

KERNELS FOR PROTEIN HOMOLOGY DETECTION

J. Dylan Spalding

School of Engineering, Computer Science and Mathematics,
University of Exeter

*Submitted by J. Dylan Spalding to the
University of Exeter
as a thesis for the degree of
Doctor of Philosophy in Computer Science
October 2009.*

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other university.

J. Dylan Spalding

Signature.

Abstract

Determining protein sequence similarity is an important task for protein classification and homology detection, which is typically performed using sequence alignment algorithms. Fast and accurate alignment-free kernel based classifiers exist, that treat protein sequences as a “bag of words”. Kernels implicitly map the sequences to a high dimensional feature space, and can be thought of as an inner product between two vectors in that space. This allows an algorithm that can be expressed purely in terms of inner products to be ‘kernelised’, where the algorithm implicitly operates in the kernel’s feature space.

A weighted string kernel, where the weighting is derived using probabilistic methods, is implemented using a binary data representation, and the results reported. Alternative forms of data representation, such as Ising and frequency forms, are implemented and the results discussed. These results are then used to inform the development of a variety of novel kernels for protein sequence comparison.

Alternative forms of classifier are investigated, such as nearest neighbour, support vector machines, and multiple kernel learning. A kernelized Gaussian classifier is derived and tested, which is informative as it returns a score related to the probability of a sequence belonging to a particular classification. Support vector machines are tested with the introduced kernels, and the results compared to alternate classifiers. As similarity can be thought of as having different components, such as composition and position, multiple kernel learning is investigated with the novel kernels developed here.

The results show that a support vector machine, using either single or multiple kernels, is the best classifier for remote protein homology detection out of all the classifiers tested in this thesis.

Contents

Abstract	2
Contents	3
List of Tables	7
List of Figures	9
Notation and Abbreviations	20
1 Introduction	21
1.1 The Biology of Proteins	22
1.1.1 Protein Sequences	22
1.1.2 Protein Structure	23
1.1.3 Protein Classification & Homology	26
1.2 Kernels & Kernel Methods	29
1.2.1 Kernels	29
1.2.2 The Kernel Trick	31
1.2.3 Kernels in Bioinformatics	32
1.3 Description of the protein databases	35
1.4 Thesis Aims	37
2 The PSST Kernel	38
2.1 The PSST Algorithm	38
2.1.1 Derivation of ω_i	40
2.1.2 Investigating the value of k	47
2.1.3 From binary to frequency representation	49
2.1.4 The Kullback Liebler Divergence	50

2.1.5	Methodology	53
2.1.6	Results	54
2.2	Ising representation	61
2.2.1	Results	65
2.3	Data derived ω	66
2.3.1	Model Derivation	67
2.3.2	Methodology	70
2.4	Discussion	73
3	The Jaro Kernel and alternative composition, position, and order kernels	75
3.1	Composition and Position kernels	77
3.1.1	The Composition Kernels	77
3.1.2	The Position Kernels	77
3.2	The Jaro Kernel	78
3.2.1	Methodology	80
3.2.2	Results	82
3.2.3	Discussion	87
3.3	Novel Kernels	88
3.3.1	Linear Kernel	88
3.3.2	Linear Scaled Kernel	89
3.3.3	Order Kernel	89
3.3.4	Jaro Wild Kernel	90
3.3.5	pos_k_n Kernel	91
3.3.6	comp_amino_20_4 Kernel	91
3.3.7	Pos-amino-20-4-k-n Kernel	92
3.3.8	Length Kernel	92
3.3.9	Methodology	93
3.3.10	Results	94

3.3.11	Discussion	101
4	A Gaussian Classifier in Kernel Space	103
4.1	Gaussian Classifier	103
4.1.1	Methodology	105
4.1.2	Implementation	107
4.1.3	Results	109
4.2	Kernelised Gaussian classifier	113
4.2.1	Methodology	117
4.2.2	Results	117
4.2.3	The Kernelized Gaussian Classifier on SCOP version 1.73	124
4.3	A kernelized Linear Classifier	129
4.3.1	Conclusion	133
5	SVM Classification	137
5.1	Support-Vector Machines	138
5.1.1	Methodology	143
5.1.2	Results	145
5.1.3	Discussion	149
5.2	Multiple-Kernel Classifiers	149
5.2.1	Methodology	153
5.2.2	Results	154
5.2.3	Kernel Weights	159
5.2.4	Comparison of the kernels with the state-of-the-art	161
5.2.5	Determination of C	163
5.2.6	Discussion	168
6	Conclusions	170
6.1	Further Work	174
6.2	Conclusion	175

