Prediction of Extracellular Matrix Proteins using SVM^{hmm} Classifier

Anitha Jose[1], Rejimoan R
Department of Computer Science and Engineering
SreeChitraTirunal College of Engineering, Kerala
India

Sivakumar KC
Bioinformatics facility,
Rajiv Gandhi Centre for Biotechnology,
Thiruvanathapuram
Kerala, India

Sathish Mundayoor
Bioinformatics facility,
Rajiv Gandhi Centre for Biotechnology,
Thiruvanathapuram
Kerala, India

ABSTRACT

Extracellular matrix (ECM) proteins are those secreted to the exterior of the cell, which act as mediators between resident cells and the external environment. These proteins not only maintain cellular structure but also play a part in diverse processes, including growth, hormonal response, homeostasis, and disease progression. Regardless of their importance, current knowledge of the number and functions of ECM proteins is limited. Deregulation of ECM proteins may cause diseases, including developmental abnormalities and cancer. Recent studies say that some of these proteins in body fluid may be considered as disease specific markers. Therefore, identification of ECM proteins is a significant step in understanding cancer progression and providing effective therapeutic targets or diagnostic markers. Despite their importance, current knowledge of the number and functions of ECM proteins is limited. Experimental identification of ECM is labor as-well-as time intensive. Here, propose a computational and novel method to predict ECM proteins. The dataset used in this study for training and testing was obtained from Uniprot database. Specific features such as PSI-BLAST derived PSSM and amino acid composition were utilized for training the models. Based on this newly generated features and the discriminatory characteristics of Hidden Markov Model were determined, which significantly improved the performance of SVM^{hmm} classification.

Keywords

extracellular matrix protein, amino acid composition, pssm, svm^{hmm}.

1. INTRODUCTION

Extracellular matrix (ECM) proteins represent a class of secreted proteins, and gather as a massive network on the cell surface. They are secreted proteins are exported to the outside of the cell membrane, and participate in communication with neighboring cells [1][13]. These proteins support the cell surface microenvironment, and additionally influence critical cell behavior, such as proliferation, survival, and differentiation. Consequently, dysregulation of ECM proteins may cause diseases, including developmental abnormalities and cancer [2][13]. Recent studies report the presence of some of these proteins in body fluid, suggesting they may have practical applications as disease-specific markers [3][13]. Therefore, identification of ECM proteins is a significant step in understanding cancer progression and providing effective therapeutic targets or diagnostic markers. A large proportion of eukaryotic protein localization has not been annotated experimentally [4]. Because experimental verification is labor-intensive and time-consuming, several computational programs have been developed to predict ECM.

With the development of genome projects, the amount of sequence data increases in an astonishing speed, and a lot of data have been accumulated. To narrow the huge gap between the enormous amount of raw sequence data and the experimental characterization of the corresponding proteins, people therefore have to find computational ways to efficiently analyze these data. Despite their importance, current knowledge of the number and functions of ECM proteins is limited. The cost, time and the incumbent limitations of experimental methods, coupled with the tremendous biological significance and mounting interest in these proteins have motivated attempts to develop computational tool to identify ECM. Compared with experimental methods, computational prediction methods that can provide fast, automatic and accurate assignment of protein subcellular location is very desirable, especially for high-through analysis of large-scale genome sequences. Many computational methods have been developed for the prediction of the subcellular location of proteins, and most of them are working as online web servers which can be accessed from Internet. Here, we propose a computational method to make a predict ECM proteins using the PSI-BLAST Position Specific Scoring Matrix as well as the discriminative features of ECM domains and repeats as input features for SVM^{hmm}.

2. MATERIALS AND METHODS

2.1 Dataset

The input data set is in FASTA format which begins with a single line description, followed by lines of sequence of data. FASTA format is a text based format for representing the biological sequences, in which nucleotides or amino acids are represented using single letter codes. This format also allows for sequence names and comments to precede the sequence and this now become a standard in the field of bioinformatics. The Universal Protein Resource (Uniprot) is a central resource for protein sequences and functional information. The data set collected from UniProt KB (http://www.uniprot.org), for the functional information on proteins, are accurate, consistent and with rich annotation. In addition to, it explains the amino acid sequence, protein name or description, taxonomic data and citation and all information added .The protein information derived from genome sequencing projects and contains large amount of information which are collected from research literature as well. In this work 325 extra cellular matrix proteins were collected from Uniprot KB and positive dataset. Similarly negative dataset consist of 325 nitrogen fixing proteins (nifu) proteins collected from Uniprot KB. Nifu proteins are nitrogen fixation bacteria proteins.

