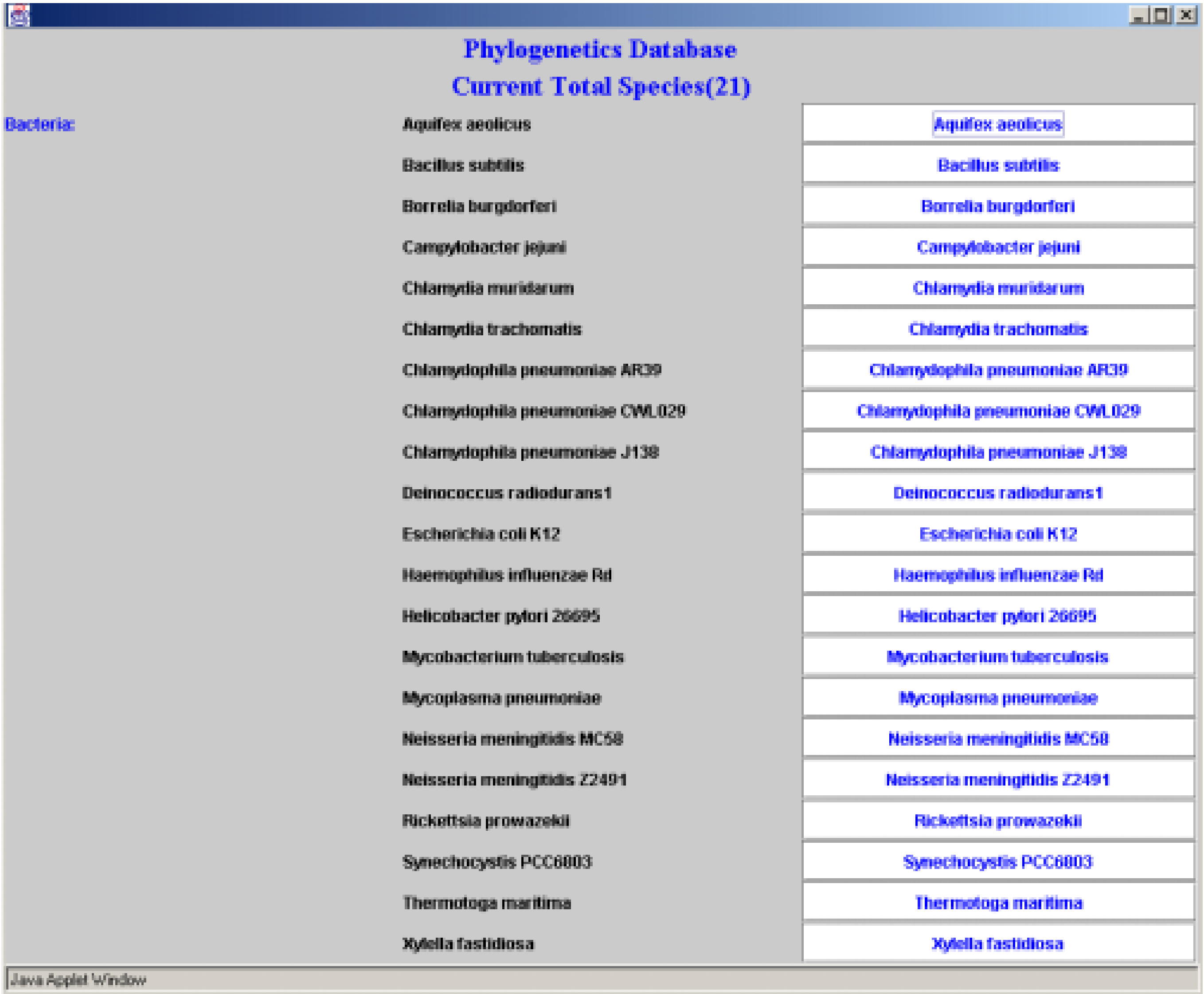
Step 1
Parse & filter: first.txt + speciesOfInterest.txt
base
http://www.ncbi.nlm.nih.gov
store gi in

Table

species	species URL	
aqua	cgi-bin/Entrez/framik?db=Genome&gi=	133
bsub	cgi-bin/Entrez/framik?db=Genome&gi=	27
bbur	cgi-bin/Entrez/framik?db=Genome&gi=	132
:	:	

