Capturing the Right Signals: String Kernels for Protein Sequence Analysis

George Karypis
Department of Computer Science & Engineering, Digital Technology Center, & Army HPC Research Center
University of Minnesota
Motivation &
Background
Biomolecular Sequences

- DNA and protein sequences are often the primary types of data that is available for various organisms and genes.
  - Today such sequences can be obtained easily and inexpensively.
- They represent the starting point for a series of analysis whose goal is to gain insights on the role and function of the underlying biomolecule.
In biomolecular sequences (DNA, RNA, or amino acid sequences), high similarity usually implies significant functional or structural similarity.

Evolution reuses, builds on, duplicates, and modifies successful structures.

Evolutionarily and functionally related molecular strings can differ significantly throughout much of the string and yet preserve the same three-dimensional structure(s), or the same two-dimensional substructure(s) (motifs, domains), or the same active sites, or the same or related dispersed residues (DNA or amino acid).
Sequence-Driven Analysis Tasks

- DNA-driven analysis
  - Gene identification
    - introns, exons, splicing cites, etc.
  - Regulatory element
- Protein-driven analysis
  - Protein family classification
  - Structure & organization
    - Domain identification, secondary & tertiary structure prediction, fold recognition
  - Protein interactions
    - protein-protein, protein-nucleic-acid

Common to these tasks is the need to derive/predict some higher-level information from the primary sequences.

These problems fall under the sequence-based
- prediction/classification algorithms and annotation algorithms in data mining and machine learning.
Machine Learning Approaches

- Nearest-neighbor classification
  - Local/global sequence alignment, BLAST, etc.
- Markov Model-based approaches
  - Markov chains, Hidden Markov Models, Profile HMMs, etc.
- Discriminating Models
  - Neural networks, Support Vector Machines, etc.
- Arbitrary combination of the above basic approaches.
It is all about the signals...

Need to tightly integrate ML with the characteristics of the various problems.
Why do we Expect Machine Learning Approaches to Work?

- Evolution reuses, builds on, duplicates, and modifies successful structures.
  - High sequence similarity usually implies significant functional or structural similarity.
    - However, in many cases the same is true for sequences with low or no sequence similarity.

- Many of the underlying processes are driven by well-understood physico-chemical principles.
How hard is the problem that we are trying to solve via ML techniques?

- Reverse engineer evolution
- Reverse engineer physics/chemistry
- Reason over an exponentially large input space
Right Signals can Help

The only way to develop effective ML techniques for sequence analysis is to:

- Not reverse-engineer evolution, but
  - identify & model how evolution manifests itself by preserving/modifying the biomolecular sequences of the various organisms

- Not reverse-engineer physics/chemistry, but
  - identify & model how the various physico-chemical principles relevant for the problem manifest themselves in terms of sequence-base signals

- Not explore the exponentially large input space, but
  - identify and focus on the regions that matter.
Gene Prediction

A direct example
Eukaryotic Gene Characteristics

- Gene Structure
  - It is made of a number of exons (4-8) separated by introns.
    - Average exon size of 150bp
    - Average intron size of 30Kbp
  - A gene starts with ATG and ends with a stop codon.
  - Average length of 1Kbp nucleotides

- Most of the DNA does not code for a gene—less than 30%.
There is a distinct nucleotide usage pattern in the exons compared to the intron/non-coding regions. This is a result of the organisms codon use bias.

- CpG islands upstream of the genes indicating a regulatory region.
- PolyA region downstream of the gene.
- There are weak sequence consensuses at the intron-exon & exon-intron boundaries.

![Diagram of gene prediction signals]

- **Donor site**: 5’—AG GTAAGT—3’
  - well conserved

- **Acceptor site**: 5’—Py Py Py Py Py Py *CAG G—3’
  - well conserved
  - T or C
Effective gene prediction algorithms:
- Capture the various known signals using different “sensors”.
- Combine these signals based on the overall gene structure.
Detecting and Fusing Signals

- These sensors are implemented using different models:
  - Exon: interpolated/inhomogeneous Markov chain
  - Acceptor/Donor sites: Profile HMMs or PSSMs
  - Intron: random model

- The sensors are combined together using different approaches:
  - Dynamic programming
  - Hidden Markov model
  - Neural networks

- There is a notion of “parsing” the DNA sequence to identify its various components:
  - Annotation
GENSCAN

- It couples the various sensors using an HMM.
  - Individual sensors are implemented using probabilistic models
    - profile HMMs
    - IIMM
    - background probabilities

- Main elements
  - $E_{\text{init}}, E_{\text{term}}$ models for initial and final exons
  - $I_0, I_1, I_2$ models for introns
    - subscripts use to model where in the last codon of the previous exon the intro starts
      - codon phase
      - $I_1$ means that the intron starts after the first base of the last codon
  - $E_0, E_1, E_2$ models for internal exons
    - subscripts capture the same codon phase and this is why we can only transition from $I_i$ to $E_i$.

