ABSTRACT

Automated Sequence Homology: Using Empirical Correlations to Create Graph-based Networks for the Elucidation of Protein Relationships

Stephen J. Bush

Chairperson: Erich J. Baker, Ph.D.

Identification of sequence homology has presented a formidable obstacle despite significant increases in both technological capability and detailed knowledge of genomes and proteomes. While PSI-BLAST remains the popular tool for the job, it often returns inaccurate results with unacceptable levels of false positives. In order to increase the sensitivity and accuracy of homology finding, we have developed a software application called Automated Sequence Homology that bypasses these shortcomings and provides reliable and precise results. The system presented here is based upon the creation of a graph-based network highlighting the relational connections between proteins using empirical correlations. It takes a step back from PSI-BLAST to the acclaimed BLAST algorithm to create a sampling of the protein relational network.

Automated Sequence Homology: Using Empirical Correlations to Create Graph-based Networks for the Elucidation of Protein Relationships

by

Stephen J. Bush, B.S.

A Thesis

Approved by the Department of Biomedical Studies

Robert R. Kane, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of
Master of Science

Approved by the Thesis Committee

Erich J. Baker, Ph.D., Chairperson

Christopher M. Kearney, Ph.D.

Myeongwoo Lee, Ph.D.

Accepted by the Graduate School
August 2008

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This thesis is the result of many hours of hard work and patience from a number of people, foremost Dr. Erich Baker. His guidance and knowledge were indispensible in the completion of this project. Thank you, Dr. Baker for putting up with all my eccentricities; You are a great mentor and a wonderful friend. Thank you to Dr. Sarah-Jane Murray for funding me throughout my master's program. Your support and friendship mean the world to me. I would also like to acknowledge my department directors who have worked with me throughout this process and helped ensure the success of this project.

To my wife Jen

and my daughter Elizabeth, whom I cannot wait to meet

CHAPTER ONE

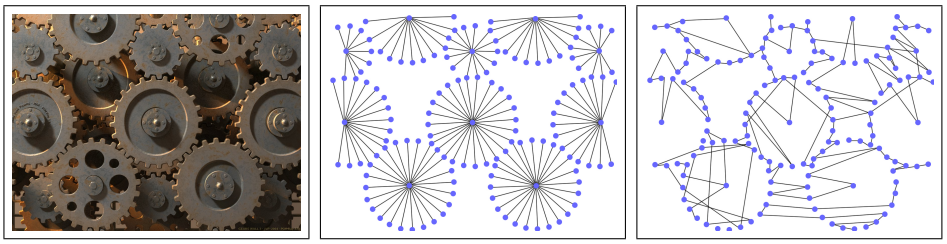
Introduction

Determining sequence homology is a difficult but remarkably common task for technically savvy biologists. Although a variety of existing programs address this issue, they often suffer from one or several of a number of drawbacks including lack of sensitivity (Shah et al., 2008; Xiaohua, 2007b; Zhang et al., 2005), false positives (Oul and Mumcuolu, 2007; Shah et al., 2008), intensive and lengthy computational requirements (Oul and Mumcuolu, 2007; Shah et al., 2008; Xiaohua, 2007b), and required precursory program training or learning (Oul and Mumcuolu, 2007; Shah et al., 2008; Zhang et al., 2005). Researchers regard NCBI's BLAST well for its ability to identify homologs (Oul and Mumcuolu, 2007; Shah et al., 2008); however, its capability to handle only one input sequence severely limits its range and sensitivity (Nikolski and Sherman, 2007; Jun Wu, 2006). Both Hidden Markov Models, such as SVM, and position specific scoring models, such as PSI-BLAST, attempt to bypass this weakness with the use of sequence profiles, yet even these are subject to false positives (Shah et al., 2008; Jun Wu, 2006; Zhang et al., 2005). Despite these shortcomings, many still continue to use PSI-BLAST as a basis for homology research (Schock et al., 2000; Jun Wu, 2006; Zhang et al., 2005).

The original BLAST programs (Altschul et al., 1990) are designed to perform sequence similarity queries on one sequence at a time and is thus severely limited for use in applications requiring concurrent, iterative, or batch operations (Frickey and Lupas, 2004; Nikolski and Sherman, 2007); however, its widespread popularity and overall success has lead to continual use and improvement upon the underlying technique resulting in a variety of new programs, including PSI-BLAST. The strategy behind PSI-BLAST involves an initial BLAST-type query where the re-

sults are used to create the Position-Specific Scoring Matrix (PSSM). PSI-BLAST then uses the matrix to seed further iterations and continually changes and updates the matrix after each iteration to reflect the increased accuracy of the result set. By creating the PSSM, PSI-BLAST has demonstrated reasonable success in overcoming the single sequence issue (Shah et al., 2008). However, low sequence similarity present in remote homologs and weak domain identification problems for multiple domain proteins may severely compromise the integrity of the results for homolog identification (Shah et al., 2008; Jun Wu, 2006). Another confounding issue with PSI-BLAST is the lack of a mature tool for visual analysis of results. Though the recent addition of phylograms to the PSI-BLAST tool set (Figure 3.2) and third-party software such as the CLANS program (Biegert et al., 2006; Frickey and Lupas, 2004; Frickey and Weiller, 2007) help alleviate the need, they are still based upon the PSI-BLAST output. While helpful, these visualizations show node-edge connections in relation to the seed and the PSSM, which may not correctly identify the proper protein domains depending on the complexity and function of the protein. However, using multiple protein sequences to seed iterative BLAST queries gives a relational perspective for each protein within the set and eliminates the distortion seen by the PSSM of PSI-BLAST.

An additional advantage to viewing sequence relationships in a graph-based format is the ability to analyze the empirically created relationships using graph theory approaches. Many graphs represent scale-free networks and are found in a variety of naturally occurring situations including the World Wide Web (Barabasi, 2007; Dinc, 2007; Li, 2007; Xiaohua, 2007a; Xin Biao, 2006), social networks (Barabasi, 2007; Dinc, 2007; Li, 2007), and even biological protein-protein interactions (Barabasi, 2007; Dinc, 2007; Li, 2007). ASH applies these properties to homology identification. Described previously (Barabasi, 2007), scale-free networks exhibit properties of a small world model and employ a high degree of organization through the use of



(a) Clock Gears (b) Scale-free Overlay (c) Random Overlay

Figure 1.1: A Simple Example of Scale-free Networks. a) An example of gear connections. b) A simplified rendition of a scale-free network based upon the gears shown in box 'a'. The network contains highly connected hubs which grant a high degree of organization. These hubs grow and shrink in a power law distribution. c) A simplified possible random network using the nodes shown from the gears. There is no organization and the distance required to traverse the graph from any one node to another varies greatly between nodes. In addition, the ratio of the number of connections from any one node to the network average remains fairly constant with little to no deviation throughout the graph.

highly connected nodes called hubs (Barabasi, 2007; Li, 2007). These hubs account for a very small portion of the network population and are often separated from each other by only one or two degrees (Barabasi, 2007; Li, 2007); however, they contain the vast majority of edges, calculable through a power law distribution (Barabasi, 2007; Dinc, 2007; Hongbo, 2006; Li, 2007), and also exhibit preferential attachment for the addition of new nodes (Barabasi, 2007; Dinc, 2007; Hongbo, 2006). For a simplified image of these complex networks, imagine the gears on the inside of a clock where nodes exist in the center and on every cog of the gears (Figure 1.1). These complex networks are highly stable and incredibly resilient to random attacks. Due to the small number of hubs versus total nodes, the probability of a random node failure occurring at a hub is increasingly small; however, a direct attack on a hub can cripple the integrity of the entire network (Barabasi, 2007). In addition, the distance required to traverse a scale-free network versus a random network is incredibly low (Young-Rae, 2006). These principles cause the pervasiveness threshold for scale-free networks to fall to zero and explain the susceptibility of populations to viral or bacterial infection while also providing new strategies for vaccine treatment (Barabasi, 2007).

This paper proposes a unique method for protein homology estimation called Automated Sequence Homology (ASH). ASH requires no *apriori* knowledge of the proteins other than the peptide sequence and does not require expansive computational resources. In addition, ASH stands a greater chance of identifying remote homology than competing solutions by creating a sample population of peptides rather than a sequence profile. This is accomplished through a bottom-top-bottom methodology using a number of iterative BLAST queries and empirical correlations to create graph-based networks, thinning the network with a top-down elimination of unwanted queries, and extracting a series of sub-networks and cliques focused around any singular protein found within the dataset. ASH may help identify an unknown protein sequence, validate a putative or hypothetical sequence, or identify previously unknown homologs. ASH takes advantage of the single sequence disadvantages of the blastp algorithm to generate a graph-based relational network centered around a single seed protein. This paper compares this approach against traditional PSI-BLAST using the calmodulin protein.

CHAPTER TWO

Methods & Materials

2.1 Computing Resources and Data

All experiments were performed on Dell PowerEdge 1550 systems with dual Pentium 4 processors with 1 GB memory running RedHat Enterprise Linux 3.0 Workstation operating systems. The primary processing machine designated huxley connected to the Baylor ECS backbone by 10/100 Mbps Ethernet. NCBI's Stand-Alone Blast (SAB) v.2.2.14 and the *Arabidopsis thaliana* RefSeq protein database containing approximately 31711 sequences was downloaded from NCBI's FTP site onto the local system. ASH accepts user input parameters for BLAST queries, and tests were run with an e-value of 1e-4, a minimum of 250 iterations, and included hypothetical proteins while discarding experimental proteins. By default, ASH collects the bit score for comparison purposes.