Step 2
Java httpUrlConnection: base URL + species URL
get the content of each species page and parse it to get the URL of protein gene table page
Folder: speciesPage
Files:
Aquifex aeolicus
:
http://www.ncbi.nlm.nih.gov/cgi-bin/
Entrez/framik?db=Genome&gi=27
Aquifex aeolicus complete genome
Protein Coding gene
Step 3
Java httpUrlConnection: protein table URL
get the content of each protein table page and parseit to get the list of PID
http://www.ncbi.nlm.nih.gov/cgi-bin/
Entrez/altik?gi=27&db=Genome
Location Length Strand
Gene ....
base
http://www.ncbi.nlm.nih.gov/cgi-
seqa?gi=***&pid=*********
Step 4
Java httpUrlConnection: base URL + giTable + PID
get the content of each BLAST page and
http://www.ncbi.nlm.nih.gov/cgi-bin/
Entrez/seqa?gi=27&pid=2632268
BLAST Hits
-
Oracle
Database

Figure 2. Data gathering and analysis pipeline



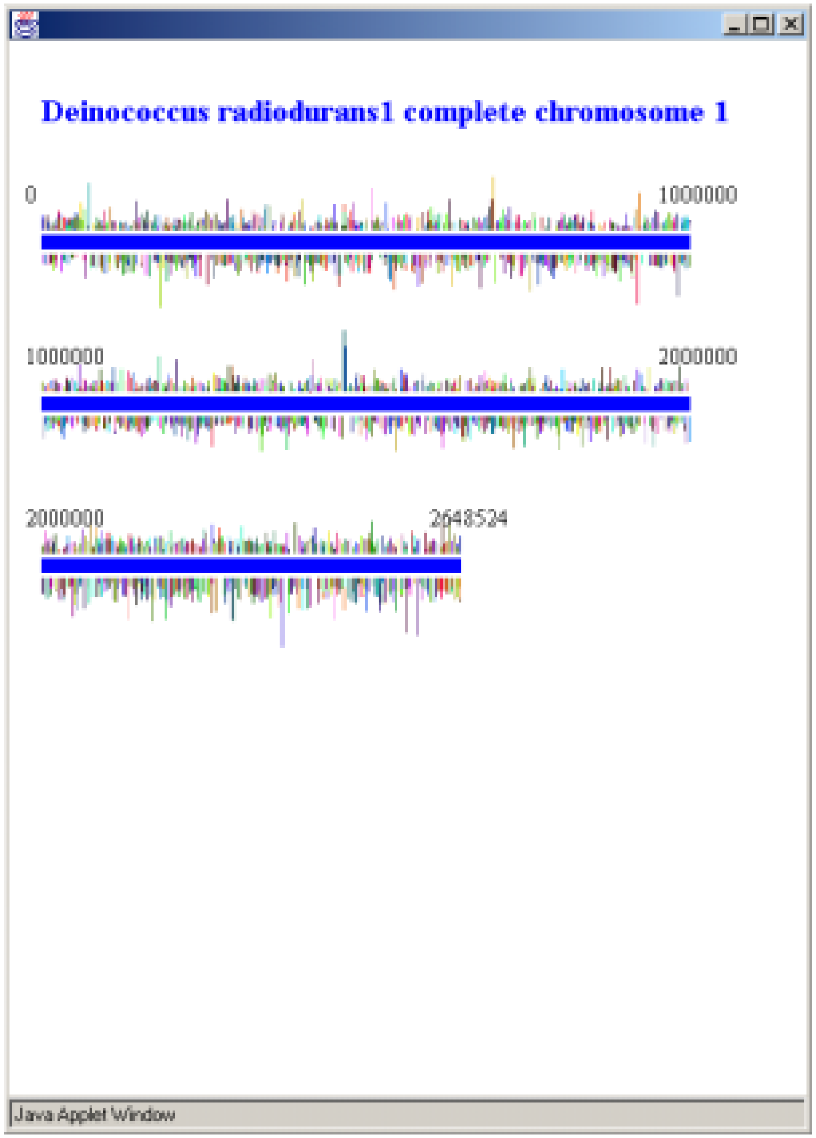
Phylogenetics Database

Current Total Species(21)

