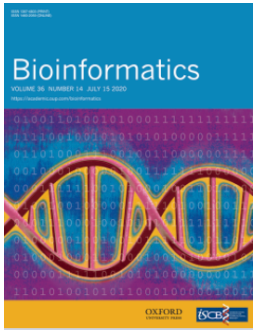
Bioinformatics

ISCB INTERNATIONAL SOCIETY FOR COMPUTATIONAL BIOLOGY

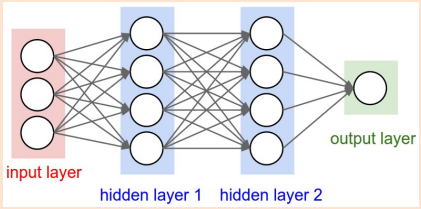
FastSK: Fast Sequence Analysis with Gapped String Kernels



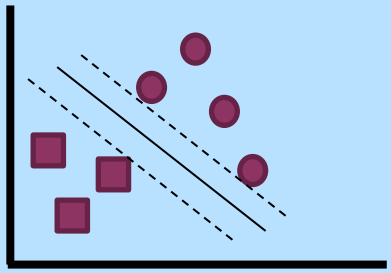
Dr. Yanjun Qi
University of Virginia
https://github.com/QData/FastSK

ECCB Parallel Proceeding Track #03
Monday, Sep 7, 2020 10:45 AM

State-of-the-art String Classification Models

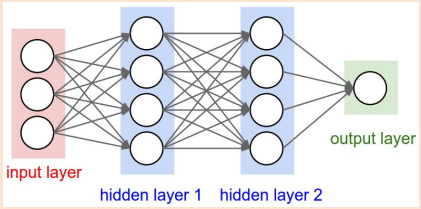


Deep Neural Network
(DNN)

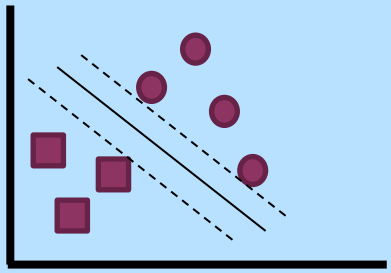


Support Vector Machine
(SVM)

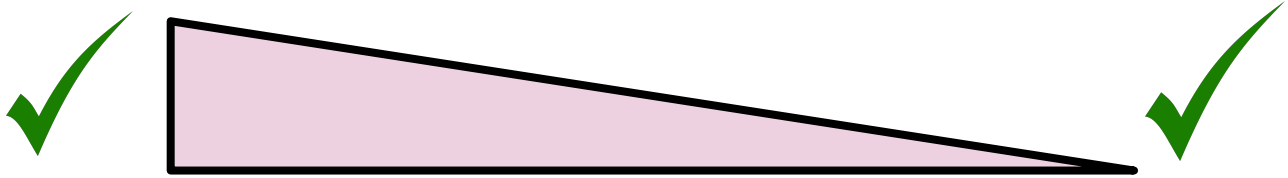
State-of-the-art String Classification Models



Deep Neural Network (DNN)



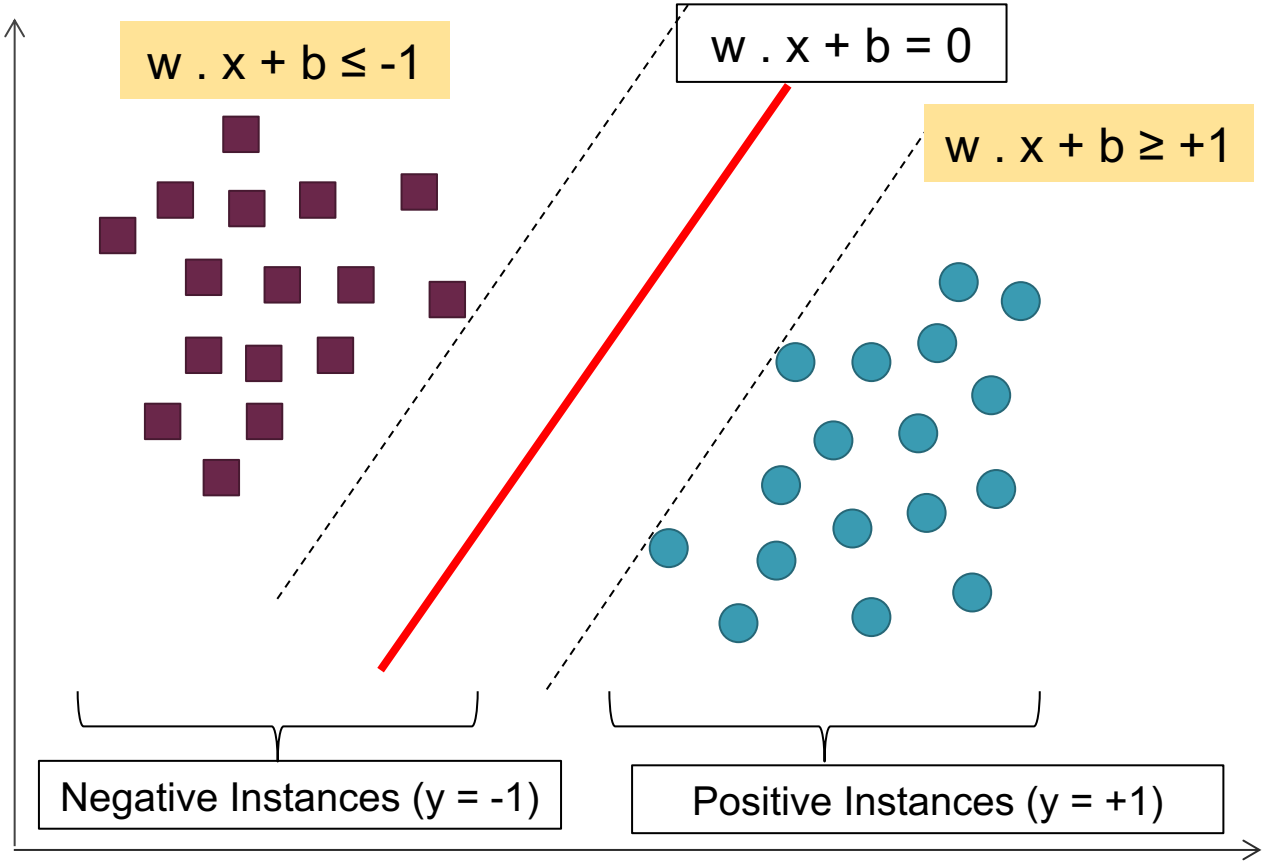
Support Vector Machine (SVM)



Number of Training Samples

Support Vector Machine (SVM)