For data correlation, ASH creates the query and calls the statistical analysis program R (Ihaka and Gentleman, 1996; Peng, 2003; Ritz and Streibig, 2005). All correlations are done using the Pearson method to provide unweighted correlations. ASH uses the program Cliquer (stergrd, 2002) to identify cliques of various sizes: it outputs the proper data in Dimacs format and feeds it to Cliquer. Graphviz (Ellson et al., 2002; Gansner and North, 2000) provides a simple graph representation of the data. The nodes and edges are listed in the Graphviz format and fed into Graphviz Neato. Graphviz requires a large amount of memory and processing capability to run on increasingly large graphs, and will take an amount of time proportional to the number of nodes and edges in the graph. For this reason and although useful it's not necessary, Graphviz does not run by default.

2.2 The Seed: Calmodulin 5, NP_850097

The Calmodulin (CaM) family was choosen for analysis due to the homology identification issues in PSI-BLAST which arise from the strong domain identification of calmodulin-dependant proteins in the dataset. From the available CaM proteins, CaM5 NP_850097 was randomly selected as the seed. CaM proteins comprise one type of the EF-hand superfamily of calcium ion binding protein (Finkler et al., 2007; Yang et al., 2004) and function in a variety of roles, including RNA and DNA binding inhibition (Delaney et al., 2006; Finkler et al., 2007), activation of the innate immune response (Ali et al., 2007; Chiasson et al., 2005), and signal pathway modulation in response to Ca^{2+} (Ali et al., 2007; Braam et al., 1997; Galon et al., 2008; Yang et al., 2004). They are also required for activation of a number of serene/threonine-specific kinases – a few of these kinases have evolved to contain their own CaM domain for this very reason. Calmodulin represents an excellent example of the differences between ASH and PSI-BLAST approaches because these kinase-CaM fusion proteins provide a strong kinase domain, which overweights the result set and limits the ability of PSI-BLAST to correctly identify the entire CaM family.

2.3 The ASH Package

ASH is a package of two drivers and six classes to run BLAST, collect and analyze the data, and output it in a user friendly format. See Figure 2.1 for a simple illustration of program flow.

2.3.1 The Collector

The ASH Collector is the central organizing metaphor of the ASH package. Given an initial protein sequence, the Collector queries the SAB database and parses through the BLAST results. The data is normalized using a special log odds equation and correlated using a Pearson method. This initial step provides an unfiltered and complete graph-based network which the Extractor uses to filter out homology data.

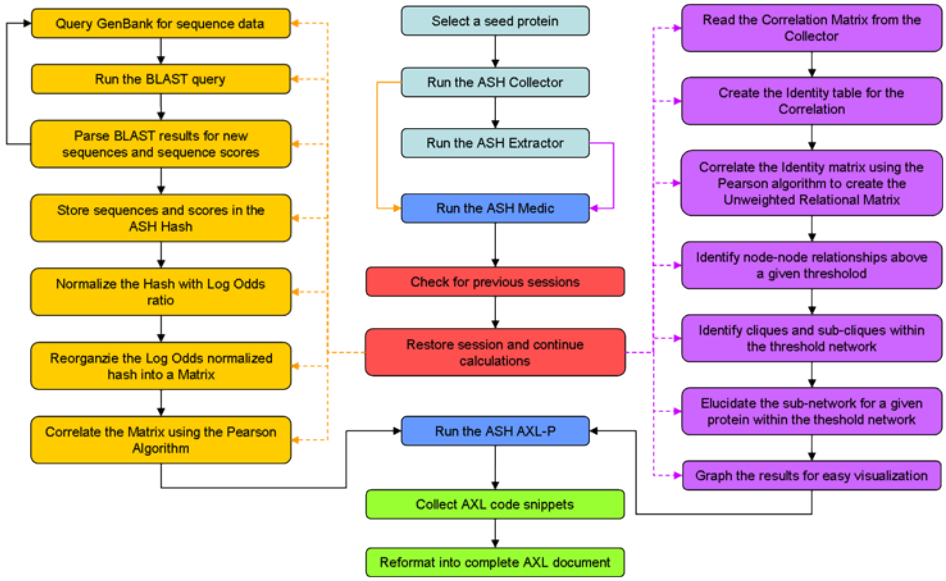


Figure 2.1. Diagram of program flow.

The Collector retrieves GenBank sequence data from NCBI's Entrez server. It first sends a single request for the seed sequence followed by batch requests for sequences identified in BLAST results. ASH is optimized for batch querying to allow for maximum sequence retrieval to avoid a possible uncatchable floating point exception from Entrez. Several factors play a role, including current Entrez load capacity, amount of time passed between queries, and the number of sequences requested in both the previous and current queries. Estimates show about a forty percent crash rate for batches over five hundred (data not shown), especially if the request is timed close to the previous one. To help provide distance between each request, new blastp queries run on the SAB between each Entrez request. By segmenting our Entrez queries into batches, ASH not only avoids a fatal error from NCBI, but it also greatly increases the Collector's efficiency and speed.

When the Collector parses the BLAST results, it collects the bit scores by default. While the e-value represents the probability a given sequence will be returned in a given set, the bit score provides the raw sequence similarity score normalized for the scoring matrix, and thus provides a better score comparison between different

queries. Unfortunately, the bit scores still contain an element of sequence length and are therefore not optimally normalized between queries. In order to correlate the sequences based upon the relational data rather than the sequence similarity, another normalization is applied across the data set in the form of a log odds ratio of a ratio of the query score for a given seqeunce α within query β over the average of β over a ratio of the average score for every instance of α within the dataset ζ over the average of ζ.

The first two steps store the data in a hash data structure to decrease time, space, and computational costs, and the Collector must reformat the hash into a matrix structure for correlation. This presents a small problem since sequences are not returned in their own BLAST query and not all sequences are returned in every BLAST result set, thus leaving empty holes in the matrix; the Collector solves this problem by doubling the maximum score in the entire result set for identity matches and quintupling the minimum score for non-existing, non-identity matches. For the last step of the Collector, ASH applies a Pearson correlation to provide an un-weighted analysis, versus a Spearmann correlation where appearance in the list denotes added value, i.e. rows 1 and 2 innately have a stronger correlation than rows 1 and 3. This correlation matrix is a table representation of graph-based network and contains node-edge connections between all of the proteins included in the BLAST sampling; despite these previous normalizations the relationships are weighted largely upon sequence similarity rather than the desired relational similarity and only loosely denote homology. To remedy this, the Correlation Matrix is run through the ASH Extractor.

2.3.2 The Extractor

The Extractor is the analysis component of the ASH program. Using the data generated by the Collector, the Extractor identifies the homology data through ap-

plication of a qualitative layer and subsequent correlation to remove the sequence bias and data mining of the Unweighted-Correlation Matrix to isolate homologies. Utilization of the ASH XML Parser (AXL-P) and Graphviz allow for easy visualization of the results.

The Extractor begins by taking the Collector's correlation matrix and stripping the sequence similarity weights by creating an Identity Table. This table is a 0-1 representation of the Correlation Matrix for edges above a given threshold. Creation of this Identity Matrix helps eliminate some of the bias inherent in the BLAST sequence comparison and allows us to analyze the data based upon a relational or qualitative standpoint: which proteins are related to the same proteins, rather than a similarity based or quantitative one, or the strength of a relationship between two proteins.

The Identity Table alone cannot tell us all of this information, and the Extractor runs another Pearson correlation on the Identity Table. This Unweighted-Correlation table highlights the connections identified by the Identity Table. This step in combination with the Log Odds score from the Collector give the Extractor its edge over PSI-BLAST because of the two layers of abstraction from the sequence similarity. Once the Extractor creates the Unweighted-Correlation matrix, it extracts all node-edge relationships with a score above a given threshold and identifies maximal and sub-maximal clique graphs, as well as sub-network graphs. These graphs are viewable both text-file and graphical formats.

2.3.3 The Medic & AXL-P

The ASH Medic exists as a series of functions integrated into the Collector and Extractor; however, to decrease computational time and prevent user frustration, it is under construction as its own entity. The Medic aids in communication between the the Collector and Extractor and allows for session recovery and completion at

any point past the initial hash creation. This gives stability through a program or hardware error, as well as eliminating redundant processing.

Also in progress for recreation as a separate entity is the ASH XML (AXL, pronounced 'axle') Parser (AXL-P). AXL-P creates bits of AXL throughout program execution and melds them into a final AXL file for easy data retrieval and manipulation without having to wait for graph file creation or sift through tables for the data (see A.3 for an AXL example). AXL-P provides graph-based analysis without needing to create the visual graph.

