Published in final edited form as:

J Supercomput. 2009 January 1; 5698: 49–59. doi:10.1007/978-3-642-03275-2_6.

Multi-granularity Parallel Computing in a Genome-Scale Molecular Evolution Application

Jesse D. Walters^{2,4}, Thomas B. Bair², Terry A. Braun^{1,2,4,5}, Todd E. Scheetz^{1,2,4,5}, John P. Robinson^{1,4}, and Thomas L. Casavant^{1,2,3,4}

¹Center for Bioinformatics and Computational Biology, University of Iowa, Iowa City, IA 52242 USA

²Coordinated Laboratory for Computational Genomics, University of Iowa, Iowa City, IA 52242 USA

³Department of Biomedical Engineering, University of Iowa, Iowa City, IA 52242 USA

⁴Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242 USA

⁵Department of Ophthalmology and Visual Sciences, University of Iowa, Iowa City, IA 52242 USA

Abstract

Previously [1], we reported a coarse-grained parallel computational approach to identifying rare molecular evolutionary events often referred to as horizontal gene transfers. Very high degrees of parallelism (up to 65x speedup on 4,096 processors) were reported, yet the overall execution time for a realistic problem size was still on the order of 12 days. With the availability of large numbers of compute clusters, as well as genomic sequence from more than 2,000 species containing as many as 35,000 genes each, and trillions of sequence nucleotides in all, we demonstrated the computational feasibility of a method to examine “clusters” of genes using phylogenetic tree similarity as a distance metric. A full serial solution to this problem requires years of CPU time, yet only makes modest IPC and memory demands; thus, it is an ideal candidate for a grid computing approach involving low-cost compute nodes. This paper now describes a multiple granularity parallelism solution that includes exploitation of multi-core shared memory nodes to address fine-grained aspects in the tree-clustering phase of our previous deployment of XenoCluster 1.0. In addition to benchmarking results that show up to 80% speedup efficiency on 8 CPU cores, we report on the biological accuracy and relevance of our results compared to a reported set of known xenologs in yeast.

1 Introduction

Historically, evolutionary biological research has proceeded from painstaking taxonomical classification according to the physical characteristics of hundreds of organisms and species. The Human Genome Project and subsequent application of massive DNA and RNA sequencing capacity has made available rich datasets capable of answering broad evolutionary questions from a molecular point of view. These trillions of nucleotides of DNA sequence data from thousands of species containing as many as 35,000 genes each makes it possible to pose biological and biomedical questions that just a few years ago would have been inconceivable. In our previous paper [1], we described XenoCluster 1.0

which addressed one such setting known as horizontal gene transfer, or jumping genes. In addition, general issues facing the use of heterogeneous networks for such problems were addressed. The results of that initial implementation showed that it was possible to efficiently harness more than 4,096 processors organized in a heterogeneous grid of modest sized compute clusters to reduce the overall computation time of a typical problem setting from more than 2 years to roughly 12 days. This paper extends that initial work to exploit fine-grained shared memory parallelization of previously unparallelized portions of that application which further reduces practical execution times by another order of magnitude. We also address the issue of biological accuracy and relevance of predictions using recently reported xenologs in *S.cerevisiae* (budding yeast). kground and Related Work

For the sake of completeness, an abbreviated overview of some background information as well as our general discovery method [1] is presented here.

2 Biological Background

Typical genes are transferred through lineages, from one generation to the next within a species. However, an alternate form of “inheritance” is possible in which genetic material crosses species boundaries. This form of inheritance is termed horizontal gene transfer. Our ability to identify patterns of horizontal gene transfer can increase our understanding of the evolution of species and the structure of the tree of life.

General features of horizontal gene transfer include higher inter-species sequence similarity between two taxa (species) that are in different clades (branches) of the consensus tree. To accomplish this, a broad set of species must be sampled. The limiting factors in this process are the availability of sequence for a large number of taxa and the capability to harness sufficient computational power to identify orthologous sequences, construct phylogenetic trees for each orthologous set of genes, and then compare the trees derived from each orthologous set to identify non-evolutionary inheritance patterns.

