A Fast, Large-Scale Learning Method for Protein Sequence Classification

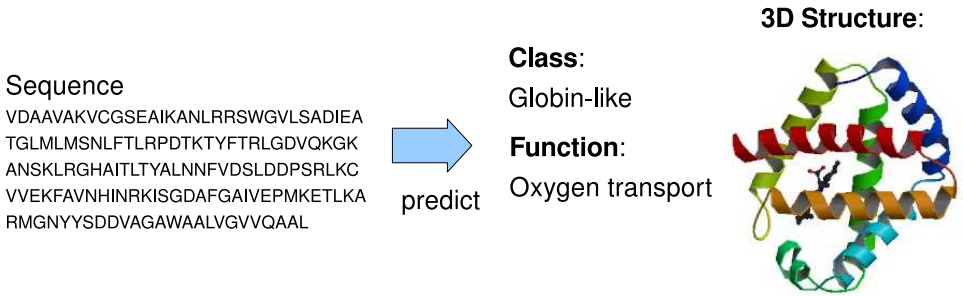
Pavel Kuksa, Pai-Hsi Huang, Vladimir Pavlovic

Department of Computer Science
Rutgers University

BIOKDD 2008

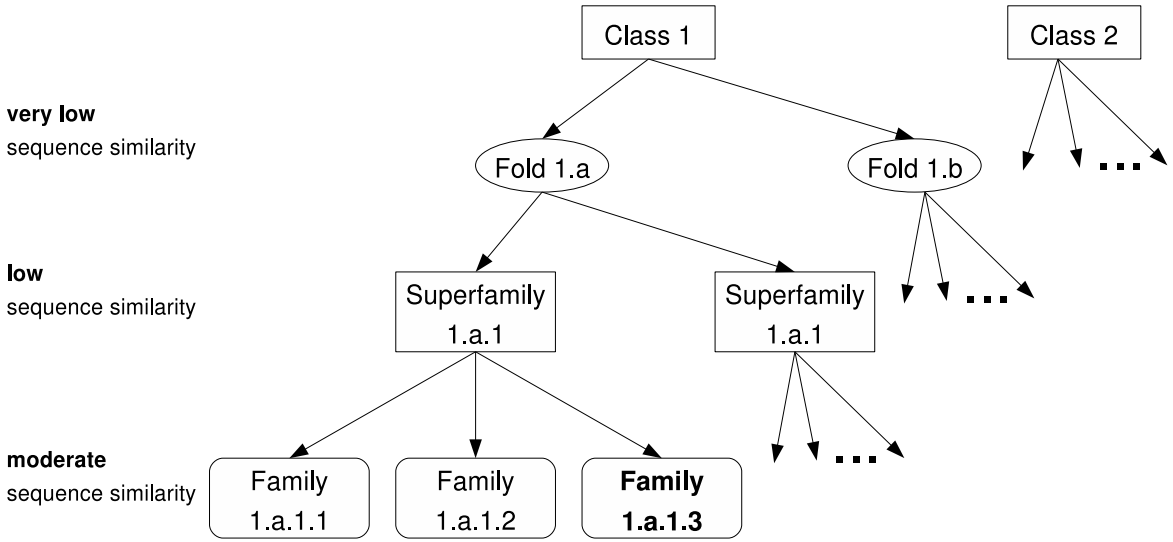
Classification of protein sequences

- Goal: sequence modeling and classification in the *remote* sequence similarity setting:
  - remote homology prediction: from a primary sequence, predict if belongs to superfamily $\Rightarrow$ need to model/recognize remote (across families) relationships
  - fold prediction: predict fold from primary sequence $\Rightarrow$ need to model/recognize remote (across superfamilies/families) relationships
  - focus on remote homology
- Challenges
  - Highly dissimilar sequences
  - limited labeled examples
  - $\Rightarrow$ hard computational and modeling problem



Remote homology prediction

- Proteins: *linear* strings of amino acids ($|\Sigma| = 20$)
- Structural Classification of Proteins [LCAH$^+$00] (SCOP)