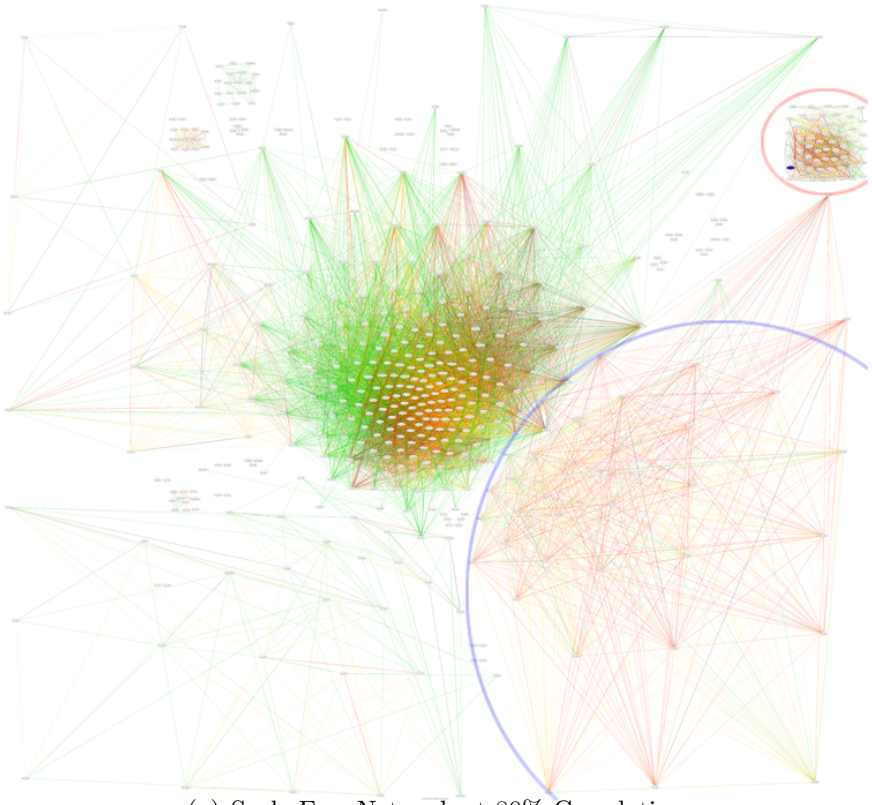
CHAPTER THREE

Results & Discussion

The Calmodulin protein NP_850097 from *Arabidopsis thaliana* was chosen for test runs of ASH as it presents an interesting problem. Calmodulin is a calcium-ion binding protein which is required for activation of many serene/threonine-specific kinase; because of this, a few kinases contain their own calmodulin domain and are returned as matches in a BLAST query. In fact, the graph-based network generated by the ASH Collector consists mainly of kinases with only a very small number of calmodulin proteins (Figure 3.1). The ASH project aims to identify just the CaM sub-network without any of the kinases seen in the PSI-BLAST results.

3.1 ASH Results

At 95% correlation (Table 3.1) ASH identified calmodulin 1, 2, 3, 4, 6, 7, and 8, a putative calmodulin, and calmodulin related proteins ATCEN 2 (centrin) and ATCAL 4 ((Table 3.2) shows the cliques and sub-networks generated). At 90%, ASH included a number of calcium-ion binding proteins, a few putative calmodulin, and a few others. At 85% and 80%, calmodulin 9 and more putative calmodulin were added along with TCH2 and 3. Since CaM9 shows a lower sequence similarity in the MSA (Table 3.3), its lower correlation shows the emphasis still placed on similarity in the ASH algorithm. Interestingly, the inclusion of so many putative CaM shows the potential of ASH to identify new and unknown proteins, and the large number of CaM-like proteins shows a great potential for domain identification.



(a) Scale-Free Network at 80% Correlation

(b) Calmodulin Subsection

(c) Sub-Network at 80% Correlation

Figure 3.1: a) The network at an 80% threshold. The desired Calmodulin family is in the top right corner circled in red. The blue circle denotes the kinase group PSI-BLAST pulls sequences from. b) The Calmodulin sub-network isolated and enlarged from the network on the left. c) The sub-network extracted from the Unweighted matrix for the seed protein. This sub-network is also a clique for the data set.

Table 3.1: List of proteins identified by threshold in ASH.

		ASH			PSI-BLAST			
Threshold		ACC	Putative	Name	4	3	2	1
	1	NP_176814		CAM4	1	1	1	1
	2	NP_180271		CAM2	1	1	1	1
	3	NP_188933	1	Calmodulin	1	1	1	1
	4	NP_189967		CAM7	1	1	1	1
	5	NP_190605		ATCEN2 (CENTRIN2)	1	1	1	1
0.95	6	NP_850860		CAM6	1	1	1	1
	7	NP_850096		CAM2	1	1	1	1
	8	NP_193200		CAM8	1	1	1	1
	9	NP_191239		CAM3				
	10	NP_198594		CAM1				
	11	NP_850344		ATCAL4; Calcium-ion Binding Protein				
	12	NP_001031798		Calcium-ion Binding Protein	1	1	1	1
	13	NP_173866	1	Polcalcin / Calcium Binding Protein	1	1	1	1
	14	NP_174504	1	Calmodulin	1	1	1	1
	15	NP_179170	1	Calmodulin-related	1	1	1	1
	16	NP_186950	1	Calmodulin	1	1	1	1
	17	NP_187405		AGD11, Calcium-ion Binding Protein	1	1	1	1
0.90	18	NP_191503	1	Calcium Binding Protein	1	1	1	1
	19	NP_193022		UNE14; Calcium-ion Binding Protein	1	1	1	1
	20	NP_195418	1	Caltractin / Centrin	1	1	1	1
	21	NP_173288	1	Calmodulin	1	1	1	
	22	NP_172089	1	Calcium-ion Binding Protein		1	1	1
	23	NP_192238	1	Calcium Binding Protein		1	1	1
	24	NP_565996		MSS3; Calcium-ion Binding Protein		1	1	1
	25	NP_001031523		TCH3	1	1	1	1
	26	NP_181642		CL Calcium Binding Protein	1	1	1	1
	27	NP_181643		TCH3	1	1	1	1
	28	NP_190760		CAM9	1	1	1	1
	29	NP_199053	1	Calmodulin-related	1	1	1	1
	30	NP_850343		TCH3	1	1	1	1
	31	NP_198593		TCH2		1	1	1
0.85	32	NP_173259	1	Calcium Binding Protein		1	1	1
	33	NP_189188	1	Calmodulin		1		
	34	NP_564874	1	Calmodulin-related			1	1
	35	NP_177504	1	Calcium Binding Protein			1	
	36	NP_197249	1	Calmodulin-related				
	37	NP_187630	1	Calmodulin				
	38	NP_849686	1	Calcium-ion Binding Protein				
	39	NP_177790	1	Calmodulin-related				
	40	NP_172695	1	Calmodulin	1	1	1	1
	41	NP_176470	1	Calmodulin	1	1	1	1
0.80								

Table 3.1 – Continued

Threshold		ACC	Putative	ASH Name	PSI-BLAST 4	3	2	1
	42	NP_199259	1	Calcium Binding Protein				1
	43	NP_177791		Calcium Binding Protein EF-hand				
	44	NP_181160	1	Calmodulin-related				
	45	NP_193810	1	Calcium Binding Protein				
0.75	46	NP_194377		Calcium-ion Binding Protein				
	47	NP_174807		ATCDPK2 (kinase)	1	1	1	1
	48	NP_177612		CPK30 (kinase)	1	1	1	1
	49	NP_190753		CPK13 (kinase)	1	1	1	1
	50	NP_192695		CPK4 (kinase)	1	1	1	1
	51	NP_564066		ATCDPK1 (kinase)	1		1	1
	52	NP_191312		CPK32 (kinase)	1		1	
	53	NP_190646	1	Calmodulin-related	1		1	
	54	NP_187677		CPK2 (kinase)	1			1
	55	NP_195257		CPK5 (kinase)	1			1
	56	NP_196107		CPK1 (kinase)	1			1
	57	NP_565411		CPK6 (kinase)	1			
	58	NP_195536		CPK26 (kinase)		1	1	1
	59	NP_192380		CPK22 (kinase)		1	1	
	60	NP_193925		CPK15 (kinase)		1		
	61	NP_001078576		CPK7 (kinase)			1	1
	62	NP_001077834		Calcium Binding Protein EF-hand		1		1
	63	NP_186993	1	Polcalcin / Calcium Binding Protein			1	1
	64	NP_197748		CDPK9 (kinase)			1	1
	65	NP_568281		CPK7 (kinase)			1	1
	66	NP_188374		RelA / SpoT domain				1
	67	NP_188676		CPK9 (kinase)				1
	68	NP_190332		Calcium Binding Protein EF-hand				1
	69	NP_175485		CPK33 (kinase)				1
	70	NP_181425		CPK20 (kinase)				1
	71	NP_192381		CPK21 (kinase)				1
	72	NP_192383		CPK23 (kinase)				1
	73	NP_680596		CPK31 (kinase)				1
	74	NP_181672	1	Calmodulin				
	75	NP_186990	1	Calmodulin-related				
	76	NP_186991	1	Calmodulin-related				
	77	NP_565064		Calmodulin-related				
	78	NP_566152	1	Calmodulin-related				
	79	NP_974207	1	Calmodulin-related				

< 0.70

Negatives

Table 3.2: Shown are the sub cliques and sub-networks from NP_850097 for each threshold from 0.95-0.70 in 0.5 increments. The sub-network nodes are colored by edge count. Dark blue is 100% of possible edges (i.e. the same number as the seed), and a lighter blue for every increment of 25%.

