Application of Modified General Regression Model to Cluster Protein Sequences

G Lavanya Devi[†], Allam Appa Rao[††], A Damodaram[†††], GR Sridhar[††††], G Jaya Suma[†]

[†] Gitam Institute of Technology, GITAM University, Andhra Pradesh, India
[††] College of Engineering, Andhra University, Andhra Pradesh, India
[†††] Jawaharlal Nehru Technological University, Andhra Pradesh, India
[††††]Endocrine and Diabetes Centre, Andhra Pradesh, India

Summary
Cluster analysis is the study of techniques for finding the most representative cluster prototypes. Linear relation of two sequences can be modeled perfectly through the classical linear regression model. Protein sequence clustering has many applications such as helps in classifying a new sequence, predicting the protein structure of unknown sequence and finding the family and subfamily relationships of protein sequences. To cluster a repository of protein sequences into groups where sequences have strong linear relationship with each other, it is prohibitively expensive to compare sequences one by one. In this paper, we have proposed a new technique named General Regression Model Technique (GRMT1) to test the linearity of the sequences. Later we have applied General Regression Model Technique Clustering Algorithm (GRMTCA) to cluster the protein sequences. The performance of the algorithm was evaluated with 50 protein sequences. We used BLAST to annotate the clusters obtained by GRMTCA. It is observed that the clusters have biological significance.

Key words:
Clustering, BLAST, General Regression Model, Protein Sequences

1 Introduction

Cluster analysis has become the utility for several practical problems in various fields such as biology, information retrieval, weather forecasting, psychology, medicine and business where the data size is very large. Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The objects within a group are similar to one another and different from the objects in other groups. The greater the similarity within a group and the greater the difference between groups, the better or more the distinct is the clustering. Clustering is mainly used for dimensionality reduction, prototype selection, or abstraction for pattern classification, data reorganization and indexing and for detecting outliers and noisy patterns. There are many types of clustering techniques namely hierarchical clustering, partitional clustering, exclusive clustering, non-exclusive clustering, and fuzzy clustering[1,2].Clustering is an active research topic in pattern reorganization, data mining, statistics and machine learning with diverse prominence.

1.1 Importance of protein sequence clustering

Proteins are large organic compounds made of the 20 amino acids arranged in a linear chain [3] and joined together by peptide bonds between the carboxyl and amino groups of adjacent amino acid residues. The sequence of amino acids in a protein is defined by a gene and encoded in the genetic code. Protein sequences have a remarkable ability to reproducibly fold into a three dimensional shape and this shape confers them to the ability to form a variety of critical for life: enzymatic catalysis, structural support, generation of motion, reception of signals between cells, and transduction of forces into chemical signals, to name a few [4]. Molecular biology has undergone an incredibly rapid development, currently yielding huge amounts of raw data that efficient computer algorithms are mandatory for data analysis. The number of unique entries in all protein sequence databases together exceeds now more than half a million. However biological evolution lets proteins fall into so called families, thus imposing a natural grouping. A protein family contains sequences that are evolutionarily related and or share a common three dimensional fold. Similar protein sequences probably have similar biochemical function and three dimensional structures. Protein sequence clustering helps in classifying a new sequence, retrieve a set of similar sequences for a given query sequence, predicting the protein structure of unknown sequence and finding the family and subfamily relationships of protein sequences.
There are various algorithms available for aligning sequences, to cluster sequences etc., [5].

1.2 Brief review of BLAST

The BLAST is (Basic Local Alignment Search Tool) is a local similarity search method that concentrates on finding short identical matches, which may contribute to total match [6].

1.3 Similarity Measures

Sequence analysis has drawn a lot of research interests with a vast range of applications. The basic research problems in this field [7] - [14], [29, 30], are matching, sub-matching, indexing, clustering, rule discovery, etc. The focal point is how to define and measure similarity. Currently, there are several popular models used to define and measure (dis)similarity of two sequences. These methods can be classified into four main categories: Lp norms [7, 8], Transforms [9, 27, 28], Time Warping [24, 25, 26], Linear relation [15,16, 22, 23].