2.1 Computational Background

Several operations are necessary to identify horizontal gene transfers, or other anomalous gene inheritance events. First, orthologous sequences must be identified. Next these sequences must be aligned, and phylogenetic trees created. Finally, the tree structures must be compared to identify atypical patterns of inheritance.

There are several computational methods for determining orthology. The most commonly accepted method is based upon a strongly connected graph constructed of nodes representing genes across species in which each element is more similar to the others in the set than to genes from outside the set. There are several previously generated set of orthologs publicly available. These include COG and KOG [2], OrthoMCL [3] and EGO [4]. The method employed in our work incorporates sequences from all organisms available in NCBI's non-redundant amino acid database (NR) and uses a straight-forward reciprocal BLAST criteria.

Phylogenetic analysis allows determination of the most likely pattern of inheritance of a gene. Programs such as PHYLIP [5] and PAUP [6] are commonly used to construct phylogenetic trees, based upon an aligned set of orthologous sequences. High-performance parallel phylogenetic inference has been extensively studied by Stamatakis in the RAXML-VI-HPC with impressive performance optimizations [7]. Due to its widespread acceptance and usage, we use PHYLIP in our work.

3 Methods and Solution

We now summarize our parallel grid of clusters solution in heterogeneous-latency networks. The algorithm is divided into 3 major phases:

1. Identification of a maximal set of orthologous genes.
2. Generation of phylogenetic trees resulting from orthologous groups.
3. Clustering of these trees into groups corresponding to genes which show consistent evolutionary behavior.

In phase 1, it is necessary to identify potential homologous genes for every gene in the union of a complex set of 1000s of species. This is accomplished by BLASTing [8] each gene against the set of all known genes in all species, and then performing a reciprocal BLAST operation to verify that the best hit for each gene hits the original gene with the highest rank score. This becomes the base set of orthologous gene groups to be used in phase 2 among which xenologs may be identified. The second phase involves the sequence trimming and multiple alignment of all members of each of the orthologous gene groups, followed by the automated generation of a phylogenetic tree for each aligned group. The final phase performs an all-pairs distance analysis of phylogenetic trees for all gene groups, and then uses a clustering technique to identify maximal sets of trees, which represent sets of genes which share a common evolutionary history. Design details of the procedure outlined above are presented in our previous paper [1], and are illustrated summarily in Figure 1. The highlights are repeated here for completeness only.

Ortholog identification was performed using the COE (Computational Orthology Engine) system, developed at the University of Iowa. COE identifies orthologous sequence groups using a reciprocal best-alignment strategy. Each mRNA RefSeq [9] sequence for a base species was BLASTed against NCBI's non-redundant amino acid database. For each BLAST result, the top hit of each species was selected, if and only if it met a stringent quality threshold criterion. If the threshold criterion was met, a reciprocal BLAST was performed with these top species hits against the RefSeq database [9] to further support the orthology inference.

Text parsing of the BLAST results was performed using custom BioPerl [10] scripts, and batch scheduling of all BLAST operations in phase 1 was performed using the Portable Batch System [11].

Once ortholog identification has been performed, the next phase is trimming, alignment, and phylogenetic tree generation. The trimming of sequences is done using a custom Perl script. Multiple sequence alignment is accomplished using the well-established clustalw software [12]. The final step in this phase is the generation of the phylogenetic trees – the PHYLIP [5] software suite was used to generate each of the trees. These programs generated the sequence distance matrices and the phylogenetic trees for each of the bootstrap replicates. Finally, the consensus program was run to obtain the consensus phylogenetic tree based upon the bootstrap replicates.

Finally, the phylogenetic tree clustering was performed from the results of the PHYLIP software package. This involves two main sub-phases – distance matrix generation, and clustering. Inter-tree distance was calculated using a modified version of the TreeRank [13] algorithm. Our adaptation of this algorithm was implemented in POSIX C with pthread support. Development of this software was done on Fedora 9.0 and Redhat Enterprise machines. The second sub-phase involved clustering, given a complete distance matrix from every tree to every other tree.