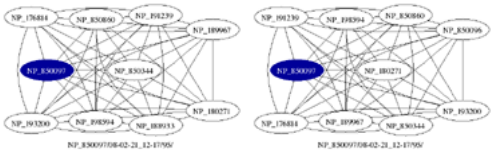
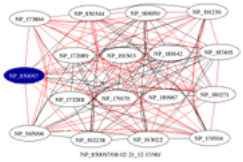
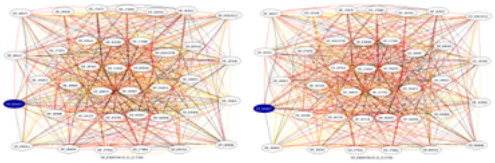
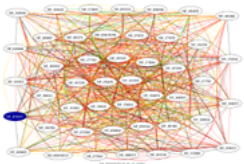
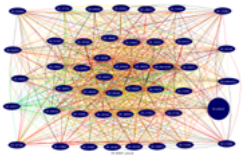
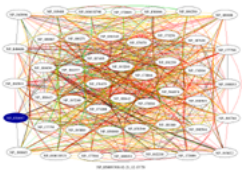
	Cliques	Sub-networks
0.95		
0.90		
0.85		
0.80		
0.75		
0.70		

Table 3.3: MSA of the known CaM proteins numbered from Table 3.1. The seed is on top, and all CaM9, at the bottom, correlate at greater than 95%. Sequence 11 is the former ATCAL4.

No.	Alignment	
(0)	-MADQLTDDQISEFKEAFSLFDKDGDGCITTKELGTVMRSLGQNPTEAELQDMINEVDADGNGTIDFPEFLNLMARKMKDTDSEEELKEAF	[91]
(1)	-.......E.....K.....	[91]
(2)	-.....	[91]
(4)	-.....	[91]
(6)	-.....	[91]
(7)	----------.....	[91]
(8)	MEETA..K...T.....C..........VE..A..I...D.....Q..H.I.T.I.S.S....E.A.......K.LQES.A.....	[91]
(9)	-.....	[91]
(10)	-.......E.....K.....	[91]
(11)	-.....	[91]
(28)	-...AF..E..Q..Y...C.I...S..F..KEK.TK..K.M.K..KAEQ..Q.MSD..IF...G.T.DD..YI..QNTSQESASD..I.V.	[91]
(0)	RVFDKDQNGFISAAELRHVMTNLGEKLTDEEVDEMIKEADVDGDGQINYEEFVKVMMAKRRGKRVMAAKRSSNSAEYKEKNGRRKSHCRIL	[182]
(1)	.....E...R.....I....----------	[182]
(2)	.....----------	[182]
(4)	.....R.....----------	[182]
(6)	.....S.....R.....----------	[182]
(7)	.....----------	[182]
(8)	K.....Y...S..S...I..........EQ.....L.....V..D....M.INID----------	[182]
(9)	.....----------	[182]
(10)	.....E...R.....I....----------	[182]
(11)	.....----------	[182]
(28)	....R.GD.L..QL..GEG.KDM.M.I.A..AEH.VR...L....FLSFH..S.M.I.ASY----------	[182]

While ASH correctly identified the CaM family members and even corroborated a few putative CaM, it included a few known non-CaM proteins. These non-CaMs are not completely false positives, rather they are members of the EF-hand super family and display a strong domain similarity with CaM, perhaps denoting an evolutionary connection. ATCAL4 and ATCEN2 both showed a 95% correlation and are not members of the CaM family. Since this data was collected, changes were made to NP_840344 on GenBank to properly identify the protein as CaM2 rather than ATCAL4. Even though ATCEN2 is not a CaM, it shows a very close sequence similarity to the known CaM sequences (Table 3.4), binds calcium, and is a member of the EF-hand superfamily along with CaM, which all account for its inclusion within the dataset. All of the other definitively non-CaM proteins ASH included contain EF-hand domains, thus further elucidating ASH as a successful tool for domain identification. Interestingly, TCHs are light and touch sensitive CaMs (Braam et al., 1997; Lee et al., 2005); in fact, TCH1 was a known CaM protein and was relabelled as CAM1 while TCH2 and 3 are known CaM-related (Braam et al., 1997; Lee et al., 2005; Sistrunk et al., 1994). NP_181643, NP_850343, NP_001031523, and NP_198593 (TCH3 and 2) are notated in GenBank as identical matches to CaM-related proteins 2 and 3.

Since even the false positives are members of the EF-hand superfamily, it is interesting to note the rejection of one putative CaM and four putative CaM-related proteins. The entire list of proteins with a correlation of 75% or greater show a strong possibility of homology, despite several non-Cam and Cam-like or related proteins (Table 3.5). This suggests these proteins are not CaM or CaM-related as predicted or are at least distant homologies. The rejection of these sequences from among a strong collection of significant sequences further illustrates ASH's ability to correctly identify homology and domain detection for unidentified proteins.

Table 3.4. MSA of the non-CaM proteins with the seed numbered from Table 3.1.

No.	Alignment	
(0)	----------MADQLTDDQISEFKEAFSLF	[109]
(5)	----------MSSIYRTVSRKEKPRRHHG..TQKKQ.I....E..	[109]
(11)	----------.....	[109]
(13)	----------MFNKNQGSNGGSSSNVGIGADSPYLQKARSGKTE.R.LEAV.KK.	[109]
(17)	----------MDQA.LARI.QM.	[109]
(19)	----------MDRG.LSRV.QM.	[109]
(20)	----------MSEAAQLRRGLKPKGKTYG..NQKRR.IR.I.D..	[109]
(24)	----------MVRIFLLYNILNSFLLSLVPKKLRTLFPLSWFDKTLHKNSPPSPSTMLPSPSSSSAPTKRIDP..L.RV.QM.	[109]
(25)	MADKLTDDQITEYRESFRLFDKNGDGSITKKELGTMMRSIGEKPTKADLQDLMNEADLDGDGTIDFPEFLCVMAKNQGHDQAPRHTKKT...K.....T.YR.S.R..	[109]
(27)	----------MMRSIGEKPTKADLQDLMNEADLDGDGTIDFPEFLCVMAKNQGHDQAPRHTKKT...K.....T.YR.S.R..	[109]
(30)	----------	[109]
(31)	----------MSSKNGVVRSCLGSMDDI.KV.QR.	[109]
(0)	DKDGDGCITTKELGTVMRSLGQNPTEAELQDMINEVDADGNGTIDFPEFLNLMAR----------KMKDTDSEEELKEAFRVFDKDQNGFISAAELRHVMTN	[218]
(5)	.T..S.T.DA...NVA..A..FEM..EQINK..AD..K..S.A...D..VHM.TAKIGERD----------TK...TK..QII.L.K..K..PDDIKRMAKD	[218]
(11)	..........----------..........	[218]
(13)	.VN...K.SS....AI.T...HEVP.E..EKA.T.I.RK.D.Y.N.E..VE.NTKGMDQND----------VL.N..D..S.Y.I.G..S...E..HE.LRS	[218]
(17)	.RN...K..KQ..NDSLEN..IYIPDKD.VQ..EKI.LN.D.YV.IE..GG.YQTIMEERD----------E..DMR...N...QNRD...TVE...S.LAS	[218]
(19)	..N...K.AKN..KDFFK.V.IMVP.N.INE..AKM.VN.D.AM.ID..GS.YQEMVEEKE----------E..DMR.......QNGD...TDE...S.LAS	[218]
(20)	.I..S.S.DAS..NVA.....FEMNNQQINELMA...KNQS.A...D..VHM.TTKFGERD----------.ID..SK..KII.H.N..K..PRDIKMIAKE	[218]
(24)	..N...R..KE..NDSLEN..IYIPDKD.TQ..HKI..N.D.CV.ID..ES.YSSIVDEHHNDG----------ETE..DM.D..N...Q.GD...TVE..KS..AS	[218]
(25)	..N...S..K...R...F...K.R.K.D....M....L..D.........Y...KNQGHDQAPRHTKKTMVDYQLTD.QIL.FR........NGD.Y.TVN...TT.RS	[218]
(27)	..N...S..K...R...F...K.R.K.D....M....L..D.........Y...KNQGHDQAPRHTKKTMVDYQLTD.QIL.FR........NGD.Y.TVN...TT.RS	[218]
(30)	----------.F...K.R.K.D....M....L..D.........Y...KNQGHDQAPRHTKKTMVDYQLTD.QIL.FR........NGD.Y.TVN...TT.RS	[218]
(31)	..N...K.SVD..KE.I.A.SPTASPE.TVT.MKQF.L....F..LD..VA.FQIGIGGGGNN----------RNDVSD.....ELY.L.G..R...K..HS..K.	[218]
(0)	LGEKLT--DEEVDEMIKEADVDGDGQINYEEFVKVMMAKRRGKRVMAAKRSSNSAEYKEKNGRRKSHCRIL----------	[326]
(5)	...NF.--.A.IR..VE...R.R..EV.MD..MRM.RRTAY.GN----------	[326]
(11)	......--..........----------	[326]
(13)	..DECS--IA.CRK..GGV.K....T.DF...KIM.TMGS.RDN..GGGPR----------	[326]
(17)	..L.QGRTL.DCKR..SKV......MV.FK..KQM.KGGGFAALGSNL----------	[326]
(19)	M.L.QGRTL.DCKK..SKV......MV.FK..KQM.RGGGFAALSSN----------	[326]
(20)	...NF.--.NDIE...E...R.K..EV.L...M.M.KRTSY.----------	[326]
(24)	...TQ.--KA.LQD..N...A....T.SFS...C..TG.MIDTQSKKETYRVVNQGQGQVQRHTRNDRAGGTNWERDIAVGVASNIIASPISDFMKDRFKDLFEALLS	[326]
(25)	..L.QGKTLDGCKK..MQV.A....RV..K..LQM.KGGGFSSSN----------	[326]
(27)	...TQ.--KA.LQD..N...A....T.SFS...C..TG.MIDTQSKKETYRVVNQGQGQVQRHTRNDRAGGTNWERDIAVGVASNIIASPISDFMKDRFKDLFEALLS	[326]
(30)	...TQ.--KA.LQD..N...A....T.SFS...C..TG.MIDTQSKKETYRVVNQGQGQVQRHTRNDRAGGTNWERDIAVGVASNIIASPISDFMKDRFKDLFEALLS	[326]
(31)	....CS--VQDCKK..SKV.I....CV.FD..K.M.SNGGGA----------	[326]