- Remote homologs: same superfamily, different families
- New subclass discovery nature: recognition of previously *unseen* families
- Goal: accurately predict if unknown sequence belongs to a *superfamily* of interest

Classification of Sequences using Kernel Methods

- Infer class labels via similarity measures (kernels) $K(x, y)$:
  - measures similarity between two objects (e.g. bio-sequences)
  - has a corresponding vector (dot-product) feature (kernel) space $\phi(x)$
  - the problem is mapped from the original space (may be arbitrary, non-vector) to *vector feature* space
  - use learning algorithm (eg. SVM) to build a kernel-based classifier

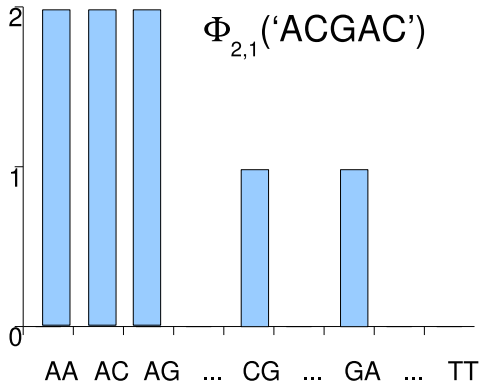
$$K(X, Y) = \Phi(X)^T \Phi(Y)$$

- kernel-based methods some of the best results in sequence analysis

State-of-the art methods for protein classification

- Spectrum-like string kernels [LEWN02]
  - measure similarity based on common contiguous fixed-length substrings (k-mers)
  - mismatch/spectrum kernels

$$\Phi(X) = \left(\sum_{\alpha \in X, |\alpha| = k} I_m(\alpha, \gamma) \right)_{\gamma \in \Sigma^k}$$



- mismatch kernel:
  - best performing string kernel for bio-sequences
  - has strong $O(k^{m+1}|\Sigma|^m)$ dependency on the alphabet size and number of mismatches
- Profile kernel: probabilistic string representation
  - each character in a string is replaced with a probability distribution over alphabet characters
  - measure similarity based on common (in *probabilistic* sense) k-mers

Summary: state-of-the-art methods for remote homology

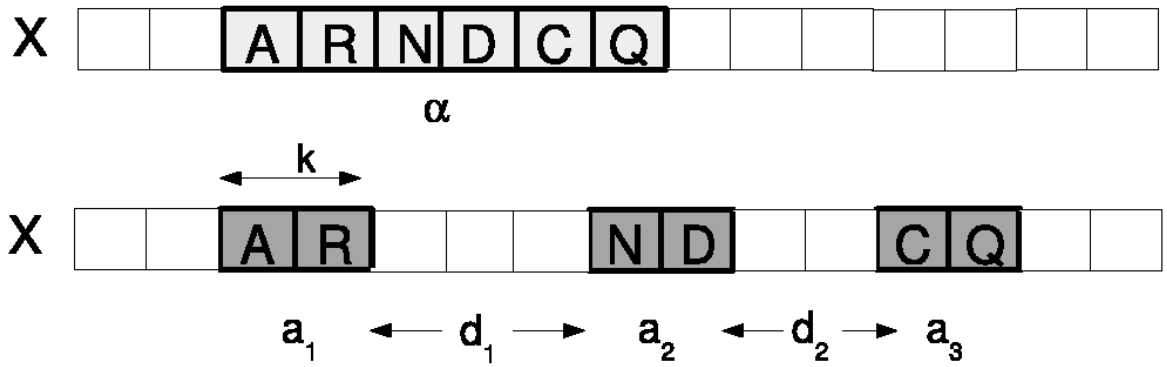
- Good performance is achieved at high computational cost
- Spatial configuration of features in the sequence is not retained
- Performance of the best available methods is still too low for reliable sequence annotation