- It has been shown to be very effective.
Sequence Alignment

A not so direct example
Determining the Similarity

In biomolecular sequences (DNA, RNA, or amino acid sequences), high similarity usually implies significant functional or structural similarity.

Evolution reuses, builds on, duplicates, and modifies successful structures.

- Similarity is determined by looking at how well the sequences align with each other.
- Key components:
  - The type of alignment.
  - The method used to score the alignment.
An one-to-one mapping between the positions of the two sequences.

Depending on the application, different types of alignments are used:
- global alignments
- local alignments, with/without gaps, end-space free variants, etc.
The quality of an alignment is determined by scoring the aligned positions.

- There is a preference in aligning certain characters against each other.

Higher scores are better

- The alignment algorithms try to maximize this score.

The scoring is usually achieved by using alphabet-weighted scoring matrices.

There is a high degree of inter-dependency between scoring matrices and the specific classification problem.

- Certain matrices are better suited for some problems than others!

\[
\text{Alignment Score} = \sum_{i=1}^{\mid X' \mid} S(x'_i, y'_i)
\]
Scoring Matrices

- An $n \times n$ matrix $S$, where $n$ is the alphabet size
  - Play the most critical role in determining how two sequences align with each other.
  - The alignment algorithm tries to find the alignment that optimizes the score.
  - Different scoring matrices can lead to dramatically different alignments.

- $S[a,b]$ is the similarity score between amino acids ‘a’ and ‘b’ and is a measure of the degree to which ‘a’ and ‘b’ can occur aligned against each other in correct alignments.
  - The scores are used to encode the likelihood of ‘a’ to be evolved into ‘b’ while still retaining the biological role in question.
  - “Biological role in question”?  
    - Depends on the particular classification problem.
      - Be part of the same protein family, have the same secondary structure, adapt the same fold, etc.

- There are many problem-specific scoring matrices, and in general constructing one for your own problem will improve your results!
Support Vector Machines, Kernels, & String Kernels
Support Vector Machines

- A widely used machine learning tool that learns a maximum margin linear binary classifier.

- A binary classifier. Within the machine learning community, classification algorithms based on discriminative models, especially kernel-based support vector machines, have become very popular as they produce very accurate classifiers.

- The key idea of these discriminative models is to build a model that separates the various classes
  - discriminate between classes as opposed to describing/generating each class.

- In the last 5+ years, similar models have been developed for sequence classification.
  - The work focused on developing string-based kernel functions that can be used within the context of support vector machines.
Linear Separators

\[ f(x) = \text{sign}(\langle w \cdot x \rangle + b) \]

There are many linear separators. Which one is better?
Maximum Margin Classifiers

How about finding the linear classifier that maximizes the margin?

- The width that the boundary could be increased by before hitting any points.

support vectors

margin
Support Vector Machines

- A widely used machine learning tool that learns a maximum margin linear binary classifier.
- The classification hyperplane is defined in terms of the support vectors:
- It can find non-linear hyperplanes via the use of the kernel-trick.

\[ f(x) = \sum_{i \in SV} y_i \alpha_i \langle x_i \cdot x \rangle + b \]

Kernel-based formulation

\[ f(x) = \sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \]

The kernel function can be considered as a measure of similarity between objects and used to encode key information about the classification problem.
String Kernels

- A class of kernels designed to operate on sequence data.
- Two widely used approaches for obtaining string kernels
  - Feature-space generation—either explicit or implicit
    - $k$-mer kernel
      - each dimension corresponds to a distinct $k$-mer
      - each sequence is represented as a frequency vector of the various $k$-mers that it contains.
    - $k$-mer mismatch kernel
      - extends the $k$-mer kernel to allow for up to $m$ mismatches
    - fisher-kernel
      - It is derived from a profile HMM
        - each dimension corresponds to the emission probabilities of the various states
    - pairwise-kernel
      - Each dimension corresponds to a sequence in the training set
      - The value to a particular dimension corresponds to the sequence similarity score
  - Similarity-based approaches—derive a kernel from a biologically relevant similarity measure
    - Convolution kernel
      - Total alignment score of all pairwise local alignments
Protein Structural Bioinformatics
Protein Structure

