### Use and Development of Shared Grid Resources for Genome Assembly, Annotation and Analysis

PI	Don Gilbert (IU, Department of Biology)
CoPI	Jeong-Hyeon Justin Choi (IU Center for Genomics and Bioinformatics)

## Summary of proposed research

This project will use TeraGrid resources for assembly, annotation and comparative analysis of new organism genomes, specifically *Daphnia* (crustacean water flea) and twelve *Drosophila* species (insect fruit flies). Research into improved methods of genome assembly and analyses are enabled by this project, as is development of genome community tools for use of shared cyberinfrastructure. Assembly and analyses of these 200 megabase eukaryote genomes are computationally intensive, and require several days to several weeks of single-cpu time. Storage requirements are in the 10s to 100s of gigabytes depending on analyses and databases needed. Our proposed use of TeraGrid will provide a model use of shared cyberinfrastructure for future genome informatics research. Outcomes of this research will be shared with the genome informatics community through Generic Model Organism Database tools, and with bioscience research communities interested in *Daphnia* and *Drosophila* genomics, through web databases and publications.

***Computational methodology:*** New methods for genome assembly assessment are being developed as part of this research. State of the art genome informatics tools are in use to assemble and analyze genomes, including assemblers (Arachne, PCAP); sequence similarity tools (BLAST, Blat, others); gene prediction (SNAP, Twinscan, others); assessment and prediction of protein functions (InterProScan, others); comparative genome alignment (MAVID, Multiz, others), gene orthology assessment (OrthoMCL). A few of these genome tools are parallelized. A data-grid approach is being developed as part of this research to effectively use parallel systems with non-parallelized tools by partitioning genome data sets.

***Requested resources**:* 150,000 service units (TeraGrid cluster, TeraGrid IU, TeraGrid roaming) are requested, based on usage for preliminary research (50,000 SU) and estimated 1-year needs to complete genome analyses. Of this total, 90,000 SU for Daphnia genome assembly and validation, and 60,000 SU will be used for comparative genome annotation and analyses. Details justifying this request are provided below.

***Local computing environment:*** IU High Performance Computing resources include one 84-cpu IBM PowerPC cluster (Libra), two IA32 Linux clusters of 208 processors each (AVID), and one 32-cpu IA64 Linux cluster (iu.teragrid), GPFS and NFS shared storage, and a 250 TB massive data storage system. PBS and Globus Grid tools are available on these systems. IU Genome Informatics Lab resources include 2 Terabyte storage dedicated to public genomics and bioinformatics datasets, including the world-collaborative Bio-mirror.net (Gilbert et al. 2004), and six dual-cpu servers with Globus Grid tools for development and bioinformatics Grid testing.

## Scientific and Intellectual Merit

A general problem in bioinformatics is effective use of shared cyberinfrastructure with biology data sets. Bioinformatics analyses in genomics, proteomics, evolution and other areas are difficult today with the available large data sets. Cluster and Grid computing in bioinformatics have followed other disciplines in parallelizing applications, but this is costly and limited to a subset of bioinformatics applications.

Parallelizing data access has potential to open many existing and new biology analyses to effective use of Grid and cluster computing. A data grid approach to bioinformatics can be usefully applied now to genome analyses, including assembly and annotation of new genomes, phylogenetic comparisons of genomes across species, computational analyses and predictions of genome and proteome functions. Data grid methods can provide bioscientists with computable access to the widely distributed, large and changing data sets needed for large-scale analyses in life sciences.

The water flea *Daphnia pulex* is a crustacean that is commonly found in shallow ponds. Water fleas are an important and environmentally sensitive member of the food chain in ponds and lakes. They feed on algae and are in turn food for other animals such as fish or other small animals including Hydra. The *Daphnia* Genomics Consortium (DGC) is an international network of investigators devoted to creating a new model system for ecological and evolutionary genomic research. The DGC in collaboration with the Joint Genome Institute (JGI) is currently undertaking the *Daphnia* Genome Project. This project aims to sequence and annotate the complete *Daphnia* genome. The sequencing has been completed as of September 2005. The challenge now is to assemble, annotate and interpret the genome sequence. Further details on *Daphnia* genomic resources are available at www.wfleabase.org and daphnia.cgb.indiana.edu (Colbourne et al. 2005).