Table 3.5. MSA of the CaM-like and putative proteins with the seed numbered from Table 3.1.

No.	Alignment	
(0)	----------MADQLTDDQISEFKEAFSLFDKDGDGCITTKELGTVMRSLGQN-PTEAELQDMINEVDADGNGTIDFPEFLNLMARKMKDTD	[140]
(2)	----------MEEIQQQQQQQQQQQQQQQQQQQQQQQE..QE..M.....C..........AD..A..I...D..-...Q.....T.I.S.....E.S.....NQLQE..	[140]
(14)	----------MSHKVSKK.DEE..N.LR.I.RS..RNK..SL.QL...SLL.A..VK-.SPDQFETL.DKA.TKS..LVE....VA.VSPELLSPA	[140]
(15)	----------MSNVSFLELQYKLSKNKMLRKPSRMFSRDRQSSGLSSPGPGGFSQPSVN.MRRV..R..L.K..K.SQT.YKV.L.A...E-RAIEDVPKIFKA..L..D.F...R..IDAYK.SGGIRS	[140]
(16)	----------MSCDGGKPAPAK.G.E.LA.LR.I.RS..QNK..SL.EL...SLL....LK-.SQDQ.DTL.QKA.RNN..LVE.S..VA.VEPDLVK--	[140]
(18)	----------MVRVFLLYNLFNSFLLCLVPKKLRVFFPPSWYIDDKNPPPP.ESETESPVDL.RV.QM...N...R..KE..NDSLEN..IF-MPDKD.IQ..QKM..N.D.CV.IN..ES.YGSIVEEKE	[140]
(21)	----------ME...RQL.DI.DR..M.A..SL.IL..AALL....LK-.SGDQIHVLLASM.SN...FVE.D.LVGTILPDLNEE-	[140]
(22)	----------MDPT.L.RV.QM...N...T..G...SETL....IY-IPDK..TQ..EKI.VN.D.CV.ID..GE.YKTI.DEE.	[140]
(23)	----------MDST.LNRV.QM.......K......NESFKN..II-IP.D..TQI.QKI.VN.D.CV.IE..GE.YKTI.VEDE	[140]
(26)	----------..NKF.RQ.....R.Q..VY..N...H...E.F.A.....L.-L.Q....EE..DS.L..D...N.T...CA..---...Y	[140]
(29)	----------MTLAKNQKSSLSRLYKKVSSKRSESSRNLEDESRTSSNSSGSSSLNVN.LRTV.DYM.ANS..K.SGE..QSCVSL..GA-LSSR.VEEVVKTS.V..D.F...E...K..EGEDG--S	[140]
(32)	----------MASANPETAKPTPATVDMANPE.L.KV.DQ..SN...K.SVL...G.FKAM.TS-Y..T..NRVLE...T.RD.Y.NLD..ST.CRSS-----	[140]
(33)	----------..STKPT...KQL.DI.AR..M.K..SL.QL..AALL....IK-.RSDQISLLL.QI.RN...SVE.D.LVVAILPDINEE-	[140]
(34)	----------MSKNVSRNCLGSMEDI.KV.QR...NN..K.SID..KD.IGA.SP.-ASQE.TKA.MK.F.L....F..LD..VA.FQISDQSSN	[140]
(35)	----------MANTNLESTNKST.PSTDM.L.KV.DK..AN...K.SVS...N.FK.M.TS-Y..E..NRVLD.I.I.CD.F.NQE..ATICRSS-----	[140]
(36)	----------MSVA.I.ERV..NK..K.SWD.FAEAI.AFSPS-I.SE.IDN.FR.I.V..DNQ..VA.YASCLMLGGEGNK	[140]
(37)	MKLAKLIPKRFFIRSKDRSTVSKSPTAFSFGSASSSSGQDCKNSGGDGGGGSVTPTSILPEVPSPYSYV.ILQ..K.I.R.N..AVSRHD.ESLLSR..PDPL..E.INV.LK...C..D...RLE.LASRVVSLDPAR.	[140]
(38)	----------MASANPETAKPTPATVDMANPE.L.KV.DQ..SN...K.SVL...G.FKAM.TS-Y..T..NRVLE...T.RD.Y.NLD..ST.CRSS-----	[140]
(39)	----------MKNTQRQLSSSFMKFLEEKN------RDLEAV.AYM.ANR..R.SAE..KKSFKT..EQ-MSDE.AEAAVKLS.I..D.ML.IN..AL.IKGNDEF-T	[140]
(40)	----------MGK.G.S...V.SM....M...T....K.APS...IL.....G.-..Q.Q.KSI.ASENLS--SPF..NR..D...KHL.TEP	[140]
(41)	----------MSK.G.SN..V.SM....M...T....K.APS...IL.....G.-...SQ.KSI.TTENLS--SPF..NR..D...KHL.TEP	[140]
(42)	----------MEINNEKKKLSRQSSSFRLRSPSLNALRLHRV.D....NN..F..VE..SQALSR..LD-ADFSD.KSTVDSFIKPDKTGLR.DD.AA.HKTLDESFF	[140]
(43)	----------MKNNTQPQSSFKKLCRKLSPKREDSAGEIQQHNSSNGEDKN------R.LEAV..YM.ANR..R.SPE..QKSFMT..EQ-LSDE.AVAAVRLS.T..D.ML..E..SQ.IKVDD----	[140]
(44)	----------MA.I.ESV..NK..K.LWD.FAEAI.VFSPQ-I.SE.IDK.FIVL.V..D.Q..DV..ASCLMVNGGGEK	[140]
(45)	----------MESNNNEKKKVARQSSSFRLRSPSLNALRLQRI.D....N...F..VE..SQALTR..L.-ADLSD.KSTVESYIQP..TGLN.DD.SS.HKTLDDSFF	[140]
(0)	S----------EEELKEAFRVFDKDQNGFISAAELRHVMTNLG--EKLTDEEVDEMIKEADVDGDGQINYEEFVKVMMAKRRGKRVMAAKRSSNSAEYKEKNGRRKSHCRIL----------	[273]
(2)	A----------D.......K........Y...S......I...-=.........Q......L.....V..D...RM..ING----------	[273]
(14)	KRTTPYT----------..Q.LRL..I..T.G....T....A.S.AK..--HA..VA.LTG......S....R..FQ..A.AINSAAFDDIWG----------	[273]
(15)	----------SDIRNS.WT..LNGD.K...E.VMS.LWK..-=.RCSL.DCNR.VRAV.A....LV.M...I.M.SSNNV----------	[273]
(16)	---CPYT----------DDQ..AI..M..R.G..Y.T....A.S.AK..-=HA..A..LTG......R....C.DFQ...QAITSAAFDNAWG----------	[273]
(18)	EGD----------MRD..N...Q.GD...TVE..NS...S..LKQGK.L.CCK...MQV.E....RV..K..LQM.KSGDFSN.S----------	[273]
(21)	---VLIN----------S.Q.L.I.KS..R.G.........AGA.AKM.--QP..YK.LT.......TN...V.SFG..ASI.AKSAVDYFGLKINS----------	[273]
(22)	EEE----------.DM....N...QNGD...TVD..KA.LSS..LKQGK.LDDCKK...KV......RV..K..RQM.KGGGFNSL----------	[273]
(23)	DEVG----------..DM....N...RNGD...TVD..KA.LSS..LKQGK.L..CRK..MQV......RV..M..RQM.KKG.FFSSLS----------	[273]
(26)	.----------.KD..KD..L..I.K........M.Y.R.I.R--W.Q....I..I..A...........R..ARL....NQ.HDTKYDTTGGTLERDLAAGVAKNIIAAPMTDFIKNLFEALFS	[273]
(29)	DEER----------RK......GMYVMEGEE..T..S..RTLSR..-=.SC.VDACKV..RGF.QND..VLSFD...LM.R----------	[273]
(32)	--SS----------AA.IRD..DLY.Q.K..L...S..HQ.LNR..--MSCSV.DCTR..GPV.A....NV.F...Q.M.TSSSLLNSNGS.APP.T----------	[273]
(33)	---VLIN----------Q.Q.M.V..S..R.G..S.T....AGS.AKM.--HP..YR.LT..MT...SN...V.SFN..SHI.AKSAADFLGLT.S----------	[273]
(34)	-NSA----------IRD.....DLY.L.R..R...N..HS..K...-=..CSIQDCQR..NKV.S....CVDF...K.M..INGSA----------	[273]
(35)	--SS----------AV.IR...DLY.QNK..L..SS.IHK.LNR..--MTCSV.DCVR..GHV.T....NV.F...Q.M.SSPELV.GTV.NS----------	[273]
(36)	-EDE----------DIVM....DLY.I.GD.K...S.IHV.LKR..-=..Q.IA.CIA.VRAV.A....FVSF...KTM.SCNNKKLQ----------	[273]
(37)	----------ST....T.EF..A.RD.L...D..LR.FSTI.D-.RC.LDDCKR..ADV.E....FVCFT..SRM.DLQ.----------	[273]
(38)	--SS----------AA.IRD..DLY.Q.K..L...S..HQ.LNR..--MSCSV.DCTR..GPV.A....NV.F...Q.M.TSSSLLNSNGS.APP.T----------	[273]
(39)	EEEK----------KRKIM....MYIA.GEDC.TPGS.KMMLMK..-=.SR.TDDCKV..QAF.LNA..VLSFD..ALM.R----------	[273]
(40)	F----------DRQ.RD..K.L..EGT..VAV.D...IL.SI.-=...EPN.F..W...V..GS..K.R..D.IAR.V..----------	[273]
(41)	F----------DRQ.RD..K.L..EGT..VAV.D...IL.SI.-=...QPS.F..W...V..GS..K.R..D.IAR.V..----------	[273]
(42)	GGEGSCCDGSP--------.SD.E...N...E.GD.....V..QK.LKK..LP.AGEI.Q.EK..VSV.SNH..RVDFF..KNM.QTVVVPS----------	[273]
(43)	EEEK----------KM...G...LYIAEGEDC.TPRS.KMMLKK..-=.SR.TDDCRV..SAF.LNA..VLSFD..ALM.R----------	[273]
(44)	DTEE----------.VVM....DLY.M.GD.K...S.IHV.LKR..-=..H.M.DCVV.VQTV.K.S..FV.F...KIM.NSNKESH----------	[273]
(45)	GGACGGGENEDDPSSAAEN.SD.A...K...ENGD.....R..QT.LKK..LP.GGEM.R.EK..VSV.RNQ..RVDFF..KNM.RTVVIPSS----------	[273]