Different Levels of Protein Structure

(a) Primary structure
- Ala - Glu - Val - Thr - Asp - Pro - Gly -

(b) Secondary structure
- $\alpha$ helix
- $\beta$ sheet

(c) Tertiary structure
- Domain

(d) Quaternary structure
Determining Protein Structure

- Experimental approaches
  - X-ray crystallography
    - The most reliable and accurate approach for determining the 3D structure of a protein.
    - Time consuming, expensive, and certain proteins are notoriously hard to crystallize.
  - NMR spectroscopy
    - Less expensive and faster that X-ray crystallography.
    - Primary suited for smaller size proteins.

- Computational Approaches
  - Predict the secondary, tertiary, and quaternary structure of a protein from its primary sequence.
    - Computational Biology
      - *Ab initio* structure prediction via molecular dynamics computations.
    - Bioinformatics/Knowledge-based Approaches
      - Employ various machine learning techniques to predict the structure of proteins by leveraging existing protein structure information.
PDB—Protein Data Bank

http://www.rcsb.org/pdb/

The rate of new “folds” discovery is decreasing

this number is significantly smaller than the million+ of known proteins

The rate of new “folds” discovery is decreasing
Prediction Problems

- There are a number of different protein structure prediction problems:
  - Domain prediction
    - Identify the regions of the protein that fold into “independent” compact structures.
  - Secondary structure prediction
    - Identify the local structural regions of the protein.
  - Contact-map prediction
    - Identify the pairs of residues that will be in “contact” in the protein’s 3D structure.
  - Fold recognition
    - Determine whether or not the protein’s tertiary structure will adopt a shape that is similar to that of a known 3D structure.
  - New-fold prediction
    - Determine the tertiary structure of a protein whose shape has not yet been encountered.

- There is a biannual competition evaluating the current state-of-the-art of the various prediction approaches
  - Critical Assessment of Structure Prediction (CASP).
Key facts about proteins

- Proteins fold in very compact/tight structures
  - The core/buried portion of the protein consists primarily of hydrophobic residues
  - The surface/exposed portion of the protein consists primarily of polar residues
Key Signals in Structural Bioinformatics

- Strong Connection between Sequence and Structural Conservation
  - However, it is a family game…
    - Analyzing a protein by itself does not provide sufficient information as to what has been conserved through evolution.
      - Protein families, profiles, and position specific scoring matrices (PSSM).

- Sequence-based signals occur at different levels and/or granularities
  - Certain aspects of the structure is determined by local information whereas other depend on interactions involving non-local sequence segments.

- Complex conservation signals
  - Covariant mutations.
Remote Homology Prediction & Fold Recognition
Remote Homology Prediction

Problem Definition

- The goal is to determine whether or not a pair of proteins are homologous (i.e., sharing a common origin and potentially similar functionality) in cases in which their amino acid sequence has significantly diverged through evolution.
  - Sequences are usually less than 30% similar.

- Existing state-of-the-art approaches utilize various techniques ranging from
  - Sophisticated profile-based pairwise alignment schemes
  - Profile hidden Markov models
  - Discriminative neural network and/or support vector machines models
Fold Recognition

Problem Definition

- Fold Recognition
  - The goal is to determine whether or not the three dimensional structure of a protein will adopt a *shape* that is similar to one of the known shapes adapted by proteins whose 3D structure has been experimentally determined.
    - Existing experimentally determined protein structures have been classified in about ~1000 different shapes (i.e., folds).
  - Existing state-of-the-art approaches rely on techniques similar to those for remote homology prediction and in addition to primary sequence information also utilize predicted local structural features such as secondary structure and solvent accessibility, and utilize fold profiles often computed via structural alignment methods.
Source of Training Data

- The training data is obtained by combining known structures in PDB with SCOP
  - SCOP provides a classification of PDB’s domains into four primary levels:
    - Class, Fold, Superfamily, & Family

![SCOP Classification Tree]

- Class: α, β, α/β, α+β
- Fold: Rossmann fold, Flavodoxin-like, α/β barrel, TIM, Trp biosynthesis, Glycosyltransferase, Rubisco (C)
- Superfamily: β-Galactosidase (3), β-Glucanase, α-Amylase (N), β-Amylase
- Family: Acid α-amylase, Taka-amylase, Cyclodextrin glycosyltransferase
- Domain: 2aaa, 6taa, 2taa, 1cdg, 1cgt, 1cgu

