

Analysis and Comparison of the Kernel Accuracy for Saccharomyces Genus Protein Sequence Classification

Hyo Jung Chun, Dong Woo Hong, Taeseon Yoon

Hankuk Academy of Foreign Studies, Yongin, Gyeonggi, South Korea

hjchun96@gmail.com, hdongwoo1479@gmail.com, tsyoon@hafs.hs.kr

Abstract— Even as the modern protein sequencing technology is uncovering great amount of new protein, such data are futile without knowledge of the functions the protein sequences encode. To overcome the limitation of experimental analysis, utilizing Support Vector Machine and kernel methods for functional prediction of unannotated protein has become a promising topic of research in the field of computational biology, inducing many researchers to develop kernels with improved accuracy and efficiency. In this paper, we assigned the Gaussian, Polynomial, and Normal kernels to each *Sensu Stricto*, *Sensu Lato*, and *Petite-Negative* groups of the *Saccharomyces* fungus species, for which the kernel showed the greatest accuracy in protein sequence classification. From the result we discovered the sequential shapes of the proteins and detected similarities of the structural linearity among the proteins species belonging in the same group. The resulting data allow us to provide an important categorization of kernels that will predict protein function with the greatest accuracy depending on the group of the *Saccharomyces* the protein belongs for future researches using sequential analysis for prediction. into which you can type your own text.

Keywords— Support Vector Machine, Kernel methods, Protein Sequence, Function prediction

I. INTRODUCTION

As the advancement in genome sequencing technologies continues to uncover great amount of new proteins, using machine learning for characterization of proteins has effectively taken off the burden experimentally analyzing and classifying thousands of new data. Such breakthrough is significant in that knowing the functions of proteins provide the basis of new drug development and advanced biological researches. Introduced by Vapnik in 1995 and 1998, SVM and Kernel functions were shown to be more effective in solving prediction problems than previous methods of classification such as the Markov chain models or neural networks [1]. Besides comparing the accuracies and functionality of the basic kernel functions such as the Linear, Polynomial, and Gaussian kernels for protein sub cellular localization and similar tasks [1] [2], much effort was put into designing and utilizing more suitable kernels for various problems. Researchers showed improvement in protein recognition when

integrating multiple data sources through fusion of various kernels, each of which can process data set that encodes a particular aspect of similarity between proteins such as gene expression or protein-protein interaction data [3]. Borgwardt [4] showed increase in prediction accuracy for protein function using similar methodology, achieved through graph kernels and hyperkernel to include specific nodal information of sequences. The work of Sharan [5] gives an overview on the types of computational approaches previous researchers took to improve prediction results for protein functions.

The fundamental basis of protein function prediction is analyzing the sequential and structural similarities of the proteins [6], and in this paper we strive to provide a guideline that will make future works of protein classification more efficient. Based on the research of Cliften [7] and Kellis [8] that explored the similarities among the *Saccharomyces* species in the *Sensu Stricto* group through sequential analysis, we predict that the proteins in the same *Saccharomyces* groups would have similar sequential shapes. Because proteins in different groups are likely to be different in the shape of the sequences, we attempt to find the kernels that will produce greatest accuracy when classifying protein sequences of each groups, and the result will guide researchers to be more effective in deciding which kernel to utilize for protein function prediction, so long as they are informed of which group the protein in hand belongs to.

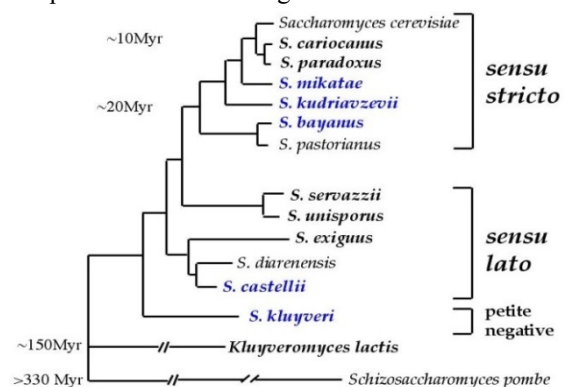


Figure 1. This chart shows the homology of different species in the *Saccharomyces* fungus

In this experiment, we will analyze protein sequences of 7 species that belong in 3 distinct groups: Sensu Stricto (*Saccharomyces Cerevisiae*, *Saccharomyces Paradoxus*, *Saccharomyces Mikatae*), Sensu Lato (*Saccharomyces Castellii*, *Saccharomyces Servazzii*, *Saccharomyces Exiguus*), and Petite-Negative (*Saccharomyces Kluyveri*). The categorization is explained in Figure 1.