Our new sequence classification method

- Inspired by a need to account for complex processes of sequence transformations (e.g. evolution) without explicitly modeling the process
- Has linear $O(cn)$ in the sequence length (n) time complexity and small alphabet-independent constants
- Based on a class of efficient string-based kernels (sparse spatial sample kernels, SSSK)
- Provides significant improvements over existing state-of-the-art methods in both performance and running times

Spatial Sample Kernels

$$K(X, Y|k, t, d) = \sum_{\substack{(a_1,d_1,a_2,\ldots,d_{t-1},a_t),\\ a_i \in \Sigma^k, 0 \leq d_i \leq d-1}} C(a_1, d_1, \ldots, d_{t-1}, a_t|X) C(a_1, d_1, \ldots, d_{t-1}, a_t|Y)$$



Contiguous k-mer feature α of a traditional spectrum/mismatch kernel (top) contrasted with the spatial sample features (bottom).

- Notation:
  - $t = 2$: double-(k,d)
  - $t = 3$: triple-(k,d)

Features Induced by Kernels

S = HKYNQLIM

spectrum-5
HKYNQ
KYNQL
YNQLI
NQLIM

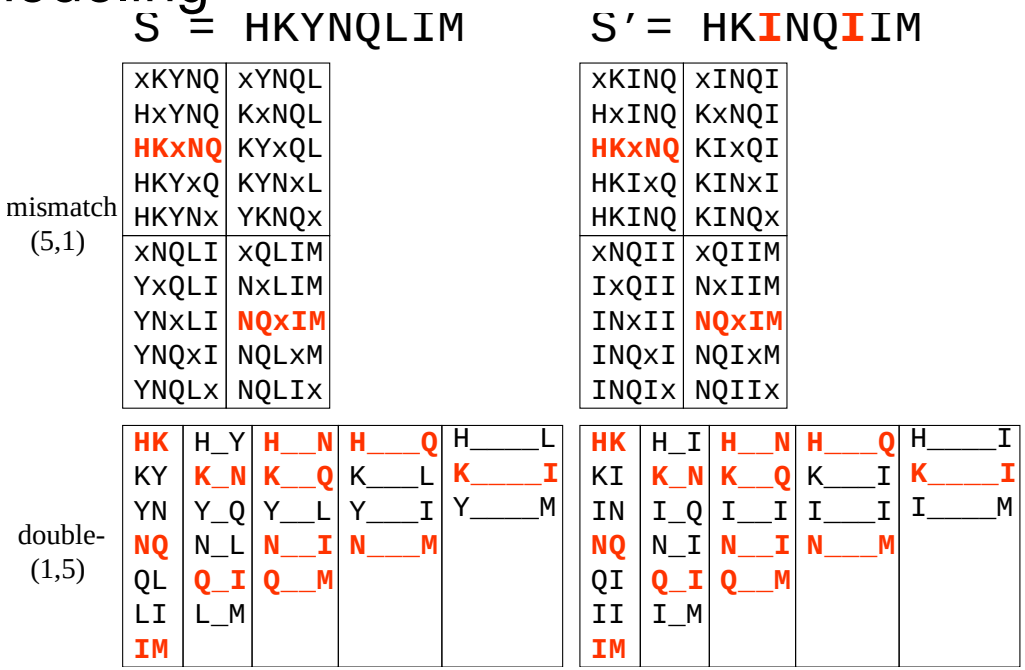
mismatch(5,1)	
xKYNQ	xYNQL
HxYNQ	KxNQL
HKxNQ	KYxQL
HKYxQ	KYNxL
HKYNx	YKNQx
xNQLI	xQLIM
YxQLI	NxLIM
YNxLI	NQxIM
YNQxI	NQLxM
YNQLx	NQLIx

double-(1,5)				
HK	H_Y	H__N	H___Q	H____L
KY	K_N	K__Q	K___L	K____I
YN	Y_Q	Y__L	Y___I	Y____M
NQ	N_L	N__I	N___M	
QL	Q_I	Q__M		
LI	L_M			
IM				

Comparison of features extracted by the spectrum-like and spatial kernels. In the mismatch representation, each feature basis with 'X' corresponds to $|\Sigma|$ features.

