Clustering on Protein Sequence Motifs using SCAN and Positional Association Rule Algorithms

Bernard Chen[1], Ben Nordin[1], Sriram Bobba[1], Devendar Singireddy[1], Brad Taylor[1], Sinan Kockara, and Mutlu Mete[2]

[1]University of Central Arkansas, Department of Computer Science

[2] Texas A&M University-Commerce, Department of Computer Science

***Abstract*— The role of protein sequence motifs is in predicting functional or structural portion of other proteins including prosthetic attachment sites, enzyme-binding sites and DNA /RNA binding sites, and so on. A fixed window size is usually predefined to discover protein sequence motifs for many algorithms and techniques. However, the predefined window size may deliver a number of similar motifs simply shifted by some bases or including mismatches. In this paper, we use the positional association rules algorithm to form motifs network and adapt a Structural Clustering Algorithm for Networks named SCAN to recognize similar motifs. Although association rule based algorithms have been widely adapted in association analysis and classification, few of those are designed as clustering methods. With the SCAN analysis, the qualities of the clusters are further improved.**

***Index Terms*— Positional Association Rules, SCAN, Protein Sequence Motifs**

I. INTRODUCTION

Bioinformatics is the science of interpreting data from observations of biological process whose data is managed and mined [2]. Unlike data generated in various fields to support a hypothesis, the biological data is generated assuming that it contains vital information, and this information might answer several important questions. [3].

One of the most important applications of data mining is in the field of bioinformatics, because of its huge mass of data and hidden patterns particularly in proteomics data. The proteomic data consisting of sequence motifs in recurring patterns has the capability to predict a protein's structure and functionalities [8]. In order to identify sequence motifs, most algorithms need to specify a fixed size for the motif in advance. These algorithms deliver a similar number of motifs since they have a fixed size (1), include mismatches, or (2) are shifted by one base [5]. The problem of mismatches is addressed by showing that some groups of protein motifs occur in recurring patterns. The first problem implies that some group motifs may be similar to one another; the second problem probably can be more easily seen in this way: If there exists a biological sequence motif with length of 12 and we set the window size to 9, it is highly possible that we discovered two similar sequence motifs where one motif covers the front part of the biological sequence motif and the other one covers the rear part [8].

The Association Rule [1, 6, 7] is used to extract important information from large repositories of data. For example, association rules can discover the support and confidence of "if A occurs then B will occur." This can be expanded to any number of item sets whether it is three, four, or more. To put forth this kind of DNA/Protein bioinformatics data into Association Rules, each protein is regarded as a transaction and the sequence motifs as items in the transaction. Some of the papers that were referenced apply the Association Rule in this manner [1, 8]. Although Association Rule plays an important role in extracting recurring patterns from protein sequences, there is still one more criteria to be considered. The motifs in a protein occur in specific distance intervals, so it is vital to discover the distance between the occurrence of motifs A and B. Therefore, a new Positional Association Rule Algorithm is proposed in [8]. The Positional Association rule is simple extension of the Basic Association rule with a new parameter named "*Distance Assurance*".

It is proved that the fixed window size problem can be solved by generating clusters with the help of the Positional Association Rules Algorithm in [8]. In this paper, Structural Clustering Algorithm for Networks (SCAN) [10], a new clustering algorithm for networks, is applied to generate clusters from the Positional Association Rules. SCAN is a popular tool for analyzing graphs. SCAN's ultimate goal is to divide the nodes in the graph into three categories: clusters, hubs, and outliers. It creates clusters from structurally similar nodes [10]. For example, social networks may suggest a friend to you because you share similar friends with that person (i.e. you both belong to the same cluster). Nodes that belong to more than one cluster may bridge the two clusters together. SCAN identifies nodes of this pattern as hubs. Finally, SCAN marks structurally dissimilar nodes as outliers, which may be discarded as noise data [10].

In this paper, we propose that one can use SCAN to refine positional association rule results in order to increase the quality of the resulting clusters. We apply proposed approach to alleviate the first problem "*include mismatches*" caused by the fixed window size approach. The set of rules produced using the positional association rule are fed into SCAN, to generate clusters, outliers, and hubs. The outliers and hubs were discarded while the clusters were retained since the primary goal is to increase the quality of the clusters that SCAN revealed. Higher-quality SCAN clusters are verified with the quality of the positional association rule clusters.