Overall, the quick and proper identification of the CaM family clearly shows the usefulness of ASH for family identification. Further identification of other EF-hand family members indicates its ability to identify homologous proteins and domains, and the inclusion of putative CaMs strongly suggests its capability to classify and find unknown proteins while weeding out mismatches. ASH's ability to identify homology rests strongly on both sequence similarity and the initial parameters given to the Collector. If the minimum number of iterations were increased past 250, it would likely have found many more homologous EF-hand proteins. The limitation of sequence similarity also limits ASH to primary homology and limited secondary homology. It cannot identify tertiary homology as it relies on BLAST queries and does not consider structure or function (Oul and Mumcuolu, 2007). Again, since ASH is designed to find homology quickly and without any previous knowledge, these are expected limitations.

3.2 ASH vs. PSI-BLAST

While the ASH results look great on their own, they achieve a much greater margin of accuracy when compared to PSI-BLAST (PSI-BLAST). After four iterations using the same parameters and database as ASH, the results included no new sequences but instead rotated two sequences in and out. From the very first iteration, PSI-BLAST acquired a number of kinases and failed to acknowledge a number of CaM and CaM-related proteins, unfortunately including both CaM1 and 3 (Table 3.1). The second and third iterations show little to no improvement, and the fourth iteration revealed an unwanted surprise – several of ASH's identified proteins with 85 and 90% correlation disappeared from the results. A quick glance at the fourth iteration tree shows an even more alarming problem: some of the proteins highly connected to the seed are not even CaM proteins (Figure 3.2). In fact, most of the tree consists of kinases identified as from the group circled in blue in Figure 3.1.

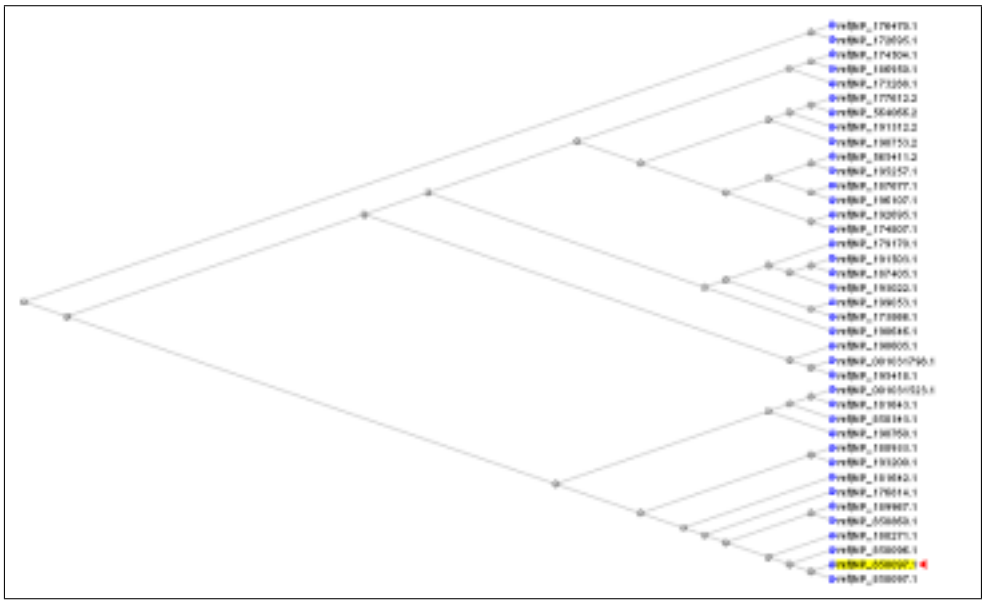
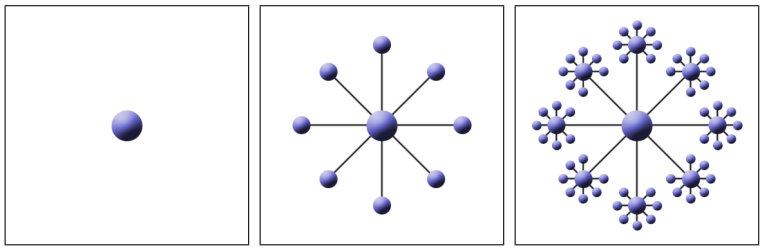


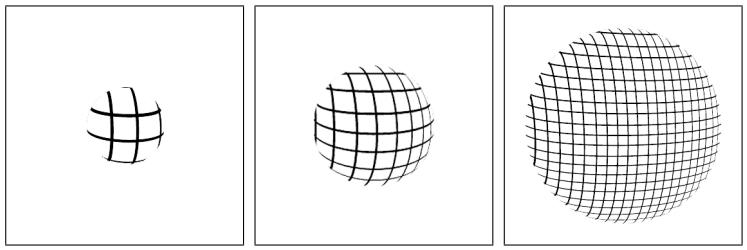
Figure 3.2. The fourth PSI-BLAST iteration on Calmodulin protein NP_850097.

Another concern is the dubitable consistency of the PSI-BLAST dataset. Slight variations surface between queries with the exact same initial parameters and, worse, between iterations of a single query. A few magically disappearing and reappearing sequences exist in the result set of one iteration, disappear in the next, and pop back up in the next without pattern or explanation. In these tests, some are in iterations one and three only, others in one, two, and four, and still others in only one or two and four. None of these proteins were included in the ASH data set and most of them were calmodulin-dependent or calmodulin-domain kinases and were correctly discarded to begin with. A couple of these were discarded before fourth iteration, but an alarming number of them showed up in the results. Since the integrity of the PSI-BLAST dataset relies on the increasing accuracy of each dataset, the fluctuating results casts uncertainty on the ability of the program to determine homology.