Lp norms as measure of (dis)similarity cannot capture similarity in the case of shifting and scaling. It is known that the mean-deviation normalization can discard the shifting and scaling factors.

The transforms are used actually for feature extraction. However, after features are extracted, some type of measure is unavoidable. If Lp norm distance is used, it inherits the drawback stated above.

Time warping has a great advantage that it can tolerate some local non-alignment of time phrase so that the two sequences do not have to be of the same length. It is more robust and flexible than Lp norms. But it is also sensitive to shifting and scaling. And the warping distance only has relative meaning, just like the Lp norms.

Linear relation though is invariant to shifting and scaling, the distance still only has relative meaning [14].

In this paper, we propose a new model, named GRMT (General Regression Model Technique) to measure the degree of the linear relation of multiple sequences at one time. In addition, we have applied the technique to a sample set of 50 sequences.

The organization of this paper is as follows: Section1 is introduction; Section 2 provides overview of regression model. Section 3 describes GRMT in detail and section 4 discusses the experimental results of GRMT clustering algorithm. Finally section 5 draws the conclusions. Section 6 is appendix.

2 Overview of Regression Model

Linear regression analysis originated from statistics and has been widely used in econometrics [33, 34]. For an instance, to test the linear relation between consumption Y and incoming X, we can establish the linear model as:

$$Y = \beta_0 + \beta_1 X + u \quad (1)$$

The variable u is called the *error term*. The regression as (1) is termed as "*the regression of Y on X*". Given a set of sample data, X = [x_1, x_2,…., x_N]and Y=[y_1, y_2,...,y_N], β_0 and β_1 can be estimated in the sense of minimum-sum-of-squared-error. That is, we try to find a line, called regression line, in the Y –X space, to fit the points (x_1, y_1), (x_2, y_2),…, (x_N, y_N) as well as possible. We need to determine β_0 and β_1 such that $\sum_{i=1}^{N} u_i^2 = \sum_{i=1}^{N}$ (y_i-β_0-$\beta_1 x_i$) is minimized.

Using first order conditions [27, 28], we can solve β_0 and β_1 as follows:

$$\beta_0 = \sum_{i=1}^{N} \bar{y} - \beta_1 \bar{x} \quad (2)$$

$$\beta_1 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})\left(y_i - \bar{y}\right)}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \quad (3)$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\frac{1}{N}\sum_{i=1}^{N} y_i$, the average of sequence Y and X respectively.

After obtaining β_0 and β_1, we have to measure how well the regression line fits these data. To resolve this, the *R-star* (R*) is defined as:

$$R^* = 1 - \frac{\sum_{i=1}^{N} u_i^2}{\sum_{i=1}^{N} \left(y_i - \bar{y}\right)^2} \quad (4)$$

The value of R^* is always between 0 and 1. The closer the value is to 1, the better the regression line fits the data points. R^* is the measure for the *Goodness-of-Fit* in the traditional regression. The regression model as (1) is called *Classical Regression Model*. It involves only one independent variable X and one dependent variable Y. More independent variables can be added to the model as shown below:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots. + \beta_K X_K + u \quad (5)$$

This is called *Multiple Regression Model*. β_0, β_1,..... , β_K can be estimated similarly using first order conditions.

3 Generalized Regression Model

3.1 Limitations of Classical Regression Model

We observed that the Classical Regression Model is excellent in testing the linear relation of two sequences. R^* is a good measure for linear relation. For an instance, $R^*(X_1, X2) = 0.95$ is statistically strong evidence that the two sequences are highly linear related to each other, thus they are very similar. We do not have to compare $R^*(X_1,X_2) > R^*(X_1,X_3)$ and say X_1 is similar to X_2 rather than X_3. Therefore, the meaning of R^* for similarity is not relative, unlike distance-based measures.

