

INTERNSHIP REPORT

COMPARE AND CONTRAST THE EFFECTS OF USING LESS STRINGENT CRITERIA IN BLASTCLUST TO A NOVEL ITERATIVE METHOD FOR IDENTIFYING GENE FAMILIES

An internship report presented in
partial fulfillment of the requirements
of the Professional Science Master's
in Computational Biosciences

Virginia Earl-Mirowski
Computational Biosciences Program
Arizona State University

Dr Michael Rosenberg Ph.D.
Internship Advisor
School of Life Sciences
Arizona State University

Internship:
From: September 2006 - May 2007

THIS REPORT IS NOT CONFIDENTIAL

Technical Report Number: 07-03
July 30, 2007

ABSTRACT

Gene family relationships evolve over time by speciation and gene duplications. In a previous study, a novel iterative method was developed utilizing a variety of techniques for identifying members of gene families through ancestral predictions. The goal of this study was to compare and contrast the effects of using less stringent threshold criteria in the *blastclust* program to the novel iterative method. Several iterations of the novel iterative method were performed setting the *blastclust* threshold criteria to various levels. The initial and post ancestral sequence clustering results were compared between the default and various alternative approaches. Results indicated that a relaxed similarity setting by itself or in conjunction with relaxed coverage setting, of *blastclust* grouped more sequences initially. This combination also clustered singletons into significant cluster sizes overlooked by the *blastclust* program with the default settings. Using a relaxed coverage option alone, this produced similar results to the default threshold setting, with few additional new sequences.

ACKNOWLEDGEMENT

I would like to thank my advisor, Dr. Michael Rosenberg, for his guidance, support, and direction regarding my project. I would also like to thank my husband Joe for his loving support and for the help he provided in doing this report, and my committee members, Dr. Jeffrey Touchman and Dr. Martín Wojciechowski; not only for their feedback for this report but also for their enthusiasm and mentoring as instructors in the Computational Biosciences Program.

TABLE OF CONTENTS

ABSTRACT.....	2
ACKNOWLEDGEMENT.....	3
GOAL.....	6
INTRODUCTION.....	7
METHODS.....	11
RESULTS.....	17
Performance/Timing Using Neighbor / hit List.....	17
Data Integrity.....	17
Initial Clustering of Data.....	19
Clustering of Data after Ancestral Sequences are Added.....	19
Comparison of Default Run Additional Sequences.....	20
Addition Sequences Captured.....	21
CONCLUSIONS AND FUTURE DIRECTIONS.....	24
REFERENCES.....	26
APPENDIX A:.....	28

LIST OF FIGURES

Figure 1 - Gene Family Extension Workflow Methodology.....	9
--	---

LIST OF TABLES

Table 1 blastclust Threshold Settings.....	12
Table 2 Software Used in Default Run.....	13
Table 3 Software Used in Threshold Runs	15
Table 4 Largest Default Cluster (after Ancestral Sequences) vs. Largest Cluster (initial) of Threshold Runs	18
Table 5 Initial Clustering of Families for All Workflow Runs.....	19
Table 6 Clustering with Ancestral Sequences of Families for All Workflow Runs	20
Table 7 Cluster of 10 Sequences Initial Clustering All Threshold Runs	21
Table 8 Clustering of 10 Sequences After Ancestral Clustering Threshold Runs 1,2 and 5	21
Table 9 New Sequences Captured Using relaxed <i>blastclust</i> Threshold Settings.....	22
Table 10 Distribution of Threshold Run 6 Additional Sequences	23
Table 11 Blastclust Parameters and Default Values	28

GOAL

The goal of this project was to compare and contrast the effects of using less stringent threshold criteria in *blastclust*[17] to the novel iterative method, which made use of the predicted ancestral sequences to identify gene families. This study also utilized the neighbor / hit list in its *blastclust* settings to determine if this reduces the processing time of the project workflow.

