Comparison between m-SVM and TargetLoc on Protein Subcellular Localization Problem

Fanchao MENG

Computer and Information Science Department, University of Delaware

Abstract

Protein subcellular localization problem is one of the most crucial problems in Bioinformatics playing an important role of making inferences about cellular processes. The computational prediction version of this problem in general can be stated as given a description of a protein, we seek a prediction of its localization in a cell. With respect to the description, various information related to protein can be used. Particularly in [1], Annette Höglund, et al. uses N-terminal targeting sequences, amino acid composition and protein sequence motifs to construct the prediction system, TargetLoc. It outperforms another two well-known methods, TargetP [3] and iPSORT [4]. Another respect is how to construct the predictor. Multi-layer framework consisted of binary classifiers is one of the major ways to implement it, and for each binary classifier, SVM so far is a dominant option. TargetLoc is one of such typical examples. However, there are alternative ways for the both respects. In [2], Alexander Zien, et al. give a different perspective to consider the amino acid composition, and specifically in a more general way; furthermore, a novel approach of constructing a single-kernel SVM predictor is proposed, which is essentially different from TargetLoc. This study will make a comparison between these two prediction systems so that to get more inspired on how to use the related protein descriptions and how to construct predictors. It will go through the comparison from both the two respects, and also the final experimental results.

Instruction

In [1], Annette Höglund, et al. treat N-terminal targeting sequences, amino acid composition and protein sequence motifs separately. From [6], we know that targeting signals are critically helpful information to predict subcellular localization. This information can be found in the polypeptide chain or in the folded protein. For the first place, the primary structure of protein is involved, and for the second place, the tertiary structure and the quaternary structure are involved. Figure 1 shows the protein structure. The continuous amino acid residues in the chain containing such kind of information are called signal peptides. They can be found in the primary structure. Two typical place where we can find signal peptides are the N-terminal extension and C-terminal extension. Moreover, there is also another type of signals, named signal patch. Unlike signal peptides, signal patches are brought by protein folding, which means they could only be found in the tertiary structure and the quaternary structure. Figure 2 signifies the signal patch. However, by unfolding, it is easy to see that a signal patch is composed of parts which are separable in the primary sequence. [9] Therefore, literally we do not have to deal with the tertiary structure itself. TargetLoc actually only uses N-terminal targeting sequences, and there is a component named SVMTarget making localization prediction using this information. [1]

With respect to using amino acid composition to make prediction, TargetLoc gives a component, SVMaac, but without specifying any detailed method, which actually leaves a flexibility to this framework. The method proposed in [5] can be an effective and typical practice. It is a SVM based method, and takes not only the amino acid compositions but also the amino acid pair compositions and the gapped amino acid compositions into account when constructing kernels for SVMs. Then all the predictors are combined together using a voting scheme. Specifically, in [5], it only includes amino acid, amino acid pairs, one gapped amino acid pairs, two gapped amino acid pairs, and three gapped amino acid pairs, these five compositions, and for each composition it builds a SVM for each localization.