Bacteria:	Aquifex aeolicus	Aquifex aeolicus
	Bacillus subtilis	Bacillus subtilis
	Borrelia burgdorferi	Borrelia burgdorferi
	Campylobacter jejuni	Campylobacter jejuni
	Chlamydia muridarum	Chlamydia muridarum
	Chlamydia trachomatis	Chlamydia trachomatis
	Chlamydophila pneumoniae AR39	Chlamydophila pneumoniae AR39
	Chlamydophila pneumoniae CWL029	Chlamydophila pneumoniae CWL029
	Chlamydophila pneumoniae J138	Chlamydophila pneumoniae J138
	Deinococcus radiodurans1	Deinococcus radiodurans1
	Escherichia coli K12	Escherichia coli K12
	Haemophilus influenzae Rd	Haemophilus influenzae Rd
	Helicobacter pylori 26695	Helicobacter pylori 26695
	Mycobacterium tuberculosis	Mycobacterium tuberculosis
	Mycoplasma pneumoniae	Mycoplasma pneumoniae
	Neisseria meningitidis MC58	Neisseria meningitidis MC58
	Neisseria meningitidis Z2491	Neisseria meningitidis Z2491
	Rickettsia prowazekii	Rickettsia prowazekii
	Synechocystis PCC6803	Synechocystis PCC6803
	Thermotoga maritima	Thermotoga maritima
	Xylella fastidiosa	Xylella fastidiosa