INTRODUCTION

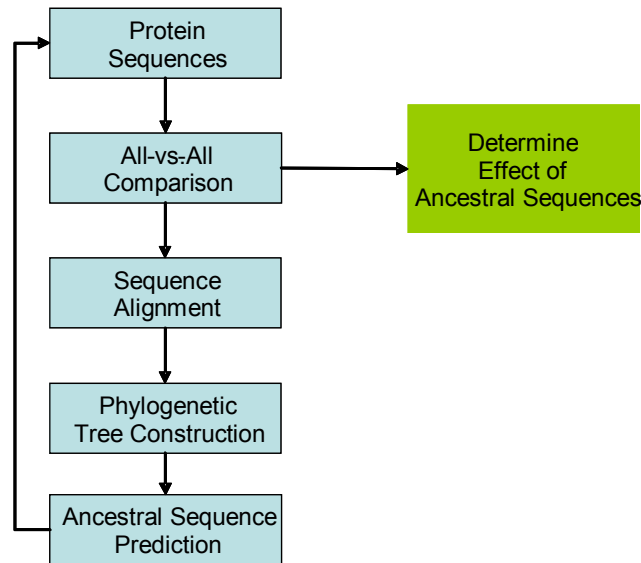
Gene families are described as a group of genes that are united by characteristics that they have in common [16]. For most researchers, this commonality is in the structure, function, and in the sharing a common ancestor. Each gene has evolved to be unique, but share common building blocks. These families exist whether these genes are within an organism, across a species and large clades [11]. The general term homolog describes such genes, their associated proteins, and DNA sequences. Paralogs are genes related by duplication in a genome on an organism. New functionalities evolve even if the new functionality is related to the original functionality. Orthologs are homolog genes that exist across a species, evolve by speciation, and retain the functionality [7]. Gene families are important to understanding the gene and protein function. When a new gene is identified, researchers can predict the new gene's functionality based on the known gene family with which it is clustered. In a study done by Liepman et al [11], it was observed that function was conserved in the CslA gene family; a gene family present in various mosses, ferns, gymnosperms and mono- and dicotyledonous angiosperms. Varieties of methods have developed to classify homologous families. [3,5,9,12]

This study is an extension of a work by former CBS student Loretta Goldberg [6] where a workflow was developed utilizing a variety of techniques that identified members of a gene families through ancestral predictions. The goal of that research paper was to show proof of concept for a new iterative method for identifying member of gene families using predicted ancestral sequences. The project workflow developed is shown in Figure

1. In the first step, the protein sequences are obtained from NCBI's nr database. An all-vs.-all comparison is used to cluster homologous sequences together using the program *blastclust* [17]. The CLUSTAL program [14] is used for the sequence alignment and is used to construct a phylogenetic tree for the aligned sequences of each cluster using the neighbor joining method. The PAML program [15] is used to reconstruct ancestral sequences using maximum-likelihood prediction. All predicted ancestral sequences are inferred for each cluster. These predicted ancestral sequences are added to the database of the protein sequences and a second all-vs.-all comparison is performed on the sequences. The effect of the predicted ancestral sequences is determined by comparing the set of clusters from both all-vs.-all comparisons. It concluded with proof that additional homologies could be captured to existing clusters using this method

In this workflow, the NCBI *blastclust* tool was used in the all-vs.-all comparison using its default settings. One of the future directions targets mentioned was to investigate relaxing the threshold criteria of *blastclust*. Researchers argue that restrictive cut offs may result in some cluster members being missed. This is especially true if sequences have large phylogenetic distances. [13] This study addresses that question and examines how the workflow results are affected by relaxing the *blastclust* threshold criteria.

Figure 1 - Gene Family Extension Workflow Methodology



Many studies of gene families use a clustering tool as initial step in their analysis and *blastclust* is widely used for this purpose. [3,12] *Blastclust* is designed to rapidly cluster sequences for further analysis. *Blastclust* groups a given set of sequences together based on similarity and coverage thresholds. Thus, *blastclust* results are very sensitive to the threshold parameters set. However, a search through literature provided no information on which particular threshold settings were used for *blastclust* in published studies.