Daphnia possesses the smallest recorded genome among crustaceans, but may be 1.3 times larger than insects. Genome science suggests that insects and crustaceans have a similar number of genes, but then what structures are influencing genome size? Preliminary data suggest that Daphnia has smaller than expected repetitive (non-coding) DNA compared to animals with similar genome sizes. A survey of Daphnia transposable elements (genomic parasites) suggest they may also be rare relative to insects. Gene introns are shorter than expected. A complete analysis of a high quality assembled genome is required to confirm hypotheses on genome evolution that would explain these patterns. The National Science Foundation has funded two interdisciplinary proposals aimed, in part, at creating the Daphnia genomic system (NSF 0328516: Causes and Consequences of Recombination; NSF 0221837: Development of Methods Linking Genomic and Ecological Responses in a Freshwater Sentinel Species).

## Annotation and analyses of invertebrate genomes

***Genome data grid methods.*** Many genome computations work iteratively over the genome base “string”, where genome substrings or contigs can be effectively analyzed independently then results collated. One major exception to this is genome assembly, which requires analyses of all genome fragments at once. Data grid methods to partition and analyze in parallel a genome can be implemented with a few steps to locate data, copy to grid, and return results:

```
1. @virtualdata= biodirectory("find protein coding sequences for Drosophila species"), as an example from a wide range of queries.
2. @realdata= biodirectory("get locators for @virtualdata split n ways"), for n compute nodes
3. for i (1.. n) { copy(realdata[i],gridcpu[i]);  results[i]=runapp(gridcpu[i]) }
4. result_table = collate( @results );
```

These steps summarize actions to find/query data directories, copy subsets to distributed computing nodes, and return results from the analyses, collated from the compute nodes. Steps 2, 3 are the core of a data-grid system. Step 3 means that analysis applications need not have any special data access methods. Data grid tools can transport appropriate data parts to each compute node. Steps 1,4 may be separate systems. E.g. any database query system could work for step 1 as long as it returned data set IDs usable for selecting data subsets. Step 4 would include many tools that assemble and summarize raw results, such as those from BioPerl. These steps should be overseen by a workflow system capable at data and compute tasks. Genome partitioning has been tested during preliminary annotation of genomes, and are equally effective to MPI-parallelized versions of BLAST for genome analysis. They also permit parallel analyses with non-parallelized applications, such as gene finding, multiple alignment and orthology analysis.

***Genome annotation using TeraGrid.*** Preliminary assessment of TeraGrid to analyze new invertebrate genomes has been performed in the context of uses for the wider genome database community. It was performed during years 2005-2006 with TeraGrid allocation BIR050001 to the PI for "Development of GMOD Genome Database Community Grid Resources". This assessment includes newly sequenced genomes for *Daphnia pulex* and twelve *Drosophila* genomes. Genome database tools from the Generic Model Organism Database (GMOD, Stein et al., 2002) project are used to organize TeraGrid computations for public access.

For each of one *Daphnia* and twelve *Drosophila* genomes, a comparison is made to a set of nine proteomes, with 217,000 proteins, drawn from source genome databases, Ensembl and NCBI. These reference proteomes are human, mouse, zebrafish, fruitfly, mosquito, bee, worm, mustard weed, and yeast. Sizes of the new genomes are in the 150 Mb to 250 Megabase range. Protein-genome DNA alignment is performed with tBLASTn, using a Grid (MPI parallel) version developed at Indiana University Technology Services. The TeraGrid run for each genome took 12 - 18 hours using 64 processors. Whole genome DNA-DNA alignments were performed for a subset of new genomes. Gene predictions with SNAP (Korf, 2004) have been generated. Over the course of 6 months, with 2 to 3 genome assembly updates each per species, and error corrections, the total TeraGrid 64-cpu usage per genome has been approximately 4 days, excluding queue waiting times. Figures 1 and 2 summarize the annotation of these genomes.