- spatial kernels: low-dimensional, sparse, few features
- mismatch kernels: high-dimensional, very many features

Sequence Modeling



Differences in handling substitutions by spectrum-like and spatial kernels

- Mismatch: very few features retained
- Spatial kernels: still significant overlap in sequence features
- For the spectrum-like kernel, the number of allowed mismatches needs to be increased (at a high computational cost)

Semi-Supervised Spatial Sample Kernel

- Semi-Supervised Learning
  - Small amount of training labeled sequences under supervised learning $\Rightarrow$ low performance
  - Can improve by enlarging training set with unlabeled sequences that are similar (homologous) to the labeled sequences
- Semi-Supervised Spatial Sample Kernel
  - Define a kernel over **string sets** $N(X)$ (instead of strings X)

$$K(X, Y) = \sum_{x \in N(X)} \sum_{y \in N(Y)} K(x, y)$$

  - String set $N(X)$ is formed by potential homologous (similar in the sense of some measure) sequences [WLI+05]

Computing Spatial Sample Kernels

- Can be efficiently computed using Sorting and Counting

Input: set of strings $S = \{s_1, \ldots, s_N\}$, parameters k, t, d

1: $\mathbf{K} = \mathbf{0}$
2: **for all** $(d_1, \ldots, d_{t-1}) \in \{1, \ldots, d\}^{t-1}$ **do**
3: Build sets $L_{s_i}, i = 1, \ldots, N$ of SSS features for each string in the sequence set
4: Construct complete set of features in the sequence set $L = \bigcup_{i=1}^{N} L_{s_i}$ with each feature containing the index of the sequence it comes from
5: Obtain a permutation π_L that lexicographically orders L using t rounds of counting sort
6: Obtain counts $\mathbf{c}(f) = \{c(f, s_i)\}_{i=1,\ldots,N}$ for each distinct feature f in L_{π_L} in a single pass over the list and update kernel $\mathbf{K} = \mathbf{K} + \mathbf{c}(f)\mathbf{c}(f)^T$
7: **end for**

Sparse Spatial Sample Methods: Summary

- efficiently model complex sequence transformations (multiple mutations, insertions, deletions, etc), variable length pattern
- explicitly model spatial configuration of features in the sequence
- captures sequence content at multiple scales
- fast $O(cn)$ evaluation, alphabet-free matching (alphabet size independent)

Experiments

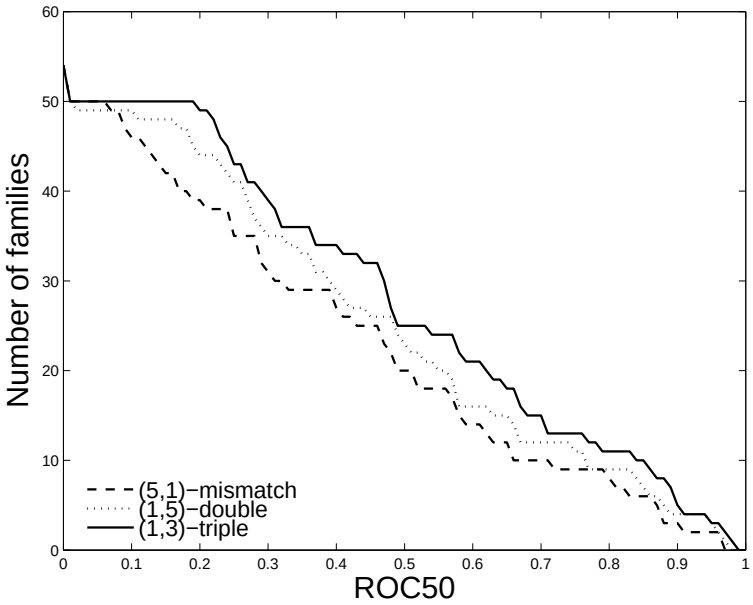
- Remote homology prediction
  - fully-supervised setting
  - semi-supervised setting
    - large-scale
- Fold prediction
- Motif discovery