2.2 PSSM (Position Specific Scoring Matrices)

PSI-BLAST (Position Specific Iterative –BLAST) derives a Position Specific Scoring Matrix (PSSM) or profile from the multiple sequence alignment of sequences detected above a given score threshold using protein-protein BLAST. This PSSM is used to further search the database for new matches, and is updated for subsequent iterations with these newly detected sequences. Thus PSI-BLAST provides a means of detecting distant relationships between proteins. PSI-BLAST is one of the most powerful and popular homology search programs currently available. The position specific scoring matrices or profile it is used in Protein Blast and obtained amino acid substitution scores which is given separately for each position in a protein multiple sequence alignment. Alignment means extract a segment from each sequence, if sequence length is smaller than the other then add gap symbols to each segment to create equal length sequence and place one padded segment over the other. The PSSM is used to get numerical value, if the numerical value is high from the previous one then that is the better alignment from the previous ones. PSSM scores are normally positive or negative integers. Positive scores indicate that the given amino acid substitution occurs more frequently in the alignment than expected by chance. PSSMs are generated by using PSI-BLAST, which finds similar protein sequences from the query sequences and then construct PSSM from the resulting alignment [5]. The dimensionality of the PSSM is multidimensional data, but here consider only one feature of PSSM.

2.2.1 PSIBLAST Algorithm

1. Perform initial alignment with BLAST using BLOSUM 62 substitution matrix.
2. Construct a multiple alignment from hits.
3. Prepare a position specific scoring matrix (PSSM).
4. Use PSSM profile as the scoring matrix for a second BLAST (run against database).
5. Repeat steps 2-4 until convergence.

2.2.2 Constructing a Position Specific Scoring Matrix (PSSM)

Dimension of a PSSM: lq × 20, where lq is the length of the query protein.

1. Run BLAST against the database (local alignment).
2. Collect database sequence segments with E-value below threshold (default is 0.01).
3. Remove similar sequences.
    - Remove sequence segments identical to a query segment.
    - Retain one copy for any rows that are >98% identical to one another.
4. Construct the multiple alignment block M with the remaining segments (length M = lq)
    - Ignore pair wise alignment columns that involve gap characters inserted into the query.
5. For each column C:
    a. Reduce M to MC (1 · C · query length)
        - Let R be the set of sequences with a residue in C.
        - Columns of MC are columns of M with all sequences in R. In other words, MC only contains those database sequences in R. Therefore, MC contains a subset of M's columns and rows (see the figure below).

    b. Compute weights for each sequence in R
    c. Compute Pi, the background frequency of residue i over MC.
        - Compute weighted frequency fi for each residue i.

    d. Estimate the relative number of independent observations NC as the mean number of different residue including gap characters.
    e. Compute pseudo count gi for each residue i (expectation based on score gi matrix).

$$gi = \sum_j (fi \times pj) \times qij \qquad (2.2.2.1)$$

$$qij = PiPje^{\gamma u Sij} \qquad (2.2.2.2)$$

Where the target frequencies are implicit in the substitution matrix, sij is the substitution matrix score for aligning each pair of amino acids i and j, and ¸u is a constant parameter for ungapped alignments.

f. Compute Qi as the weighted sum of fi and gi.

$$Qi = \frac{\alpha fi + \beta gi}{\alpha + \beta} \qquad (2.2.2.3)$$

$$\alpha = Nc - 1 \qquad (2.2.2.4)$$

B=10(empirically) (2.2.2.5)

2.2.3 Matrices Reported in a PSSM Output File

The PSSM can be saved to a file by using the -Q switch of blastpgp. A PSSM file contains two matrices. The first one is the regular PSSM that contains the log-odds ratios rounded down to the nearest integer. This matrix is the one that is computed in the last PSIBLAST iteration. The second matrix is the weighted observed percentages rounded down to the nearest integer (i.e., 100 × fi values).