When we need to test only two sequences, the Classical Regression Model is suitable. However, when more than two sequences are involved in some applications such as clustering, the Classical Regression Model has to run regression between each pair of sequences. The performance cannot be efficient. We cannot apply R^* in the multiple regression model to test whether multiple sequences are similar to each other or not, because it only means the linear relation between Y and the *linear combination* of $X_1,X_2,\ldots,X_K$. Furthermore, R^* in the multiple regression is sensitive to the order of sequences. If we randomly choose X_i to substitute Y as dependent variable and let Y be independent variable, then the regression becomes

$X_i = \beta_0 + \beta_1 X_1 + \ldots\ldots + \beta_i Y + \ldots\ldots + \beta_K X_K + u$. The R^* here will be different from that of (5), because they have different meanings.

From a geometrical point of view, equation (5) describes a hyper-plane instead of a line in $(K+1)$-dimensional space. To test the similarity among multiple sequences, we need a line in the space instead of a hyper-plane. [17, 18, 19, 20, 21].

Generalizing the idea of Classical Regression Model to multiple sequences, we propose the General Regression Model Technique (GRMT).

3.2 GRMT : Generalized Regression Model Technique

Given K ($K \geq 2$) sequences $X_1,X_2,\ldots\ldots,X_K$ and

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots x_{1N} \\ x_{21} & x_{22} & \cdots x_{2N} \\ \vdots & \vdots & \vdots \\ x_{K1} & x_{K2} & \cdots x_{KN} \end{pmatrix}$$

Initially we organize them into N points in the K dimensional space:

In the traditional regression, the error term is defined as:

$u_i = y_i - (\beta_0 + \beta_1 x_{1i} + \ldots\ldots + \beta_K x_{Ki})$ (6)

It is the distance between y_i and the regression hyper-plane in direction of axis Y. This makes sequence Y unique from any X_i (i = 1, 2, … , K). In GRMT, we define the error term u_i as the *vertical* distance from point $(x_{1i}, x_{2i}, \ldots\ldots , x_{Ki})$ to the regression line. It is to be noted that there is no Y here anymore, because no sequence is special among its community. To ensure that the regression line exists uniquely, we need to adhere to the hypothesis below:

• *Hypothesis1.* No sequence is constant. It guarantees the scatter matrix has eigenvector.

• *Hypothesis2. N* points determine a line uniquely.

In real applications, it is highly unlikely that a random sequence is constant or all K sequences are exactly the same. Therefore, the assumptions will not limit the applications of GRMT. Similar to the traditional regression, after determining the regression line, we need a measure for Goodness-of-Fit. We define:

$$GR^* = 1 - \frac{\sum_{i=1}^{N} u_i^2}{\sum_{j=1}^{K}\sum_{i=1}^{N}\left(x_{ji} - \overline{x_j}\right)^2} \qquad (7)$$

If the value of GR^* is close to1,the K sequences have a high degree of linear relationship with each other.

3.3 Application of GRMT

The procedure of applying GRMT to measure the linear relation of multiple sequences is described by algorithm GRMT1.

GRMT1: Testing linearity of multiple sequences

• Organize the given K sequences with length N into N points $p_1, p_2, \ldots, p_K$ in K-dimensional space as shown in section3.2.

• Determine the regression line. First, calculate the average $m = \frac{1}{N}\sum_{i=1}^{N} p_i$ calculate the scatter matrix $S = \sum_{i=1}^{N} (p_i - m)(p_i - m)^t$.

Then, determine the maximum eigen value λ and corresponding eigenvector e of S.

• Calculate GR^* according to property shown in #6 Appendix.

• Draw conclusion. For instance if we only accept linearity with confidence no less than C (say, C = 95%). If $GR^* \geq C$, we can conclude that the K sequences are linear to each other with confidence GR^*.

3.4 Apply GRMT1 to cluster massive sequences

When hundreds or thousands of random sequences are tested by algorithm GRMT1, one can foresee that GR^* cannot be close to 1 before really calculating it, because hundreds or thousands of random sequences are highly unlikely to be linear to each other. The significance of GRMT1 in testing massive sequences is can make use of it to obtain heuristic information for clustering sequences.