A	Graphs from Chapter 4	176
A.1	Kernelized Gaussian Classifier on SCOP_A	176
A.2	The Kernelized Gaussian Classifier on SCOP version 1.73	183
B	Graphs from Chapter 5	191
B.1	Single kernel SVM	191
B.2	Multiple Kernel Classifiers	199
	Bibliography	202

List of Tables

1.1	Table showing both the single and three letter codes for the 20 proteinogenic amino acids. The last columns shows the amino acid classification, as defined by Lehinger <i>et. al.</i> [48]. Negative, positive, & polar describe the charge on the non-polar R group, with polar being uncharged.	23
1.2	The composition of the sequence database created from SCOP 1.73.	36
1.3	The composition of the SCOP_A sequence database.	36
2.1	Table showing the average values for ρ_i for the miniPRINTS database at various values of k	46
2.2	Table of the maximum accuracy performance and associated value of k for the PSST algorithm with the binary representation and $1/\omega_i$ weighting (S_N), and the frequency representation with $1/\pi_i$ weighting (S_f) for both the SCOP and miniPRINTS datasets. . .	60
2.3	Results for the data-driven algorithm, using the miniPRINTS datasets with $k = 2$, compared to the results for both the binary (S_N) and frequency (S_f) algorithms.	73
3.1	Accuracy of the various Jaro kernels on the SCOP dataset.	82
3.2	ROC50 scores for the various Jaro kernels on the SCOP dataset. .	82
3.3	Amino acids and their new groups in the comp-amino-20-4 kernel based on the classification of their R group, where N is non-polar, P is polar, A is negative, and B is positive.	92
4.1	Table comparing the accuracy of the Gaussian classifier using both $\bar{\bar{\Sigma}}_f$ and $\bar{\bar{\Sigma}}$ at superfamily classification.	106

4.2	Accuracy results for the family + query Gaussian model using the SCOP_A data-set.	110
4.3	Maximum accuracy of the Gaussian classifier for the SCOP_A data-set, using a test set of 100 query sequences.	112
5.1	Table showing the component kernels of the seven novel multiple kernels.	154
5.2	Table showing the kernels and weights (where $\beta_k > 0$) for the G multiple kernel on data-set number 1.	159
B.1	Table showing the kernels used in Figure B.16.	201

List of Figures

1.1	Protein sequence of HBA1_PLEWA from the ALPHAHAEM family from the PRINTS [6] database.	23
1.2	A motif from the alpha-haemoglobin (ALPHAHAEM) family [6, 7]. This family has a total of 5 motifs. The first column is the motif, and the second column is the sequence code for the particular protein sequence within the ALPHAHAEM family.	25
1.3	Diagram of the hierarchial nature of the structural classification of proteins.	25
1.4	Diagram showing the support vectors (circled) and the maximum margin hyperplane for a linear SVM.	34
1.5	Diagram showing how a SVM can use a polynomial kernel (of factor 3) to find the decision boundary.	34
2.1	Diagram showing the theoretical distributions of the mean values of both the true positive (S_{TP}) and true negative (S_R) distributions for the PSST algorithm. The optimization attempts to increase the distance between the means of the two distributions.	41
2.2	Comparison of the Miller & Attwood, variance, and optimal (from Equation 2.1.26) weights. It can be seen that as $\rho_i \rightarrow 0$ they all approximate each other.	46
2.3	Values of $D(n)$ recorded and kernel smoothed at $k = 3$	52
2.4	Values of $D(n)$ recorded and kernel smoothed at $k = 4$	52
2.5	Accuracy of the PSST algorithm with binary representation and standard Miller and Attwood weighting ($1/\omega_i$) at various values of k , for the four versions of the miniPRINTS database.	55

2.6	Accuracy of the PSST algorithm with frequency representation and $1 / \pi_i$ weighting at various values of k , for the four versions of the miniPRINTS database.	56
2.7	Accuracy of both the Binary and Frequency the PSST algorithm on the miniPRINTS database.	57
2.8	Graph showing the ROC50 score for various values of k for the Binary (S_n) PSST algorithm on the four versions of the miniPRINTS dataset.	58
2.9	Graph showing the ROC50 score for various values of k for the Frequency (S_f) PSST algorithm on the four versions of the miniPRINTS dataset.	58
2.10	Accuracy of the PSST algorithm with the binary representation and $1/\omega_i$ weighting (S_N), and the frequency representation with $1/\pi_i$ weighting (S_f).	60
2.11	Graph showing the ROC50 score for various values of k for both the Binary (S_N) and Frequency (S_f) versions of the PSST algorithm on the SCOP dataset.	61
2.12	Graph showing the accuracy for various values of k for the Ising PSST algorithm on the four versions of the miniPRINTS dataset.	66
2.13	Example of a Q–Q plot of eigenvalues from a random versus real database, with $k=2$ using the mid-length miniPRINTS database. This graph suggests that only eigenvalues > 1 should be used to calculate the inverse.	72
3.1	Graph comparing the ROC-n scores for PSI-BLAST and the Jaro_2 kernel at $k = 2$ for various values of n	83
3.2	Graph comparing the accuracy scores of the wildcard kernel with 1 wildcard and various values of λ on the SCOP dataset to the jaro_4 kernel.	85