Blastclust is a program that is part of the NCBI-BLAST [1] standalone distribution available for download. It is designed to rapidly cluster related protein or nucleotides sequences. *Blastclust* utilizes single-linkage clustering algorithms, the blastp algorithm for proteins, and the MegaBlast algorithm for nucleotide sequences.[16] The clustering algorithms use pair-wise matches to place a sequence in a cluster. If a sequence is

considered a “neighbor” to at least one sequence in the cluster then it will be placed in that cluster.

BLAST similarity searches are based on the heuristic technique of filtration. [7] The concept behind filtration is that good alignments contain short stretches that have a high degree of similarity. When these short stretches of sequences are located, they are used as the basis for expansion to a longer good alignment. BLAST uses matrices to improve the efficiency of filtration.

The High Scoring segment Pair (HSP) is a segment pair with the best score over all other segment pairs in the two sequences. In the *blastclust* program, clustering is based on two criteria: the coverage threshold and the similarity threshold. [16] The coverage of each sequence is equal to the HSP length of the sequence divided by sequence length. The coverage threshold criterion is based on the maximum or minimum coverage value, between the coverage of the two sequences. This value depends on the setting of the ‘-b’ option, which is set to True or False. If the ‘-b’ option is set to true then both sequences must be greater than or equal to the coverage and similarity thresholds, while only one sequence must meet this requirement if ‘-b’ is set to false. The similarity threshold is based on either the BLAST score density or the percentage of identical residues. The BLAST score density is defined as the BLAST score divided by the minimum HSP length of the two sequences; while the percentage of identical residues is equal to the number of identical residues per alignment length. For two sequences to be considered neighbors, both the similarity and the coverage thresholds have to be greater than or

equal to the threshold value. *Blastclust* has options to adjust these threshold values. The similarity option (-S) can be set to BLAST score density range of 0.0 to 3.0 or to a percentage of identical residues 3 to 100. The default value is a score density of 1.75. The minimum length coverage option (-L) has a default value of .9 but has a range of 0.0 to 1.0. The default parameter settings of *blastclust* are listed in Appendix A.

METHODS

For this study, the software processing methodology was executed as outlined in the previous study by Goldberg [6]. NCBI's non-redundant (nr) protein database was downloaded on 7 September 2006. To better understand the predicted ancestral sequence workflow, the yeast.aa database was used. This species database is comprised of sequences from *Saccharomyces cerevisiae* (baker's yeast) and *Schizosaccharomyces pombe* (fission yeast) and is relatively small. For examining the effects of the relaxed *blastclust* threshold setting, the species database for *Rattus norvegicus* (rat), was used because it was more complex organism and the database was small enough to avoid processing bottlenecks. Computing was performed locally on a Dell Optiplex GX620 PC with Intel® Pentium® 4 CPU 3.40 GHz, 1.00 GB of RAM running Microsoft Windows XP Professional Version 2002 Service Pack 2. All additional programs were written in Perl.

A baseline of the rat species sequences extension was executed using the same process as the previous study (default run 0), where all *blastclust* threshold options used the default threshold settings with a neighbor / hit list created in the initial execution *blastclust*. Next, the workflow was executed several times (threshold runs 1-6) using the neighbor/ hit list

and different settings of the threshold options in *blastclust* as seen in Table 1. A search of related literature for guidance on appropriate *blastclust* threshold settings for this study was conducted but none could be found. Therefore, conservative adjustments were made to the threshold settings that attempted to balance relaxed criteria to yield new relationships but not so relaxed that it was hard to determine the relationships between sequences in a cluster.

Table 1 blastclust Threshold Settings

Run #	Similarity setting (-S)	Min. Length Coverage (-L)	Both/One coverage (-b)
0	default	default	both
1	default	0.8	both
2	default	default	one
3	1.65	default	default
4	1.65	0.8	default
5	default	0.7	default
6	1.55	0.7	default

The ancestor prediction processing or the default run was performed using the processing steps as listed in Table 2. The additional step to this processing methodology is the `removeNonAnc.pl` program, which move files from the processing directory that failed to produce ancestral information.