Remote homology prediction

- SCOP [LCAH$^+$00] 1.59 dataset
  - used as benchmark in many studies [WLI$^+$05]
  - $7,329$ sequences (2862 labeled, 4467 unlabeled)
  - 54 *binary* classification problems simulating a remote homology detection problem:
    - test whether the method can detect a new (unseen) family from a given superfamily
  - Fully-supervised (2862 seq)
  - Semi-supervised setting
    - SCOP (4467 seq)
    - Large-scale semi-supervised (PDB, Swiss-Prot, NR)
    - Construct neighborhood $N(X)$ for each sequence X (train + test) $N(X) = \{X' : eValue(X, X') \leq 0.05\}$ using 2 PSI-BLAST iterations or BLAST (single pass)

Comparison on SCOP dataset

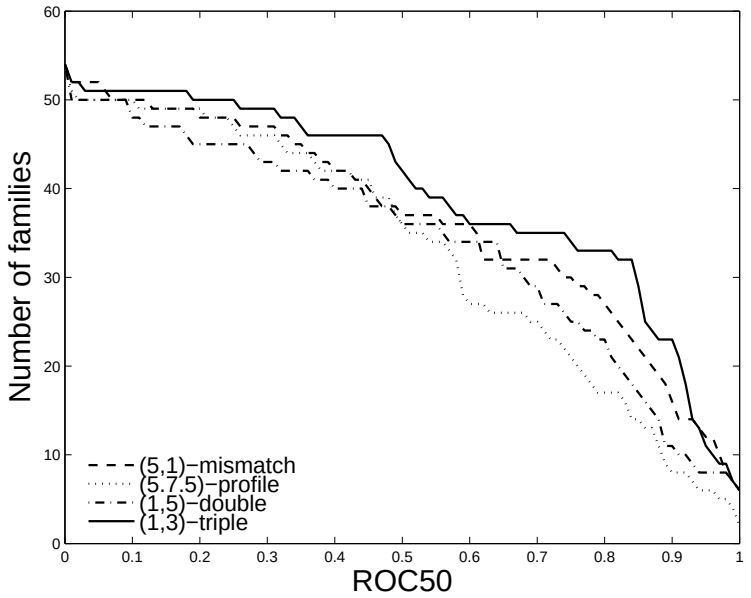
- *Supervised* setting (labeled only, 2862 total)



Method	ROC	ROC50	# dim.	Time (s)
(5, 1)-mismatch	0.8749	0.4167	3200000	938
SVM-pairwise	0.8930	0.4340	-	-
gapped(7,3)[LK04]	0.8540	0.3953	8000	297
(1,5) double	0.8901	0.4629	2000	54
(1,3) triple	**0.9148**	**0.5118**	72000	112

- increase ROC50 by 20%
- increase speed 10-20 fold

- *Semi-Supervised* (labeled+unlabeled, 7329 total)



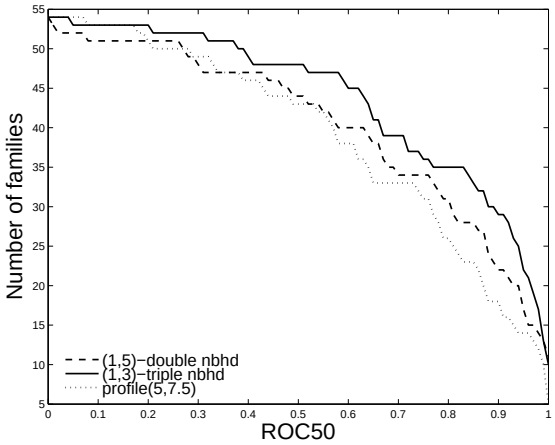
Method	ROC	ROC50
(5, 1)-mismatch	0.9093	0.6745
(5,7.5)-profile	0.9190	0.6069
(1,5)-double	0.9282	0.6383
(1,3)-triple	**0.9382**	**0.7262**