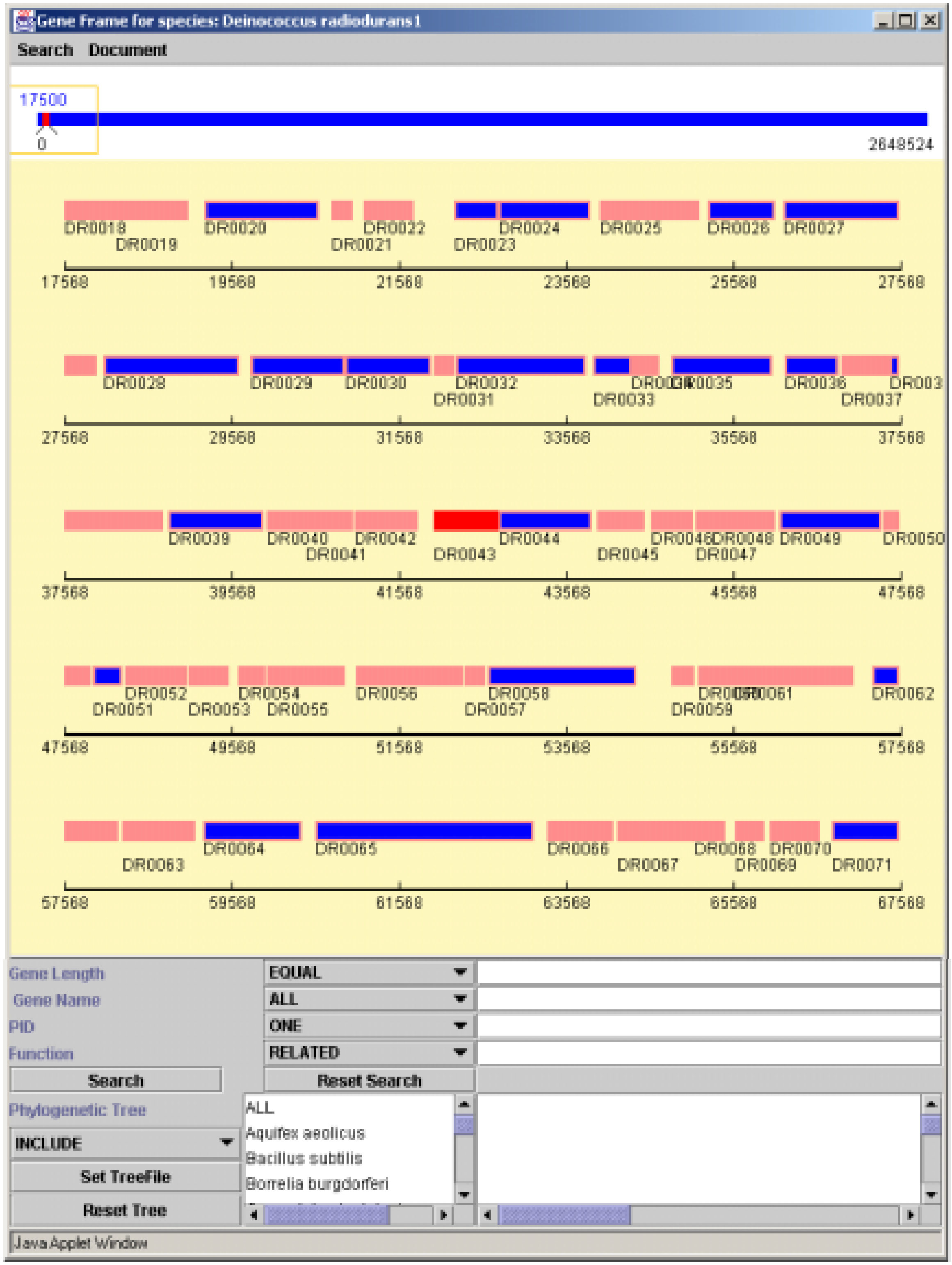
Java Applet Window

Figure 3.1. Web Interface Main Page



Deinococcus radiodurans1 complete chromosome 1

Figure 3.2. Global View of a Genome



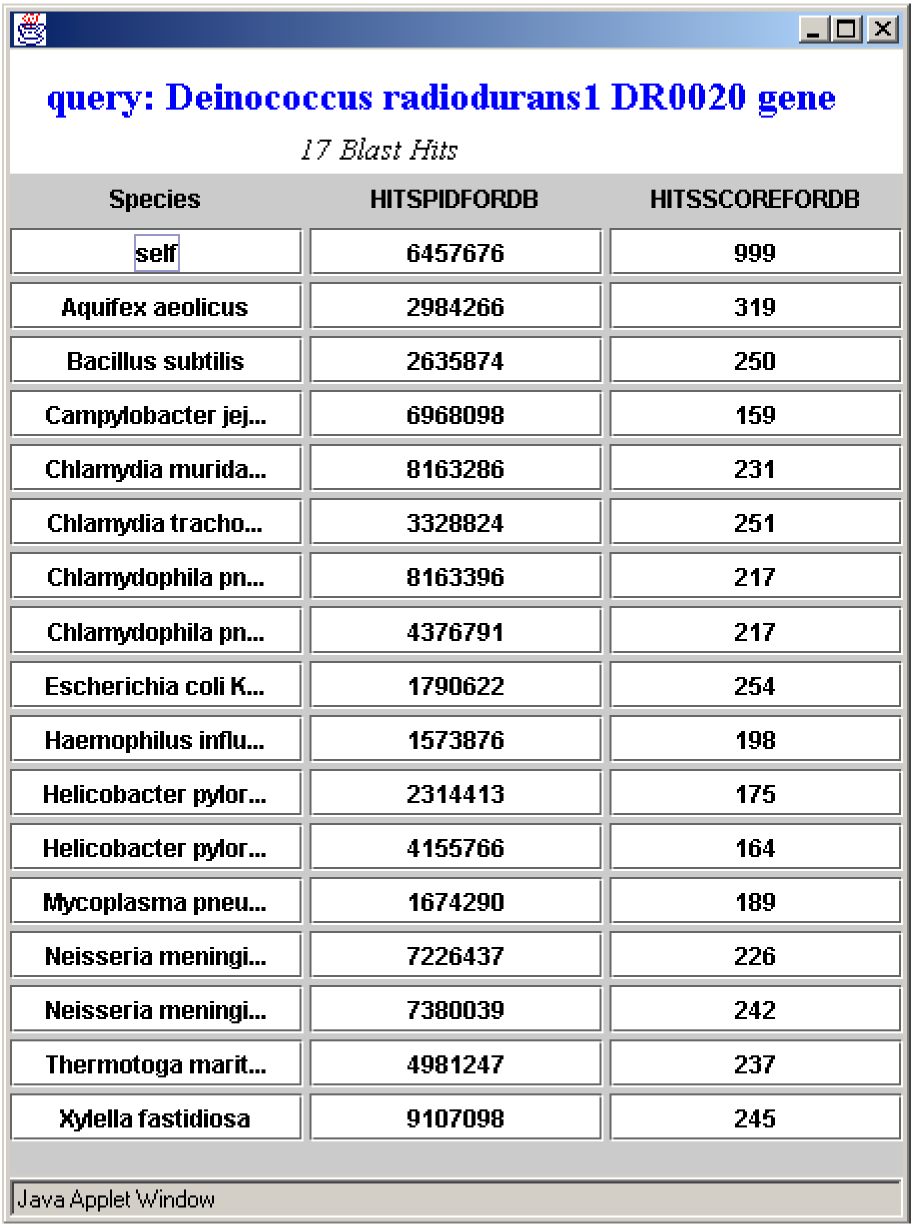
Gene Length: EQUAL

Gene Name: ALL

PID: ONE

Function: RELATED

Figure 3.3. Object Viewer

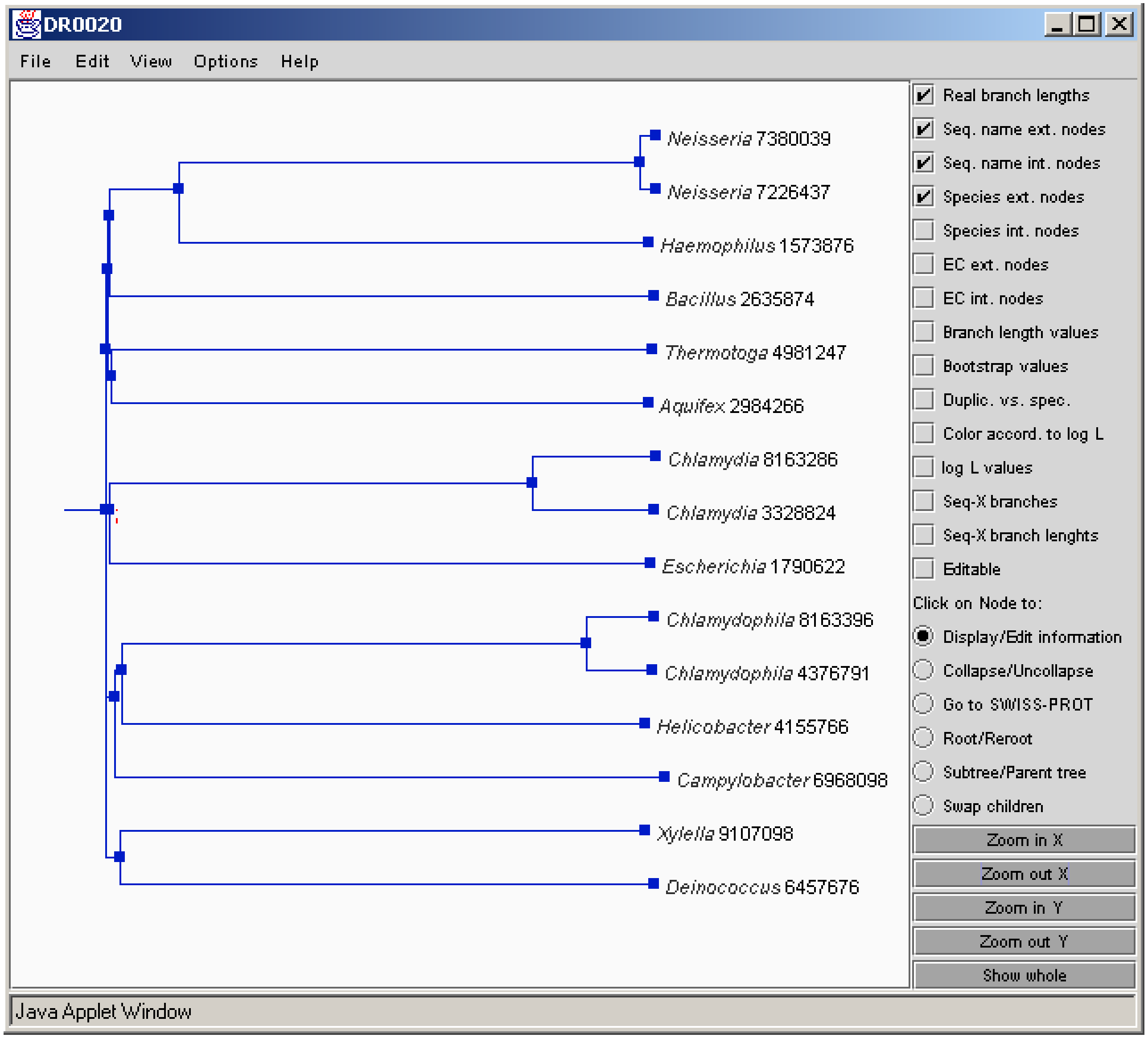


query: Deinococcus radiodurans1 DR0020 gene

17 Blast Hits

Species	HITSPIDFORDB	HITSSCOREFORDB
self	6457676	999
Aquifex aeolicus	2984266	319
Bacillus subtilis	2635874	250
Campylobacter jej...	6968098	159