3.3	Graph comparing the accuracy scores of the wildcard kernel with 2 wildcards and various values of λ on the SCOP dataset to the jaro_4 kernel.	85
3.4	Graph comparing the ROC50 scores of the wildcard kernel with 1 wildcard and various values of λ on the SCOP dataset to the jaro_4 kernel.	86
3.5	Graph comparing the ROC50 scores of the wildcard kernel with 2 wildcards and various values of λ on the SCOP dataset to the jaro_4 kernel.	86
3.6	Graph showing the accuracy scores of the Jaro and linear kernels using the SCOP dataset.	94
3.7	Graph showing the ROC50 scores of the Jaro and linear kernels using the SCOP dataset.	94
3.8	The accuracy scores achieved by the position kernels on the SCOP dataset.	95
3.9	The ROC50 scores achieved by the position kernels on the SCOP dataset.	95
3.10	Graph showing the accuracy scores of the comp_amino_20_4 kernel and the wildcard kernel with 1 wildcard and various values of λ on the SCOP dataset.	96
3.11	Graph showing the ROC50 scores of the comp_amino_20_4 kernel and the wildcard kernel with 1 wildcard and various values of λ on the SCOP dataset.	96
3.12	Graph showing the accuracy scores of the wildcard kernel with 2 wildcards and various values of λ on the SCOP dataset.	97
3.13	Graph showing the ROC50 scores of the wildcard kernel with 2 wildcards and various values of λ on the SCOP dataset.	97
3.14	Bar chart showing the accuracy of the various kernels at superfamily classification using the nearest neighbour classifier.	99

3.15	Bar chart showing the ROC50 scores of the various kernels at superfamily classification using the nearest neighbour classifier. . . .	100
4.1	Graph showing the results by family for both the standard model and the family + query model.	110
4.2	Graph showing how the performance of the standard Gaussian classifier at family classification, with varying numbers of eigen-pairs used in the calculation of $\bar{\Sigma}^+$	112
4.3	Graph showing how the performance of the standard Gaussian classifier at superfamily classification, with varying numbers of eigen-pairs used in the calculation of $\bar{\Sigma}^+$	112
4.4	Performance of the linear kernel within a k GMM with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	118
4.5	Performance of the linear kernel within a k GMM with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	118
4.6	Performance of the linear-scaled kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	119
4.7	Performance of the linear-scaled kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	119
4.8	Performance of the jaro kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	121
4.9	Performance of the jaro kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	121

4.10	Performance of the pos_k_1 kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	122
4.11	Performance of the pos_k_1 kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	122
4.12	Performance of the pos_k_2 kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	123
4.13	Performance of the pos_k_2 kernel within a kernelised Gaussian classifier with varying numbers of eigen-pairs used in the calculation of Σ^{-1}	123
4.14	Bar chart showing the best performing wildcard and non-wildcard kernels for each value of k in a kernelized Gaussian classifier. . . .	124
4.15	Bar chart showing the best accuracy achieved at each value of k and the respective kernel and the accuracy of PSI-BLAST. The error bars indicate the 95% confidence intervals.	127
4.16	Bar chart showing the best ROC50 score achieved at each value of k and the respective kernel and the accuracy of PSI-BLAST. The error bars indicate the 95% confidence intervals.	127
4.17	Graph showing the accuracy of each of the kernels at their optimal k against the maximum number of eigenpairs used to calculate the covariance matrix for the kernelized Gaussian classifier. The error bars indicate the 95% confidence intervals.	128
4.18	Graph showing the accuracy of each of the kernels at their optimal k against the maximum number of eigenpairs used to calculate the covariance matrix for the kernelized Gaussian classifier. The error bars indicate the 95% confidence intervals.	129