The rest of the paper is organized into four more sections. Section II provides a detailed explanation of the algorithm.

Section III follows with details about the Experiment. Section IV shows the results of this work. Finally, the paper is concluded with Section V.

II. ALGORITHM

2.1 Positional Association Rules Algorithm

Algorithm: Positional Association Rule with the Apriori Concept
Input: Database, D, (Protein sequences as Transactions and Sequence Motifs as items), min_support, min_confidence, and min_distance_assurance
Output: P, positional association rules in D.
Method:

```
(1)  L = find_frequent_itemsets(D, min_support)
(2)  S = find_strong_association_rules(L, min_confidence)
(3)  for (k=2; Sk ≠ Ø; k++)
(4)     for each strong association rule, r ∈ Sk
(5)        antecedent_motif = Apriori_Motif_Construct(r_ant)
(6)        consequence_motif = Apriori_Motif_Construct(r_con)
(7)        if antecident_motif == NULL or consequence_motif == NULL:
(8)           goto Step (4)
(9)        for each protein sequence, ps ∈ D
(10)          for (ant_position=1; |ps| ; ant_position++)
(11)             if antecedent_motif start appear on ps[ant_position]:
(12)                r_ant_count++
(13)                for (con_position=1; |ps| ; con_position++)
(14)                   if consequent_motif start appear on ps[con_position]:
(15)                      distance = ant_position – con_position
(16)                      r_distance ++
(17)       Pk = { r_distance | r_distance > min_distance_assurance * r_ant_count }
```

Apriori_Motif_Construct(itemset)

```
(1)  if |itemset| == 1:
(2)     return itemset
(3)  else:
(4)     for each positional association rules in P|itemset|
(5)        if all items in the itemset appear in the positional association rule:
(6)           return the new motif constructed by the positional association rule
(7)     return NULL
```

Figure 1 The Pseudocode of Positional Association Rule with the Apriori concept

The Association Rule in Data Mining generates item sets which occur frequently with certain rules occurring in a particular format, say (X=>Y) i.e. "if X occurs then Y occurs" with the condition that all of these item sets must pass a minimum support and confidence. A new Positional Association Rule, proposed in [8], has another parameter called "*distance assurance*." The Positional Association Rule identifies a frequent item set with a certain frequent distance (d) and applies this distance once it obtains strong Association rules with a minimum confidence and minimum support. Where support and confidence is defined as:

$$Support(X \Rightarrow Y) = \frac{|X \cup Y|}{|T|}$$

$$Confidence(X \Rightarrow Y) = \frac{|X \cup Y|}{|X|}$$

Where $|T|$ is the total number of transactions, $|X|$ is the number of transactions in T that contains at least one X, $|X \cup Y|$ is the number of the transactions in T that contain both X and Y. The newly proposed "*distance assurance*" is defined as:

$$Dis.Assurance(X \overset{d}{\Rightarrow} Y) = \frac{\left\| X \overset{d}{\cup} Y \right\|}{\|X\|}$$

Where $\|X\|$ is the total number of times that X appears in T, d indicates the distance, $-\infty < d < \infty$. Where $X \overset{d}{\Rightarrow} Y$ denotes "if X appears, then after the distance of d, Y appears," $\|X \overset{d}{\cup} Y\|$ is the total number of times in T that when X occurs and after the distance of d, Y occurs. Figure 1 shows pseudo code for the Positional Association Rule Algorithm and a detailed description is available in [8].

2.2 SCAN Algorithm

SCAN is short for Structural Clustering Algorithm for Networks. While many algorithms find just the clusters in a network, SCAN finds the hubs and outliers. The identification of hubs is the real strength of SCAN, as hubs bridge clusters, and spread its influence from cluster to cluster. The usefulness on identifying outliers on the other hand, is simply in knowing that the outliers can be ignored. Outliers have little influence on their connected cluster, or on the cluster's network.