We chose *Saccharomyces* (yeast) because although this genus has been researched for decades as a model organism and can provide us with abundant data, it is still at the center of interest in the scientific community while scientists struggle to classify the remaining 2/3 of discovered proteins.

II. MATERIALS & METHODS.

A. Support Vector Machines and Kernel Methods

Support Vector Machine is a supervised learning algorithm which learns a linear classifier from given data. The concept of SVM is to map the training data in the input space onto a higher-dimensional vector space, called a feature space F , and achieve linear classification in such space. The map $\varphi: \mathcal{X} \rightarrow F$, $x \rightarrow X := \varphi(x)$ is used. In the feature space, the SVM constructs a hyper plane defined as $w^T x + b = 0$ (Figure.2) with the greatest margin and classifies the data with the corresponding decision function $f(x) = \text{sgn}(\langle w, x \rangle + b) = \text{sgn}(w^T x + b)$. This function is the most basic Normal Kernel.

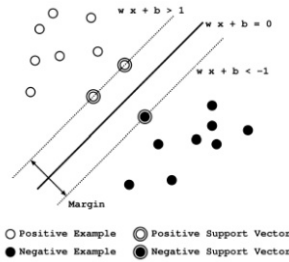


Figure 2. The hyperplane, margin, and support vectors (black and white circles) are signified in the picture.

Generally for data that are not linearly separable, the generalization error becomes too huge when the hard margin classifiers try to correctly classify all outliers, even after transforming the data into a higher-dimension feature space using kernels. The soft margin classifier that allow some vectors to be misclassified or violate the margin is far more commonly used in experiments dealing with real data because it is much more practical and produces realistic results.

Given the data set $D = \{x_n, y_n \mid x_n \in R^d, y_n \in \{-1, 1\}\}_{n=1}^N$ with input vector $\mathbf{X}$ in d dimensional space, all labeled as either 1 or -1, the optimization problem for maximum margin ($w^T x + b > 1, w^T x + b < -1$) is:

$$\begin{aligned} & \min_{w, b, \xi} \frac{1}{2} w^T x_n + \\ & C \sum_{n=1}^N \xi_n \quad (1) \\ & \text{Subject to} \\ & y_n(w^T x_n + b) \geq 1 - \xi_n, \quad (n = 1, \dots, N) \end{aligned}$$

$$\xi_n \geq 0$$

$$w \in R^d, b \in R, \xi \in R^N$$

where vector $\mathbf{W}$ is the weight vector and scalar b the bias, both of which are known as primal variables. ξ is the slack that quantifies the violation of the margin, and C a trade-off parameter that defines the relative importance of maximizing the margin and minimizing the total violation.

By turning the equation into Lagrange formation, solving it using Quadratic Programming, and simplifying it into dual form, we arrive at the solution using Quadratic Programming:

$$\alpha = \alpha_1, \dots, \alpha_N \rightarrow w = \sum_{n=1}^N \alpha_n y_n x_n \quad (2)$$

Taking into consideration the KKT condition: for $n = 1, \dots, N$

$$\alpha_n (y_n (w^T x_n + b) - 1) = 0 \quad (3)$$

the margin support vectors that satisfy the margin, satisfying $y_n (w^T x_n + b) = 1$ when $0 \leq \alpha_n \leq C$ and $\xi_n = 0$, and the non-margin support vectors that may violate the margin, satisfying $y_n (w^T x_n + b) < 1$ when $\alpha_n = C$ and $\xi_n > 0$ are all considered as acceptable parts of the SVM.