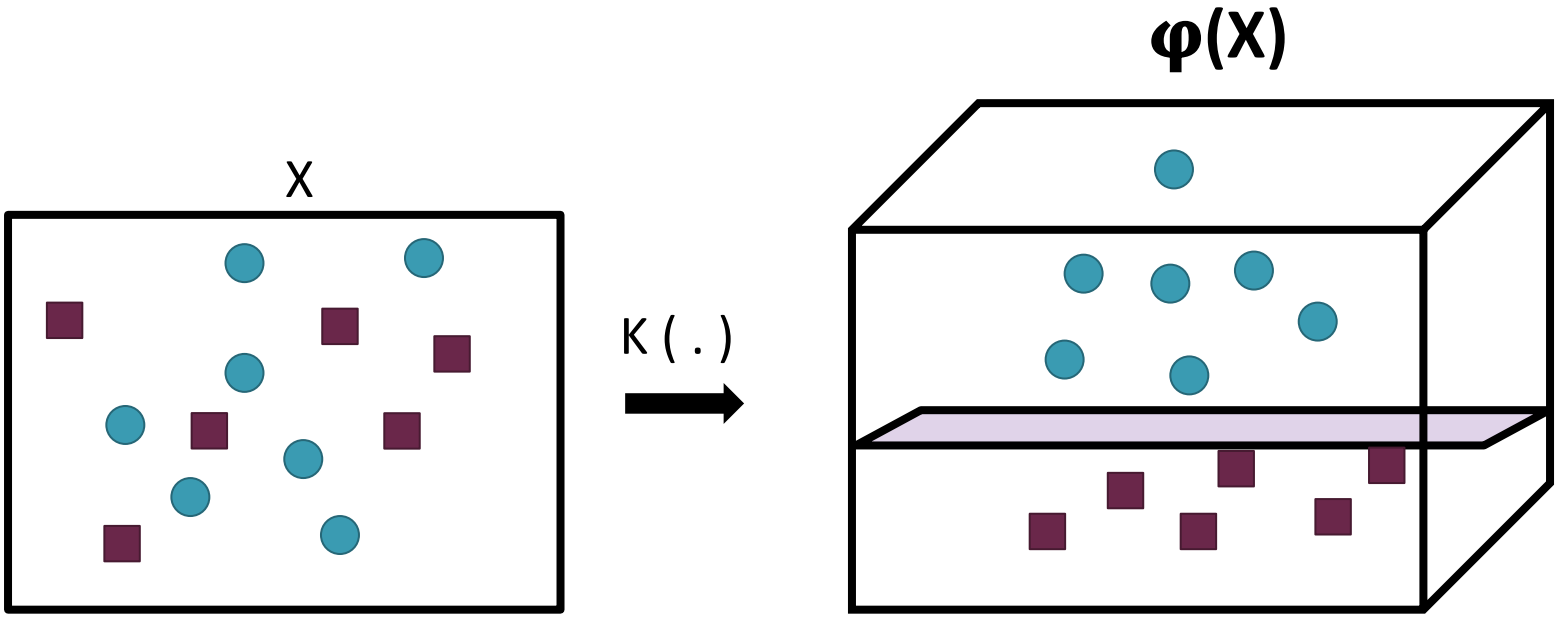
$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$$



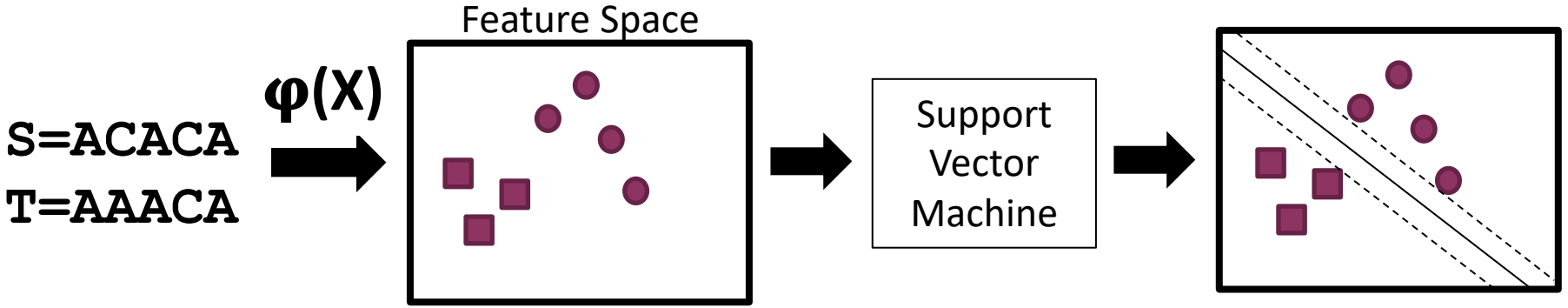
Kernel

X

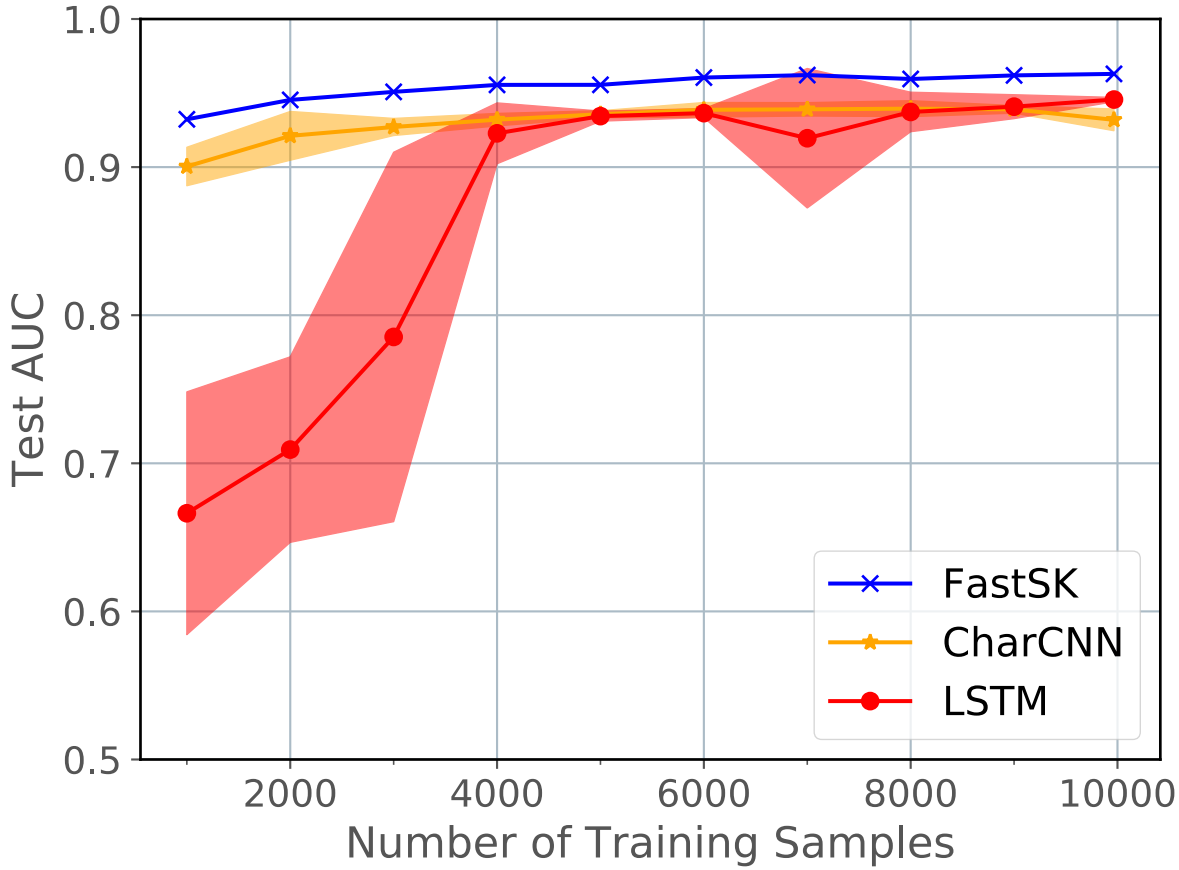
$$f(x) = w \cdot \Phi(x) + b.$$



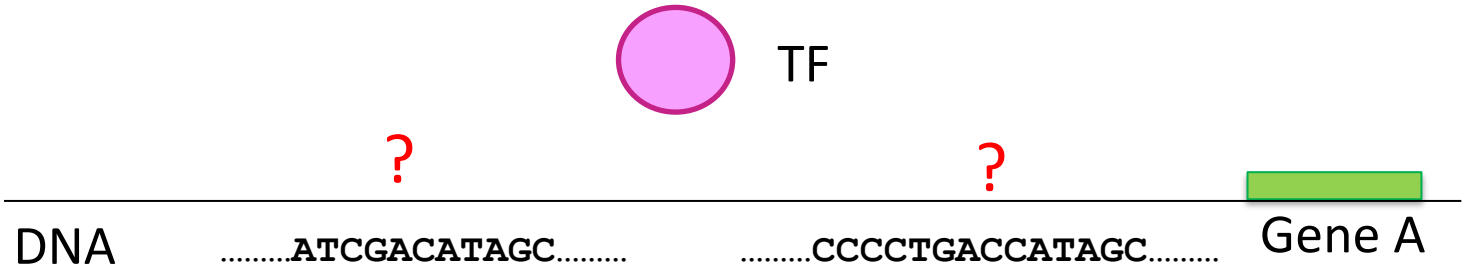
String Kernel + SVM Framework



String Kernel-SVM Predicts Well with Low Variance



[Application I]: Does TF bind to this DNA sequence?