SCAN works by looking at the neighborhood of vertices instead of only their direct connections. This allows the detection of hubs and outliers. Not only is the algorithm useful, but it is also efficient with a running time of O(n).

When running SCAN, the algorithm labels a newly found vertex as unclassified. From here it checks to see if this vertex has a minimum amount of connections in a cluster. If so, it uses this new found core as a springboard to search for more vertices. Finally, once SCAN visits all vertices, it identifies the vertices that connect to two or more clusters as hubs, and vertices that connect to only one cluster as outliers. The more connections a vertex has to a cluster, the more influence that vertex has on the cluster.

2.3The combination of Positional Association Rules algorithm and SCAN Algorithm

In this paper, in order to alleviate the first problem "include mismatches" caused by the fixed window size approach, we combine the positional association rules algorithm with SCAN to identify protein sequence motifs that similar to each other. First of all, positional association rule algorithm with distance equals to zero is implemented to identify protein sequence motifs that occur on the same position. The rationale behind this is that if two (or more) motifs occur on the same position frequently enough (pass the minimum distance assurance), they should be similar to one another. As the result, the network-like graph such as Figure 2 is generated. Next, the associations were converted into two columns of data for input into SCAN (as showed in *Results*). The data was used to run SCAN multiple times for each distance assurance with different values of μ and ε. Finally, clusters are generated by the proposed approach, which combines the positional association rules algorithm and SCAN. Secondary structure information is taken and analyzed the quality of each SCAN-generated cluster. Detail results with different parameters are available in *Results*.