Given a set of sequences $S = \{X_i \mid i = 1, 2,\ldots,K\}$,algorithm GRMTCA (General Regression Model Technique Clustering Algorithm) works as follows:

GRMTCA: Clustering of massive sequences

• Apply Algorithm GRMT1 to test whether the given sequences are linear to each other or not. If yes, all the sequences can go into one cluster and we can stop, otherwise, go to next step.

• After GRMT1, we have eigenvector $[e_1, e_2,\ldots. , e_K]t$. Create a feature value sequence $F =(\sigma(X_1)/e_1, \sigma(X_2)/e_2,\ldots. , \sigma(X_K)/eK)$ and sort it in increasing order. After sorting, suppose $F = (f_1, f_2,\ldots..,f_K)$.

• Start from the first feature value $f1$ in F. Suppose the corresponding sequence is Xi. We only check the linearity of Xi with the sequences whose feature values in F are close to $f1$. Here"close" means $fj/f1 \leq \xi$ (According to our experience, ξ =0.95 is enough).We collect those sequences which have linearity with Xi with confidence $\geq C$ into cluster $CM1$. Delete all the sequences in this cluster from set S, then repeat the similar procedure to obtain next cluster until S becomes empty. The most time-consuming part in GRMT1 and GRMTCA is to calculate the maximum eigen value and corresponding eigen vector of scatter matrix S. Fast algorithm [31, 32] can do it with high efficiency.

4 Experiments

Our experiment primarily focuses on to test whether GR^* is a good measure for testing the linearity. Secondly we would like to check the accuracy of GRMTCA.

4.1 Normalization of protein data

Proteins are strings of combination of the twenty amino acids. Each of the amino acid is given a random weight. Also all the N sequences that are to be clustered may not have the same length. The sequence that has maximum length X_m is considered and all other (N-1) sequences are to be padded with a neutral value. Truncating the sequences to a fixed length may lead to loss of useful information. The procedure followed above prevents us from loosing such information.

4.2 Sample data

The protein sequences are retrieved through Protein Protein Blast in NCBI (National Center for Biotechnology Information) [35]. We have chosen AAP59031: BchE [Thiocapsa roseopersicina] as our query. The length of the sequence is 551.With BLAST tool we obtained the sequences that are similar to the query sequence. This reverse process was adopted to test our results obtained by applying GRMTCA. We have considered BLAST as measure of scale to our experiments as it is widely used tool to find similarities. For the test case we considered the data set given in Table1. By applying GRMT1 we have obtained the GR* value 0.3263. Maximum eigen value λ is obtained from the 50×50 scatter matrix S. The eigen vectors for the maximum λ value are shown in Eigen Vectors column of Table1. The feature values are shown in Feature Values column. The clusters generated vary each time GRMTCA is executed, if the weights to the amino acids are given through random number generation. We have other choice which allows us to give chosen values to amino acids. We have chosen the former one. We have performed 75 iterations and observed that the 50 protein sequences were divided into 12 clusters. Table1 shows one iteration values. Due to the limitation of space, the values obtained for other iterations are not shown.

Our programs were written using MATLAB 6.5.

Table1. Data set of the experiments

Sr.No	*Protein ID*	Eigen Vectors	Feature Values
1	AAP59031	0.16352	9.1601
2	ABQ89351	0.097928	9.1078
3	YP_001431695	0.10456	7.8317
4	AAC84027	0.09501	7.531
5	ZP_01516182	0.095265	7.415
6	YP_001637247	0.11596	8.3333
7	AAG15204	0.12201	8.0441
8	YP_001679875	0.12215	8.1454
9	YP_374182	0.10601	8.2271
10	ZP_00512932	0.084266	7.1503
11	YP_001129840	0.14673	9.0866
12	YP_912730	0.13929	9.4771
13	YP_378550	0.10339	9.4232
14	ZP_01386524	0.078863	9.6961
15	ZP_00591039	0.079834	9.6536