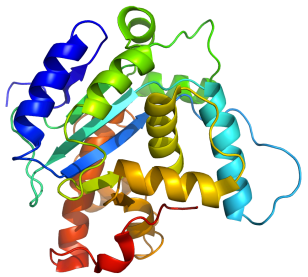
ATCGAATCCG ✓
CGCTGAATCG ✗
ATCGCTATCG ✓
ATCCCGCTCG ✗

$\sum=4$

Minimum Number of Training Samples

1000

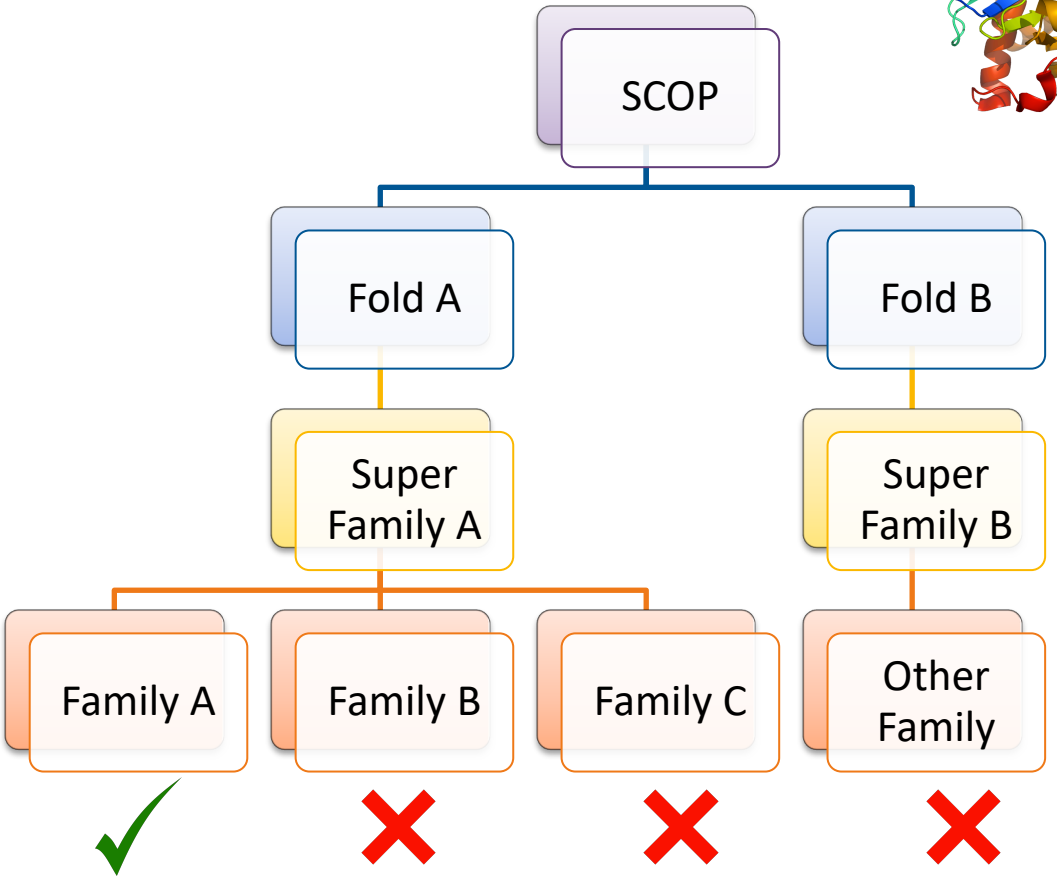
[Application III]: What family does a protein sequence belong to?



VQGGHACCAKKQQQ

∑=20

Minimum Number of Training Samples
500



[Application III]: Language Classification:

X

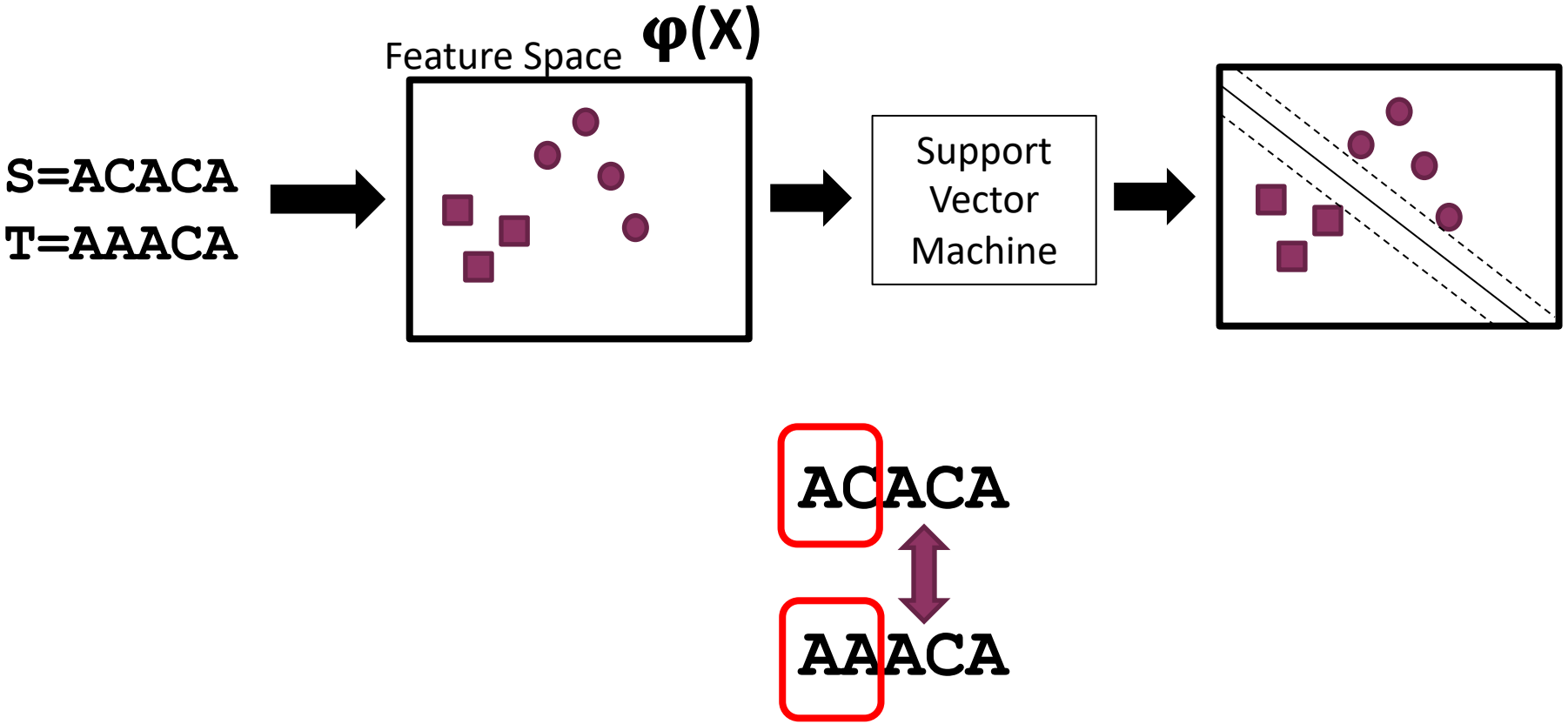
This Food is not good.