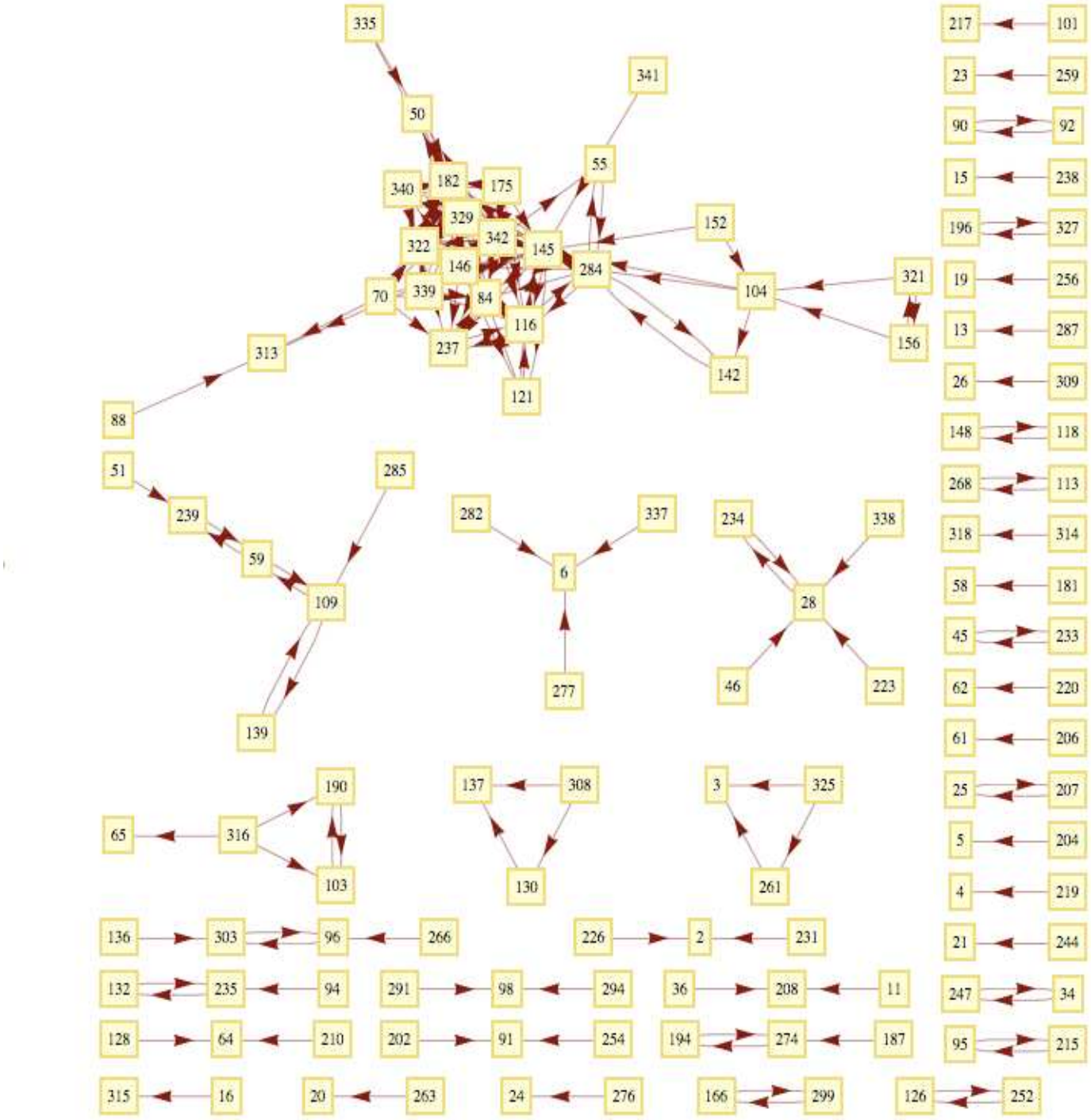

Figure 2: Directed graph generated from positional association rules based on minimum support, confidence, and distance assurance equal to 20%, 70% and 50% respectively.

III. Experiment and Parameters Setup

3.1 Dataset

The original dataset used in this work includes 2710 protein sequences obtained from Protein Sequence Culling Server (PISCES) [11]. It is the dataset that was used in [8,12] to generate protein sequence motifs. No sequence in this database shares more than 25% sequence identity. The frequency profile from the HSSP [13] is constructed based on the alignment of each protein sequence from the protein data bank (PDB) where all the sequences are considered homologous in the sequence database. For the frequency profiles (HSSP) representation for sequence segments, each position of the matrix represents the frequency for a specified amino acid residue in a sequence position for the multiple sequence alignment. Twenty rows represent 20 amino acids and 9 columns represent each position of the sliding window. Secondary structure was also obtained from DSSP [14], which

is a database of secondary structure assignments for all protein entries in the Protein Data Bank, for evaluation purposes. DSSP originally assigns the secondary structure to eight different classes. According to previous related research [12, 15], those eight classes were converted into three based on the following method: H, G and I to H (Helices); B and E to E (Sheets); all others to C (Coils). 343 different sequence motifs with window size of nine generated from previous work [12] are included in this paper. The dataset actually used in this work comes from [8] and contains more than 2000 protein sequence as transactions vary in amount of motifs (items). Each transaction sequence is sorted and organized by distance value, the items on the same line having a distance of zero from one another. The secondary structure data contained nine values for each 343 motifs, each value corresponding to its H, E, or C secondary structure percentage.

3.2 Positional Association Rule

The protein sequences are treated as transactions and the sequence motifs are treated as items of the transaction. Firstly, the association rules are generated from the data. As we mentioned in section 2.1, only tradition association rules are not sufficient due to the protein motifs occurring at positions. "*Distance assurance*" measure is incorporated. In this paper only a distance measure of zero is taken into account which means the protein sequence motifs which occur at same positions are considered.

3.3 Running SCAN for refining clusters

The SCAN proposed in [10] is used to generate clusters from the rules generated as described in section 3.2. When a member of the generated clusters is identical to a neighboring cluster, their combined structure will add up to a bigger cluster. So, the number of common neighbors is normalized by the geometric mean of the two neighborhood sizes.

$$\sigma(v,w)=\frac{|\Gamma(v)\cap\Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}}$$

where, $\Gamma(v)$ and $\Gamma(w)$ denotes the neighborhood of v and w respectively. When assigning a member to a cluster a threshold ε is applied to the computed structural similarity. Also μ number of neighbors with a structural similarity and exceeding the neighborhood threshold ε is required to decide whether a vertex is a core.