The main reason for the difference between the two result sets originates from the algorithms the programs use. As mentioned earlier, PSI-BLAST initially runs a basic BLAST query on the given seed sequence and uses the results to generate a position-specific scoring matrix. It then uses the matrix to seed further iterations and



(a) ASH Progression



(b) PSI-BLAST Progression

Figure 3.3: a) ASH begins with a single query and branches out to achieve a true-to-form relational network. b) PSI-BLAST increases the range and specificity of its search contents based upon a position-specific scoring matrix, which skews the connection between the seed and resultant sequences.

strengthens the matrix with each subsequent iteration (Shah et al., 2008; Jun Wu, 2006; Zhang et al., 2005). Problems arise when a protein contains multiple domains or if a protein is often associated with a different domain without actually containing the domain itself, as is the case with CaM and kinase proteins. Unfortunately with CaM, the kinase domain is not only larger, but more prevalent and thus masks the EF-hand domain the PSSM by the fourth iteration of PSI-BLAST and incorrectly identifies unrelated proteins as close connections to the seed. ASH instead uses the original BLAST program blastp for homology detection; however, it overcomes the single sequence limitation by using iterative querying of SAB with the sequences returned from previous BLAST queries. This creates a true-to-form graph-based network with a scope based upon the user-input parameters. A simple way to imagine the difference between the two algorithms is to think of it in terms of sampling the aquatic life in a large lake: PSI-BLAST anchors in one spot with a small, generic net while throwing back mismatches, all the while increasing the net

size and mesh density; ASH instead takes an initial reading at one spot and branches out from each previous cast (Figure 3.3).

CHAPTER FOUR

Conclusion

The tests using the *Arabidopsis thaliana* Calmodulin protein showed increased reliability and performance of ASH over the current popular program PSI-BLAST. The ASH results are more accurate and provide the same quality of data for any other protein included in the set without the need for further computations. Our current project focuses on v-Abl using the techniques described here in order to provide further evidence of ASH's performance, and to help elucidate the homology of v-Abl. Although ASH cannot identify remote homology due to the the strong emphasis on sequence similarity, if the relational scores generated by ASH are combined with known structural and functional data, it could potentially elucidate relational links between homologous proteins with little sequence similarity. These links could provide sequences for further iterations in the Collector, thus greatly enhancing the specificity for the dataset and possibly allowing for homology identification with a greater clarity than current methods.

In addition, ASH shows increased potential over PSI-BLAST in simple domain finding versus a position-specific approach. The generated dataset shows an unusual potential for identifying evolutionary and functional relationships on many levels. For starters, the higher correlation of some non-CaM proteins with the seed CaM5 over CaM9 suggests a different evolutionary route for the CaM9 protein. It could also denote a slight difference in function from CaM9 versus the other identified CaMs. In addition, the inclusion of kinases into the network highlights a functional relationship on some level. Although the CaM-kinase relationship is already known, this tool could help identify other unknown functional relationships for other homologies. This connection also questions their evolutionary relationship since at least one

fusion protein containing both individual proteins is needed for either protein to show up in the other's BLAST results. On another level, the scale-free network generated at a threshold of 80% shows one dense group of kinases with lower correlation scores and two sparser, more highly correlated groups of proteins. These relationships uncover different types of functional kinases. As a whole, the ASH program creates a graph-based analysis of protein sequences, which could allow for deeper analysis into other homologous and evolutionary relationships.

A smaller application of ASH might involve improvement of the integrity of the RefSeq database. The cliques and sub-networks produced at very high thresholds (greater than 95%) could identify possible duplicate proteins in the database. Once candidate proteins are identified, a simple MSA or pair-wise alignment of these few sequences rather than a MSA of the whole dataset would show whether or not the sequences were identical or simply very closely related mutations of one another. On a more practical level, ASH may help identify new proteins for vaccine research or drug targets. Identification of proteins similar to know disease-causing proteins such as oncogenes or uncovering of viral proteins attempting to mimic native proteins can help with understanding the exact mechanism underlying the disease. With the improved homology detection of ASH, these tests are a more viable solution. Other future projects using the ASH program include attempts to merge the correlation scores with pre-existing data to produce tertiary homology relationships and testing homology identification for proteins with multiple domains.

CHAPTER FIVE

Licensing

ASH is licensed under the GNU Public license (GPL), Academic Free license (AFL), and Educational Community license (ECL). The complete source code and documentation for ASH is available for download from SourceForge at <http://sourceforge.net/project/ash-protein/>.

APPENDICES

APPENDIX A

Selections of Perl Code from ASH

A.1 Parsing of BLAST Resutls

```
sub ash_sabParse($) {
   my $self = shift;
   my $infile = shift;
   my $scores = ();
   my $count = 0;
   my $acc = 0; my $bs = 0; my $ev = 0;

   ##### read in the BLAST file data
   open(BF, $infile);
   my @blast_file = <BF>;
   close(BF);
   splice(@blast_file,0,20);

   ##### parse through the data
   foreach my $line (@blast_file) {
      if($line =~ m/^>/ || $line =~ m/Matrix\:/) {
         @blast_file = {};
         last; }
      if($line =~ /(hypothetical)/ && !$self->{FLAGS}{HYPOTHETICALS}) {
         next; }

      ##### The first expression picks almost every line out
      ##### The second picks out lines with an accession right at the front
      ##### The third should pick out any leftovers we might want
      ##### The fourth seems only to pick out blank lines
      elsif($line =~
m/^ref\|(NP_\d{3,})([\w\W[:punct:]]*)*\s+([\d\.]+)\s+(\d*(e-|\.)\d*)/) {
         $acc = $1;
         $bs = $3;
         $ev = $4;
      } elsif($line =~
m/^(\w+\|)?(\w+\d*_\w+)?(\w+\d{3,})\s*([\w\W[:punct:]]*)*
   \s+([\d\.]+)\s+(\d*(e-|\.)\d*)/) {
         $acc = $3;
         $bs = $5;
         $ev = $6;
      } elsif($line =~
m/^[\w\d\_]*\s\((\w+[\w\d]*)\).*\s+([\d\.]+)\s+(\d*(e-|\.)\d*)/) {
         $acc = $1;
         $bs = $2;
         $ev = $3;
      } elsif($line =~
m/^.*\(?(\w+\d{3,})\)?.*([\d\.]+)\s+(\d+(e-|\.)\d*)/) {
         print WARN "NOT_USED: $line";
      } elsif($line =~ /\s*/) {
      } else { print ERR "Not_sure_how_we_got_here: $line"; }

      # saving the good data or printing to error file
      if(!$acc || !$ev || !$bs) { next;
      } elsif($acc =~ m/XP_/i) {
         print WARN "NOT_USED, experimental: $line";
         next;
      } elsif(exists($self->{BAD_SEQUENCES}{$acc})) { next;
      } elsif($self->{FLAGS}{SCORE_TYPE} eq 'bitscore') {
         $scores->{$acc} = $bs;
```

```
            $count++;
        } elsif($self->{FLAGS}{SCORE_TYPE} eq 'evalue') {
            $scores->{$acc} = $ev;
            $count++;
        } else {}
    }

    $count > 0 ? return $scores : return 0;
}
```

A.2 Identification of Relational Sub-networks

```
sub find_network {
    my $self = shift;

    if(!exists($self->{CORRELATION_I}) ||
          !exists($self->{CORRELATION_II}) ||
          !exists($self->{FLAGS}{FIND})) {
        return 0; }

    print "Compiling full $self->{SEED}{ACC} network...\n";

    my $sub_network = {};
    my $sub_network_count = {};
    my $sub_network_omit = {};

    $sub_network->{$self->{SEED}{ACC}} = {};

    ##### OBTAIN a list of all nodes with a valid edge connected to our FIND
    foreach my $leaf (keys %{$self->{CORRELATION_I}{$self->{SEED}{ACC}}}) {

        ##### VALIDATE the edge
        if($self->{CORRELATION_II}{$self->{SEED}{ACC}}{$leaf}
              >= $self->{FLAGS}{THRESHOLD}) {

            ##### SAVE the edge (from both nodes)
            $sub_network->{$self->{SEED}{ACC}}{$leaf}
                = $self->{CORRELATION_I}{$self->{SEED}{ACC}}{$leaf};
            $sub_network->{$leaf}{$self->{SEED}{ACC}}
                = $self->{CORRELATION_I}{$leaf}{$self->{SEED}{ACC}};

            #### ADD a node to our list
            $sub_network_omit->{$leaf} = 1; }    }

    ##### ITERATE through the list to find other connections in sub network
    while((my $branch,my $junk) = each %{$sub_network_omit}) {

        ##### DELETE the current node -- the graph is symmetrical
        delete $sub_network_omit->{$branch};

        #### ITERATE through the list
        foreach my $leaf (keys %{$sub_network_omit}) {
            ##### VERIFY and VALIDATE the edge
            if(exists($self->{CORRELATION_II}{$branch}{$leaf}) &&
                  $self->{CORRELATION_II}{$branch}{$leaf}
                     >= $self->{FLAGS}{THRESHOLD}) {

                ##### SAVE the edge (from both nodes)
                $sub_network->{$branch}{$leaf}
                    = $self->{CORRELATION_I}{$branch}{$leaf};
                $sub_network->{$leaf}{$branch}
                    = $self->{CORRELATION_I}{$leaf}{$branch};
            }
        }
    }
```

```
    ##### ARRANGE by seed highest correlation
    my @order = sort { $sub_network->{$self->{SEED}{ACC}}{$b}
          <=> $sub_network->{$self->{SEED}{ACC}}{$a} }
       keys %{$sub_network->{$self->{SEED}{ACC}}};
    unshift(@order, $self->{SEED}{ACC});

    ...

    return 1;
}
```