Chlamydia murida...	8163286	231
Chlamydia tracho...	3328824	251
Chlamydophila pn...	8163396	217
Chlamydophila pn...	4376791	217
Escherichia coli K...	1790622	254
Haemophilus influ...	1573876	198
Helicobacter pylor...	2314413	175
Helicobacter pylor...	4155766	164
Mycoplasma pneu...	1674290	189
Neisseria meningi...	7226437	226
Neisseria meningi...	7380039	242
Thermotoga marit...	4981247	237
Xylella fastidiosa	9107098	245

Figure 3.4. BLAST Result View

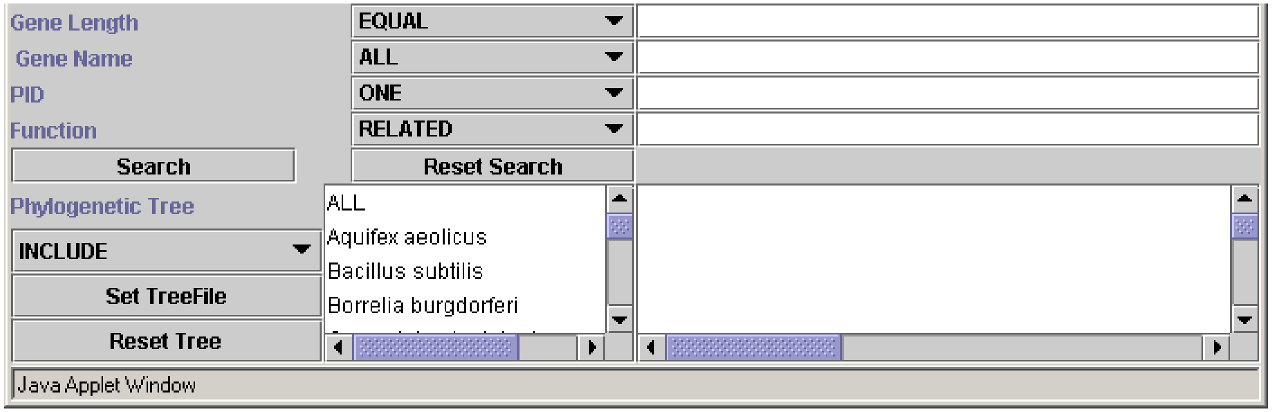


☑ Real branch lengths
☑ Seq. name ext. nodes
☑ Seq. name int. nodes
☑ Species ext. nodes
☐ Species int. nodes
☐ EC ext. nodes
☐ EC int. nodes
☐ Branch length values
☐ Bootstrap values
☐ Duplic. vs. spec.
☐ Color accord. to log L
☐ log L values
☐ Seq-X branches
☐ Seq-X branch lenghts
☐ Editable

Click on Node to:

● Display/Edit information
○ Collapse/Uncollapse
○ Go to SWISS-PROT
○ Root/Reroot
○ Subtree/Parent tree
○ Swap children

Figure 3.5. Tree View using ATV



Gene Length: EQUAL

Gene Name: ALL

PID: ONE

Function: RELATED

Phylogenetic Tree: INCLUDE

ALL
Aquifex aeolicus
Bacillus subtilis
Borrelia burgdorferi

Figure 3.6. Search Window