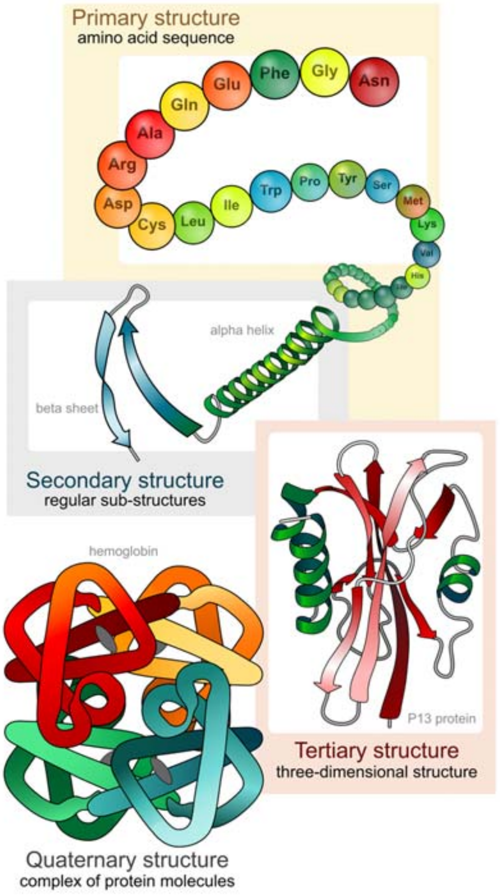


Figure 1 Protein Structure

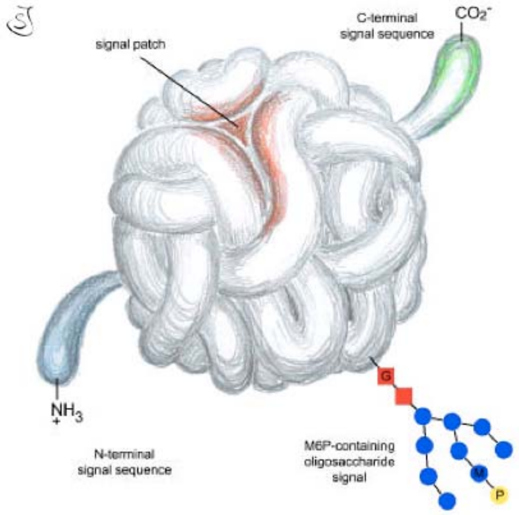


Figure 2 Signal Patch

Sequence motifs are also used in [1], but different from SVM bases classifier, it relies on searching in PROSITE and NLSdb, these two databases. Sequence motif in fact is an amino acid sequence pattern. For example, “KRx{10}KKKL” [11] listed in NLSdb, where ’x’ means any amino acid, and ‘{10}’ means for ten of ‘x’.

All the three parts of results then will be formed into a vector called PPV (protein profile vector), and be sent into another SVM bases component as input to do the final step prediction. This component is of a “one-vs-one” scheme [10] making multi-class classification. Figure 3 demonstrates the general framework presented in [1].

In general, it is easy to see that signal sequences, amino acid compositions and sequence motifs could literally all be considered as amino acid combinations with patterns. Therefore, the most intuitional question when thinking about how to use them is whether there exists a uniform way to handle them all. In [2], Alexander Zien, et al. propose such a method to generalize these combinations. Moreover, they figured out a way to construct a single kernel based on these information, and integrate it into SVM.

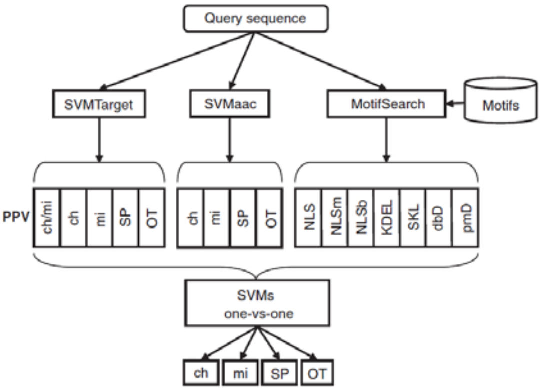


Figure 3 TargetLoc Framework

Amino Acid Compositions

Astrid Reinhardt, et al. in [12] have presented that the amino acid composition is providing useful information for subcellular localization. However, the entire protein sequence is way too specific for the prediction. Therefore some generalization ways emerge then. The first direction is to use subsequences instead of the entire protein sequences, since many important indications of localization are not global. For example, using N-terminal signal peptide sequence with some specific properties is going down this way. The other direction is to extract dependencies between two or more amino acids. For example, in [5] mentioned above, various amino acid compositions could represent such dependencies; and in [14] the distributions of consecutive subsequences of a given length are considered, which is also essentially focusing on the dependencies, only in a different way to describe these information.