3.1 Grid/Cluster Implementation and Benchmarking Details

Each of the phases described above were implemented in a LINUX environment (2.2GHz dual Athlon with 2GB RAM running Fedora Redhat 9.0), and benchmark executions were performed using the largest set of human genes in April of 2005. For this analysis, and all benchmarks, this consisted of the set of all 20,364 known human RefSeq mRNA sequences. Runtime estimates for the first phase of the computation, which involved the COE system, varied significantly with system threshold parameters. The initial iteration of the system yielded an average of 588 cpu seconds per RefSeq mRNA. Variations of the aforementioned match length, alignment score and e-value thresholds can change the number of reciprocal BLASTs performed and therefore the average runtime. Thus, the values reported in Table 1 are an average taken across the entire set of 20,364 genes. The COE results yielded an average of 12.6 orthologs per human RefSeq mRNA. More relevant to performance, an average of 39 reciprocal BLASTs were performed for each RefSeq mRNA. Thus, approximately 32% of the reciprocal alignments were considered “true” orthologs by our method.

In our previous work [1], we discussed in detail the effects of deployment of XenoCluster on a large-scale grid of compute clusters. We utilized the figure of 588 seconds to estimate and bound the runtime of the entire dataset through the COE system. To confirm the accuracy of our execution-time predictive model, 20,364 mRNAs at an average of 588 seconds would yield 3,326 CPU hours of compute-time. A benchmark was then performed on a 16-node Linux cluster where the observed execution time was shortened to 207 cluster hours. This was very close to the expected time of approximately 12 days. Runtimes for PHYLIP were 579 cpu seconds at 100 bootstrap iterations, while an average of 2,518 cpu seconds was achieved at 500 bootstrap iterations. The tree clustering phase (UIPTC) results were extrapolated to reveal the estimated overall runtime of the full set of 20,364 genes.

Table 1 summarizes the detailed times (in wall-clock time units of seconds) of 5 computational and 2 communication elements of XenoCluster. Details are provided in our previous paper. Note that the first four computational elements parallelize cleanly across all genes. However, in the final phase, the times for tree clustering (UIPTC) do not show the effects of parallelization.

4 Results and Discussion

In [1] we presented the performance results of the coarse-grained grid-based parallelization of the approach outlined above. Now we present the results of fine-grained parallelization of the Tree Clustering (UIPTC) phase, as well as the biological validation of our method. In our benchmarking results shown above, UIPTC requires approximately 12.2 days of computational time. While this amounts to only 2% of the serial execution time as shown in Table 1, this phase now comprises 99% of the best-case parallel execution time as shown in Table 2.

The fine-grained parallelization of the UIPTC algorithm was done using the pthread multi-threading package for Linux 2.4 based kernels [14]. This package allows for several threads or light-weight processes to share memory but maintain independent execution paths. The most obvious place to harness multiprocessor capabilities in UIPTC was in the treeSim() function [1]. In this function, $(N \times N)/2$ independent comparisons of the trees to one another is performed. These comparisons were randomly partitioned into sets. Each set is then run in a different thread.

The distribution of work to threads was accomplished by assigning a unique ID to each thread and providing the total number of threads in the process to each thread. The ID and

number of threads allowed each thread to process a given row I in the $N \times N$ matrix and perform the following operation: $\text{mod}(I, \text{numThreads}) == \text{ID}$. If this statement was true, the thread takes local responsibility for all the computation in that row.