2.3 Composition of Position specific Scoring Matrix (PSSM 400)

For better accuracy and to get correct position of amino acid convert the PSSM into PSSM 400 units. In PSSM 400 , 20 amino acids in row and 20 amino acids in column. Each and every element in this vector was divided by the length of sequence. The resultant matrix with 400 elements was used as input feature of SVM^{hmm} . In this work performance can be increased with more metrics using physiochemical properties and other amino acid compositions .But consider these properties we get more reliable results in PSSM and PSSM 400.Therefore,PSSM 400 as input feature of our machine learning technique

2.4 SVM^{hmm}

SVM^{hmm} is an implementation of structural SVMs for sequence tagging using the training algorithm described in and the new algorithm of SVMstruct V3.10 .This program is used for scientific purpose and for complex structures containing interactions between elements. It can easily handle tagging problems with millions of words and millions of features and it can train higher order models with arbitrary length dependencies for both transitions and emissions.

2.4.1 Algorithm and Model

SVM^{hmm} discriminatively trains models that are isomorphic to a kth-order hidden Markov model using the Structural Support Vector Machine (SVM) formulation. In particular, given an observed input sequence $x = (x_1 \dots x_l)$ of feature vectors $x_i \dots x_l$ the model predicts a tag sequence $y = (y_1 \dots y_l)$ according to the following linear discriminate function [6].

$$y = \operatorname{argmax}_y\{\Sigma_{i=1\dots l}\begin{bmatrix} \Sigma_{j=1\dots k}\left(x_i \cdot w_{yi-j\dots yi}\right) \\ +\varphi_{trans}\left(y_{i-j,\dots,}y_i\right) \cdot w_{trans}\end{bmatrix}\}$$

SVM^{hmm} learns one emission weight vector $w_{yi-k\dots yi}$ for each different kth-order tag sequence $y_{i-k\dots}y_i$ and one transition weight vector w_{trans} for the transition weights between adjacent tags. $\varphi_{trans}(y_{i-j,\dots,}y_i)$ is an indicator vector that has exactly one entry set to 1 corresponding to the sequence $y_{i-j,\dots\dots,}y_i$. It is contrast to a conventional HMM; the observations $x_1 \dots x_l$ can naturally be expressed as feature vectors, not just as atomic tokens. Also note that a kth-order model includes all the parameters of the lower-order models as well. During training, SVM^{hmm} solves the following optimization problem given training examples $(x^1,y^1)\dots(x^n,y^n)$) of sequences of feature vectors

$x^j = \left(x_1^j, \dots x_l^j\right)$ with their correct tag sequences

$y^j = (y_{1,\dots\dots}^j y_l^j)$.The following optimization problem corresponds to a model with first-order transitions and zeroth-order emissions, but it should be obvious how it generalizes to higher order models given the discriminate function from above. As the loss function $\Delta\left(y^i,y\right)$ the number of misclassified tags in the sentence is used. The resulting hard margin optimization problem and allow errors in the training set, we introduce slack variables and propose to optimize a soft margin criterion. Several ways to implement slack variables, according to Crammer and Singer introduce one slack variable for every nonlinear constraint which will result in an upper bound of the empirical risk and offers some additional advantages. Adding a penalty term, that is linear in the slack variables to the objective results in the quadratic program [7][8][9].

$$\min 1/2 w * w + \frac{C}{n}\Sigma_{i=1\dots n}\xi i$$

show that for all y:

$$[\Sigma_{i=1\dots l}(x_i^1 \cdot w_{yi}^1 + \varphi_{trans}\left(y_{i-1}^1, y_i^1\right) \cdot w_{trans}] \geq$$

$$[\Sigma_{i=1\dots l}\left(x_i^1 \cdot w_{yi}\right) + \varphi_{trans}(y_{i-1}, y_i) \cdot w_{trans}] + \Delta(y^1,y) - \xi_1$$

…

for all **y**:

$$[\Sigma_{i=1\dots l}\left(x_i^n \cdot w_{yi}^n\right) + \varphi_{trans}(y_{1-1}^n, y_i^n) \cdot w_{trans}] \geq$$

$$[\Sigma_{i=1\dots l}\left(x_i^n \cdot w_{yi}\right) + \varphi_{trans}(y_{i-1}, y_i) \cdot w_{trans}] + \Delta(y^n,y) - \xi_n$$

C is a parameter that trades off margin size and training error or trading off slack vs. magnitude of the weight vector. A good value for C must be selected via cross-validation, ideally exploring values over several orders of magnitude. C >0 is a constant that controls the trade-off between training error minimization and margin maximization. Epsilon (e) specifies the precision to which constraints are required to be satisfied by the solution. The smaller EPSILON, longer and more memory training takes, but the solution is more precise. However, solutions more accurate than 0.5, typically do not improve prediction accuracy.