TargetLoc makes use of information generalized from the both ways, and combines prediction results derived from various features together to make prediction. Here by feature, it means a description of a protein from a specific perspective. For instance, N-terminal signal sequence, or amino acid compositions, or sequence motifs. However, one of its most severe shortages is that it did not apply a uniform approach to deal with all these information together, instead for each feature TargetLoc builds an individual prediction component. That is why TargetLoc has a relatively complex structure, at least compared to a single-kernel SVM.

On the other hand, in [2], Alexander Zien, et al. are figuring a more generalized way to use the amino acid compositions. In m-SVM, a composition pattern could be consisted of any certain number of amino acids being placed within a given length of sequence. For example, “ ■ □ □ ■ ■ ” could be a pattern with an amino acid triplets and two gaps inside. Then for any given pattern, it computes the empirical distribution of corresponding motifs from a given amino acid sequence, and the result is a histogram of occurrences of each possible r-mer sequence, where r-mer means there are r amino acids in the sequence. One thing needs to be noticed is here, in [2] when talking about m-SVM, “motif” is of a different concept from the conventional one that means those sequence patterns truly found, or sometimes more being used as consensus sequences. Instead, it means a possible “motif”.

This method can be understood in the following way. Given two sequences, our task is to tell the similarity between them in a comparable and quantitative way. More rigorously, the similarity needs to be discussed in a metric space. Then any single subsequence pattern that means temporarily regardless the exact amino acids will contribute to the similarity in a certain amount. And based on this subsequence pattern similarity, if the exact amino acids are appearing in a similar way such as what they are, or what the order they are of, then they will further contribute to the overall similarity in another certain amount. The more the similar patterns and amino acid combinations, the greater the overall similarity. Nevertheless, how to define the “certain amount” becomes a good question. Statistics is going to play an empirically important role here. For each subsequence pattern, in a given sequence, we can know how many times it appears, and also for a pattern, we can do the same thing to the amino acid combinations. We use these statistical data as the likelihood estimation. And all the data forms the distributions mentioned above. Therefore, it measures the compositions in a similar way as used in [14].

By this way, it is easy to understand that as long as a descriptive feature of a protein is expressed in an amino acid composition way, then the method mentioned above would cover all the useful sequence-based cases theoretically at least. In the m-SVM section, we will see how it be done in a feasible way.

Multi-Layer Prediction System

In TargetLoc, it mainly uses the multi-layer prediction scheme to construct the system. There are two heuristics to construct a multi-layer classification system. The first one is “one-vs-rest”. It means a single classifier is trained for each class to discriminate this class from all other classes, and then the one with the highest confidence score is picked as the final prediction result. The other one is “one-vs-one”. For a M classes problem, it will build a binary classifier for each distinct pair of classes. Then through another voting scheme to determine the final classification result, for example, the one with the largest votes from M(M-1)/2 classifiers wins. [15] Specifically, the general framework of TargetLoc is using the second heuristic for its second layer classification, i.e. the SVM outputs the final prediction results. Its SVMTarget component is using this heuristic as well, Figure 4 shows the details about it, whereas SVMaac might use the first heuristic if doing it as in [5].

This structure actually has its advantages, and one of the most noticeable advantages is it will be easy to implement. Since for each classifier, it is a binary one, for instance, typically, by using SVM, there is no additional requirements or computation except the regular binary SVM training. Moreover, usually the voting schemes are all easy to implement too. Hence, the multi-layer prediction system is the most straightforward way to conduct multi-class classification.

However, on the other hand, it could not be the best. In a general perspective, each SVM can be considered as a perceptron making discriminating one group of objects from the other group with respect to one specified feature. The classification results generated from the first layer are collected and formed into another descriptive vector representing the object, then the vector as input is sent to the second layer to do another classification. Obviously there would be some information lost between the two layers if thinking about this procedure in the information theory perspective. Because each classification empirically will be an approximation if the classification result is telling a definitive result, i.e. the input object should be in which class, which is the place where the original information of the similarity between the input object and others is partially lost. Or the classification only gives a probability assignment for each possible class. But in this case, the classifier in fact does not conduct an explicit classification. In other words, the result does not discriminate the given object from other “dissimilar” ones in a better way than describing it by the similarity measure of it defined in a feature space. Thereby, another question if we could keep the original similarity information as much as possible meanwhile we could discriminate a given object from others is arisen.