Table 2 Software Used in Default Run

Input	Software Used	Output
NCBI's nr database	spFilter.pl	speciesDB (FASTA)
speciesDB(FASTA)	dbFormatter.pl (formatdb)	speciesDB (BLAST)
speciesDB(BLAST)	bClust.pl (blastclust)	cluster list neighbor/hit-list
cluster list	countClust.pl	summary of cluster sizes
cluster list	collectClust.pl	cluster files (FASTA)
cluster list (FASTA)	buildClustalBat.pl	bat file (clustalw)
cluster list (FASTA)	bat file (clustalw)	alignment files (PHYLIP-C) tree files (PHYLIP-C)
alignment files (PHYLIP-C)	modifyAlignFiles.pl	alignment files (PHYLIP-P)
tree files (PHYLIP-C)	modifyTreeFiles.pl	tree files (PHYLIP-P)
alignment files (PHYLIP-P) tree files (PHYLIP-P)	buildCtlFiles	control files bat file(codeml)
alignment files (PHYLIP-P) tree files (PHYLIP-P) control files	bat file (codeml)	ancestral sequences files ancestral sequences
ancestral sequence files	removeNonAnc.pl	(FASTA)
ancestral sequence files	extractAnSeq.pl	ancestral sequences (FASTA)
ancestral sequences (FASTA)	concatenate files together	speciesDB + AnSeq(FASTA)
speciesDB + AnSeq(FASTA)	dbFormatter.pl (formatdb)	speciesDB + AnSeq(FASTA)
speciesDB + AnSeq(BLAST)	bClust.pl (blastclust)	speciesDB + AnSeq(BLAST)
cluster list + AnSeq cluster list	compareClust.pl	summary of clusters combined and sequences maintained and added

Processing for the different threshold cases were performed in a similar methodology as the default case and are listed in Table 3. The `bClustThres.pl` program is modified version of the `bClust.pl`, which uses neighbor / hit list to reduce the initial clustering processing time. The `bClustThres.pl` program allows the user to select and specify which of the threshold options are used, as well as specifying if the neighbor / hit list is required.

The *blastclust* program output is a file that consists of clusters of sequences identifiers, sorted from largest clusters to the smallest, one cluster to a line. Phylogenetic trees and ancestral sequences are generated for each cluster. To ensure that phylogenetic trees would be constructed with multiple generations of ancestral sequences, clusters of six or greater were used for subsequent processing in the workflow for all threshold runs. This cluster size window was based on that the number of ancestral sequences for a phylogenetic tree is two less than the number of starting sequences.

For the default and threshold runs, the sequence alignment and phylogenetic tree reconstruction steps are performed using the program CLUSTAL and the ancestral sequence prediction were performed using the PAML program. All output files and processing were as described in [3]. The ancestral sequences are given unique identifiers, starting with '9999' plus a serial count, so that they can be distinguished from the NCBI sequences and so they can be traced back to their derived cluster. The ancestral sequences are loaded in the original database for each run and using *formatdb*, and a new BLAST database was constructed. For the all-vs.-all clustering with the ancestral sequences, the `bClustThres.pl` was executed for threshold runs 1-6 without the neighbor / hit list as to

have avoid predetermined restrictions. The bClust.pl program was used for the default run for its all-vs.-all clustering with ancestral sequences.