2.4.2 Input File Format

AG qid: EXNUM FEATNUM: FEATVAL EATNUM: FEATVAL...

2.5 Leave-one-out cross validation (LOO-CV)

This is deemed as the most objective and rigorous mode of evaluation wherein one dataset sequence is singled out for testing, while the rest are used to generate the model. This iterates on each sequence till each sequence becomes the testing data exactly once. This is a stringent case of k-fold cross-validation where k equals the total number of sequences. The best parameters (λ and C) as measured by the various performance measures are taken and then averaged to get overall assessment of the model. Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against. The basic form of cross-validation is k-fold cross-validation. Other forms of cross-validation are special cases of k-fold cross-validation or involve repeated rounds of k-fold cross-validation .In our work around 320 extra cellular matrix proteins are taken as positive data set. Similarly 325 nifu sequences are taken as negative data set.

2.5.1 Procedure for cross validation

1. Make Positive and Negative datasets in two files. eg- N number sequences for positive and N number for negative sequences.
2. Combine these two file in two a single file. e.g. - N+N=2N

The sequences in dataset should have minimum sequence similarity but it decrease size of dataset significantly. A confusion matrix was employed to quantify the efficiency of classification between ECM proteins and non-ECM proteins using TP (True positive- known and predicted ECM proteins), TN (True negative- known and predicted non-ECM proteins) FN (known ECM proteins and predicted non-ECM proteins), and FP (False positive-known non-ECM proteins and predicted ECM proteins).

Two types of parameters, Threshold dependent and Threshold independent

In threshold dependent case includes

$$\text{Sensitivity} = \frac{TP}{TP+FN} \times 100$$

$$\text{specificity} = \frac{TN}{TN+FP} \times 100$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

These are used for evaluating model performance.

3. RESULTS

3.1 Performance of similarity-based searches

Position-Specific Iterative-Basic Local Alignment Search Tool (PSI-BLAST) is usually the first method of choice for the functional annotation of proteins. We carried out the PSI-BLAST analysis on the non-redundant positive dataset of ECM proteins in a manner like leave-one-out cross-validation (LOO CV), with the cut-off E-value (-e option of blastpgp) of 0.001 and the number of iterations as 3. Each sequence was used as the query sequence once with the rest forming the target database, thus iterating, for each sequence. Herein, no significant hits were obtained for 125 out of 300 sequences, which signify that homology-based searches alone are not sufficient to identify these proteins.

3.2 Performance of SVM^{hmm} model

We performed LOO CV of Position-Specific Scoring Matrix (PSSM) based classifier, trained using the SVM^{hmm} function kernel. Table 1 depicts the performance of the SVM^{hmm} classifier as observed in the LOO CV. With SVM^{hmm} classifier we obtained an accuracy of 95.65% % with PSI-BLAST PSSM based model. It is clear that such an sequence composition based models can provide sufficient information to for good discrimination between ECM and non-ECM proteins. Apart from encapsulating residue composition, the PSSM profiles capture useful information about conservation of residues at crucial positions within the protein sequence, because in evolution the amino acid residues with similar physico-chemical properties tend to be highly conserved due to selective pressure. PSSM profiles have been employed for training SVMs for a legion of classification problems, like prediction of cyclins [10], nucleic acid binding residues [11], protein subcellular localization [12] etc.

4. DISCUSSION

We developed a SVM^{hmm} based model using PSSM profiles to facilitate the identification of ECM proteins.. Firstly, we proposed a novel SVM^{hmm} classifier which incorporates the strength of SVM and HMM algorithms in the characterization of ECM proteins, which proved remarkably useful for the classification process. Secondly, we built a highly accurate classifier for ECM proteins using SVM^{hmm} and a reduced feature set. Consequently, 12 candidate ECM proteins were predicted using our classification engine (Table 2). ECM maintains cell shape and controls the communication with the environment. While most cellular proteins are concealed by the lipid bilayer membrane, ECM proteins are displayed on the surface, and often serve as specific markers for cell status or therapeutic targets. However, the number of annotated ECM proteins is too limited at present for further analysis. In this study, we attempted to predict ECM proteins in extracellular space. The identification of ECM proteins should be helpful in the analysis of ECM-related function and disease.