English

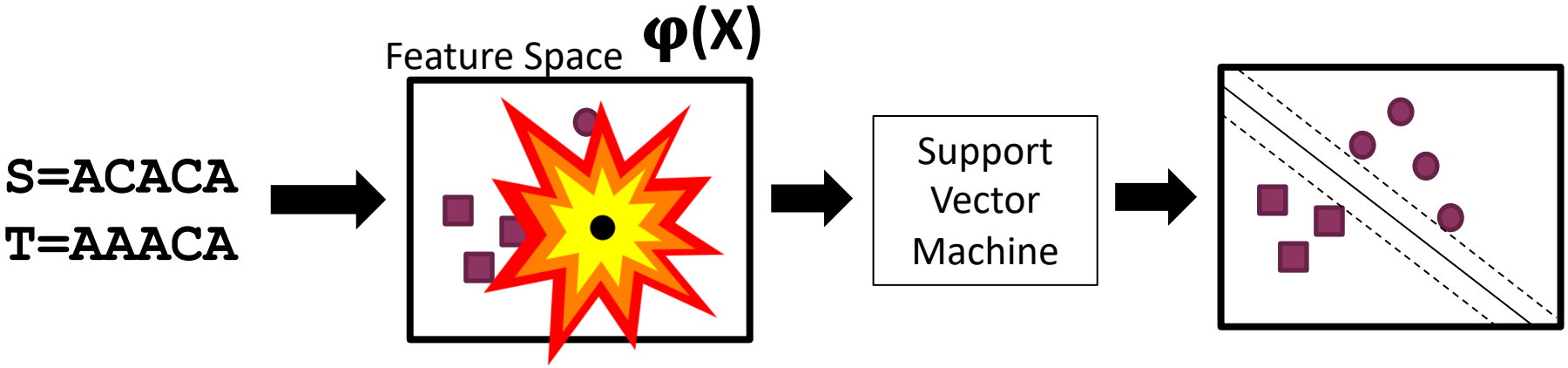
بالمناسبة ، اسمي هيوشي ايجيما .	CAI
بالمناسبة ، إسمي هيروشي ايجيما .	BEI
على فكرة ، اسمي هيروشي ايجيما .	DOH
بالمناسبة ، سميتي هيروشي إيجيما .	RAB
على فكرة ، اسمي هيروشي ايجيما .	TUN
بالمناسبة ، اسمي هيروش إيجيما .	MSA

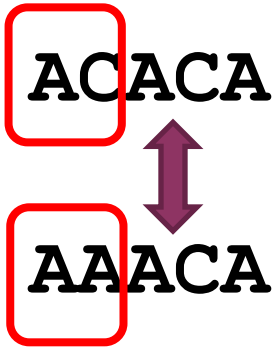
Can we identify a writer's region using just text info?

String Kernels

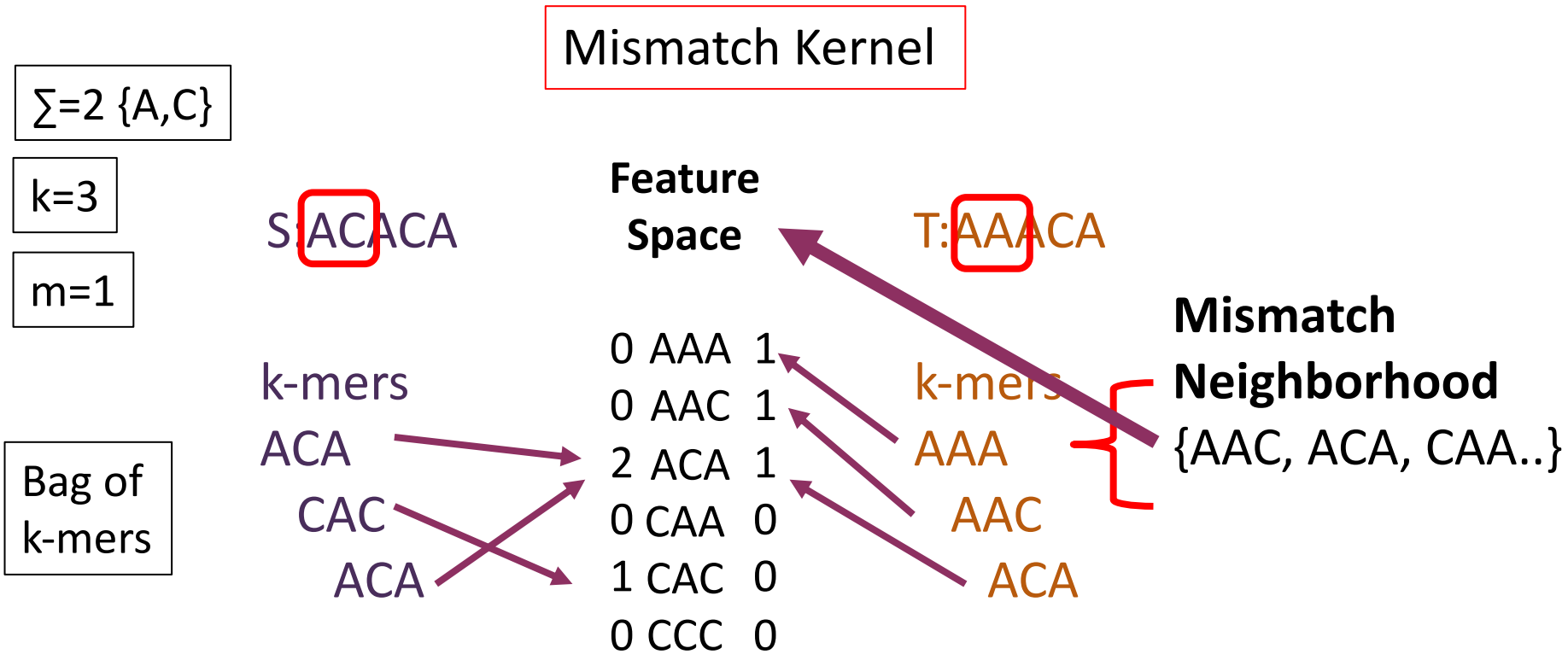


Challenge : SK-SVM methods are slow





String Kernel

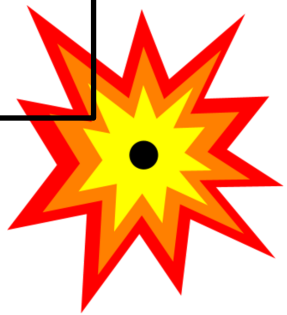


Computational Challenge

∑=20

k=10

Feature Space $\sum^k$



$(20)^{10} \sim= 10^{13}$

Sparse!

AAA	1
AAC	1
ACA	0
CAA	0
CAC	0
CCC	0

Related Work: Gapped k-mer Kernel (gkm-SVM)

∑=2 {A,C}

g=3

k=2

S:ACACA

g-mers	k-mers
ACA	{AC,CA,…}
CAC	
ACA	

T:AAACA

g-mers	k-mers
AAA	{AA,AA,…}
AAC	
ACA	

Gaps=g-k=1

Feature Space $g(|S| + |T|) \ll \sum^k$

Ghandi M, Lee D, Mohammad-Noori M, Beer MA. “Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features”.

Related Work: Gapped k-mer Kernel (gkm-SVM)

(1) $$K(x, x') = \sum_{\gamma \in \Theta_g} c_x(\gamma) \cdot c_{x'}(\gamma)$$

(2) $$K(x, x') = \sum_{i=0}^{l_1} \sum_{j=0}^{l_2} h_{gk}(g_i^x, g_j^{x'})$$

(3) $$K(x, x') = \sum_{m=0}^{g-k} N_m(x, x') h_m$$

Ghandi M, Lee D, Mohammad-Noori M, Beer MA. “Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features”.