**High structural similarity**

**Low sequence similarity**

**High sequence similarity**
Our Approach

- Learn a yes/no classifier for each of the folds/super-families using SVM.
- Assign a sequence to a class based on its distance from the hyper-plane for the various classifiers.
- Our research focused on how to design effective kernel functions for these two problems.
Similarity-based Kernel Functions

- Developed two novel classes of directly constructed kernel functions that combine
  - Automatically generated sequence profiles
    - Profiles were constructed using PSI-BLAST.
  - Effective schemes for scoring the aligned profile positions
    - The scoring scheme combines both position specific scoring and position specific frequency matrices.
  - New and existing approaches for determining the similarity between pairs of protein sequences.
    - Window-based kernels
    - Local alignment-based kernels
      - The similarity is determined using a Smith-Waterman alignment
        - The gap opening/extension and zero-shift parameters of the scoring system have been optimized for the problem at hand.
**Window based kernels:**

- **k-mer Concept:**
  
  A *k*-mer is a contiguous subsequence of length *k*.

The similarity is determined by considering sequence windows of size $2w + 1$ ($w$mers) centered at each residue.
All Fixed k-mer Scoring Scheme

- Sum up all the $k$-mer residue pairs between two sequences and use it as a similarity measure.
- Measures the “overall” similarity between two sequences.

$$AF\text{-PSSM}_{X,Y}(w) = \sum_{(wmer_X(i), wmer_Y(j)) \in P_w} wscore_{X,Y}(i, j).$$
Best Fixed k-mer Scoring Scheme

- Identify the highest scoring non-overlapping k-mers and use the sum of their scores as the similarity measure.
- Measures the similarity by taking into account the best local regions of each protein, irrespective of their overall order.

\[
\text{BF-PSSM}_{X,Y}(w) = \sum_{(\text{wmer}(X,i), \text{wmer}(Y,j)) \in \mathcal{P}_w} \text{wscore}_{X,Y}(i,j).
\]
Best Variable $k$-mer Scoring Scheme

- This scheme relaxes the $k$-mer length
- Picks $k$-mers from one to the max specified by the user for each position
We evaluated the various direct kernels on a standard benchmark for remote homology prediction and fold recognition derived from SCOP.

Remote homology prediction was simulated by learning a model for a particular superfamily by using sequences from only one of its families as positive train and from another one of its families as positive test.

- 54 different classification problems with at least 10 positive training examples and 5 positive test examples.
A similar approach was used to derive a training/test set for fold recognition:
- Fold recognition was simulated by learning a model for a particular fold by using sequences from only one of its superfamilies as positive train and from another one of its superfamilies as positive test
- 23 different classification problems with at least 10 positive training examples and 5 positive test examples

The performance was assessed using ROC50 values, which is the area under the ROC curve up to the first 50 false positives and provides a good operational measure of the classifiers performance.
Results: Remote Homology

<table>
<thead>
<tr>
<th>Kernel</th>
<th>ROC</th>
<th>ROC50</th>
<th>mRFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-Fisher</td>
<td>0.773</td>
<td>0.250</td>
<td>0.204</td>
</tr>
<tr>
<td>SVM-Pairwise</td>
<td>0.896</td>
<td>0.464</td>
<td>0.084</td>
</tr>
<tr>
<td>LA-eig(β = 0.2)</td>
<td>0.923</td>
<td>0.661</td>
<td>0.064</td>
</tr>
<tr>
<td>LA-eig(β = 0.5)</td>
<td>0.925</td>
<td>0.649</td>
<td>0.054</td>
</tr>
<tr>
<td>SVM-HMMSTR-Ave</td>
<td>0.872</td>
<td>0.400</td>
<td>0.084</td>
</tr>
<tr>
<td>Mismatch</td>
<td>0.974</td>
<td>0.756</td>
<td>0.013</td>
</tr>
<tr>
<td>Profile(4,6)</td>
<td>0.980</td>
<td>0.794</td>
<td>0.010</td>
</tr>
<tr>
<td>Profile(5,7.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- AF-PSSM(2)                         | 0.978| 0.816 | 0.013|
- BF-PSSM(2)                         | 0.980| 0.854 | 0.015|
- BV-PSSM(2)                         | 0.973| 0.855 | 0.018|
- SW-PSSM(3,0,0.750,1.50)            | 0.982| 0.904 | 0.015|
- AF-GSM(6)                          | 0.926| 0.549 | 0.048|
- BF-GSM(6)                          | 0.934| 0.669 | 0.053|
- BV-GSM(6)                          | 0.930| 0.666 | 0.052|
- SW-GSM(B62,5.0,1.0,0.5)            | 0.948| 0.711 | 0.039|