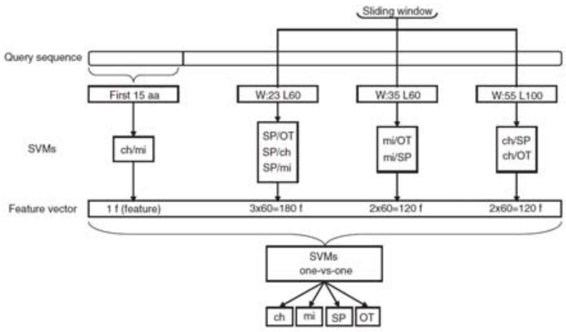


Figure 4 SVMTarget component in TargetLoc

m-SVM and Multiple Kernel Method

With respect to the question brought up in the previous section, in [2] Alexander Zien, et al. give a positive answer. As mentioned above, for any given amino acid composition pattern, it will contribute a part of similarity between two protein sequences. Therefore, the general idea is to use a set of such kind of amino acid composition patterns to describe a protein sequence. And each pattern will be considered as a feature, then a joint feature space is formed in which the overall similarity is measured.

To measure the overall similarity, in [2], it starts with computing the similarity between two amino acid. Since as well known, a substitution matrix describes the rate at which one character in a sequence changes to other character states over time. [16] BLOSUM62 [17] is one of the most popular substitution matrices. Figure 5 shows the BLOSUM62 matrix. BLOSUM62 is computed by $S_{ij} = \left(\frac{1}{\lambda}\right)\log\left(\frac{p_{ij}}{q_i \cdot q_j}\right)$, where p_{ij} is the probability of two amino acids i and j replacing each other in a homologous sequence, and q_i, q_j are the background probabilities of finding the amino acids i and j in any protein sequence. [17] Hence, obviously p_{ij} represents the relationship between i and j, so in [2] we use $K_1^{AA}(a,\ b) = \sum_c p_{ac} - p_{ab}$ as the kernel function to compute the similarity between amino acids a and b. This kernel function actually is the graph Laplacian. The graph Laplacian is also called Laplacian matrix which could be understood as an analogue of the Laplacian in multivariable calculus. It provides a measure of difference between a value of a function at a given point and values at nearby points, which is of the similar meaning that the divergence of the gradient of a function on Euclidean space. [18]

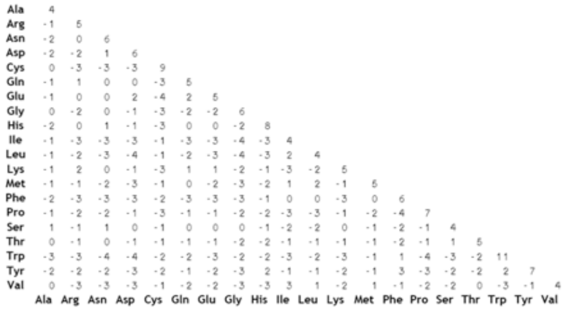
As an extension, for r-tuples of amino acids (here in [2] it is called “motifs”), the kernel is represented by $K_r^{AA}(s,\ t) = \sum_{i=1}^{r} K_1^{AA}(s_i, t_i)$, $s, t \in \mathcal{A}^r$ where $\mathcal{A}$ is the set of 20 amino acids mentioned in BLOSUM62. So far we have the way to compute the similarity with respect to the amino acid combinations.