Kernel functions are the tools used to transform the hyperplane into a higher dimensional, even infinite, space that can best execute the task when the given set of training data is not linearly separable. Defined as $K(x, x') := \langle x, x' \rangle = \varphi(x)^T \varphi(x')$, they are convenient because by replacing the dot products involved in computing the decision function with kernel functions, as shown in the following,

$$\begin{aligned} f(x) &= \text{sgn}(\langle w, x \rangle + b) = \text{sgn} \left(\sum_{n=1}^N y_n \alpha_n \langle x, x_n \rangle + b \right) \\ &= \text{sgn} \left(\sum_{n=1}^N y_n \alpha_n \langle \varphi(x), \varphi(x_n) \rangle + b \right) \\ &= \sum_{n=1}^N y_n \alpha_n k(x, x_n) + b \quad (4) \end{aligned}$$

the kernels make it possible to describe the high-dimensional space and the hyper plane in terms of the training data in the input vector space $\mathcal{X}$. This makes attempting to understand or compute the solution in terms of the vectors in the infinite space is unnecessary in solving the problem. Only the validation of the kernel is prerequisite, which is possible through mathematical proof.

Along with the soft margin classifier, the kernel functions are frequently used when dealing with a non-linear classification problem. The kernel functions that are used in this paper will be explained in depth in the following sections, which are the Polynomial and the Gaussian kernels. [9][10]

B. Polynomial Kernel

Polynomial kernel is the fundamental model of a kernel that can raise the feature space to dimension d for separation by the hyper kernel of $d - 1$ dimension. By using polynomial kernel, we can create a hyperplane or vector space

which linearly separates data and is useful when the set of data is linearly non-separable. The polynomial kernel is defined as:

$$K(x, y) = (x^T y + c)^d, (\{x, y\} \in R^k, d > 0, d \text{ is natural number}) \quad (5)$$

Constant c trades off the exponents of high-order in response to low-order terms in the polynomial equation and d represents the dimensions that the kernel expands the vector space into.

When the value of c is zero, the kernel becomes homogeneous ($K(x, y) = (x^T y)^d$), for which the sums of exponents for each nonzero term are all equal and thus not carry a trade off parameter. There is also a non-homogeneous polynomial kernel, which is more frequently used than the homogenous kernel in actual data sorting. The feature map $\varphi(x)$ contains monomials whose exponents is lower or equal to d and this kernel contains the trade off constant c . The non-homogenous kernel is defined as:

$$(x^T y + c)^d = (x_1 y_1 + \dots + x_k y_k + \sqrt{c} \sqrt{c})^d = \sum_{n_i \geq 0, \sum_{i=1}^{k+1} n_i = d} \binom{d}{n_1, \dots, n_k} x_1^{n_1} y_1^{n_1} \dots x_k^{n_k} y_k^{n_k} \cdot c^{\frac{n_{k+1}}{2}} \quad (6)$$

Thus, the entries of the vector $\varphi(x)$ are:

$$\varphi(x) = \left(\sqrt{\binom{d}{n_1, \dots, n_k}} x_1^{n_1} \dots x_k^{n_k} \cdot c^{\frac{n_{k+1}}{2}} \right)_{n_i \geq 0, \sum_{i=1}^{k+1} n_i = d} \quad (7)$$

For instance, if we let $d=k=2$:
 $(x^T y + c)^2 = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 + 2cx_1 y_1 + 2cx_2 y_2 + c = \varphi(x)^T \varphi(y)$,
 where $\varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2}cx_1, \sqrt{2}cx_2, \sqrt{c})$,
 we have mapped $\varphi(x)$ from R^2 to R^6 space and the hyperplane that satisfies $\varphi(x)^T \varphi(x) - b = 0$ in R^6 will correspond to conics shape at R^2 field. Since $(x_1 + \dots + x_k)^d = \sum_{n_i \geq 0, \sum_{i=1}^k n_i = d} \binom{d}{n_1, \dots, n_k} x_1^{n_1} \dots x_k^{n_k}$, we can calculate the multinomial coefficient following $\binom{d+k-1}{d} \binom{d+k-1}{k-1} = \frac{(d+k-1)!}{d!(k-1)!}$ (Binomial coefficient) This coefficient makes it possible to estimate the dimension of vector space $\varphi(x)$ where $x \in R^k$.