The values of ε and μ are varied to generate various clustered files. The ε is varied from 0 to 0.5 and μ is varied between 1 and 2 only although various values of μ has been used they, all proved to be ineffective.

3.4 Dissimilarity Measure

The following formula is used to calculate the dissimilarity between two sequence segments:

$$\text{Dissimilarity}=\sum_{i=1}^{L}\sum_{j=1}^{N}\left|F_k(i,j)-F_c(i,j)\right|$$

Where *L* is the window size and *N* is 20 which represent 20 different amino acids. *Fk(i,j)* is the value of the matrix at row *i* and column *j* used to represent the sequence segment. *Fc(i,j)* is the value of the matrix at row *i* and column *j* used to represent the centroid of a give sequence cluster. The lower dissimilarity value is, the higher similarity two segments have.

3.5 Structural Similarity Measure

Cluster's average structure is calculated using the following formula:

$$\frac{\sum_{i=1}^{ws}\max(\,p_{i,H}\,,p_{i,E}\,,p_{i,C}\,)}{ws}$$

Where *ws* is the window size and *Pi,H* shows the frequency of occurrence of helix among the segments for the cluster in position *i*. *Pi,E* and *Pi,C* are defined in a similar way. If the structural homology for a cluster exceeds 70%, the cluster can be considered structurally identical [13]. If the structural homology for the cluster exceeds 60% and lower than 70%, the cluster can be considered weakly structurally homologous [15].

IV. RESULTS

The positional association rule runs six times with distance assurance values of 10%, 20%, 30%, 40%, 50%, and 60%; while the minimum support and confidence is set as 20% and 70% based on the optimal parameter setup of previous work [1]. Once complete, the file was translated into a two-column format representing the associations. For example, $A \overset{0}{\rightarrow} B$ would become line "*A B*." The two column files were then fed into SCAN. An example is given in Figure 3 with minimum distance assurance equals to zero.

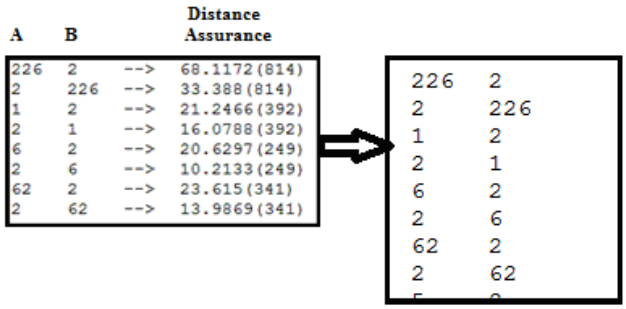

Figure 3: Conversion of the Positional Association Rules output to SCAN input

However, besides the data, SCAN requires two other parameters: ε and μ. μ is varied between 0 and 3 with step-size of 1. ε is between 0 and 1 to generate various clustering files and optimum clustered data is chosen. In the first run of SCAN, some limitations on the parameters were determined. First, Mu seems to only be effective at values 1 or 2. A value of zero results in all clusters and no outliers, a value higher than two results in all outliers and no clusters. SCAN produces hubs with values of ε greater than 0.5, so ε was restricted to lower values.

Hubs were determined to be an undesirable component in this research because they were not included with the clusters. This caused isolation of major cluster components. For example, Figure 4 shows four motifs that should belong to the same cluster. If ε was set too high, Motif #6 would be

classified as a hub, removing it from the cluster. Since 282, 337, and 277 are not associated with any other motifs, they are removed as outliers.

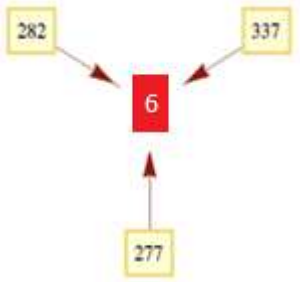

Figure 4: A Cluster with a Potential Hub

In the end, SCAN was run with distance assurance between 10% and 60%, M between 1 and 2, and E between 0.1 and 0.5. To ease the process of running SCAN on all of these parameter combinations, a script was created to run them in batch. The SCAN algorithm is a pre-packaged Java application. the algorithm was called with the appropriate combination of parameters and it gave the output files containing the clusters, hubs, and outliers obtained from the association rule data.