Table 3 shows the results of parallelization of UIPTC on a set of HP workstations with dual-core Opteron processors. For small numbers of CPUs, almost linear speedup is observed. The use of the rudimentary load balancing as mentioned earlier was adequate, and provided threads with desirable computational load. Although excellent speedup was achieved, most users will be limited to 2 or 4 CPUs per system simply because most x86 architectures do not scale well above 8 CPUs. Thus, 16 and 32 thread implementations would be unlikely to be efficient without additional message passing infrastructure [15]. Figure 2 and Table 3 show runtime speedup and efficiency results for UIPTC with different numbers of threads. Note that efficiency begins to decrease as the number of CPUs approaches 8. This can be attributed to memory latency and system bus bottlenecks. However, speedup is achieved as one increases the number of threads from 7 to 8, which means additional speedup could be achieved by simply using more CPUs. Note that memory usage was less than 10 megabytes and therefore not a limiting factor, however, memory demand grows at an $O(n^2)$ rate, which means it may be something to consider as datasets grow. Benchmarking was performed using HP workstation using 4 2.2 GHz Dual Core Opteron Processors. The benchmarking dataset was the complete yeast tree build described above.

4.1 Biological Validation and Interpretation of Results

Results for the COE phase of the XenoCluster system applied to the *S.cerevisiae* (yeast) species yielded on averaged 30 orthologs per yeast RefSeq gene. These orthologs were, on average, 361 amino acids long. A total of 128,190 orthologs were identified from 4,234 of 7,001 RefSeq genes. The 2,767 genes which did not yield orthologs, were most likely alternative transcripts which already were represented in the original gene set. This is a reasonable explanation as it is estimated that *S.cerevisiae* has approximately 5,000 genes. Of the 4,234 yeast gene ortholog sets identified above, 2,650 trees were produced from our PHYLIP pipeline. Approximately 2,000 trees were not created because they lacked the minimum criteria of 5 orthologous species. This criterion was utilized because trees with less than 5 species may mis-represent an HGT event.

Several iterations of phylogenetic clustering of the yeast genome were then performed utilizing several different similarity threshold values. These thresholds were the minimum TreeRank similarity needed to classify a tree in question, into a tree/gene cluster. Threshold values varied from 60 % up to 97 percent. Figure 3 shows the number of clusters formed for each of the threshold values used. Most of the clusters remain very small, while one or two large “consensus” clusters represent the majority of the sequences. For example, for the 95% identity clustering, the 87 resulting clusters consisted of 1 cluster of size 2129 trees, and 86 relatively small clusters consisting of fewer than 5 trees each.

To validate the utility of this approach, a list of 10 independently identified *S.Cerevisiae* horizontal gene transfer (HGT) cases were identified from the published in biological literature [16] and observed in the XenoCluster dataset. Each of the 10 RefSeq identifiers were BLASTed against the latest RefSeq release to identify their current RefSeq ids. We then examined the position of these genes in the resulting clustering. The results of this analysis are shown in Table 4. Four of the ten genes (YFR055W, YMR090W, YOL164W, YJL217W) lacked sufficient homology to other available species to meet our criteria of 5 species for reciprocal BLAST hits, and thus could neither be confirmed nor denied as valid xenologs. Of the remaining six, two were identified as belonging to singleton clusters (i.e., lacked similarity to the phylogenetic trees of any other yeast genes) at thresholds of 85% (YDR540C) and 90% (YJL218W). These two examples provide the strongest evidence of

the validity of our approach. However, the remaining four cases (YPL245W, YKL216W, YNR057C, and YNR058W) were as yet unaccounted for.

In addition to a percent identity threshold, a secondary clustering criterion was implemented to improve HGT detection. The secondary criterion specifies a minimum number of trees that must meet the percent identity threshold in order for the tree in question to be incorporated into the cluster. Currently, if a single tree-tree comparison yields a result greater than or equal to the percent identity threshold, the tree in question will become part of that cluster. By applying this criteria to the single largest cluster (the one that represents the canonical structure the tree illustrating the true evolutionary position of yeast in the tree of life), we are able to distill trees (or genes) whose similarity to this dominant structure only consists of links to very few other members of that cluster. Our hypothesis was that these genes would also be excellent candidates as the result of GHT events.