Related Work: Gapped k-mer Kernel

gkm-SVM

∑=2 {A,C}

g=3

k=2

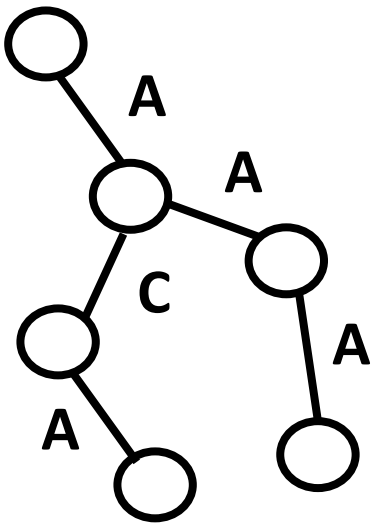
S:ACACA

g-mers
ACA
CAC
ACA

T:AAACA

g-mers
AAA
AAC
ACA

Trie



Mismatch Profile

Coefficient

$$K(S,T) = \sum_{m=0}^{g-k} h_m N_m$$

m=0 $N_m = 2$

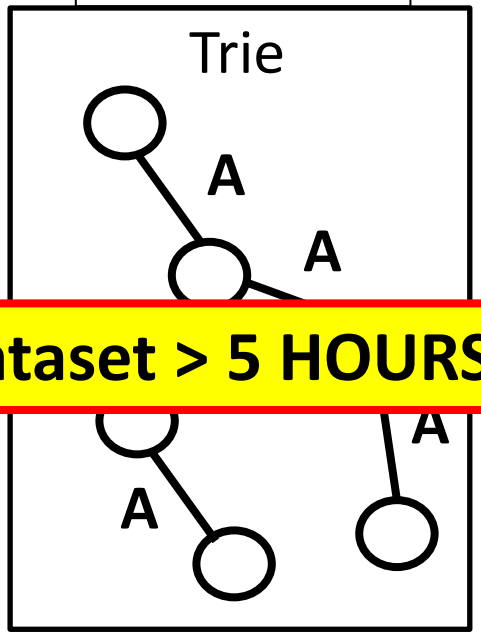
m=1 $N_m = 3$

Gaps=g-k=1

Ghandi M, Lee D, Mohammad-Noori M, Beer MA. “Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features”.

Related Work: Gapped k-mer Kernel

gkm-SVM



∑=2 {A,C}

g=3

Dependence on term $(\sum)^m$

S:ACACA

T:AAACA

Time taken to compute Kernel for Protein dataset > 5 HOURS

ACA

CAC

ACA

Mismatch Profile

Coefficient

AAA

AAC

ACA

$$K(S,T) = \sum_{m=0}^{g-k} h_m N_m$$

m=0 $N_m = 2$

m=1 $N_m = 3$

Gaps=g-k=1

Ghandi M, Lee D, Mohammad-Noori M, Beer MA. “Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features”.

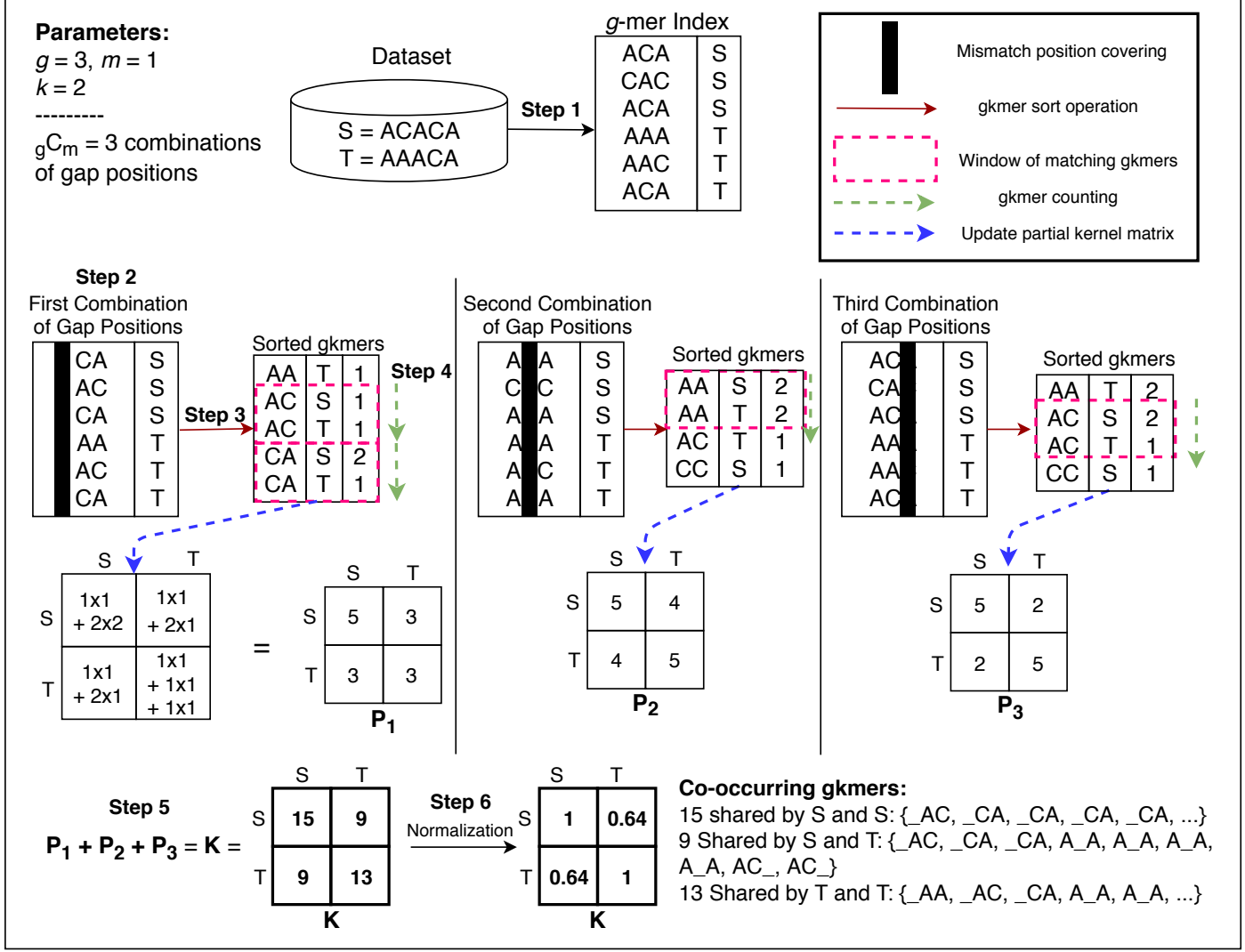
Proposed: Three Key Strategies in FastSK