16	ZP_00588879	0.14874	8.8023
Sr.No	*Protein ID*	Eigen Vectors	Feature Values
17	NP_662836	0.1323	9.3513
18	YP_001203767	0.059418	9.4232
19	YP_001242220	0.096307	9.3938
20	Q7X2C7	0.10937	9.3415
21	YP_428629	0.042848	9.5539
22	YP_001235395	0.10313	9.2108
23	YP_001003191	0.11534	9.2745
24	YP_001770400	0.14939	9.1912
25	ZP_01037514	0.078621	9.4297
26	YP_533665	0.15264	9.1422
27	ZP_01878043	0.16445	8.9771
28	YP_782841	0.1387	9.2222
29	NP_947014	0.17551	8.7288
30	YP_487467	0.16014	8.6111
31	YP_568625	0.14481	8.951
32	YP_001167220	0.17503	8.9118
33	ZP_02303776	0.17527	8.9118
34	P26168	0.17574	9.2255
35	YP_001533971	0.17681	9.1062
36	YP_353355.1	0.17768	9.268
37	AAF24279	0.17674	9.116
38	CAB38729	0.16926	9.2026
39	ZP_01902748	0.1762	9.2271
40	ZP_01387586	0.17417	9.1683
41	YP_383565	0.17464	9.183
42	ZP_01389153	0.17676	9.1536
43	NP_953930	0.16005	9.1422
44	YP_001232341	0.16289	9.0621
45	YP_001471340	0.16289	9.0719
46	ZP_01594925	0.14416	9.384
47	YP_384134.1	0.13705	9.5392
48	YP_965803	0.17322	8.7353
49	YP_375565	0.17346	8.7631
50	YP_902300	0.16352	9.1601

5 Conclusion

We have proposed GRMT by generalizing the Classical Regression Model. GRMT gives a measure GR^*, which is a novel measure for linearity of multiple sequences. The meaning of GR^* for linearity is not relative. Based on GR^*, algorithm GRMT1 can test the linearity of multiple sequences at a time and GRMTCA can cluster massive sequences with high accuracy as well as high efficiency.

6 Appendix

Properties of GR^*

1. $GR^* = \dfrac{\lambda}{\sum_{i=1}^{N} \left\| p_i - m \right\|^2}$ and $1 \geq GR^* > 0$

2. $GR^* = 1$ means the K sequences have eact linear relationship to each other.
3. GR^* is invariant to the order of $X_1, X_2, \ldots\ldots X_K$, i.e., we can arbitrarily change the order of the K sequences and the value of GR^* does not change.

Acknowledgment

The authors would like to express their cordial thanks to Dr. A.Chandra Sekhar, Department of Mathematics, Gitam Institute of Technology, Gitam University, Visakhapatnam, India, for his valuable advice.

References

[1] M.R. Anderberg. *Cluster Analysis for Applications*. Academic Press, New York, December 1973.
[2] J.Han, M.Kamber, and A.Tung. Spatial Clustering Methods in Data Mining: A review. In H.J. Miller and J.Han, editors, *Geographic Data Mining and Knowledge Discovery*, pages 188-217. Taylor and Francis, London, December 2001.
[3] T.K.Atwood and D.J.Parry-Smith. *Introduction to Bioinformatics*. Addison Wesley Longman, 1999.
[4] Zhang Y, Skolnick J. (2005). The protein structure prediction problem could be solved using the current PDB library. *Proc Natl Acad Sci USA* 102(4):1029-34.
[5] Gusfield D. *Algorithms on Strings, Trees and Sequences*. New York: Cambridge University Press, 1997.
[6] Altschul,S.F., Gish., W., Miller, W., Myers, E.W. and Lipman, DJ.(1990) Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215, 403-410.
[7] R. Agrawal, C. Faloutsos and A. Swami, *Efficient Similarity Search in Sequence Databases*, Proceedings of the 4th Intl. Conf. on Foundations of Data Organizations and Algorithms (FODO) (1993), pp. 69–84.
[8] B. Yi and C. Faloutsos, *Fast Time Sequence Indexing for Arbitrary Lp Norms*, The 26th International Conference on Very Large Databases(VLDB) (2000), pp. 385–394.
[9] D. Rafiei and A. Mendelzon, *Efficient Retrieval of Similar Time Sequences Using DFT*, Proceedings of the 5th International Conference on Foundations of Data Organizations and Algorithms (FODO) (1998), pp. 69–84.
[10] R. Agrawal, K. I. Lin, H. S. Sawhne and K. Shim, *Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases*, Proc. of the 2Ist VLDB Conference(1995), pp. 490–501.