4.19	Accuracy of the Gaussian classifier with 4 kernels used as a binary and multi-class classifier. The error bars indicate 95% confidence intervals and d is the number of dimensions in $\bar{\Sigma}$	132
4.20	AROC50_B scores of the Gaussian classifier with 4 kernels used as a binary and multi-class classifier. The error bars indicate 95% confidence intervals and d is the number of dimensions in $\bar{\Sigma}$	132
4.21	Accuracy of the binary Gaussian classifier with the 4 kernels versus the 4 optimal binary SVM classifiers and PSI-BLAST. The error bars indicate 95% confidence intervals.	134
4.22	AROC50_B scores of the Gaussian classifier with the 4 kernels versus the 4 optimal binary SVM classifier and PSI-BLAST. The error bars indicate 95% confidence intervals.	134
5.1	Diagram illustrating the hyperplane separating two linearly separable classes.	139
5.2	Diagram illustrating the hyperplane separating two non-linearly separable classes. ξ is the slack variable for the mis-classified point.	141
5.3	Graph showing the accuracy of the Jaro, Jaro_wild, linear, linear_scaled, wild_1 ($\lambda = 0.75$), and wild_2 ($\lambda = 0.75$) kernels within a SVM versus PSI-BLAST. The error bars indicate the 95% confidence intervals.	145
5.4	Graph showing the accuracy of the pos_k_0, pos_k_1, length, order, pos_amino_20_4_0, and pos_amino_20_4_1 kernels within a SVM versus PSI-BLAST. The error bars indicate the 95% confidence intervals.	145
5.5	Graph showing the ROC50 performance of the linear SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	146

5.6	Bar chart showing the ROC50 performance of the top 25 scoring kernels within a SVM. The error bars indicate 95% confidence intervals.	148
5.7	Bar chart showing the AROC50_B performance of the multiple kernels within a SVM and the top 5 performing single kernels, shown in red. The error bars indicate 95% confidence intervals. Table 5.1 details the kernel types used.	157
5.8	Bar chart showing the accuracy performance of the multiple kernels within a SVM and the top 5 performing single kernels, shown in red. The error bars indicate 95% confidence intervals.	158
5.9	Bar chart showing the frequency with which a multiple kernel with one to 5 kernels occurs for the G multiple kernel across all 12 data-sets.	160
5.10	Box graph showing the spread of weights for each kernel in all runs with the G multiple kernel. 1 is the dominant kernel and 5 is the least dominant kernel.	161
5.11	Bar chart showing the AROC50_B performance of the profile, SW-PSSM, and top- n -gram kernels versus the best performing eight kernels and PSI-BLAST. The red bars indicate single experiments using SCOP version 1.53, while the blue bars the mean performance from 12 experiments using SCOP version 1.73.	162
5.12	Bar chart showing the accuracy performance of the SVM with optimized C versus the default C	166
5.13	Bar chart showing the AROC50_B performance of the SVM with optimized C versus the default C	166
5.14	Bar chart showing the accuracy performance of the MKL with optimized C versus the default C	167
5.15	Bar chart showing the AROC50_B performance of the MKL with optimized C versus the default C	167

A.1	Performance of all the non-wildcard kernels within a kernelised Gaussian classifier with $k=2$ at superfamily classification.	177
A.2	Performance of all the non-wildcard kernels within a kernelised Gaussian classifier with $k=3$ at superfamily classification.	177
A.3	Performance of all the non-wildcard kernels within a kernelised Gaussian classifier with $k=4$ at superfamily classification.	178
A.4	Performance of all the non-wildcard kernels within a kernelised Gaussian classifier with $k=5$ at superfamily classification.	178
A.5	Performance of all the single-wildcard kernels within a kernelised Gaussian classifier with $k=2$ at superfamily classification.	179
A.6	Performance of all the single-wildcard kernels within a kernelised Gaussian classifier with $k=3$ at superfamily classification.	179
A.7	Performance of all the single-wildcard kernels within a kernelised Gaussian classifier with $k=4$ at superfamily classification.	180
A.8	Performance of all the single-wildcard kernels within a kernelised Gaussian classifier with $k=5$ at superfamily classification.	180
A.9	Performance of all the double-wildcard kernels within a kernelised Gaussian classifier with $k=2$ at superfamily classification.	181
A.10	Performance of all the double-wildcard kernels within a kernelised Gaussian classifier with $k=3$ at superfamily classification.	181
A.11	Performance of all the double-wildcard kernels within a kernelised Gaussian classifier with $k=4$ at superfamily classification.	182
A.12	Performance of all the double-wildcard kernels within a kernelised Gaussian classifier with $k=5$ at superfamily classification.	182
A.13	Graph showing the accuracy of all the kernels within a kernelized gaussian classifier with $k=2$. The error bars indicate the 95% confidence intervals.	185