Next, a second script was ran, which fed each SCAN output file into the quality algorithm. The quality algorithm implements the Structural Similarity Measure discussed in section 3.5. The algorithm takes the SCAN output file and file containing motif structure information as parameters. Once complete, the algorithm produced an output file containing a percentage on each line representing a cluster's quality. Finally, a third, simple script was run to summarize the quality results and place them into range groups including >80, 70-80, 60-70, and <60. An example summary is shown in Figure 5.

DA	Mu	EPS	Cluster #	Quality	Q80	Q70	Q60	Qlow
50	1	0.1	1	0.742129	0	1	0	0
50	1	0.1	2	0.714095	0	1	0	0
50	1	0.1	3	0.615433	0	0	1	0
50	1	0.1	4	0.694561	0	0	1	0
50	1	0.1	5	0.739924	0	1	0	0
50	1	0.1	6	0.694304	0	0	1	0

Figure 5: Sample Quality Summary

Initially, all of the summary files were combined to determine which parameters gave the best results. The most favorable combination was a distance assurance of 50%, M of 1, and E of 0.3. Distance assurance had the most significant impact on cluster quality. ε, as shown in Figure 6, has little or no effect on quality.

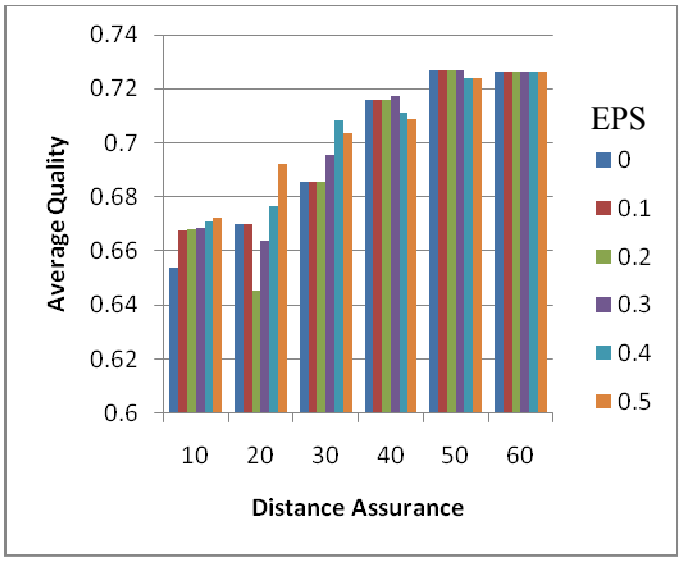

Figure 6: Dist. Assurance & ε Quality, M = 1

M has a slight effect on quality, but still does not compare to distance assurance. Figure 7 shows M's effect.

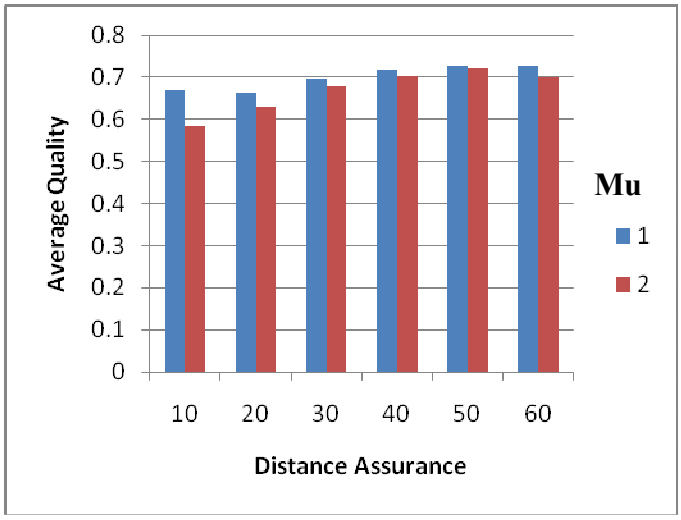

Figure 7: Dist. Assurance & μ's Quality, EPS = 0.3

With these new findings, it can be concluded that the SCAN parameters ε and μ have little effect on cluster quality as long as they stay within the range tested above. Distance assurance's effect on the result demonstrates the impact of the positional association rule analysis method. Running SCAN on the data provided two important pieces of information: clusters and outliers. Each cluster provides a graph structure containing the original associations. This allows observations to be made on groups of associations rather than one at a time. The outliers remove noise, or associations of little significance. The positional association rule does a good job of eliminating outliers based on occurrence statistics, but SCAN takes it a step further and analyzes relationship structures.