A.3 Structure of an AXL Document

```
<!-- ROOT -->
<ash>

    <!-- SEED -->
    <seed>
       <accession></accession>
       <length></length>
       <sequence></sequence>
    </seed>

    <!-- SESSION -->
    <session>
       <id></id>
       <directory></directory>
    </session>

    <!-- COLLECTOR -->
    <collector>
       <blast>
          <e_value></e_value>
          <database></database>
          <program></program>
          <method></method>
       </blast>

       <flags>
          <iterations></iterations>
          <score_type></score_type>
          <hypotheticals></ihypotheticals>
          <blast_files></blast_files>
          <data_files></data_files>
       </flags>
    </collector>

    <!-- EXTRACTOR -->
    <extractor>
       <threshold></threshold>
       <maximal></maximal>
       <submaximal></submaximal>
       <network></network>
       <subnetwork></subnetwork>
       <find>
          <accession></accession>
          <clique></clique>
          <network></network>
       </find>

       <flags>
          <maximal></maximal>
          <submaximal></submaximal>
          <network></network>
          <subnetwork></subnetwork>
          <graph></graph>
```

```
      <find></find>
   </flags>
</extractor>

<!-- RECOVERY -->
<recovery>
   <hash></hash>
   <logodds></logodds>
   <matrix></matrix>
   <extract>
      <threshold></threshold>
      <correlation_01></correlation_01>
      <identity></identity>
      <correlation_02></correlation_02>
   <extract>
   <last_clique></last_clique>
   <last_network></last_network>
</recovery>

<!-- TABLE OF CONTENTS -->
<table_of_contents>
   <protein>
      <accession></accession>
      <id></id>
      <genus></genus>
      <species></species>
      <description></description>
      <length></length>
      <sequence></sequence>
   </protein>
</table_of_contents>

<!-- CLIQUES -->
<clique>
   <id></id>
   <type></type>
   <depth></depth>
   <threshold></threshold>
   <count_nodes></count_nodes>
   <count_edges></count_edges>
   <seed>
      <accession></accession>
      <number_of_edges></number_of_edges>
   </seed>
   <element>
      <accession></accession>
      <number_edges></number_edges>
   </element>
   <edge>
      <accession></accession>
      <accession></accession>
      <weight></weight>
   </edge>
</clique>

<!-- NETWORKS -->
<network>
   <id></id>
   <type></type>
   <depth></depth>
   <threshold></threshold>
   <count_nodes></count_nodes>
   <count_edges></count_edges>
   <seed>
      <accession></accession>
      <number_of_edges></number_of_edges>
   </seed>
   <element>
```

```
            <accession></accession>
            <number_edges></number_edges>
        </element>
        <edge>
            <accession></accession>
            <accession></accession>
            <weight></weight>
        </edge>
    </network>

</ash>
```

BIBLIOGRAPHY

Ali, R., Ma, W., Lemtiri-Chlieh, F., Tsaltas, D., Leng, Q., von Bodman, S., and Berkowitz, G. A. 2007. Death don't have no mercy and neither does calcium: Arabidopsis cyclic nucleotide gated channel2 and innate immunity. *Plant Cell* 19:1081–1095.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215:403–410. Ed167 Times Cited:23598 Cited References Count:22.

Barabasi, A. L. 2007. The architecture of complexity. ID: 1 Generic.

Biegert, A., Mayer, C., Remmert, M., Soding, J., and Lupas, A. N. 2006. The mpi bioinformatics toolkit for protein sequence analysis. *Nucl. Acids Res.* 34:W335–339.

Braam, J., Sistrunk, M. L., Polisensky, D. H., Xu, W., Purugganan, M. M., Antosiewicz, D. M., Campbell, P., and Johnson, K. A. 1997. Plant responses to environmental stress: regulation and functions of the arabidopsistch genes. *Planta* 203:S35–S41.

Chiasson, D., Ekengren, S., Martin, G., Dobney, S., and Snedden, W. 2005. Calmodulin-like proteins from arabidopsis and tomato are involved in host defense against pseudomonas syringae pv. tomato. *Plant Molecular Biology* 58:887–897. 10.1007/s11103-005-8395-x.

Delaney, K. J., Xu, R., Zhang, J., Li, Q. Q., Yun, K.-Y., Falcone, D. L., and Hunt, A. G. 2006. Calmodulin interacts with and regulates the rna-binding activity of an arabidopsis polyadenylation factor subunit. *Plant Physiol.* 140:1507–1521.

Dinc, D. 2007. The effect of euclidean distance in directed scale-free network generation. ID: 1 Generic.

Ellson, J., Gansner, E., Koutsofios, L., North, S. C., and Woodhull, G. 2002. Graphviz - open source graph drawing tools. *Graph Drawing* 2265:483–484. Bw21m Times Cited:3 Cited References Count:1 Lecture Notes in Computer Science.

Finkler, A., Ashery-Padan, R., and Fromm, H. 2007. Camtas: Calmodulin-binding transcription activators from plants to human. *FEBS Letters* 581:3893–3898.

Frickey, T. and Lupas, A. 2004. Clans: a java application for visualizing protein families based on pairwise similarity. *Bioinformatics* 20:3702–3704.

Frickey, T. and Weiller, G. 2007. Analyzing microarray data using clans. *Bioinformatics* 23:1170–1171.

Galon, Y., Nave, R., Boyce, J. M., Nachmias, D., Knight, M. R., and Fromm, H. 2008. Calmodulin-binding transcription activator (camta) 3 mediates biotic defense responses in arabidopsis. *FEBS Letters* 582:943–948.

Gansner, E. R. and North, S. C. 2000. An open graph visualization system and its applications to software engineering. *Software-Practice and Experience* 30:1203–1233. 353KZ Times Cited:72 Cited References Count:53.

Hongbo, L. 2006. Intrinsic fitness and preferential attachment in scale-free networks. ID: 1 Generic.

Ihaka, R. and Gentleman, R. 1996. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5:299–314. Copyright 1996 American Statistical Association, Institute of Mathematical Statistics and Interface Foundation of America ArticleType: primary_article / Full publication date: Sep., 1996 / Copyright 1996 American Statistical Association, Institute of Mathematical Statistics and Interface Foundation of America.

Jun Wu, G. H. M. K. U. S. z. T. 2006. Identification of new claudin family members by a novel psi-blast based approach with enhanced specificity. *Proteins: Structure, Function, and Bioinformatics* 65:808–815. 1097-0134 10.1002/prot.21218 Journal Article.

Lee, D., Polisensky, D. H., and Braam, J. 2005. Genome-wide identification of touch- and darkness-regulated arabidopsis genes: a focus on calmodulin-like and xth genes. *New Phytologist* 165:429–444.

Li, G. 2007. Mining association rules in scale-free networks. ID: 1 Generic.

Nikolski, M. and Sherman, D. J. 2007. Family relationships: should consensus reign?–consensus clustering for protein families. *Bioinformatics* 23:e71–76. 1460-2059 Journal Article.

Oul, H. and Mumcuolu, E. . 2007. A discriminative method for remote homology detection based on n-peptide compositions with reduced amino acid alphabets. *Biosystems* 87:75–81. Journal Article.

Peng, R. 2003. Multi-dimensional point process models in r. *Journal of Statistical Software* 8:1–27. CODEN: JSSOBK 1548-7660 Journal Article.

Ritz, C. and Streibig, J. C. 2005. Bioassay analysis using r. *Journal of Statistical Software* 12:1–22. CODEN: JSSOBK 1548-7660 Journal Article.

Schock, I., Gregan, J., Steinhauser, S., Schweyen, R., Brennicke, A., and Knoop, V. 2000. A member of a novel arabidopsis thaliana gene family of candidate mg2+ ion transporters complements a yeast mitochondrial group ii intron-splicing mutant. *The Plant Journal* 24:489–501. M3: doi:10.1046/j.1365-313x.2000.00895.x Journal Article.

Shah, A. R., Oehmen, C. S., and Webb-Robertson, B.-J. 2008. Svm-hustle - an iterative semi-supervised machine learning ap-proach for pairwise protein remote homology detection. *Bioinformatics* . 1460-2059 Journal Article.

Sistrunk, M. L., Antosiewicz, D. M., Purugganan, M. M., and Braam, J. 1994. Arabidopsis tch3 encodes a novel ca2+ binding protein and shows environmentally induced and tissue-specific regulation. *Plant Cell* 6:1553–1565.

stergrd, P. R. J. 2002. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics* 120:197–207.

Xiaohua, H. 2007a. Data mining and predictive modeling of biomolecular network from biomedical literature databases. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* 4:251–263. ID: 1 Generic.

Xiaohua, H. 2007b. A novel approach for mining and fuzzy simulation of subnetworks from large biomolecular networks. *Fuzzy Systems, IEEE Transactions on* 15:1219–1229. ID: 1 Generic.

Xin Biao, L. 2006. Consensus in scale-free networks. ID: 1 Generic.

Yang, T., Chaudhuri, S., Yang, L., Chen, Y., and Poovaiah, B. W. 2004. Calcium/-calmodulin up-regulates a cytoplasmic receptor-like kinase in plants. *J. Biol. Chem.* 279:42552–42559.

Young-Rae, C. 2006. Efficient modularization of weighted protein interaction networks using k-hop graph reduction. ID: 1 Generic.

Zhang, Z., Kochhar, S., and Grigorov, M. G. 2005. Descriptor-based protein remote homology identification. *Protein Science* 14:431–444. Journal Article.