Then another part of the overall similarity should be the composition patterns. As mentioned in the amino acid composition section, this similarity could be computed by histograms, and after normalization they are probability distributions. [2] Therefore, the similarity for this part literally is comparing distributions. However, since these histograms data are not arbitrary vectors but carrying a special structure, Gaussian RBF will not be appropriate. [2] Then another popular way to compare distributions steps up, Jensen-Shannon divergence [19] which is derived from Kullback-Leibler divergence [20] but symmetric.

The combination of these two parts of similarities will be

$$K_r^{JS}(p,\ q) = \sum_{s\in\mathcal{A}^r}\sum_{t\in\mathcal{A}^r} K_r^{AA}(s,\ t)\cdot\left(p(s)\cdot\log\frac{p(s)}{p(s)+q(t)} + q(s)\cdot\log\frac{q(s)}{p(s)+q(t)}\right).$$

And for all such kernels, a linear combination, more rigorously a convex combination, will be good enough to integrate them together to form a uniform final kernel. It is represented by $K(\boldsymbol{x}, \boldsymbol{x}') = \sum_i \beta_i \cdot K_i(\boldsymbol{x}, \boldsymbol{x}')$. Then we get an amino acid composition based kernel function, and it is fair enough to cover any given length of patterns theoretically. This is the core achievement in m-SVM.



	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Figure 5 BLOSUM62

The underlying essence of this kernel is to construct a join feature space as mentioned above. A joint feature space means for an object, there are several different ways to describe it. For each way, it is called a feature, and it is required to have a metric to measure it meanwhile for any two given measures, they can be compared in a defined way, which means first the feature space needs to be a normed space, and second it needs a way defined for comparing two elements in the space, such as inner product. Thereby, the feature space, in terms of the kernel method, should be a Hilbert space. Furthermore, for all features, we combine them in a normalized weighted linear manner, by which the object is described across all the features and their significance. In other words, we do not have to make comparison on each feature then followed by a classification anymore. Instead, we use a multiple component kernel to construct a single classifier. And that is the essential difference from TargetLoc. Additionally, this kernel can be considered in two different ways with respect to the joint feature space and the multiple kernels respectively. Figure 6 signifies this idea.

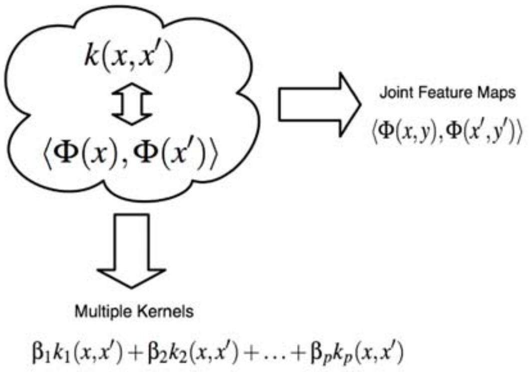

$k(x,x')$
$\langle\Phi(x),\Phi(x')\rangle$
Joint Feature Maps
$\langle\Phi(x,y),\Phi(x',y')\rangle$
Multiple Kernels
$\beta_1 k_1(x,x')+\beta_2 k_2(x,x')+\ldots+\beta_p k_p(x,x')$

Figure 6 Two perspectives of the multiple kernels idea

Since m-SVM somehow is a classifier, how to do the classification is required to be discussed. In the protein subcellular localization problem, the classifier should be a multi-class one. It is based on the confidence functions. Specifically, for each class, there will be a confidence function that is indicating the extent of confidence to which a given object is classified to this class. And the class that is of the highest confidence is considered as the classification result.

So for each class u,

$f_u(\boldsymbol{x}) = \langle \boldsymbol{w}_u, \sum_i \beta_i \Phi_i(\boldsymbol{x}) \rangle =$ $\sum_j \alpha_{ju} \sum_i \beta_i K_i(\boldsymbol{x}, \boldsymbol{x_j})$, where $\boldsymbol{w}_u = \sum_i \alpha_{iu} \Phi_i(\boldsymbol{x})$ due to the Representer Theorem [2][15].