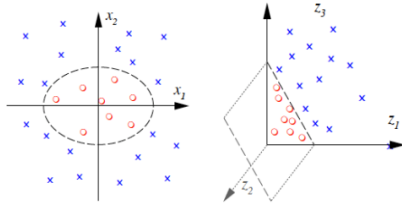


Figure 3. This figure shows a case in which the polynomial kernel is used to separate the non-linearly separable data in the R^2 field a by creating the hyperplane on R^3 field.

$$\varphi : R^2 \rightarrow R^3, (x_1, x_2) \rightarrow (z_1, z_2, z_3) := (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$$

C. Gaussian Kernel

Gaussian function kernel is defined as:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) = \exp(-\gamma\|x - x'\|^2), \left(\gamma = \frac{1}{2\sigma^2}\right) \quad (8)$$

As shown above, Gaussian kernel depends on the distance between two distinct vectors. By considering distance as similarity, Gaussian kernel shows us how similarity measures of different sets of data can be seen through certain pattern.

The role of the variables σ can be seen in the following example: consider that the value of $\sigma \rightarrow 0$

$$K(x, x'; \sigma) \rightarrow \begin{cases} 0 & \text{if } x \neq x' \\ 1 & \text{if } x = x' \end{cases} \quad (9)$$

This exemplifies a case in which Gaussian kernel is used as similarity measure. An important characteristic of the Gaussian kernel is that it is infinite dimensional. This allows all kinds of complex data to be linearly separable by mapping them into the feature space with infinite dimensions. For example, the left figure in Figure 4 appears to have no linear boundary of the dataset. However, the Gaussian kernel automatically draws a non-linear boundary around the dataset in the input space, which corresponds to a linear boundary in the feature space.

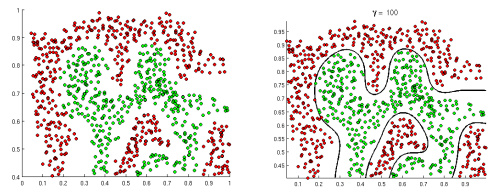


Figure 4. (Left) green dot represents positive values and red represents negative. (Right) non-linear boundary of the dataset with $\gamma = 100$ which are the Polynomial and the Gaussian kernels. [9][10]

D. Implementation

In our experiment, we used input data extracted from the MIPS yeast database (mips.helmholtz-muenchen.de/genre/proj/yeast/) and Saccharomyces Genome Database (www.yeastgenome.org). By sampling from SGD, we found ORF (Open Reading Frame: part of sequence that renders one functional protein) data from Washington University of Saint Louis and MIT for each Saccharomyces species. We selected 2 ORFs in each species that had similar function according to the MIPS database. (cf. YEL064c and YBL089w: cellular transporters.)

For actual implementation of the experiment comparing the accuracy of the kernels, we utilized the SVMlight software, which is often used in large-scale SVM classification. In order to use the SVM machine, we changed the form of our data and binarized the sequences of ORFs, one as positive data and the other as negative data, since we are comparing two sequences. We used 10-fold cross validation data (training 9 data and testing 1) and trained the kernels provided by SVMlight:

Normal, Polynomial, and Gaussian. The kernels were pre-programmed to implicitly select the values of the variables (σ and d) so that the greatest accuracy would be achieved for each classification. We divided the sequences up by every 5, 7, and 9 consecutive amino acids (referred to as different windows) which would include different amount of information regarding the compositions and naturally entail different results in the experiments.

III.RESULTS & DATA

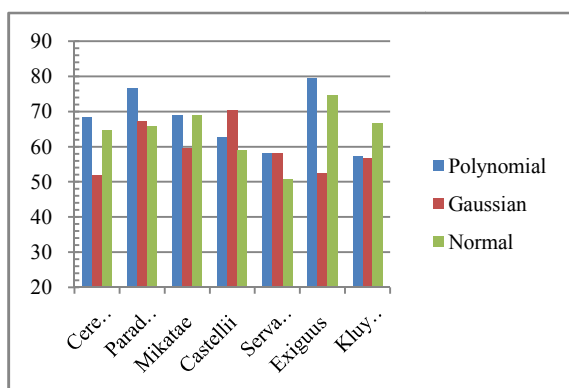


TABLE 1. THE AVERAGE VALUES ($\frac{5W+7W+9W}{3}$) OF THE ACCURACIES FOR THREE KERNELS ARE SHOWN.

