Genome Grid Aims

This project aims to create a usable package of genome data analysis with cyberinfrastructure: methods, protocols, documentation, suited for genome informaticians.

The thrust of this work is parallelizing genome data, not software, to run as many separate 1-cpu jobs as is suitable to the task and resources. It focuses on data management, transport to/from, indexing, and splitting data transparently from several source data sets to compute sites, and collating results to return to the scientist.

The poster-child task is a gene homology Blast analysis of any genome, but use of several other genomics programs from gene predictors, EST assemblers, phylogeny analyses, etc. are part of the project goal. Most of these work fine on any size of data set, and subset results can be added together.

One way to do this is as a kind Teragrid science gateway project, where the authentication, admin., grid resource finding are contained in the gateway components. Parts that the user genomicist sees are for data and analysis tool selection. Many desired genome tools are available at some Teragrid sites, but methods to transparently copy and parallelize data sets are not.

Find more background in the References, or Google: genome teragrid

Genome analysis and annotation via Grid computing

This subproject builds re-usable tools and workflows for genome analyses and annotation, using shared cyberinfrastructure (Grids or clusters). Here within are collections of scripts, documents and workflows for employing existing genome analysis tools (BLAST, homology tools, predictors, comparative and phylogenetic analyses) on available cyberinfrastructure. One emphasis here is on simplified use of grids and genome tools, to make it feasible for new genome projects to take advantage of these readily.

Target customer and tasks

The customers for this project are small to medium genome database projects, and individual bioscience research labs. We expect some familiarity with bioinformatics data and analyses. The customers generally have genome data in hand, in common formats of which Fasta (sequence) and GFF (annotation) are most common. Customers also will need to draw on public bio-data from the usual suspects (NCBI, EBI, Uniprot, UCSC, common genome databases). Often the project will need a one-time set of analyses on a new genome, or to test a new idea with existing genomes. Other times projects want to update analyses, re-running them with current data sets a few times per year. The customer often has skills with Unix command line systems, Perl and/or Java languages (with python, ruby and others mixed in). Moving data around by FTP, http, rsync and such are common skills. Using available bio-packages for parsing data, such as BioPerl (lesser BioJava), EMBOSS, some commercial products, is also a common skill. Sometimes the customer has access to a local computer cluster, or a university managed one, and would spend effort toward his/her analyses on these systems. Analysis pipelines may be involved (more common at large sequencing centers), but are often home-grown creations without a standard operation.
A genome grid gateway would support the common usage of these customers by offering access to grid resources for the same computations, with Unix command-line, Perl and Java bindings at least. Web-based front-ends are an option, but often the user data resides on unix systems along with data parsing and application tools that we would like to integrate with remote grid access.

Genome Grid components

Most of the potential parts of this package are available, and need to be assessed and combined. Our goal is not to develop new components, but to combine existing methods of genome data analyses and grid usage, adding middleware (perl, java, python, etc.) code where needed. Collecting and documenting the best practices, with working examples for genome analyses is a goal. These include in no particular order

- TeraGrid Science Gateway tutorials and simple gateway code sources
- EvidenceModeler: Perl package with good basic genome data splitting methods
- BioMart: a transaction oriented bio-database (MySQL, others) that understands range of biodata, selecting subsets, and has interfacing with the Taverna workflow project (Taverna is java-based; BioMart a mix of perl, java and various RDBMS methods).
- Lucene and Lucegene: Bio-data indexer that understands various genome data formats, now in use for grid data splitting. Lucene/Lucegene are in java. Advantage is that nothing needs be compiled, indices and software can be distributed with data to compute nodes rather easily.
- OGSA-DAI: a Grid data access project, mostly revolving around relational data access, and maybe too heavy-weight for genome informatics needs.
- A Grid Application Hosting Environment: This looked like an interesting and practical package for a grid gateway suited to bioinformaticians when I saw it in 2006, but haven't evaluated it in detail.

- Ergatis and Galaxy are GMOD related projects with genome analysis workflow systems that are relevant to this.

Downloads

See starter project in SourceForge or in package form at euGenes.

This package includes scripts for genome data partitioning, running parallel genome analysis jobs and collating partial results. It is being used successfully on TeraGrid clusters for analyzing several arthropod genomes (Daphnia, Pea aphid, 12 Drosophila, and others). It should work "as-is" on computer clusters with PBS or LoadLeveler batch queues (TeraGrid is not required). Dongilbert 19:56, 26 June 2008 (EDT)

References

Contact

Don Gilbert

Support provided by a grant from NSF BIO Database Activities