Public access to the outcomes of this research, in the form of genome maps (GBrowse), similarity searches (web BLAST), data mining (BioMart), along with genome summaries, are provided at web databases wfleabase.org (Daphnia) and insects.euGenes.org (Drosophila). This work has enabled many bioscientists to have rapid, usable access to new genomes, facilitating new science discoveries and understanding of the evolution, comparative biology, and genomics of these model organisms.

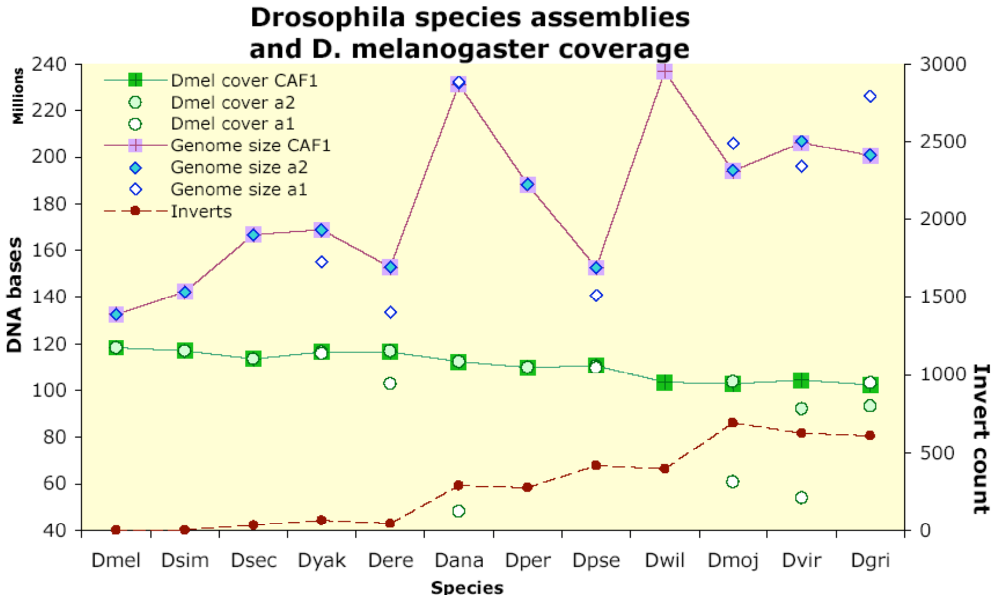


**Figure 1.** DNA coverage of *Drosophila* species assemblies to *D. melanogaster* genome, size of assembly and counts of inverted segments. Coverage for earlier assemblies along with latest assemblies (CAF1) are shown.

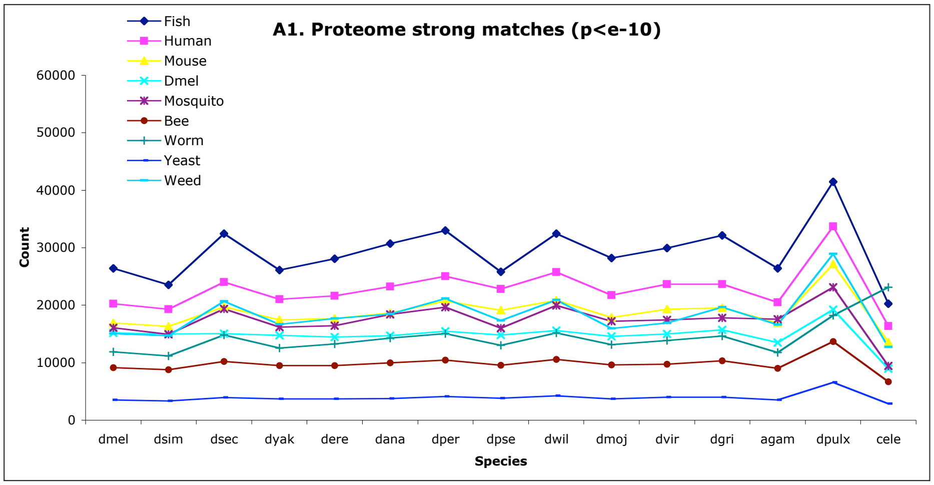


**Figure 2.** Gene matches in new invertebrate genomes. tBLASTn is used to match query proteomes to target genomes. Target genomes are twelve *Drosophila* species (Dmel .. Dgri), the mosquito *Anopholes gambia* (Agam), the crustatean *Daphnia pulex* (Dpulx), and worm *C. elegans* (Cele). These counts include many duplicate matches, to different as well as same genome locations.