And the classification is $\hat{y}(\boldsymbol{x}) = argmax_u f_u(\boldsymbol{x})$. Figure 7 represents the structure of the classifier.

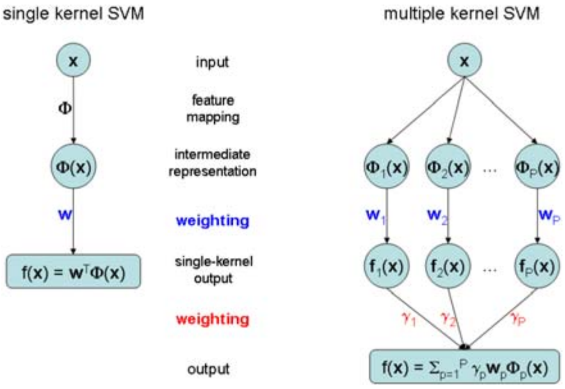
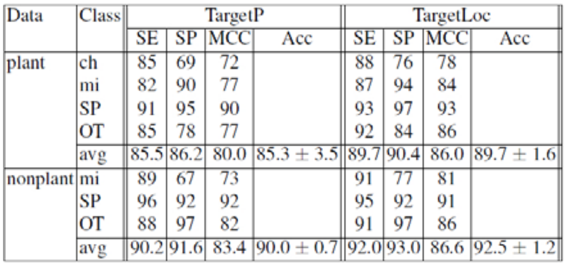


Figure 7 m-SVM classifier (right hand side) and the comparison between m-SVM and typical single kernel SVM (left hand side)

Experimental Results

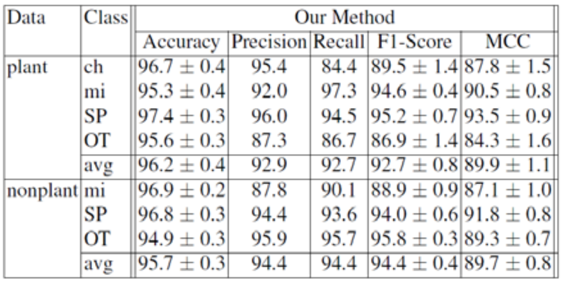
The dataset used for the experimentation is TargetP [21]. The measures are accuracy and MCC.

For the m-SVM experimentation, motif kernel length is up to 5, motifs are extracted from 4 specific places, namely the first 15 and 60 amino acids from the N-terminus and the 15 amino acids from C-terminus ($4 \times 2^{5-1} = 64$ in all), and another 3 BLAST similarity kernels, namely linear kernel on pairwise E-Value, Gaussian kernel on E-Value width 1000, and Gaussian kernel on log E0-Value width 1e5, and also 2 phylogenetic kernels, namely a linear kernel and a Gaussian kernel width 300. [2][22] Figure 8 shows the experimental results of TargetLoc. And Figure 9 shows the results of m-SVM.



Data	Class	TargetP				TargetLoc			
		SE	SP	MCC	Acc	SE	SP	MCC	Acc
plant	ch	85	69	72		88	76	78	
	mi	82	90	77		87	94	84	
	SP	91	95	90		93	97	93	
	OT	85	78	77		92	84	86	
	avg	85.5	86.2	80.0	85.3 ± 3.5	89.7	90.4	86.0	89.7 ± 1.6
nonplant	mi	89	67	73		91	77	81	
	SP	96	92	92		95	92	91	
	OT	88	97	82		91	97	86	
	avg	90.2	91.6	83.4	90.0 ± 0.7	92.0	93.0	86.6	92.5 ± 1.2