Table 3 Software Used in Threshold Runs

Input	Software Used	Output
speciesDB(BLAST) neighbor / hit-list blastclust options to use	bClusThres.pl (blastclust)	cluster list
cluster list	countClust.pl	summary of cluster sizes
cluster list	collectClust.pl	cluster files (FASTA)
cluster list (FASTA)	buildClustalBat.pl	bat file (clustalw)
cluster list (FASTA)	bat file (clustalw)	alignment files (PHYLIP-C) tree files (PHYLIP-C)
alignment files (PHYLIP-C)	modifyAlignFiles.pl	alignment files (PHYLIP-P)
tree files (PHYLIP-C)	modifyTreeFiles.pl	tree files (PHYLIP-P)
alignment files (PHYLIP-P) tree files (PHYLIP-P)	buildCtlFiles	control files bat file(codeml)
alignment files (PHYLIP-P) tree files (PHYLIP-P) control files	bat file (codeml)	ancestral sequences files
ancestral sequence files	removeNonAnc.pl	ancestral sequences (FASTA)
ancestral sequence files	extractAnSeq.pl	ancestral sequences (FASTA)
speciesDB(FASTA) ancestral sequences (FASTA)	concatenate files together	speciesDB + AnSeq(FASTA)
speciesDB + AnSeq(FASTA)	dbFormatter.pl (formatdb)	speciesDB + AnSeq(FASTA)
speciesDB + AnSeq(BLAST) blastclust options to use	bClusThres.pl (blastclust)	speciesDB + AnSeq(BLAST)
cluster list + AnSeq cluster list	compareClust.pl	summary of clusters combined and sequences maintained and added

The CompareClust.pl program was executed for the default and the threshold runs. This program used a minimum cluster size of six (6) compares the two cluster files, initial and post ancestral sequence additions. The program counts the number of sequences of all clusters in the two files and compares them to decide if additional sequences were added to a cluster as a result of the ancestral sequences. It also determines if new sequences in a cluster after ancestral sequences were added existed anywhere within the initial clustering.

The analysis of the results of the less stringent settings of blastclust centered on the output of the initial and post-ancestral sequence all-vs.-all comparisons. A data integrity check was performed on a subset of clusters, to verify that the sequence clustering was consistent for the different threshold settings.

RESULTS

Performance/Timing Using Neighbor / hit List

The neighbor / hit list captured in the initial clustering of the *blastclust* for the default run was used for the initial clustering of all the subsequent threshold runs. Using the neighbor / hit list, the initial clustering time was significantly reduced for the threshold runs from 13 hours to 45 minutes.

Data Integrity

The sequences in the clusters captured from the workflow use the *blastclust* with the threshold changes should be similar to sequences clustered in the workflow that used the *blastclust* default settings. To verify this assumption, the CiA program was developed to compare one cluster to another regardless of cluster size or threshold setting. The CiA program accepts as input the cluster file name and the cluster size of the two clusters to be compared. The program excludes all '9999' sequences in the cluster, determines what sequence ID's are unique to the two clusters, and compares which sequences overlap. As an example, the largest cluster from each threshold run of initial clustering was compared to the largest cluster of the default run of clustering after ancestral sequences. As seen in Table 4, the various runs capture similar data to the default ancestral run with the exception of Threshold Run 3, which had no intersection of sequences. For cases where multiple clusters had the same size, UNIX utilities such as *grep* and *diff*, were used to compare specific clusters across various threshold runs to each other. While this check

was not applied to every cluster generated, it does provide evidence that for the Threshold Run excluding Threshold Run 3 that sequences are clustering similarly.

Table 4 Largest Default Cluster (after Ancestral Sequences) vs. Largest Cluster (initial) of Threshold Runs

RUN #	Largest Cluster Size	# of intersecting ids with Run 0 (default)	# of id unique to Run 0 (default)	# of ids unique to threshold run largest cluster
0 (default)	232 *	-	-	-
1	240	232	0	8
2	354	0	232	354
3	260	232	0	28
4	270	232	0	38
5	270	232	0	38
6	272	232	0	40

Initial Clustering of Data

The initial clustering of data was captured for each run of the *blastclust* setting changes. Clusters with a size of six (6) sequences or more were used to constructing ancestral sequences. Table 5 shows the results of initial clustering for each threshold run.