- increase ROC50 by 10-20%
- increase ROC50 by 40% compared to fully-supervised

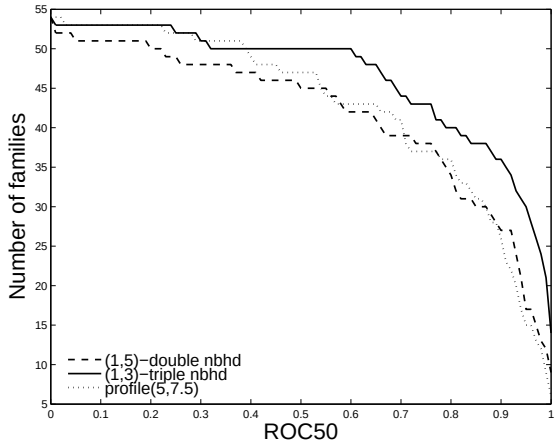
Large-Scale Protein Remote Homology Detection

Dataset	# Seq	# Neighbors (mean/median/max)
PDB	17,232	16/5/334
Swiss-Prot	101,602	56/28.5/385
NR	534,936	114/86/490

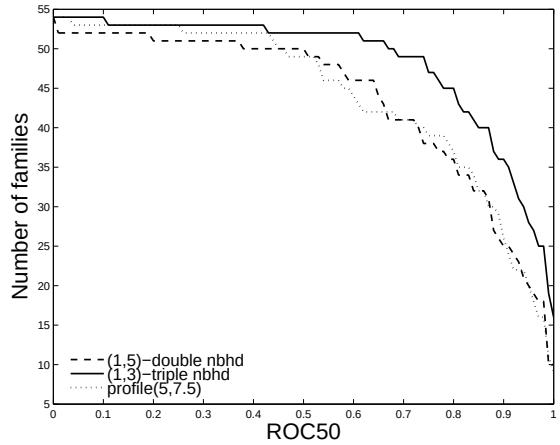
Comparison on SCOP under large-scale semi-supervised setting