![Graph showing ROC and ROC50 values for different kernels.](Image)
## Results: Fold Recognition

<table>
<thead>
<tr>
<th>Kernel</th>
<th>ROC</th>
<th>ROC50</th>
<th>mRFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA-eig($\beta = 0.2$)</td>
<td>0.847</td>
<td>0.212</td>
<td>0.129</td>
</tr>
<tr>
<td>LA-eig($\beta = 0.5$)</td>
<td>0.771</td>
<td>0.172</td>
<td>0.193</td>
</tr>
<tr>
<td>Profile(4.6)</td>
<td>0.912</td>
<td>0.305</td>
<td>0.071</td>
</tr>
<tr>
<td>Profile(5.7.5)</td>
<td>0.924</td>
<td>0.314</td>
<td>0.069</td>
</tr>
<tr>
<td>AF-PSSM(4)</td>
<td>0.911</td>
<td>0.374</td>
<td>0.067</td>
</tr>
<tr>
<td>BF-PSSM(4)</td>
<td>0.918</td>
<td>0.414</td>
<td>0.060</td>
</tr>
<tr>
<td>BV-PSSM(4)</td>
<td>0.941</td>
<td>0.481</td>
<td>0.043</td>
</tr>
<tr>
<td>SW-PSSM(3.0,0.750,2.0)</td>
<td>0.936</td>
<td>0.571</td>
<td>0.054</td>
</tr>
<tr>
<td>AF-GSM(6)</td>
<td>0.770</td>
<td>0.197</td>
<td>0.217</td>
</tr>
<tr>
<td>BF-GSM(6)</td>
<td>0.822</td>
<td>0.240</td>
<td>0.157</td>
</tr>
<tr>
<td>BV-GSM(7)</td>
<td>0.845</td>
<td>0.244</td>
<td>0.133</td>
</tr>
<tr>
<td>SW-GSM(B62,5,1.0,0.5)</td>
<td>0.826</td>
<td>0.223</td>
<td>0.176</td>
</tr>
</tbody>
</table>
Protein Secondary Structure Prediction: YASSPP Algorithm

Can anybody guess what YASSPP stands for?
Protein Secondary Structure

- Secondary structure elements:
  - alpha-helix, beta-strand, coil

- alpha-helices & beta-strands
  - 50% of the residues are in alpha-helices and beta-strands.
  - Local structural regions of a protein.
  - Basic building blocks of the overall protein structure.
  - Formed early during the folding process.
  - Exhibit little flexibility with respect to each other.
alpha-Helix (H)

- 3.6 residues per turn
- 5-40 residues in length
  - 10 residues on the average
  - about 3 turns
- One phase of the helix is typically positioned against the hydrophobic core and the other interacts with the solvent
  - leads to certain pattern of hydrophobic and polar residues use
- Proline residues do not occur in middle segments
  - can appear in the first two positions of the helices
- A alpha-helix specific amino acid composition
beta-strands fold into beta-sheets that consist of a number of beta-strands connected together side-by-side
- Antiparallel & Parallel beta-sheets
- beta-strands are typically shorter than a-helices
  - as small as 2-3 residues
- a beta-strand specific amino acid composition
  - somewhat less strong than that for alpha-helices

The hydrophobic nature of each strand is different depending on whether or not it is internal or external to the sheet.

<table>
<thead>
<tr>
<th></th>
<th>Edge strand</th>
<th>Buried strand</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>○ ● ○ ● ● ○ ● ○ ●</td>
<td>○ ● ● ● ● ● ○ ○</td>
</tr>
<tr>
<td>hydrophobic residues</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Coils (C)

- Not well-defined geometric structure
  - lines/loops
- Highly variable regions of sequence conservation when viewed in multiple sequence alignments
- They contain a high proportion of small polar residues
  - they tend to be at the surface/exposed side of the protein
YASSPP’s prediction framework:

- Utilizes a pairs of models referred to as sequence-to-structure and sequence+structure-to-structure models.
- Each pair of models consists of three binary SVM-based models for each of the C, E, and H states.
- These models are trained using one-vs-rest framework.