Table 5 Initial Clustering of Families for All Workflow Runs

Run #	Similarity setting (-S)	Min. Length Coverage (-L)	Both/One coverage (-b)	Clusters Size ≥ 2	Clusters Size ≥ 6	Largest Cluster	Largest Cluster processed	Sequences Processed	Ave. # Sequences/Cluster
0	default	default	both	4458	164	195	195	1780	11
1	default	0.8	both	5183	206	240	240	2287	11
2	default	default	one	6331	387	354	354	4963	13
3	1.65	default	default	4543	211	260	260	2308	11
4	1.65	0.8	default	5258	261	270	270	2891	11
5	default	0.7	default	5650	230	270	270	2702	12
6	1.55	0.7	default	5753	348	272	272	4106	12

Clustering of Data after Ancestral Sequences are Added

It was observed that a significant number of clusters encountered errors during the ancestral sequence prediction step in the workflow regardless of threshold setting. As a result, for these clusters the ancestral sequence reconstruction was aborted. The ancestral sequences that were generated were added to the database and another all-vs.-all comparison was performed. The results of the clustering after the predicted ancestral sequences are in Table 6.

Table 6 Clustering with Ancestral Sequences of Families for All Workflow Runs

Run #	Similarity Setting (-S)	Min. Length Coverage (-L)	Both / One coverage (-b)	Clusters Size ≥ 2	Clusters Size ≥ 6	Largest Cluster	Ancestral Sequences Added	Sequences Processed	Additional Sequences Collected	Ave. # Sequences / Cluster
0	default	default	both	4454	162	403	466	2256	10	14
1	default	0.8	both	5180	205	402	459	2755	9	13
2	default	default	one	6326	384	354	512	5537	12	14
3	1.65	default	default	4535	209	437	584	2922	30	14
4	1.65	0.8	default	5248	259	447	626	3555	38	14
5	default	0.7	default	5647	229	446	497	3207	12	14
6	1.55	0.7	default	5743	346	453	844	4987	37	14

Comparison of Default Run Additional Sequences

The default run of the gene family extension iterative workflow gained ten (10) sequences using the predicted ancestral sequences. The cluster size location of these 10 sequences was tracked in the initial clustering for the threshold runs shown in Table 7. Threshold Runs 3, 4, and 6 capture all ten of the sequences in cluster sizes ≥ 6 in their initial clustering. All three of these threshold runs have a relaxed similarity setting in common. In Threshold Runs, 1, 2, and 5 the initial clustering had the sequences in small cluster sizes. After the reclustering with the ancestral sequences, Threshold Runs 1, 2, and 5 added the sequences with a similar clustering as the default processing results in slightly larger clusters as shown in Table 8. For Threshold Runs 3,4 and 6 the initial clusters for these sequences remained constant after the ancestral sequences were added to the database.

Table 7 Cluster of 10 Sequences Initial Clustering All Threshold Runs

Sequences Gained in Default run (cluster size after anc.)	Cluster Size Found in Threshold Run 1 Initial Cluster	Cluster Size Found in Threshold Run 2 Initial Cluster	Cluster Size Found in Threshold Run 3 Initial Cluster	Cluster Size Found in Threshold Run 4 Initial Cluster	Cluster Size Found in Threshold Run 5 Initial Cluster	Cluster Size Found in Threshold Run 6 Initial Cluster
47576151 (31)	1	1	20	20	1	23
47576153 (31)	1	1	20	20	1	23
47576157 (31)	1	3	20	20	3	23
47577501 (17)	2	2	17	17	2	21
47577509 (17)	2	2	17	17	2	21
47577549 (17)	3	3	17	17	3	21
47577553 (17)	3	3	17	17	3	21
47577567 (17)	3	3	17	17	3	21
47578043 (15)	1	1	10	10	1	15
51261002 (12)	1	18	8	9	1	9

Table 8 Clustering of 10 Sequences After Ancestral Clustering Threshold Runs 1,2 and 5