	Polynomial	Gaussian	Normal
Cerevisiae	68.47%	51.85%	64.81%
Paradoxus	76.62%	67.36%	65.65%
Mikatae	69.08%	59.56%	69.08%
Castellii	62.67%	70.37%	58.97%
Servazzii	58.06%	58.06%	50.86%
Exiguus	79.36%	52.38%	74.60%
Kluyveri	57.24%	56.67%	66.67%

We analyzed and compared the classification accuracy measures of each kernel function in order to characterize the sequential shape of proteins in each group and determine the most suitable kernel for the classification task of sequences in each *Saccharomyces* species. As prediction accuracy of less than 70% generally has small significance in showing the efficacy of the kernel, as some data in the Table I are, the analysis of the data was done in a relative sense by comparing the differences of the accuracies. Furthermore, the accuracy measures for one specific window for which all three kernels showed significant drop or increase were disregarded in comparing the kernel functionalities and interpreted as a biased result induced by the characteristic of the sequence that was divided.

TABLE 2. THE CHART CATEGORIZES TP, FP, FN, TN. THE PROGRAM COMPUTES THESE VALUES AND RETURNS THE PREDICTION ACCURACY MEASURE, GIVEN AS $A = \frac{TP+TN}{TP+FP+TN+FN}$

	Predicted positive	Predicted negative
Real positive	True positive, TP	False negative, FN
Real negative	False positive, FP	True negative, TN

As seen in Table 1, the Polynomial and Normal kernels both performed similarly in classifying sequence data of *S.*

Cerevisiae proteins, showing that the sequences are linearly separable just as much as they are non-linearly separable. The low percentages show that the sequences are not perfectly classified by either kernels. However the classification accuracy of 100% for the Normal kernel in the 9 Window experimentation, a high value which can not be overlooked, allows us to conclude that *S. Cerevisiae* protein sequences may be dominantly linear in shape, although they may have some non-linear features. For the sequences of proteins of *S. Paradoxus*, the small difference between the average accuracy measures of the Polynomial and Normal kernel of only 10.9% and the accuracy measures of the Gaussian and Normal kernels being the same for all three windows signify linearity. However the high accuracy of Polynomial kernel makes it highly probable that such sequences are non-linear to some degree. For *S. Mikatae* protein sequences, the accuracy measures for all three kernels were same for two windows and Polynomial and Normal the same across all three types of divided sequences, as shown in Table 5. This means that $d = 1$ and $\sigma = 0$ for Polynomial and Gaussian kernel, respectively, and thus a linear kernel was judged to be the best fit for classification. The low measure by the Gaussian kernel for 9 Window further corroborates the conclusion that the *S. Mikatae* sequences are linear in shape. Overall, we were able to make a somewhat blunt yet justified generalization that since all the protein sequences of the species in the *Sensu Lato* group were considerably well classified by Normal kernels, they are likely to be linear in shape, although they could certainly have varying degrees of non-linearity as some of the high accuracies of the multi-dimensional kernels attest.

The protein sequences of *S. Castellii*, *S. Servazzii*, and *S. Exiguus* in the *Sensu Lato* group displayed slightly more non-linear tendencies. For *S. Castellii*, the results of the 9 Window for which the accuracies of all three kernels significantly dropped were disregarded for reasons previously explained. However the consistently high accuracy measures for both non-linear kernels of nearly 84.00% for the other two windows lead us to the conclusion that the protein sequence will most likely be non-linear in shape. The accuracy measures for classification of *S. Servazzii* by the linear and non-linear kernels are less distinct, with the 9 Window results even being the same across all three kernels. However, unlike the results of *S. Mikatae* in the first group, for which the accuracy measures by the Polynomial and Normal kernels were the same for all three windows, those of *S. Servazzii* are relatively higher for the non-linear kernels in 5 and 7 Windows. Thus we can conclude that the sequential shape of this species of proteins could be slightly non-linear. As shown in Table VIII, the Polynomial and Normal kernels scored the same for the classification of *S. Exiguus* in both 7 and 9 Window. Nonetheless, the Polynomial kernel performed better than the Normal kernel in the 5 Window with a 14.29% higher accuracy, and overall produced a credible average accuracy measure higher than that of the Normal kernel. The results suggest that *S. Exiguus* include protein sequences that may are not likely to be linear, contrary to what all three species in *Sensu Stricto* overall seem to do.