***TeraGrid Basics.*** The assessment shows that use of TeraGrid for shared genome computations is feasible. This use of TeraGrid for genome-sized computations has aided the NIH-sponsored *Drosophila* species sequencing with a quick assessment of assembly qualities for gene annotation. Improved assemblies have more gene matches, and fewer duplicate matches. Hurdles to wider use by genome informaticians include a large learning and setup cost in time and effort. Additionally, failed runs were a significant portion (TeraGrid outages, software and data errors), and required personal attention to correct. The major part of the human effort involves preparing data, distributing to grid nodes, retrieving volumes of results, and combining and summarizing those. It is this aspect where data grid tools can facilitate uses of TeraGrid and Grid resources for genome informatics.

Basic steps for using TeraGrid for genome data are not overly complicated, but require learning and trial and error for the new user. Difficulties in these steps are being addressed by TeraGrid developers. Some of these can be streamlined for specific needs of genome informatics.

- Preparation
  1. obtain TeraGrid account
  2. establish certificates, grid-security entries, test Grid and local certificate use
  3. locate computational biology software, compile needed tools
- Processing
  4. locate and prepare genome data; partition and randomize
  5. transfer input data to TeraGrid
  6. configure, run job, with attention to run errors and queuing
  7. return results; post-process to combine results from multiple nodes; e.g. GFF for map view of genome annotations.

Of these, steps 4 to 7 represent bioinformatics needs this project will address. Data selection, preparation, transport to TeraGrid, and return of results, in collated form, to the scientist are the special needs. Methods for step 6 are in the realm of workflow tools developed elsewhere and applied in this project.

**Resources justification for invertebrate genome analyses**

The preliminary assessment found that a basic genome annotation, including sequence similarity (BLAST) and gene finding (SNAP), used 1500 SU per genome assembly, excluding error runs. A total of 15 organism genomes were analyzed. Over the course of a year, 5 of these had three improved assemblies, which were partially analyzed, and 3 had two improved assemblies, or a total of 22 complete genome analysis runs using 32,000 SU. Error runs include a range of factors, such as trial-and-error tests, TG cluster failures, and refinement of methods. These accounted for an additional 15% in service units. Over the proposal period, we estimate at least one round of improved genome assemblies for 15 organisms, with two or more for some organisms, including Daphnia (see below). We plan to add new genome analyses not performed in the preliminary research, by developing genome partitioning and data grid methods that allow efficient use of TeraGrid resources with non-parallelized analysis tools. Comparative alignment of genomes, and prediction of protein functions are two of these categories of analyses. These are estimated to double the per-genome service unit need, to 3,000 SU, totaling 45,000 SU for 15 genome-runs and 15,000 SU for development work and additional assemblies.