V. Conclusion

For data mining in the field of bioinformatics, the ability to find recurring patterns in proteomics data enables the discovery of a protein's structure and functionality. Most enumerative algorithms require the size of the motif to be set in advance. This can cause errors such as mismatches and bases that are off by one. However, the *Positional Association Rule* can be used as a remedy to these problems through the use of a distance assurance.

It is known that Association Rules can already be well used in Classification techniques, and *Chen et al.* [1] proved that it can also be used for Clustering purposes. In this paper, we further combine the positional association rules algorithm with the SCAN algorithm. With the SCAN data sorted, concentration solely on the clusters further increased the cluster quality.

Acknowledgment

The work of Bernard Chen was supported in part by UCA's University Research Council (URC). The work of Sinan Cockara was supported in part by UCA's University Research Council (URC).

References

[1] Bernard Chen, Michael Miller, Timothy Montgomery, Terrance Griffin, "Clustering Using Positional Association Rules Algorithm on Protein Sequence Motifs", International Conference on Bioinformatics & Computational Biology (BIOCOMP2010), Las Vegas, USA, pp.75~80.

[2] Lonardi Stefano, Chen Jake, "Biological Data Mining": Chapman and Hall/CRC 2010 Computational Biology and Bioinformatics, IEEE/ACM Transactions on April-June 2010.

[3] Arno Siebes, V. Hlavac, K.G.Jeffery and J. wiedermann (Eds): SOFSEM 2000, LNCS 1963, PP.54-55, 2000@ springer –Verlag Berlin Heidelberg 2000.

[4] Interestingness of Discovered Association Rules in terms of Neighborhood-Based Unexpectedness (1998), Guozhu Dong, Jinyan Li.

[5] Ohler u. & Niemann, H. (2001), Identification and analysis of eukaryotic promoters: Recent Computational Approaches, Trends in Genetics. 17,56-60.

[6] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", VLDB 1994.

[7] R. Agrawal, T. Imielinski and A. Swami, "Mining Associations between Sets of Items in Large Databases", *ACM SIGMOD Int'l Conf. on Management of Data,* Washington D.C., May 1993.

[8] Bernard Chen, and Sinan Kockara, "Mining Positional Association Super-Rules on Fixed-Size Protein Sequence motifs", *IEEE BIBE 2009, Taichung, Taiwan,* proceeding pp. 1-8

[9] Haoudi, Abdelali; Bensmail, Halima "Bioinformatics and data mining in proteomics" Expert Review of Proteomics, Volume 3, Number 3, June 2006 , pp. 333-343(11)

[10] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, Thomas A. J. Schweiger, "SCAN: A Structural Clustering Algorithm for Networks", Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, August 12-15, 2007, San Jose, California, USA

[11] Wang, G. & Dunbrack, R. L. (2003) PISCES: A Protein Sequence Culling Server in Bioinformatics pp. 1589-1591, Oxford Univ Press.

[12] Chen, B., Tai, P. C., Harrison, R & Pan, Y.(2006) FGK Model : An Efficient Granular Computing Model for Protein Sequence Motifs Information Discovery, Iasted Proc. International Conference on Computational and Systems Biology (CASB), Dallas.

[13] Sader, C. & Schneider, R. (1991) Database of Homology-Derived Protein Structures and the Structural Meaning of Sequence Alignment, Proteins: Structure, Function & Genetics. 9, 56-68.

[14] Kabsch, W. & Sander, C. (1983) Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features, Biopolymers. 22, 2577-2637.

[15] Zhong W., Altun G., Harrison R., Tai P. C. and Pan YI, (2005) Improved Kmeans Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property, IEEE Trans. On Nanobioscience, Vol 4, pp. 255-265.