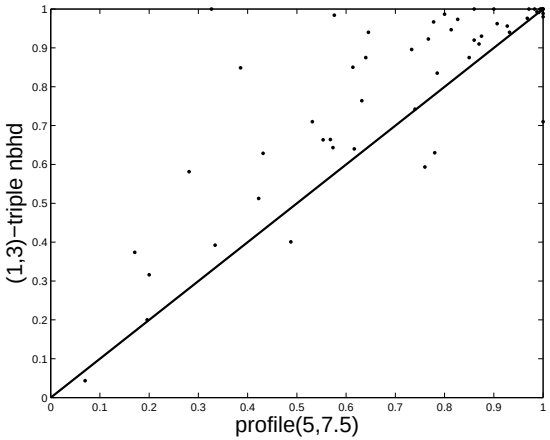
(a) PDB



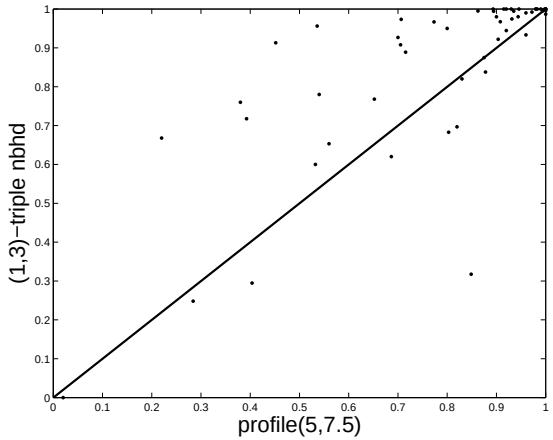
(b) Swiss-Prot



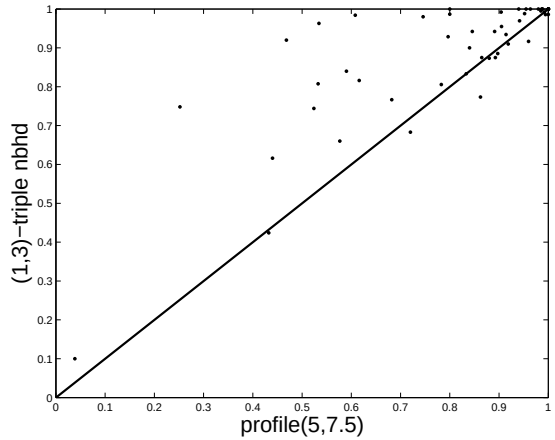
(c) NR



(d) PDB



(e) Swiss-Prot



(f) NR

Comparison on SCOP under large-scale semi-supervised setting

- ROC50 Scores

method	PDB	Swiss-Prot	NR
profile	72.05	79.14	81.51
double	74.66	77.01	80.76
triple	**82.40**	**86.05**	**89.44**

- significantly better accuracy compared to state-of-the-art
- increase ROC50 by 10-12%

Non-iterative neighborhood construction

- PSI-BLAST (iterative) procedure requires construction of profiles $\Rightarrow$ computationally demanding.
- BLAST (non-iterative, only one pass through unlabeled dataset) runs a lot faster

	PSI-BLAST			BLAST		
Data set	ROC	ROC50	# neighbors	ROC	ROC50	#neighbors
PDB	96.91	82.40	16/51/334	95.57	75.35	6/3/95
Swiss-Prot	97.32	86.05	56/28.5/385	96.40	81.44	16/9/177
NR	98.61	89.44	114/86/490	97.87	86.47	40/23/232

- Similar performance with BLAST (faster)
- 3-fold reduction in the number of sequences used for neighborhood construction $\Rightarrow$ faster training/testing

Preliminary results for fold prediction (Ding and Dubchak benchmark dataset [DD01])

- *multi-class* fold detection problem
- used as benchmark in many studies
- contains sequences from 27 folds divided into two *independent* sets (27**-way classification**)
  - training and test sequences share less than 35% sequence identities
  - within training set, no sequences share more than 40% sequence identities

Preliminary results for fold prediction (Ding and Dubchak)

Method	Balanced Error	Top 5 Balanced Error	Recall	Top 5 Recall	F1	Top5 F1
Supervised						
SVM(D&D)†	56.5	-	-	-	-	-
Mismatch(5,1)	53.22	28.86	46.78	71.14	61.68	81.45
Double(1,5)	46.19	23.92	53.81	76.18	57.57	77.97
Triple (1,3)	**44.99**	**21.09**	**55.01**	**78.91**	**65.33**	**83.74**
Semi-supervised (Non-redundant data set)						
Profile(5,7.5)	32.17	16.73	67.83	83.27	77.16	88.71
Double(1,5)	24.74	**11.6**	75.26	**88.4**	75.63	87.62
Triple(1,3)	**22.38**	11.79	**77.62**	88.21	**80.69**	**89.8**
Profile NR(Perceptron)‡	26.5	-	-	-	-	-

All measures are presented as percentages.
†: quoted from [DD01]; ‡: quoted from [MIW+07]

- decrease error rate by 15% compared to state-of-the-art

Complexity and Running time analysis

- efficient $O(cn)$
- alphabet size independent
- order-of-magnitude faster than existing algorithms

Method	Time complexity	Running time (s)	
Semi-supervised setting		Sup.	Semi-sup.
Triple kernel	$O(d^2HnN + d^2\|\Sigma\|^3N^2)$	112	327
Double kernel	$O(dHnN + d\|\Sigma\|^2N^2)$	**54**	**67**
Mismatch	$O(k^{m+1}\|\Sigma\|^mHnN + \|\Sigma^k\|N^2)$	948	-
Profile kernel	$O(kM_\sigma nN + \|\Sigma\|^kN^2)$	-	10 hours†