Figure 8 Experimental Results of TargetLoc



Data	Class	Our Method				
		Accuracy	Precision	Recall	F1-Score	MCC
plant	ch	96.7 ± 0.4	95.4	84.4	89.5 ± 1.4	87.8 ± 1.5
	mi	95.3 ± 0.4	92.0	97.3	94.6 ± 0.4	90.5 ± 0.8
	SP	97.4 ± 0.3	96.0	94.5	95.2 ± 0.7	93.5 ± 0.9
	OT	95.6 ± 0.3	87.3	86.7	86.9 ± 1.4	84.3 ± 1.6
	avg	96.2 ± 0.4	92.9	92.7	92.7 ± 0.8	89.9 ± 1.1
nonplant	mi	96.9 ± 0.2	87.8	90.1	88.9 ± 0.9	87.1 ± 1.0
	SP	96.8 ± 0.3	94.4	93.6	94.0 ± 0.6	91.8 ± 0.8
	OT	94.9 ± 0.3	95.9	95.7	95.8 ± 0.3	89.3 ± 0.7
	avg	95.7 ± 0.3	94.4	94.4	94.4 ± 0.4	89.7 ± 0.8

Figure 9 Experimental Results of m-SVM

From the results we can see m-SVM outperforms TargetLoc on the TargetP dataset based on current settings. However, these result might not be so adequate to corroborate the m-SVM will definitely outperform TargetLoc. Since firstly in its experiments, there are another 5 classifiers introduced into the m-SVM model cooperating with the motif kernels, it is hard to tell the significance of these 5 classifiers, then the significance of the motif kernels cannot be determined. Another thing is Alexander Zien et al in [2] did not conduct corresponding experiments to show if m-SVM could outperform MultiLoc proposed in [1]. Since MultiLoc literally is constructed with the exact same idea as TargetLoc except another component named SVMSA is added and the SVMaac component is enhanced so that more localizations can be covered in the model. It is easy to see that so far in this experimentation the improvement is not that significant, so I doubt if it could really do better than the general framework in [1].

Discussion

From the comparison it can be seen that a different perspective of using the feature information could lead to an at least comparable classification result to the multi-layer classification system. A more important insight of the multiple kernel model is now that various features, as long as the corresponding feature space is a Hilbert space, could be integrated together by a convex combination, then it means the features would not be limited to amino acid compositions. In other words, for example, some text-based features can also be introduced as sub-kernels. And in the experiments for m-SVM, we actually have seen another 5 kernels being used along with motif kernels. Furthermore, by this way, if text-based classifiers [23][24] could be improved becomes a very interesting question.

However, the multiple kernel method still has some problems. First of all, it is worth to notice that the kernel method proposed in [2] is quite restricted by the length of motifs. People have to explicitly specify the length of the target "motifs", for instance 5 is specified in its experiments. But in fact real motifs are usually longer than 5. Hence, how to choose an effective length is a problem.

Another problem is by this method, it is hard to tell which kernel contributes to which class more. But this information could be very useful for the future prediction.

The classification result is also a questionable place. Since we have seen that the result is derived from an “argmax”, which means it would not care too much about the significance, then that is the problem. Especially, when people cannot tell which kernel is in charge of which class, then the classification result could be really sensitive to the training set. But this problem is hard to solve by the current model. Hence, this could be a future work.

The training procedure of the m-SVM model would be complicated. Then I wonder if the Boosting method could be applicable here instead of using regular SVM training approaches.

Conclusion

This study compares two very different multi-class classification systems, the multi-layer system in [1] and the multi-kernel system in [2] respectively. Nevertheless, their original motivations are the same, namely using various features to do the prediction. The essential difference between them is located at two parts. First, how to deal with amino acid compositions, and second, how to express them as features. For the first part, in [2], m-SVM makes a further generalization by unifying all amino acid composition bases useful sequence into the “amino acid combinations + composition patterns” manner. And for the second part, the multiple kernel method is proposed instead of the “binary classifiers + multi-layer” framework.

A more meaningful achievement in [2] is the emphasis of joint feature space which provides a theoretical framework to integrate various kinds of features together to construct a single kernel classifier. Therein the feature selection is directly mapped to the kernel design.