Sequence ➔ Structure

- Computes for each sequence position a weight indicating the likelihood that this position will be in one of the three secondary structure states \{C, E, H\}.
- For each position, these weights are determined by considering the amino acid composition of a local sequence window centered around it.

Sequence+Structure ➔ Structure

- Computes for each sequence position the predicted secondary structure state from one of the \{C, E, H\} states.
- These predictions are determined by considering both the amino acid composition of the local sequence window and the predictions computed by the sequence-to-structure model.
YASSPP’s Input Coding and Structure

Position Specific Scoring Matrix

BLOSUM62

Sequence ➔ Structure Model

H vs Rest
E vs Rest
C vs Rest

Sequence+Structure ➔ Structure Model

H vs Rest
E vs Rest
C vs Rest

Input Vector: 15x(20+20+3)

final prediction

Input Vector: 15x(20+20)

Input Vector: 15x(20+20+3)
YASSPP uses a kernel function that combines a normalized second order kernel function with an exponential function.

\[
K(x, y) = \exp \left( 1.0 + \frac{K_1(x, y)}{\sqrt{K_1(x, x) K_1(y, y)}} \right)
\]

\[
K_1(x, y) = K_2^{ss}(x, y) + (K_2^{ss}(x, y))^2
\]

This kernel function captures the primary information for each position as well as their inter-dependencies.

The contribution of each position is inversely related to its distance from the central position.
YASSPP was evaluated on a set of proteins obtained from the EVA server, which performs a real-time evaluation of secondary structure prediction performance.

Its performance, measured via many standard assessment measures is superior to that obtained by all existing state-of-the-art secondary structure prediction algorithms.

- It achieves a Q3 score of 79.34% and an SOV score of 78.65% which are higher than the next best-performing scheme by 2.4% and 3.6%, respectively.
The combined use of position- and non-position specific information improves the prediction accuracy in cases in which PSI-BLAST failed to generate correct alignments for certain positions of the sequence.

<table>
<thead>
<tr>
<th></th>
<th>$P &amp; PB$</th>
<th>$P &amp; \neg PB$</th>
<th>$\neg P &amp; PB$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w$</td>
<td>$C$</td>
<td>$E$</td>
<td>$H$</td>
</tr>
<tr>
<td>0</td>
<td>0.71</td>
<td>0.75</td>
<td>0.67</td>
</tr>
<tr>
<td>1</td>
<td>0.70</td>
<td>0.75</td>
<td>0.66</td>
</tr>
<tr>
<td>2</td>
<td>0.69</td>
<td>0.75</td>
<td>0.66</td>
</tr>
<tr>
<td>3</td>
<td>0.69</td>
<td>0.75</td>
<td>0.67</td>
</tr>
</tbody>
</table>

The average information per position of different length $w$mers centered at each residue that was correctly predicted by both methods ($P \& PB$), correctly predicted only by YASSPP $P_{+PS}$ ($P \& \neg PB$), and correctly predicted only by YASSPP $PB_{+PBS}$ ($\neg P \& PB$). The results are presented based on the secondary structure state of the central residue. The $w = 0$ results correspond to the $w$mer consisting of just the position itself. The average information for longer $w$mers was computed by first computing the average information for each $w$mer and then reporting the average of these averages.
The YASSPP algorithm is available via a web-accessible server at http://yasspp.cs.umn.edu
SVMs with appropriately designed string kernels are becoming a powerful tool for solving various problems in Bioinformatics.

However, there are many challenges to overcome:

- Statistical significance of predictions
  - Biologists are interested in knowing what are the chances of getting a similar score just by chance alone.

- Computationally scalable learning and classification algorithms
  - For certain problems, there are a lot of training data.

- Effective multi-class extensions
  - Fold-recognition is a 1000+-way classification problem.
Thanks

- Fold Prediction/ Remote Homology Recognition
  - *Profile Based Direct Kernels for Remote Homology Detection and Fold Prediction* by Huzefa Rangwala & George Karypis (BIOINFORMATICS to appear)

- Secondary Structure Prediction
  - YASSPP:
    - Server: [http://yasspp.cs.umn.edu](http://yasspp.cs.umn.edu)

- Group Website:
  - [http://www.cs.umn.edu/~karypis](http://www.cs.umn.edu/~karypis)

- Email:
  - karypis@cs.umn.edu