1. Reduced feature space
2. Direct counting algorithm
3. Approximation algorithm

$$K_{FSK} = \sum_{i=1}^{gCm} P_i$$

$$P_i(x, y) = \sum_{\gamma \in \Theta_i} c_x(\gamma) c_y(\gamma)$$

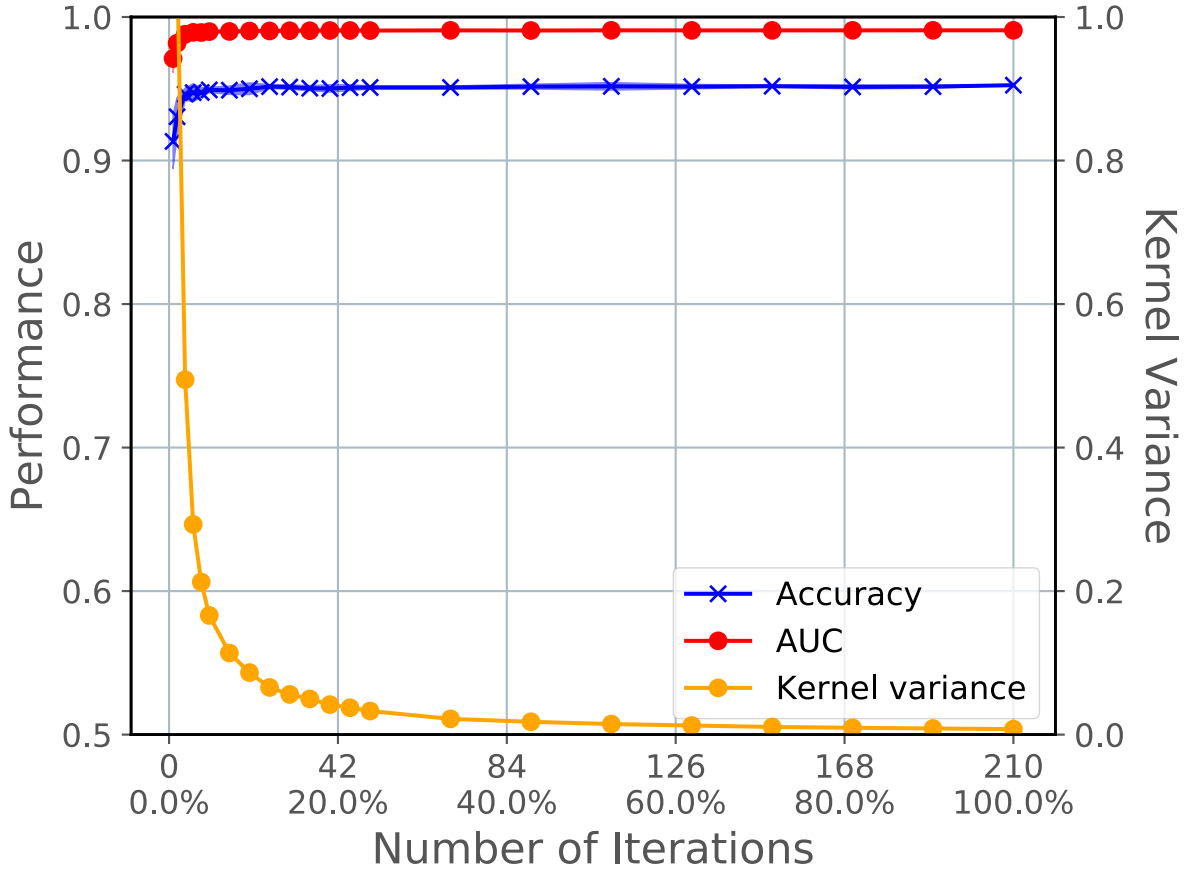
FastSK -Exact



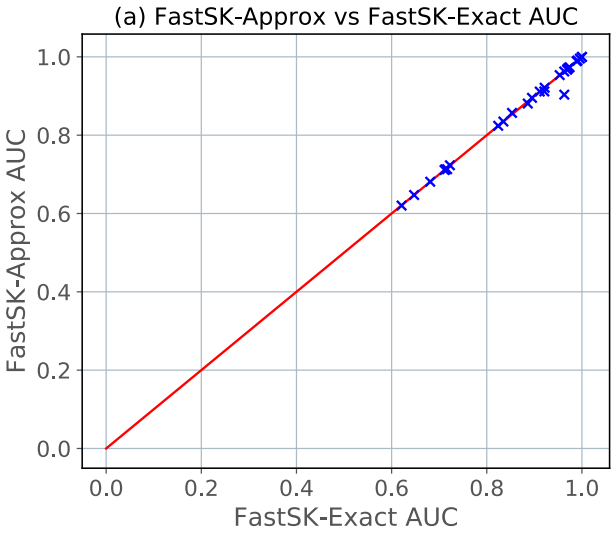
FastSK -Approx

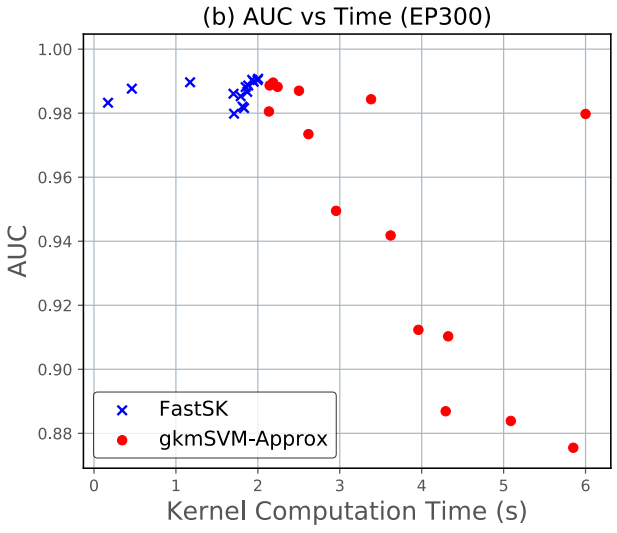
To compute FastSK -Approx, we sample possible mismatch combinations for up to $I_{max} \leq \binom{g}{m}$ iterations. That is, at iteration

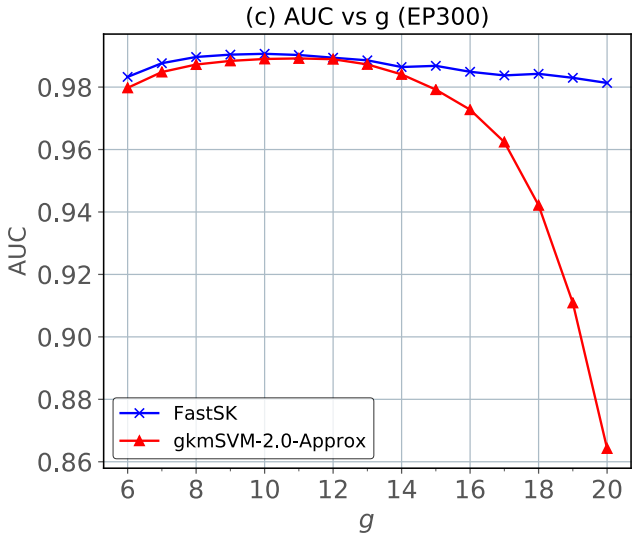
Fig. 2: Only a small proportion of the possible mismatch combinations are needed for strong performance. We use the EP300 TFBS (DNA) dataset with g = 10 and m = 4, hence 210 possible mismatch combinations. On the x-axis, we show the number of mismatch positions used and the percentage of the total possible below each one.



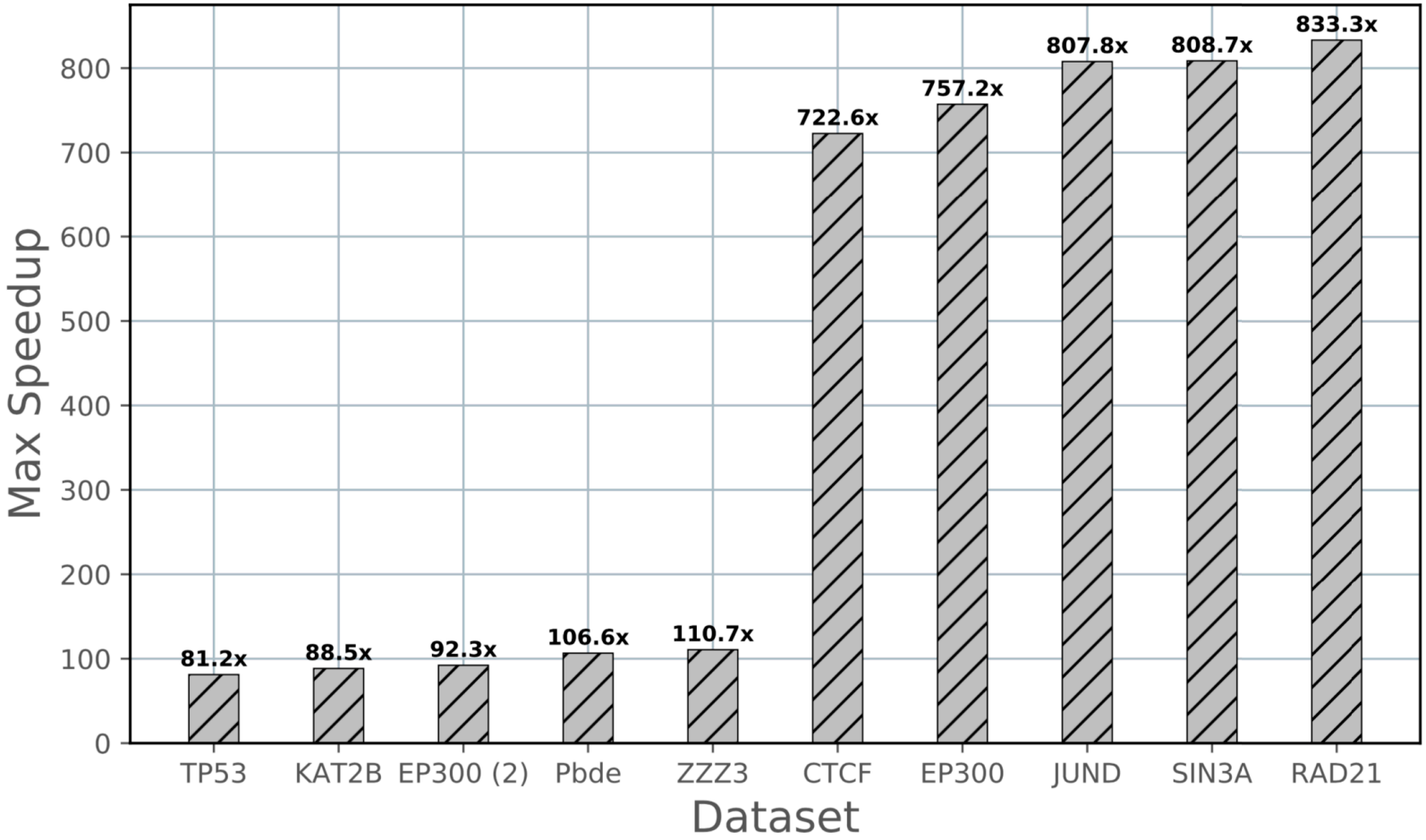
FastSK-Exact vs. FastSK-Approx vs. gkm-svm-Appro