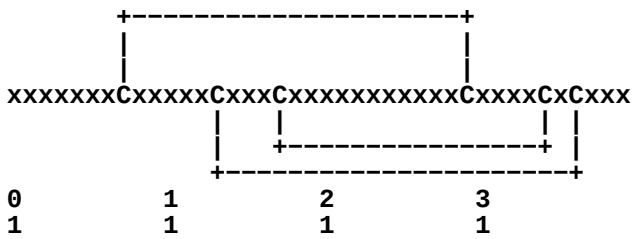
† the running time is quoted from [KIW+05]

Notations: N-number of sequences, n-sequence length, H-sequence neighborhood size, $|\Sigma|$ - alphabet size k, m - mismatch kernel parameters ($k = 5, 6$ and $m = 1, 2$ in most cases) M_σ - profile neighborhood size, $M_\sigma \leq |\Sigma^k|$

# seq.	10	30	50	100
time ratio	81	93	95	115

Discovering discriminative sequence motifs with spatial information

- Previous biological studies (*e.g.* [KFG02]) suggest that distance information between conserved key positions could play critical roles in structural/functional classification
- E.g. *Scorpion toxin-like* superfamily: several families characterized by disulphide bridges formed by conserved 'C' residues with conserved relative positions:



- Can we discover similar patterns?

Discovering short sequence motifs with spatial info (2)

- Held-out family: *short-chain scorpion toxin*.
  - 16 positive labeled training sequences (16 SVs)
  - 1067 negative labeled training sequences (88 SVs)
  - Feature 'C___C' has highest weight (consistent with established biological knowledge)
- Corresponding feature 'CC' in the gapped kernel representation ranked 38/400 as 39 out of 43 negative SVs have it
- Integrating spatial information out might lead to sub-optimal performance:
  - ROC50 for double(1,5) kernel: 76.61
  - ROC50 for gapped(2,4) kernel: 28.35

Key Contributions

- a fast, semi-supervised method for sequence classification based on a class of efficient string-based kernels (SSSK)
- significantly improved state-of-the-art classification accuracy in both fully-supervised and semi-supervised settings
- low computational complexity, order-of-magnitude running time improvements over existing algorithms
- can work with very large datasets
- All presented methods can be applied to a wide range of applications in biological and other domains

Future work

- still room for improvement; notice some hard to classify superfamilies
- joint training / feature sharing framework to recover (or exploit) tree hierarchy in sequences
- multi-class problems
- general sequence classification: text documents, images, word utterance, . . ., etc.

References

Chris H.Q. Ding and Inna Dubchak.

Multi-class protein fold recognition using support vector machines an d neural networks .
Bioinformatics, 17(4):349–358, 2001.

Alexander E. Kister, Alexei V. Finkelstein, and Israel M. Gelfand.

Common features in structures and sequences of sandwich-like proteins.
PNAS, 99(22):14137–14141, 2002.

Rui Kuang, Eugene Ie, Ke Wang, Kai Wang, Mahira Siddiqi, Yoav Freund, and Christina Leslie.

Profile-based string kernels for remote homology detection and motif extraction.
J Bioinform Comput Biol, 3(3):527–550, June 2005.

L. Lo Conte, B. Ailey, T.J. Hubbard, S.E. Brenner, A.G. Murzin, and C. Chothia.

SCOP: a structural classification of proteins database.
Nucleic Acids Res., 28:257–259, 2000.

Christina S. Leslie, Eleazar Eskin, Jason Weston, and William Stafford Noble.

Mismatch string kernels for svm protein classification.
In *NIPS*, pages 1417–1424, 2002.

Christina Leslie and Rui Kuang.

Fast string kernels using inexact matching for protein sequences.
J. Mach. Learn. Res., 5:1435–1455, 2004.

Iain Melvin, Eugene Ie, Jason Weston, William Stafford Noble, and Christina Leslie.

Multi-class protein classification using adaptive codes.
J. Mach. Learn. Res., 8:1557–1581, 2007.

Jason Weston, Christina Leslie, Eugene Ie, Dengyong Zhou, Andre Elisseeff, and William Stafford Noble.