[11] T. Bozkaya, N. Yazdani and Z.M. Ozsoyoglu, *Matching and Indexing Sequences of Different Lengths*, Proc. Of the 6th International Conference on Information and Knowledge Management(1997), pp. 128–135.
[12] E. Keogh, *A fast and robust method for pattern matching in sequences database*, WUSS(1997).
[13] E. Keogh and P. Smyth. *A Probabilistic Approach to Fast Pattern Matching in Sequences Databases*, The 3rd Intl. Conf. on Knowledge Discovery and DataMining(1997), pp. 24–30.
[14] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, *Fast Subsequence Matching in Time-Series Databases*, International Proceedings of the ACM SIGMOD Conference on management of Data(1994), pp. 419–429.
[15] C. Chung, S. Lee, S. Chun, D. Kim and J. Lee, *Similarity Search for Multidimensional Data Sequences*, Proceedings of the 16th International Conf. on Data Engineering(2000), pp. 599–608.
[16] D. Goldin and P. Kanellakis, *On similarity queries for time-series data: constraint specification and implementation*,
The 1st International Conference on the Principles and practice of Constraint Programming (1995), pp. 137–153.
[17] C. Perng, H. Wang, S. Zhang and D. Parker, *Landmarks: a New Model for Similarity-based Pattern Querying in Sequences Databases*, Proc. of the 16th
International Conference on Data Engineering (2000)
[18] H. Jagadish, A. Mendelzon and T. Milo, *Similarity-Based Queries*, The Symposium on Principles of Database Systems(1995), pp. 36–45.
[19] D. Rafiei and A. Mendelzon, *Similarity-Based Queries for Sequences Data*, Proc. of the ACM SIGMOD Conference on Management of Data(1997), pp. 13–25.
[20] C. Li, P. Yu and V. Castelli, *Similarity Search Algorithm for Databases of Long Sequences*, The 12th International Conference on Data Engineering (1996), pp. 546–553.
[21] G. Das, D. Gunopulos and H. Mannila, *Finding similar sequences*, The 1st European Symposium on Principles
of Data Mining and Knowledge Discovery(1997),pp. 88–100.
[22] K. Chu and M. Wong, *Fast Time-Series Searching with Scaling and Shifting*, The 18th ACM Symp. On Principles of Database Systems (PODS 1999), pp. 237–248.
[23] B. Bollobas, G. Das, D. Gunopulos and H. Mannila, *Time-Series Similarity Problems and Well-Separated Geometric Sets*, The 13th Annual ACM Symposium
on Computational Geometry(1997), pp. 454–456.
[24] D. Berndt and J. Clifford, *Using Dynamic Time Warping to Find Patterns in Sequences*, Working Notes of the Knowledge Discovery in Databases Workshop(1994), pp. 359–370.
[25] B. Yi, H. Jagadish and C. Faloutsos, *Efficient Retrieval of Similar Time Sequences Under Time Warping*, Proc.
of the 14th International Conference on Data Engineering(1998), pp. 23–27.
[26] S. Park, W. Chu, J. Yoon and C. Hsu, *Efficient Similarity Searches for Time-Warped Subsequences in Sequence Databases*, Proc. of the 16th International
Conf. on Data Engineering (2000).
[27] Z. Struzik and A. Siebes, *The Haar Wavelet Transform in the Sequences Similarity Paradigm*, PKDD(1999).
[28] K. Chan and W. FU. *Efficient Sequences Matching by Wavelets*, The 15th international Conf. on Data Engineering(1999).
[29] G. Das, K. Lin, H. Mannila, G. Renganathan and P. Smyt, *Rule Discovery from Sequences*, Knowledge Discovery and Data Mining(1998), pp. 16–22.
[30] G. Das, D. Gunopulos, *Sequences Similarity Measures*, KDD-2000: Sequences Tutorial.
[31] I. Dhillon, *A New O(n2) Algorithm for the Symetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. Thesis. University of. California, Berkerley, 1997.
[32] R. Duda, P. Hart and D. Stork, *Pattern Classification. 2nd Edition*, John Wiley & Sons, 2000.
[33] J. Wooldridge, *Introductory Econometrics: a modern approach*, South-Western College Publishing, 1999.
[34] F. Mosteller and J. Tukey, *Data Analysis and Regression: A Second Course in Statistics*, Addison-Wesley, 1977.
[35] www.ncbi.nlm.nih.gov