The protein sequences of *S. Kluyveri* species in the Petite-Negative group are also likely to be more non-linear than linear, since the Gaussian and Polynomial kernel both scored over 70% for 5 and 7 window and higher than the Normal kernel. The significant drop in the measures for the higher-dimensional kernels in the 9 window experiment that brought down the average accuracy measures for those kernels was overlooked. In the context of all 7 types of sequences analyzed in this paper, the Sensu Lato group protein sequences seemed to be non-linear in an intermediate degree between those of Sensu Stricto and Petite-Negative.

1) Sequences Classification Result of Protein in Sensu Stricto Group

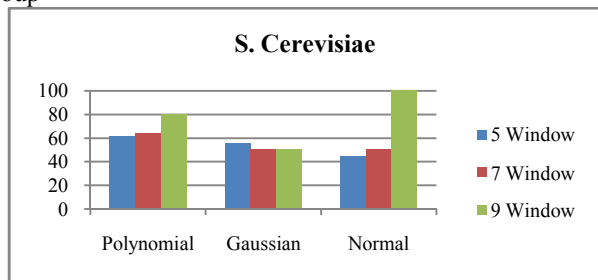


TABLE 3. COMPARISON CHART OF ACCURACY FOR *S. CEREVISIAE*, USING 3 KERNEL METHODS AND 3 DIFFERENT WINDOWS.

	5Window	7Window	9Window
Polynomial	61.11%	64.29%	80.00%
Gaussian	55.56%	50.00%	50.00%
Normal	44.44%	50.00%	100.00%

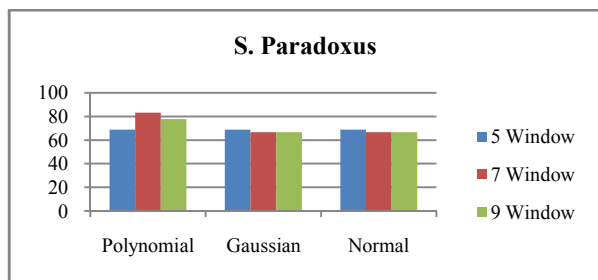


TABLE 4. COMPARISON CHART OF ACCURACY FOR *S. PARADOXUS*, USING 3 KERNEL METHODS AND 3 DIFFERENT WINDOWS.

	5Window	7Window	9Window
Polynomial	68.75%	83.33%	77.78%
Gaussian	68.75%	66.67%	66.67%
Normal	68.75%	66.67%	66.67%

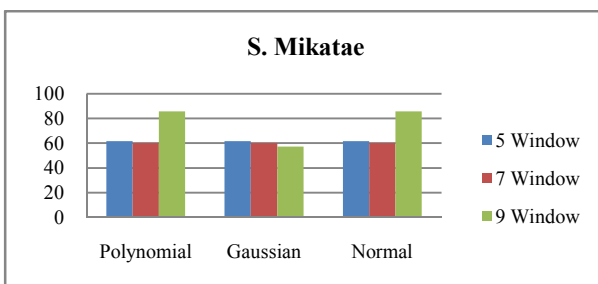


TABLE 5. COMPARISON CHART OF ACCURACY FOR *S. MIKATAE*, USING 3 KERNEL METHODS AND 3 DIFFERENT WINDOWS.

	5Window	7Window	9Window
Polynomial	61.54%	60.00%	85.71%
Gaussian	61.54%	60.00%	57.14%
Normal	61.54%	60.00%	85.71%

2) Sequence Classification Result of Protein in Sensu Lato Group

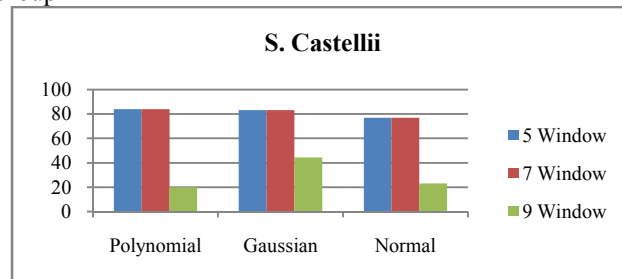


TABLE 6. COMPARISON CHART OF ACCURACY FOR *S. CASTELLII*, USING 3 KERNEL METHODS AND 3 DIFFERENT WINDOWS.