With our current clustering results, we have observed several strongly related trees within the large consensus cluster. These trees may have as many as 1,005 tree-tree comparisons with other trees in the same cluster that are greater than or equal to the 95% percent identity threshold. However, many of the trees in this large cluster have only 1 tree-tree comparison that is equal to or greater than the selected percent identity threshold. By requiring more than a single tree-tree match greater than or equal to the specified percent identity, we were able to identify clusters that contain only trees that are strongly related to each other. Figure 4 displays in sorted order, the number of tree-tree comparisons from the consensus cluster that have greater than or equal to the 95 percent identity threshold. The four documented HGT cases that are in this cluster all have 25 or fewer tree-tree comparisons that meet the 95% identity threshold. The strongest of these is YNR058W, which shows 95% similarity to only one other gene in the large cluster. Thus, of the six known cases of HGT in yeast which have sufficient orthology information, all are known to be either very unique with respect to the phylogenetic tree structure, or to only have weak evidence for similarity to other genes.

References

1. Walters, J.; Casavant, T.; Robinson, J.; Bair, T.; Braun, T.; Scheetz, T. XenoCluster: A Grid Computing Approach to Finding Ancient Evolutionary Anomalies. In: Malyshkin, VE., editor. PaCT 2005. LNCS. Vol. vol. 3606. Springer; Heidelberg: 2005. p. 355-366.
2. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, Koonin EV, Krylov D, Mazumder R, Mekhedov SL, Nikolskaya AN, Rao BS, Smirnov S, Sverdlov AV, Vasudevan S, Wolf YI, Yin JJ, Natale DA. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*. 2003; 4(1):41. [PubMed: 12969510]
3. Li L, Stoeckert C Jr. Roos DS. OrthoMCL. Identification of Ortholog Groups for Eukaryotic Genomes. *Genome Res*. 2003; 13:2178–2189. [PubMed: 12952885]
4. Lee Y, Sultana R, Pertea G, Cho J, Karamycheva S, Tsia J, Parvizi B, Cheung F, Tonescu V, White J, Holt I, Liang F, Quackenbush J. Cross-referencing eukaryotic genomes: TIGR orthologous gene alignments (TOGA). *Genome Research*. 2002; 12(3):493–502. [PubMed: 11875039]
5. Felsenstein J. PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*. 1989; 5:164–166.
6. Swofford, D. LPAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4. Sinauer Associates; Sunderland, Massachusetts: 2003.
7. Alexandros S. RAXML-VI-HPC: Maximum Likelihood-based Phylogenetic Analyses with Thousands of Taxa and Mixed Models. *Bioinformatics*. 2006; 22(21):2688–2690. [PubMed: 16928733]
8. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J. Mol. Biol.* 1990; 15:403–410. [PubMed: 2231712]
9. Pruitt KD, Katz K, Sicotte H, Maglott DR. Introducing RefSeq and LocusLink: curated human genome resources at the NCBI. *Trends Genet.* 2000; 16(1):44–47. [PubMed: 10637631]

10. Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, Fuellen G, Gilbert JGR, Korf I, Lapp H, Lehvaslaiho H, Matsalla C, Mungall CJ, Osborne BI, Pocock MR, Schattner P, Senger M, Stein LD, Stupka ED, Wilkinson M, Birney E. The Bioperl Toolkit: Perl modules for the life sciences. *Genome Research*. 2002; 12(10):1611–1618. [PubMed: 12368254]
11. PBS Pro. <http://www.pbspro.com/>
12. Thompson JD, Higgins DG, Gibson TJ. Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Res*. 1994; 22:4673–4680. [PubMed: 7984417]
13. Wang, JTL.; Shan, H.; Shasha, D.; Piel, WH. TreeRank: A Similarity Measure for Nearest Neighbor Searching in Phylogenetic Databases. *Proceedings of the 15th International Conference on Scientific and Statistical Database Management (SSDBM 2003)*; Cambridge, Massachusetts. 2003. p. 171-180.
14. Nichols, B.; Buttlar, D.; Farrell, JP. *Pthreads Programming A POSIX Standard for Better Multiprocessing*. O'Reilly; Sebastopol: 1996.
15. Squyres, JM.; Lumsdaine, A. A Component Architecture for LAM/MPI. In: Dongarra, J.; Laforenza, D.; Orlando, S., editors. *EuroPVM/MPI 2003*. LNCS. Vol. vol. 2840. Springer; Heidelberg: 2003. p. 379-387.
16. Hall C, Brachat S, Dietrich FS. Contribution of horizontal gene transfer to the evolution of *Saccharomyces cerevisiae*. *Eukaryot Cell*. 2005; 4(6):1102–1115. [PubMed: 15947202]