Lavanya Devi Golagani received B.Tech degree from Nagarjuna University in 2000. She received M.Tech degree from Andhra University in 2002. After working as assistant professor from 2002 to 2005, she has registered at Jawaharlal Nehru Technological University for research. Currently she is working as Senior Assistant Professor in the Department of Computer Science and Engineering, Gitam Institute of Technology, GITAM University, Visakhapatnam, Andhra Pradesh, India. Her research interest includes Network Security, Data Mining, Bioinformatics.
www.gitam.edu

Dr. Allam Appa Rao has received PhD in Computer Engineering from Andhra University, Visakhapatnam, Andhra Pradesh, India. Currently he is the Professor in Bioinformatics & Computational Biology,Department of Computer Science and Systems Engineering &Principal, Andhra University College of Engineering (AUTONOMOUS). His research interest includes Bioinformatics, Software Engineering and Network Security. He is a member of professional societies like IEEE, ACM and a life member of CSI **and** ISTE
www.allamapparao.net

Dr A Damodaram has PhD in Computer Science & Engineering. He is Professor in the Department of Computer Science and Engineering Jawaharlal Nehru Technological University, Hyderabad, and Vice Principal of Jawaharlal Nehru Technological University, Hyderabad. His research interest includes Computer Networks Software Engineering and Network Security.

Dr G R Sridhar, an endocrinologist, is Adjunct Professor, Bioinformatics, Andhra University College of Engineering. He was Chairman, Scientific Committee Annual Conference of RSSDI (2005).He is currently Chairman, Indian Chapter, American Association of Clinical Endocrinologists (2005-7). Dr Sridhar was the founder Editor, Indian Journal of Endocrinology and Metabolism, (1997-2000),' Widely published, he contributed chapters to 'RSSDI Textbook of Diabetes' and to 'API Textbook of Medicine. A fellow of Madras Science Foundation, he was honored with RSSDI Oration, 2007, the Hoechst Senior lecturer ship in diabetes (2002) and Boehringer Knoll lecturer ship in Diabetes (1997). Dr Sridhar's major areas of research interest are in Clinical informatics, computational biology and bioinformatics, psychosocial aspects of diabetes.
www.diabetes.org.in

Jaya Suma G received B.Tech degree from Andhra University in 1998. She received M.Tech degree from Andhra University in 2002. After working as assistant professor from 2002 to 2005, she has registered at Andhra University for research. Currently she is working as Senior Assistant Professor in the Department of Computer Science and Engineering, Gitam Institute of Technology, GITAM University, Visakhapatnam, Andhra Pradesh, India. Her research interest includes Data Mining, Network Security, Artificial Intelligence and Machine Learning.
www.gitam.edu