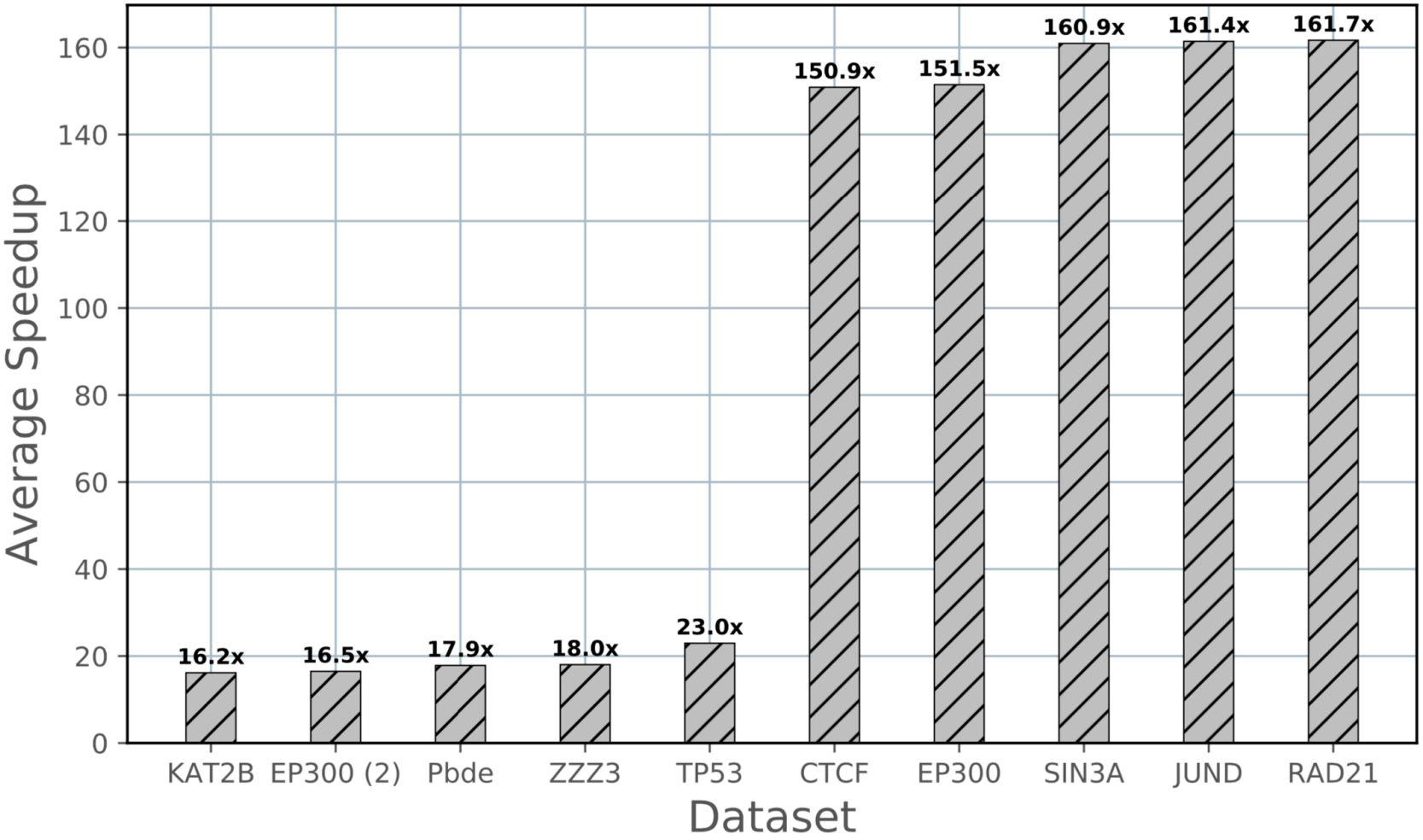




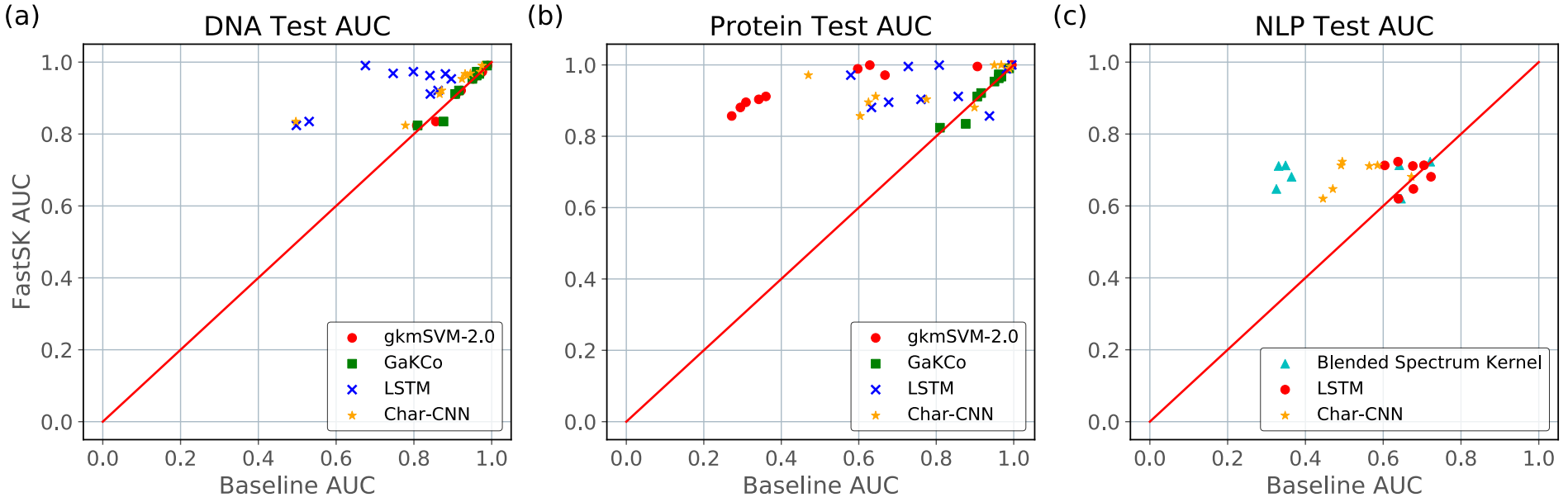
FastSK is faster!