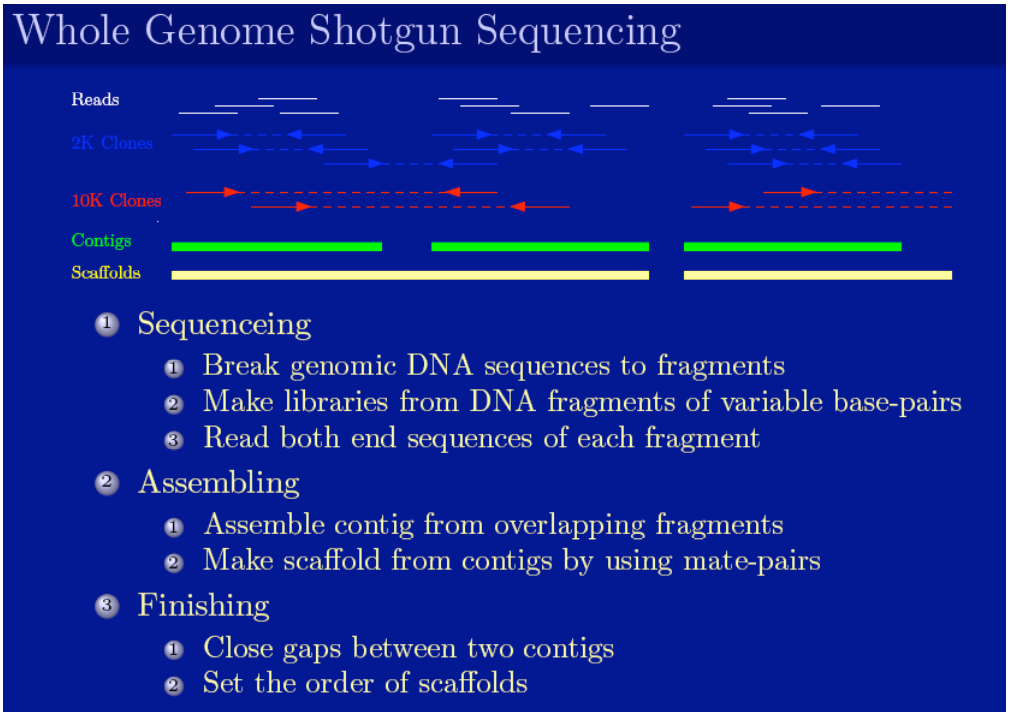
**High quality assembly of the Daphnia genome**

While "the genome sequence" of an organism is a commonly used phrase, in practice the known genomes have been composed or assembled from millions of small partially overlapping sequences. The *Daphnia* genome consists of 12 chromosomes, totaling approximately 200 million bases, and assembled from approximately 2.5 million sequencer reads. Assembly algorithms must match overlapping DNA sequences into sequence scaffolds to compose a representation of the organism's chromosomes. This computation is susceptible to errors: many parts of a eukaryote genome share similarities in low complexity, repetitive regions, or among recently duplicated regions. An assembly represents only a likely chromosome representation that requires validation. Obtaining a correct chromosome representation is critical if investigators are to draw conclusions about the evolution of genomic structures.

The most widely used method to detect misassembled regions of a genome sequence is called the fragment coverage analysis, which finds regions where the number of aligned fragment is greater than expected. Misassembled regions are detected where fragment coverage is significantly higher than predictive models. This approach has two problems. First, there are normally many high coverage regions, thus discriminating truly misassembled regions is not trivial. Second, some misassembled regions may not exhibit high coverage. Two of our collaborators, Tang and Kim, have developed computational methods for detecting misassembled regions, based on pattern statistics and clone coverage statistics (Kim et al. 2001). These algorithms accurately detect false assemblies in microbial genomes. These are being refined to provide a scaffold junction probability index for the more complex eukaryotic genomes, including *Daphnia*. A significant outcome of the proposed research is the creation and implementation of a support index along localized regions of genome sequence assemblies to indicate regions of high and low confidence.

Assembling whole-genome shotgun sequence utilizing additional experimental data for *Daphnia* (physical and genetic maps) remains a challenge. We propose to combine traditional assembly protocols and other experimental data using a simple two-step approach that builds on existing algorithms. Starting from the pre-assembled contiguous sequence fragments (called contigs), (1) we validate contigs using a support index and reduce weakly supported large contigs into smaller ones; (2) we then build scaffolds of the validated contigs, based on the additional experimental results (genetic maps and tiled BACs). We will therefore produce a strongly supported assembly of the *Daphnia* genome after having explored alternative arrangements guided by all available data.