Sequences Gained in Default run (cluster size after anc.)	Cluster Size Found in Threshold Run 1 After Reclustering	Cluster Size Found in Threshold Run 2 After Reclustering	Cluster Size Found in Threshold Run 5 After Reclustering
47576151 (31)	31	37	37
47576153 (31)	31	37	37
47576157 (31)	31	37	37
47577501 (17)	17	17	17
47577509 (17)	17	17	17
47577549 (17)	17	17	17
47577553 (17)	17	17	17
47577567 (17)	17	17	17
47578043 (15)	15	15	15
51261002 (12)	1	18	1

Addition Sequences Captured

The workflow iterations with relaxed *blastclust* threshold criteria captured sequences overlooked in the workflow iteration using the default *blastclust* threshold criteria. There were also cases of overlap in the sequences captured between the alternative settings as shown in Table 9. In Threshold Runs 2 and 5, only two sequences are captured that the

default run did not capture with one sequence common to both threshold runs. This common sequence clustered into their respective cluster size 37; the same cluster that clustered three sequences identified by the default run. Threshold Run 1 did not provide any new sequences. In Threshold Runs 3, 4, and 6 all sequences added after reclustering were not captured in the default run 0. Threshold Runs 3 and 4 provided an additional 30 and 38 sequences respectively. Threshold Runs 3 and 4 had the same similarity setting of 1.65 with Threshold Run 4 and a coverage extension of 0.8. Twenty-seven (27) of these sequences were the same in both runs with Threshold Run 4 initially clustering the remaining sequences in Threshold Run 3 into a cluster size of twelve (12).

Table 9 New Sequences Captured Using relaxed *blastclust* Threshold Settings

Threshold Run #	# of Additional Sequences Other Default Sequences	# of Sequence Overlap with Other Threshold Runs	Threshold Run #
1	0	0	n/a
2	2	1	5
3	30	27	4
4	38	27	3
5	2	1	2
6	37	0	n/a

In Threshold Run 6, all thirty-seven (37) were unique to its run. There were many instances where groups of five sequences were into the same cluster, as illustrated in Table 10. Two of these cluster groups were assigned to clusters that were composites of two previous clusters. All sequences additional sequences added for Threshold Run 6 were singletons in the default run.

Table 10 Distribution of Threshold Run 6 Additional Sequences

Cluster Size	# of Additional Sequences Added	Composite Cluster ?	Family
37	5	Yes	Zinc Finger Protein
37	5	No	Vomeronasal organ pheromone receptor family
34	5	Yes	G-protein-coupled olfactory receptor
31	3	No	G-protein-coupled olfactory receptor
30	5	No	G-protein-coupled olfactory receptor
22	2	No	G-protein-coupled olfactory receptor
21	5	No	Immunoglobulin domain variable region
18	2	No	G-protein-coupled olfactory receptor
16	3	No	G-protein-coupled olfactory receptor

CONCLUSIONS AND FUTURE DIRECTIONS

Multiple runs using the predicted ancestral sequence methodology were executed first using the default settings of the *blastclust* program and later with various adjustments to the similarity threshold and coverage threshold options of this program. Relaxing the coverage threshold of *blastclust* alone produced results that were very similar the results using the default *blastclust* value with few new sequences acquired. Using less stringent similarity criteria in *blastclust*, alone or coupled with relaxed coverage criteria, the initial clustering captures sequences that the default *blastclust* run captured in its post-ancestral sequence grouping. Clustering after the ancestral sequences were predicted and added to the database, detected singletons were clustered in groups of size six or greater which were previously overlooked by the gene family extension iterative methodology using the default *blastclust* threshold values.

It has been stated that developing useful criteria settings for BLAST tools should be considered an experiment in itself [4]. The nuance of determining how relaxed the parameters should be may require several runs of the ancestral processing methodology. If so, using a neighbor / hit list will be very useful with decreasing the processing time of the initial clustering of the threshold runs.

The CiA program was very useful to compare clusters between the alternative *blastclust* approaches but is usable only if there is one cluster of that size for any given cluster file. Functionality improvements would reduce the analysis time.

In the analysis done thus far, there is no insight into potential rates of Type I and Type II errors present. Simulation could be used to determine this information and to perform other statistical tests on how accurate the predicted ancestral sequences extend gene families.