Fig. 1.
A detailed flowchart of the XenoCluster approach

Fig. 2.
UIPTC runtime and efficiency results from Table 3

Fig. 3.

Tree similarity threshold versus the number of resulting clusters. At very low thresholds, all genes/trees form a single cluster, while at high thresholds the number of clusters increases, yet most genes are found to be contained in a small number of relatively large clusters.

Fig. 4. Analysis of the number of tree-tree comparisons meeting the similarity criteria specified. The largest number of connections is a tree with similarity of over 97% to 1,004 other trees. There are approximately 150 trees with fewer than 25 links to other trees

Table 1

Benchmark timings on 20,364 human genes for the component phases of XenoCluster run with 1 dual CPU node (cluster size 1). Timings taken on a 2.2GHz dual Athlon with 2GB RAM running Fedora Redhat 9.0.

Phase/component	Time (Seconds)	# of Iterations	Total (Seconds)
Intra Cluster IPC	124	1	124
Inter Cluster IPC	311	1	311
Initial BLAST	301	20,364	6,129,564
Reciprocal BLAST	12	794,196	9,530,352
Sequence Alignment	33	20,364	672,012
PHYLIP tree generation	2,518	20,364	51,276,552
Tree Clustering	1,036,800	1	1,036,800
Total	1,040,099	855,291	68,645,715
Days to completion			794.5106

Wall-clock Time Breakdown N=1 K=1

Phase	Percentage
Initial Blast	2%
Reciprocal Blast	9%
Sequence Alignment	14%
PHYLIP	74%
Tree Clustering	1%

Table 2

Execution times with a coarse-grained cluster-parallel implementation utilizing K=128 Linux Clusters, and N=32 CPUs/Cluster. A total of 4,096 processors in all for a net speedup of 65.

Phase/component	Time (Seconds)	# of Iterations	Total (Seconds)
Intra Cluster IPC	124	128	15,872
Inter Cluster IPC	311	32	9,952
Initial BLAST	301	5	1,505
Reciprocal BLAST	12	194	2,328
Sequence Alignment	33	5	165
PHYLIP tree generation	2,518	5	12,590
Tree Clustering	1,036,800	1	1,036,800
Total	1,040,099	242	1,079,212
Days to completion			12.5

Table 3

UIPTC parallel runtime and efficiency results for 1 through 8 CPUs

Number of CPUs	Observed Runtime	Theoretical Runtime	Efficiency%
1	7434	7434	100
2	3781	3717	98.27818133
3	2548	2478	97.17514124
4	1995	1858.5	92.65536723
5	1699	1486.8	85.72773742
6	1403	1239	86.76351897
7	1227	1062	84.46327684
8	1116	929.25	79.9031477

Table 4

Validation of Xenocluster using “known” yeast Xenologs [16]. %FS is the percentage to form a singleton cluster. %JL is the percentage to join the large cluster. #LL is the number of links to other trees in the largest cluster.

Yeast Gene ID	Gene name	%FS	%JL	# LL
YFR055W	Hypoth Protein			
YMR090W	Hypoth Protein			
YOL164W	BDS1			
YJL217W	Hypoth Protein			
YDR540C	Hypoth Protein	85%		
YJL218W	Hypoth Protein	90%		
YPL245W	Hypoth Protein		97%	25
YKL216W	URA1		97%	12
YNR057C	BIO4		97%	7
YNR058W	BIO3		97%	1