***Initial Daphnia assembly***. The steps in whole genome shotgun sequencing are outlined in Figure 3. Sequencing of 8 to 9 fold WGS (Whole Genome Shotgun) has been completed by DOE JGI (Joint Genome Institute), and the traces desposited in the NCBI Trace archive, with 2.7 million traces, of 2.7 billion bases total length. We have produced assemblies from these full sequencing data as preliminary to validating the underlying genome structure. Preliminary research using TeraGrid resources has been performed in early 2006 by Co-PI Jeong-Hyeon Choi using TeraGrid DAC grant BIR050001. Available genome assemblers include Arachne (Jaffe et al., 2003) and PCAP (Huang et. al 2006). These are widely used for eukaryote genome assembly. Additional assemblers of interest are JAZZ, which the JGI sequencing center will use to create a *Daphnia* assembly, as well as others.



Whole Genome Shotgun Sequencing

1. Sequenceing
   1. Break genomic DNA sequences to fragments
   2. Make libraries from DNA fragments of variable base-pairs
   3. Read both end sequences of each fragment
2. Assembling
   1. Assemble contig from overlapping fragments
   2. Make scaffold from contigs by using mate-pairs
3. Finishing
   1. Close gaps between two contigs
   2. Set the order of scaffolds

**Figure 3**. Whole Genome Shotgun sequencing steps. The assembly steps produce scaffolds from sequencing reads.

Assembly using PCAP requires these steps: system configuration, collection and preparation of sequence fragment data (trace files), assembly of the 9-fold WGS data set and assembly of an earlier 4-fold data set from JGI. PCAP is a suite of several programs that produce an assembly in stages. Its component tasks include sublapjobs, run using 60 cpus (IU.teragrid); runtigcode, a non-parallel step of 1 cpu using 11Gb of memory (NCSA.teragrid); subsenjobs using 60 cpus (IU.teragrid); and a summarizing step, runstatcode, 1 cpu (IU.teragrid). Table 1 summarizes the results of PCAP assembly of Daphnia.

**Table 1.** Daphnia genome assemblies produced with PCAP under four conditions from 2.7 million sequencing traces.

| Component | no-Mate-pair | Mate-pair | Screen+Trim | Distant-Constraint |
|---|---:|---:|---:|---:|
| No. Contigs | 71,519 | 81,860 | 66,747 | 74,521 |
| No. Scaffolds | 71,519 | 69,790 | 55,159 | 61,858 |
| N50 scaffold | 10,934 | 797 | 536 | 376 |
| Big scaffold | 17,207 | 1,240,503 | 1,233,017 | 1,894,251 |
| All scaffolds | 236,665,088 | 249,840,836 | 228,822,207 | 239,506,399 |
| Total job hours | 1,305 | 1,596 | 1,318 | 1,333 |
| No. Jobs | 120 | 70 | 120 | 120 |

The Arachne assembler is perhaps most widely used for eukaryote genome assembly, yeilding high quality results. It requires roughly 2 weeks to fully assemble a genome of an insect *Drosophila*. Assembling the *Daphnia* genome with Arachne has proven difficult for us, with 3 trials failing due to memory limits. This assembler is provided as a binary executable for the Alpha processor, and it is not parallelized. Our runs were performed at rachel.psc.teragrid to use its Alpha architecture, using increasing memory allocations from 20 GB to 70GB. The PCAP assembler is parallelized, and has proven capable of assembly in 24 hrs using 120 cpus from TeraGrid sites. A total of 32,000 TeraGrid Service Units have been used in this preliminary Daphnia assembly.