During the ancestral prediction step, several cluster sizes encountered error in PAML and as a result were not processed. The exclusion of any predicted ancestral sequences could affect the results of the second all-vs.-all clustering step in the iterative methodology. The cause of these errors needs to be investigated and resolved.

REFERENCES

- [1] Altschul S.F., Gish W., Miller W., Myers E.W., Lipman D.J., *Basic Local Alignment Search Tool*, **Journal of Molecular Biology**, 1990,215:403-410.
- [2] Altschul S.F., Madden T.L., Schaffer A.A., Zhang J., Zhang Z., Miller W., Lipman D.J., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*, **Nucleic Acids Research**, 1997, 25(3):3389-3402.
- [3] Anantharaman V., Koonon E., Aravind L., *Comparative genomics and evolution of proteins involved in RNA metabolism*. **Nucleic Acids Research**, 2002, 30(7):1427-1464
- [4] Durbin, R., Eddy S., Krogh A., Mitchison G., *Biological sequence analysis*, Cambridge University Press, 2004.
- [5] Edwards, R.V., Sheilds D.C., *BADASP: predicting functional specificity in protein families using ancestral sequences*. **Bioinformatics**, 2005 , 21(22):4190-4191.
- [6] Goldberg L., Rosenberg M., *Extending Gene Families Via Predicted Ancestral Sequences*, Internship Report, April 28 2006
- [7] Henikoff S., Greene E.A., Pietrokovski S., Bork P., Attwood T.K., Hood L., *Gene Families: The Taxonomy of Protein Paralogs and Chimeras*,**Science**, 1997;278, 609-614.
- [8] Korf, I., Yandell M., Bedell J., *BLAST*; O'Reilly & Associates, CA 2003
- [9] Masseroli, M., Bellistri E., Franceschini A., Pinciroli F., *Statistical Analysis of genomic protein family and domain controlled annotations for functional investigation of classified gene lists*, **BMC Bioinformatics**, 2007, 8(Supp 1):S14.
- [10] Jones, N.C. and Pevzner, P.A., *An Introduction to Bioinformatics Algorithms*, A Bradford book, Massachusetts 2004.
- [11] Liepman, A.H. et al. *Functional Genomic Analysis Supports Conservation of Function among Cellulose Synthase-Like A Gene Family Members and Suggests Diverse Roles of Mannans in Plants*, **Plant Physiology**, 2007,143:1881-1893.
- [12] Tanaka N., Abe T., Miyazaki S., Suawara H., *G-InforBio: integrated system for microbial genomics*, **BMC Bioinformatics**, 2006, 7:368
- [13] Tatusov R L, Koonin E V, Lipman D J, *A Genomic Perspective on Protein Families*, **Science**, 1997, 278, 631 -637
- [14] Thompson J.D., Higgins D.G., Gibson T.J., *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-*

specific gap penalties and weight matrix choice, **Nucleic Acids Research**, 1994, 22(22):4673-4680.

[15] Yang Z., *PAML: a program package for phylogenetic analysis by maximum likelihood*, **CABIOS**, 1997, 13(5):555-556

[16] <http://genomes.ucsd.edu/gaasterlandlab/manuals/blast/blastclust.html>

[17] <http://www.ncbi.nlm.nih.gov/Web/Newsltr/Spring04/blastlab.html>

[18] <http://ghr.nlm.nih.gov/handbook/howgeneswork/genefamilies;jsessionid=728605737242A6C8FA97CD4FD5450BD0>

APPENDIX A:

Table 11 Blastclust Parameters and Default Values

Parameter	Value
Score Density Similarity Threshold	1.75
Min. Length Coverage Threshold	0.9
Required coverage on both sequences	TRUE
e-value threshold	1.00E-06
<i>Protein</i>	
Matrix	BLOSUM62
Gap Opening Cost	11
Gap Extension Cost	1
Low-complexity Filtering	F
<i>Nucleotide</i>	
Match Reward	1
Mismatch Penalty	-3
Wordsize	28