Table -1 Performance of SVM^{hmm} classifier in LOO-CV

Performance Parameter(c=10 e=.1)	SVM^{hmm}
Accuracy	95.65%
Sensitivity	100%
Specificity	92%
λ	0.001
MCC	91.65%

Table -2 Unknown ECM in Swiss-Prot were analyzed with our method. Using SVM^{hmm} classifier with PSSM features, 12 ECM protein candidates were obtained with a cut-off value of 0.5.

Uniprot Accession No:	Gene Name	ECM score
P01871	IGHM	0.96
Q3MIW9	DPCR1	0.93
P01857	IGHG1	0.91
Q7ZUA6	LMBR1	0.90
Q9UM47	NOTCH3	0. 84
P06312	IGKV4-1	0.80
Q8BPB5	Efemp1	0.78
Q9H8J5	MANSC1	0.65
Q7RTP0	NIPA1	0.62
Q9GZY6	LAT2	0.62
O14493	CLDN4	0.52
P10039	TNC	0.50

5. ACKNOWLEDGMENTS

The authors acknowledge BTISNet, Department of Biotechnology, Government of India for the Bioinformatics Facility..

6. REFERENCES

[1] J.M. Jacobs, K.M. Waters, L.E. Kathmann, D.G. Camp, H.S. Wiley, R.D. Smith, and B.D. Thrall, 2008, The mammary epithelial cell secretome and its regulation bysignal transduction pathways, J. Proteome Res, vol. 7, pp. 558–569.

[2] S.M. Pupa, S. Ménard, S. Forti, E. Tagliabue, 2002, New insights into the role of extracellular matrix during tumor onset andprogression, J. Cell. Physiol, vol. 192, pp. 259–267.

[3] M. Gronborg, T.Z. Kristiansen, A.Iwahori, R. Chang, R. Reddy, N. Sato, H. Molina, O.N. Jensen, R.H. Hruban, M.G. Goggins, A. Maitra, and A. Pandey, 2006, Biomarker discovery from pancreatic cancer secretome usinga differential proteomic approach, Mol. Cell Proteomics, vol. 5, pp. 157–171.

[4] Nair, R., and Rost, B., 2005, Mimicking cellular sorting improves prediction of subcellular localization, J. Mol. Biol., vol. 348, pp. 85–100.

[5] Schaffer A.A., Aravind L., Madden T.L., Shavirin S., Spouge J.L., Wolf Y.I., Koonin E.V., Altschul S.F., 2001 Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements, Nucleic Acids Res., vol. 15, pp. 2994-3005.

[6] Y. Altun, I. Tsochantaridis, T. Hofmann, 2003, Hidden Markov Support Vector Machines, International Conference on Machine Learning (ICML).

[7] Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, Large Margin Methods for Structured and Interdependent Output Variables, 2005, Journal of Machine Learning Research (JMLR), vol. 6, pp.1453-1484.

[8] T. Joachims, T. Finley, Chun-Nam Yu, Cutting-Plane Training of Structural SVMs, 2009, Machine Learning Journal, vol. 77(1), pp. 27-59.

[9] Y. Altun, I., Tsochantaridis, T. Hofmann, Hidden Markov Support Vector Machines, 2003, International Conference on Machine Learning (ICML).

[10] M.K. Kalita, U.K. Nandal, A. Pattnaik, A. Sivalingam, G. Ramasamy, M. Kumar, G.P.S. Raghava, and D.Gupta, 2008, CyclinPred: A SVM-Based Method for Predicting Cyclin Protein Sequences, PLoS ONE, vol. 3, pp. e2605.

[11] Manish K., Gromiha M.M., Raghava G.P.S., 2008, Prediction of RNA binding sites in a protein using SVM and PSSM profile, Proteins: Structure, Function, and Bioinformatics, vol. 71, pp. 189–194.

[12] Guo J., Lin Y., 2006, TSSub: eukaryotic protein subcellular localization by extracting features from profiles, Bioinformatics, vol. 22, pp. 1784–1785

[13] Juhyun Jung,Taewoo Ryu,Yongdeuk Hwang,Eunjung Lee,and Doheon Lee "Prediction of extracellular matrix proteins based on distinctive sequence and domain characteristics" Journal of computational biology ,Volume 17, Number 1, 2010