A.14	Graph showing the accuracy of all the kernels within a kernelized gaussian classifier with $k=3$. The error bars indicate the 95% confidence intervals.	185
A.15	Graph showing the accuracy of all the kernels within a kernelized gaussian classifier with $k=4$. The error bars indicate the 95% confidence intervals.	186
A.16	Graph showing the accuracy of all the kernels within a kernelized gaussian classifier with $k=5$. The error bars indicate the 95% confidence intervals.	186
A.17	Graph showing the accuracy of all the kernels within a kernelized gaussian classifier with $k=6$. The error bars indicate the 95% confidence intervals.	187
A.18	Graph showing the accuracy of all the kernels within a kernelized gaussian classifier with $k=7$. The error bars indicate the 95% confidence intervals.	187
A.19	Graph showing the ROC50 scores of all the kernels within a kernelized gaussian classifier with $k=2$. The error bars indicate 95% confidence intervals.	188
A.20	Graph showing the ROC50 scores of all the kernels within a kernelized gaussian classifier with $k=3$. The error bars indicate 95% confidence intervals.	188
A.21	Graph showing the ROC50 scores of all the kernels within a kernelized gaussian classifier with $k=4$. The error bars indicate 95% confidence intervals.	189
A.22	Graph showing the ROC50 scores of all the kernels within a kernelized gaussian classifier with $k=5$. The error bars indicate 95% confidence intervals.	189

A.23	Graph showing the ROC50 scores of all the kernels within a kernelized gaussian classifier with $k=6$. The error bars indicate 95% confidence intervals.	190
A.24	Graph showing the ROC50 scores of all the kernels within a kernelized gaussian classifier with $k=7$. The error bars indicate 95% confidence intervals.	190
B.1	Graph showing the ROC50 performance of the linear SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	193
B.2	Graph showing the ROC50 performance of the linear_scaled SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	193
B.3	Graph showing the ROC50 performance of the Jaro SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	194
B.4	Graph showing the ROC50 performance of the Jaro_wild SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	194
B.5	Graph showing the ROC50 performance of the wild_1 ($\lambda = 0.75$) SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	195
B.6	Graph showing the ROC50 performance of the wild_2 ($\lambda = 0.75$) SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	195
B.7	Graph showing the ROC50 performance of the comp_amino_20_4 SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	196
B.8	Graph showing the ROC50 performance of the order SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	196

B.9	Graph showing the ROC50 performance of the pos_k_0 SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	197
B.10	Graph showing the ROC50 performance of the pos_k_1 SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	197
B.11	Graph showing the ROC50 performance of the pos_amino_20_4_0 SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	198
B.12	Graph showing the ROC50 performance of the pos_amino_20_4_1 SVM and PSI-BLAST. The error bars indicate the 95% confidence intervals.	198
B.13	Graph showing the ROC50 performance of the multiple kernel SVM with six different kernels. The error bars indicate the 95% confidence intervals.	200
B.14	Graph showing the ROC50 performance of the multiple kernel SVM with six different kernels. The error bars indicate the 95% confidence intervals.	200
B.15	Graph showing the ROC50 performance of the multiple kernel SVM with all kernels at each value of k . The error bars indicate the 95% confidence intervals.	201
B.16	Graph showing the ROC50 performance of the multiple kernel SVM with a selection of kernels at all values of k . See Table B.1 for the legend key. The error bars indicate the 95% confidence intervals.	201

Notation and Abbreviations

λ	Eigenvalue or wildcard penalty
$\bar{\bar{\Lambda}}$	Matrix of singular values from SVD
$\bar{\bar{\Sigma}}$	Covariance matrix
$\bar{\bar{U}}$	$m \times m$ unitary matrix from SVD of a $m \times n$ matrix
$\bar{\bar{V}}$	$n \times n$ unitary matrix from SVD of a $m \times n$ matrix
$\bar{\bar{X}}$	Matrix X
$\bar{\bar{X}}^+$	Pseudo-inverse of matrix X
μ	Arithmetic mean
ν	Eigenvector
\mathbf{X}	Vector X
\mathbf{X}^\top	Vector X transpose
k	Length of the k -mer
k -mer	Contiguous oligomer, or subsequence, of length k
k NN	k nearest neighbour classifier
$Tr \bar{\bar{X}}$	Trace of matrix X
AROC	Area under Receiver Operating Characteristic curve
GMM	Gaussian Mixture Model
kGMM	kernelised Gaussian Mixture Model
kPCA	kernel Principal Component Analysis
NN	Nearest neighbour classifier
PCA	Principal Component Analysis
PSST	Probabilistic Sequence Search Tool
ROC	Receiver Operating Characteristic
SVD	Singular Value Decomposition
wild_n	Wildcard kernel with n wildcards