	5Window	7Window	9Window
Polynomial	84.00%	84.00%	20.00%
Gaussian	83.33%	83.33%	44.44%
Normal	76.92%	76.92%	23.08%

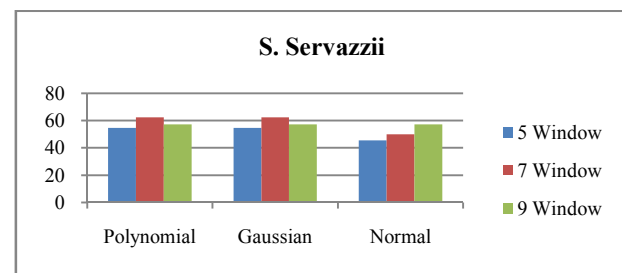


TABLE 7. COMPARISON CHART OF ACCURACY FOR *S. SERVAZZII* USING 3 KERNEL METHODS AND 3 DIFFERENT WINDOWS

	5Window	7Window	9Window
Polynomial	54.55%	62.50%	57.14%
Gaussian	54.55%	62.50%	57.14%
Normal	45.45%	50.00%	57.14%

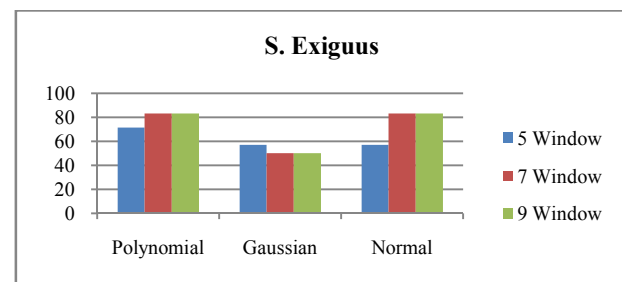


TABLE 8. COMPARISON CHART OF ACCURACY FOR *S. EXIGUUS* USING 3 KERNEL METHODS AND 3 DIFFERENT WINDOWS.

	5Window	7Window	9Window
Polynomial	71.43%	83.33%	83.33%
Gaussian	57.14%	50.00%	50.00%
Normal	57.14%	83.33%	83.33%

3) Sequence Classification Result of Protein in Petite-Negative Group

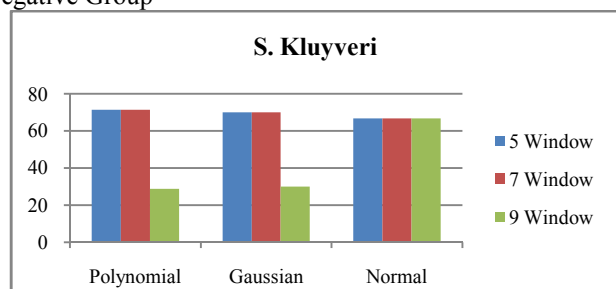


TABLE 9. COMPARISON CHART OF ACCURACY FOR S. KLUYVERI, USING 3 KERNEL METHODS AND 3 DIFFERENT WINDOWS.

	5Window	7Window	9Window
Polynomial	71.43%	71.43%	28.87%
Gaussian	70.00%	70.00%	30.00%
Normal	66.67%	66.67%	66.67%

IV. DISCUSSION & FUTURE WORKS

As previously explained in the **RESULTS & DATA** section, our data result for the proteins of 7 different species can be used to make evidence-based deduction about the shape of the protein sequences. Based on previous research [7] and [8] in comparing the sequential alignment and similarities of protein sequences of various species in the same group, we expanded the experiment to compare the spatial shapes of the sequences and characterized the sequences not only within a group but also in terms of different groups. The experiment results showed that, firstly, the *Saccharomyces* proteins in various groups may have evolved from the same origin but have distinct sequential shapes. The clear distinction between the *Sensu Stricto*, *Sensu Lato*, and *Petite-Negative* group proteins on the best classifying kernels supports this statement. As shown in Figure 1, the different time lapse in the evolvement of the three groups of the *Saccharomyces* genus clearly differed the sequential shapes of proteins in different groups, the sequences of species becoming gradually more non-linear as going down the list. More importantly, the result evinces our hypothesis that all the proteins used in the experiment generally share similar sequential shape with others belonging in the same group. This allows us to state that using different kernels depending on the group the protein belongs in will make an improvement in making a more accurate function prediction.