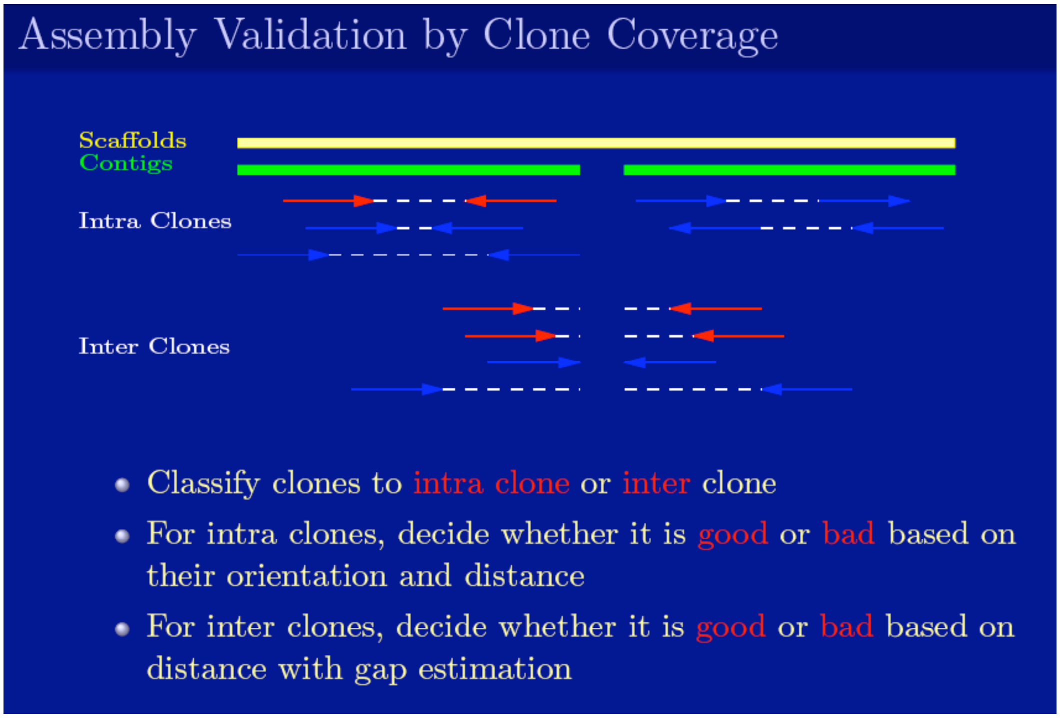
***Assembly Validation.*** A focus of this request is for validation of Daphnia assemblies. Why is assembly validation is required? Automated genome assemblers are susceptible to producing errors because of many regions of low complexity (repetitive sequence), sequencing errors, and uncloned regions (Kim et al 2001; Bartels et al, 2004). *De novo* validation involves examination of clone coverage, and fragment distribution in the assembly, that will indicate regions of high and low quality. Comparison of two or more assemblies using different coverage, assemblers, and parameters provides a statistical validation and can identify regions of uncertainty. However, assemblies can agree and still fail to model the underlying biological genome, due to common computational assumptions or sequencing problems. More compelling validation incorporates additional biological data: genetic maps, physical (BAC) maps, and cDNA libraries, to show where an assembly matches the underlying biological genome model. The strategy to be followed uses a hybrid of all these methods. Figure 4 outlines steps for validating assemblies by clone coverage.

## Resources justification for Daphnia assembly and validation

Assembly validation requires comparisons of the available Daphnia assemblies, including PCAP versions, an assembly produced by JGI using their proprietary JAZZ software. If we succeed with Arachne, that provides a third assembly. These can produce substantially different models of the underlying biological genome. One method of validation uses contig-level comparison of two assemblies. Estimated analysis time is 1/2-cpu hour for one contig. With roughly 70,000 contigs in the Daphnia assemblies, this will require 35,000 SU. At least two such cross-assembly validations will be needed. Following validation and analysis, re-assembly with adjusted methods is expected in order to produce a more valid genome representation. This will use an estimated 10,000 SU based in preliminary work.

***Validation details***. We will start the validation with the assemblies made by JAZZ, PCAP, and Arachne. We plan a final trial for obtaining the Arachne assembly, and if this succeeds we will use it, otherwise we will do without. Comparing two assemblies can use three methods that map contigs on one assembly to contigs on another assembly and depend on how a contig is represented: (1) common set, (2) longest common subsequence, (3) longest common overlapping interval. The last method would be the best. After refining this algorithm, it takes an estimated 100 SU in comparing one JAZZ assembly to PCAP assembly for 9X coverage. Comparison of 9 combinations of assemblies, using contigs and scaffolds will require 900 SU.