From the experimentation, m-SVM shows the framework at least could accomplish a comparable result to the method proposed in [1]. However, it is still too early to give a definite conclusion that m-SVM will outperform TargetLoc or even MultiLoc, since the new amino acid composition method has not been proven effective due to those 5 additional kernels. Hence, more experiments are needed for m-SVM.

Besides the experiments, there are also some other problems on the air mentioned above. Each of them could lead to a dedicated study. Therefore, extensive application and verification of this method will be a direction in the future.

Acknowledgement

I thank Dr. Shatkay for helping me on the topic selection and great advice about how to conduct this study.

References

[1] Höglund, Annette, et al. "MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs and amino acid composition." Bioinformatics 22.10 (2006): 1158-1165.

[2] Ong, Cheng Soon, and Alexander Zien. "An automated combination of kernels for predicting protein subcellular localization." Algorithms in Bioinformatics. Springer Berlin Heidelberg, 2008. 186-197.

[3] Emanuelsson, Olof, et al. "Predicting subcellular localization of proteins based on their N-terminal amino acid sequence." Journal of molecular biology 300.4 (2000): 1005-1016.

[4] Bannai, Hideo, et al. "Extensive feature detection of N-terminal protein sorting signals." Bioinformatics 18.2 (2002): 298-305.

[5] Park, Keun-Joon, and Minoru Kanehisa. "Prediction of protein subcellular

locations by support vector machines using compositions of amino acids and amino acid pairs." Bioinformatics 19.13 (2003): 1656-1663.

[6] Nakai, Kenta, and Paul Horton. "PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization." Trends in biochemical sciences 24.1 (1999): 34-35.

[7] Protein Structure. http://en.wikipedia.org/wiki/Protein_structure

[8] Signal Patch. http://en.wikipedia.org/wiki/Signal_patches

[9] Protein Targeting. http://en.wikipedia.org/wiki/Protein_targeting

[10] Scholkopf, Bernhard, and Alex Smola. "Learning with kernels." (2002).

[11] KRx{10}KKKL. https://rostlab.org/services/nlsdb/detail.php?keyword=KRx{10}KKKL

[12] Reinhardt, Astrid, and Tim Hubbard. "Using neural networks for prediction of the subcellular location of proteins." Nucleic acids research 26.9 (1998): 2230-2236.

[13] Guda, Chittibabu, and Shankar Subramaniam. "TARGET: a new method for predicting protein subcellular localization in eukaryotes." Bioinformatics 21.21 (2005): 3963-3969.

[14] Yu, Chin‐Sheng, Chih‐Jen Lin, and Jenn‐Kang Hwang. "Predicting subcellular localization of proteins for Gram‐negative bacteria by support vector machines based on n‐peptide compositions." Protein Science 13.5 (2004): 1402-1406.

[15] Scholkopf, Bernhard, and Alex Smola. "Learning with kernels." (2002).

[16] Substitution Matrix. http://en.wikipedia.org/wiki/Substitution_matrix

[17] BLOSUM62. http://en.wikipedia.org/wiki/BLOSUM

[18] Laplace Operator. http://en.wikipedia.org/wiki/Laplace_operator

[19] Jensen-Shannon Divergence. http://en.wikipedia.org/wiki/Jensen%E2%80%93Shannon_divergence

[20] Kullback-Leibler Divergence. http://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

[21] TargetP. http://www.cbs.dtu.dk/services/TargetP/

[22] Pellegrini, Matteo, et al. "Assigning protein functions by comparative genome analysis: protein phylogenetic profiles." Proceedings of the National Academy of Sciences 96.8 (1999): 4285-4288.

[23] Brady, Scott, and Hagit Shatkay. "EpiLoc: a (working) text-based system for predicting protein subcellular location." Pacific Symposium on Biocomputing. Vol. 13. 2008.

[24] Shatkay, Hagit, et al. "SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data." Bioinformatics 23.11 (2007): 1410-1417.