Our next step in research would be to further corroborate our suggested method of improving protein function prediction and expand its usage by 1) increasing the amount of sequence data used for the experiment and incorporating additional information such as weight factors for increased classification accuracy 2) analyzing the results of function prediction conducted following our suggested method 3) applying the methodology for groups in protein genus other than *Saccharomyces* 4) experimenting with more kernels that each have distinct advantages. Further support of our hypothesis will set an important guideline for bioinformatics

researchers seeking for more efficient ways to predict protein functions mainly based on protein sequence similarities, along with additional information such as protein-protein interaction data and sequence motif to enhance the accuracy of the basic kernels. By knowing which kernel will best serve in classifying proteins in various groups and species, they will be able to achieve greatest accuracies in functional predictions of proteins.

REFERENCE

- [1] S. Hua et al., "Support vector machine approach to protein subcellular localization prediction", *Oxford Journals*, Vol.17, pp 721-728, 2001
- [2] K.J. Park et al., "Prediction of protein subcellular locations by support vector machines using compositions amino acids and amino acid pairs", *Bioinformatics*, Vol.19, pp 1656-1663, 2003
- [3] G. R. G. Lankriet et al., "Kernel-based data fusion and its application to protein function prediction in yeast", *Pacific Symposium on Biocomputing*, Vol.9, pp. 300-311, 2004
- [4] K.M. Borgwardt et al., "Protein function prediction via graph kernels", *Oxford Journals*, Vol. 21, pp i47-i56, 2005
- [5] R. Sharan et al., "Network-based prediction of protein function", *Molecular Systems Biology*, Vol. 3, pp.1-13, 2007. 3
- [6] D. Lee et al., "Predicting protein function from sequence and structure", *NATURE REVIEWS*, Vol.8, 995-1005, 2007. 12
- [7] P. Cliften et al., "Surveying *Saccharomyces* Genomes to Identify Functional Elements by Comparative DNA Sequence Analysis", *Genome Research*, Vol 11. No. 7, pp. 1175-1186, 2001
- [8] M. Kellis et al., "Sequencing and comparison of yeast species to identify genes and regulatory elements", *NATURE*, Vol. 423, pp. 241-254, 2003. 5
- [9] B. Schölkopf et al., "Learning with Kernels". Cambridge, MA: MIT Press, 2002, pp. 1-21
- [10] A. Ben-Hur et al., "Support Vector Machines and Kernels for Computational Biology", *PLoS Computational Biology*, vol.4 no.10, pp.1-19, 2008.10



Hyo Jung Chun was born in Seongnam, South Korea in 1996. This author has previously studied abroad in Boulder, Colorado and is currently studying at Hankuk Academy of Foreign Studies in Yongin, Gyeonggi. This author has participated in another conference related to computational science and wishes to study artificial intelligence, robotics, and business in the future.



Dongwoo Hong was born in Busan, South Korea, in 1996. He is currently studying at Hankuk Academy of Foreign Studies in Yongin, Gyeonggi, South Korea. He wants to major molecular biology and biotechnology in the future.



Taeseon Yoon was born in Seoul, Korea, in 1972. He was Ph.D. Candidate degree in Computer education from the Korea University, Seoul, Korea, in 2003. From 1998 to 2003, he was with EJB analyst and SCJP. From 2003 to 2004, he joined the Department of Computer Education, University of Korea, as a Lecturer and Ansan University, as an Adjunct professor. Since December 2004, he has been with the Hankuk Academy of Foreign Studies in Yongin, Gyeonggi, where he is a Computer Science and Statistics Teacher. He was the recipient of the Best Teacher Award of the Science Conference, Gyeonggi-do, Korea, 2013.