After determining the weakly assembled regions, we will reassemble using algorithm modifications determined from these comparisons. Re-assembly by PCAP of contigs broken at weak regions and unplaced reads will improve the genome representation. This will require half of the computational time as the first assembly, with potentially more as we develop the modifications to assembly algorithms.   Another method for validation is to use clone coverage.   An initial test of this, using the PCAP 9X assembly, required 500 SU.  Each assembly could be analyzed three times in the level of contig, scaffold, assembly, for 4500 SU.  Use fragment distribution for validation, will require similiar time to clone coverage.   The methods TAMPA (Dew et al 2005) and ThurGood (Shatkay et al 2004) will also be used to validate assemblies.  We do not yet have time estimates for these algorithms. We also use the BLAST program to map two assemblies, which is requires about 300 SU for a genome of this size.



Assembly Validation by Clone Coverage

- Classify clones to intra clone or inter clone
- For intra clones, decide whether it is good or bad based on their orientation and distance
- For inter clones, decide whether it is good or bad based on distance with gap estimation

**Figure 4.** Assembly validation using clone coverage.  This shows how well assembled scaffolds match underlying clone fragment data, The fragments in red are collapsed because of repeated sequence.

## References

Bartels D, Kespohl S, Albaum S, Druke T, Goesmann A, Herold J, Kaiser O, Puhler A, Pfeiffer F, Raddatz G, Stoye J, Meyer F, Schuster SC. 2004 BACCardI--a tool for the validation of genomic assemblies, assisting genome finishing and intergenome comparison. **Bioinformatics**. 2005 Apr 1;21(7):853-9.

Colbourne, J.K., Singan, V.R., Gilbert, D.G. 2005. wFleaBase: the Daphnia genome database, **BMC Bioinformatics**, 6:45 doi:10.1186/1471-2105-6-45 URL: wfleabase.org

Dew IM, Walenz B, Sutton G. 2005. A tool for analyzing mate pairs in assemblies (TAMPA). **J Comput Biol**. Jun;12(5):497-513.

Gilbert, D.G., Y Ugawa, M Buchhorn, T Tan Wee, A Mizushima, H Kim, K Chon, S Weon, J Ma, Y Ichiyanagi, D Liou, S Keretho, and S Napis, 2004. Bio-Mirror project for public bio-data distribution. **Bioinformatics**, 20:3238-3240. DOI: 10.1093/bioinformatics/bth219 URL: http://www.bio-mirror.net/

Huang,X., Wang,J., Aluru,S., Yang,S.-P. and Hillier,L. (2003) PCAP: a whole-genome assembly program,. **Genome Res**., 13, 2164–2170.

Huang X, Yang SP, Chinwalla AT, Hillier LW, Minx P, Mardis ER, Wilson RK, 2006 Application of a superword array in genome assembly. **Nucleic Acids Res**. Jan 5;34(1):201-5.

Jaffe,D.B., Butler,J., Gnerre,S., Mauceli,E., Lindblad-Toh,K., Mesirov,J.P., Zody,M.C. and Lander,E.S. (2003) Whole-genome sequence assembly for mammalian genomes: ARACHNE 2. **Genome Res**., 13, 91–96.

Kim, Sun, Li Liao, and Jean-Francois Tomb, 2001. A Probabilistic Approach to Sequence Assembly Validation, ACM SIGKDD Workshop on Data Mining in Bioinformatics (**BioKDD2001**), pp 38-43

Korf, Ian 2004. Gene finding in novel genomes. **BMC Bioinformatics** 2004, 5:59 URL: http://www.biomedcentral.com/1471-2105/5/59

Shatkay H, Miller J, Mobarry C, Flanigan M, Yooseph S, Sutton G. 2004. ThurGood: evaluating assembly-to-assembly mapping. **J Comput Biol.** ;11(5):800-11

Stein L.D., Mungall, C., Shu, S., Caudy, M., Mangone, M., Day, A., Nickerson, E., Stajich, J.E., Harris, T.W., Arva, A., and Lewis, S. 2002. The generic genome browser: a building block for a model organism system database. **Genome Res.** 12: 1599-610. URL: www.gmod.org