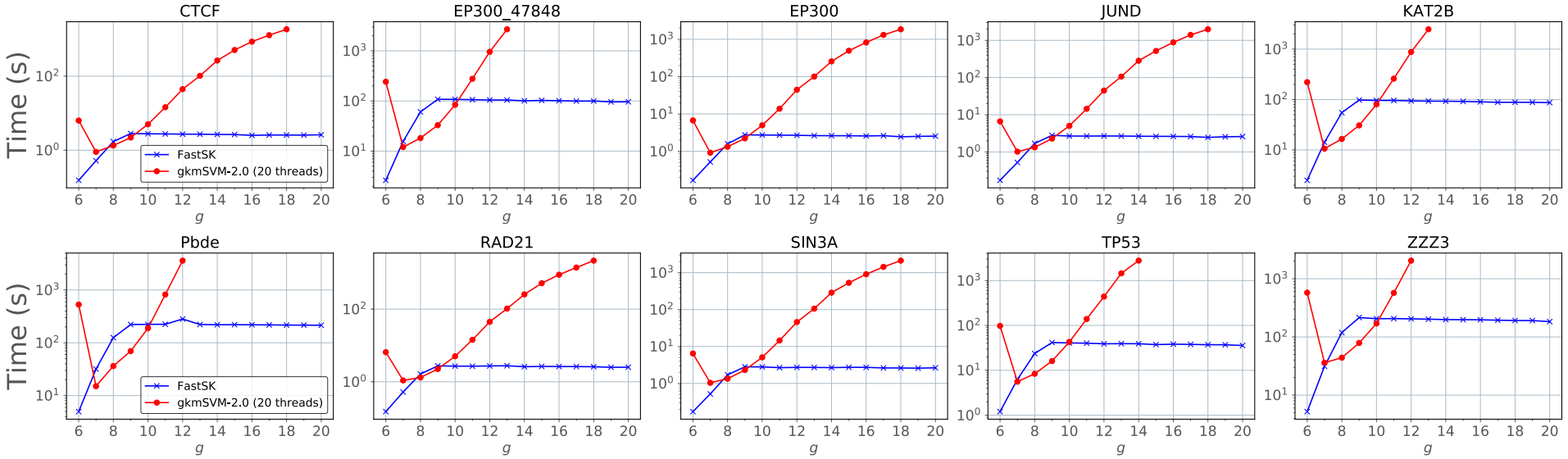
FastSK is faster!



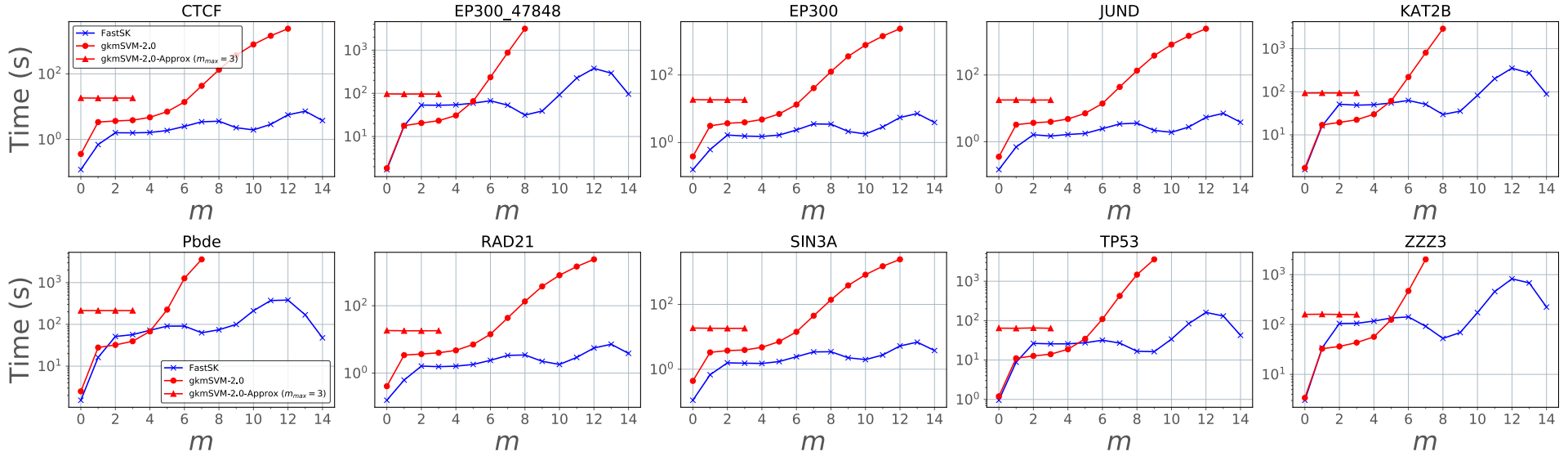
FastSK Predicts Well!



FastSK Scales well with increasing g



FastSK Scales well with increasing m



Summary:

- (A) Strong prediction performance
  (B) Scale with Σ—allow any sequence task
  (C) Scale with g—allow longer features
  (D) Scale with m—allow more mismatch versatility
  (E) Modern software architecture + usable toolbox
- Bonus property: conceptually simple kernel algorithm

- Code and datasets available at:
  https://github.com/QData/FastSK

- bioArXiv:
  https://www.biorxiv.org/content/10.1101/2020.04.21.053975v1

Thank you

FastSK vs. gkm-SVM

$$K_{FSK}(x,y) = \sum_{i=1}^{gCm} \sum_{\gamma \in \Theta_i} c_x(\gamma) c_y(\gamma)$$

$$K_{GSK}(x,y) = \sum_{\gamma \in \Theta_{g,m}} c_x(\gamma) c_y(\gamma)$$

Theorem 1

Theorem 1. *The FastSK kernel function is equivalent to the gkmSVM kernel function [6, 7].*

Proof. Because $h_{gk}(d) = 0$ for $d > m = g - k$, equation 15 is equal to

$$K_{gkm}(x, y) = \sum_{d=0}^{m=g-k} N_d(x, y) h_{gk}(d) = \sum_{d=0}^{m=g-k} \binom{g-d}{k} N_d(x, y) \tag{17}$$

To *directly* count the gkmers, we replace $\binom{g-d}{k} N_d(x, y), d \in 0, 1, 2, ..., m$ with $\binom{g}{m}$, the number of possible mismatch positions. Therefore, at the ith mismatch position, we simply count the gapped k-mers in the set Θ_i. Therefore, we have

$$K_{gkm}(x, y) = \sum_{d=0}^{m=g-k} \binom{g-d}{k} N_d(x, y) = \sum_{i=1}^{gCm} \sum_{\gamma \in \Theta_i} c_x(\gamma) c_y(\gamma) \tag{18}$$

as desired. Our method is both faster and simpler because we cut out the middleman: we directly count the gkmers. It also allows us to create a better approximation algorithm than what is possible using equation 15, because we can sample mismatch positions to estimate the kernel function.

GaKCo-SVM:

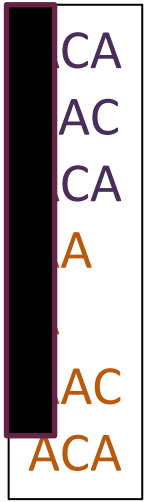
Counting Implementations GaKCo [26] is similar to FastSK in that it uses uses a sort-and-count algorithm. However, it differs from FastSK in that it follows the mismatch statistic formulation from [6]. The result is that GaKCo's time complexity has a $\sum_{d=0}^{m=g-k} \binom{g}{d}$ coefficient. In contrast, we simply have a coefficient of $\binom{g}{m}$.

S:ACACA

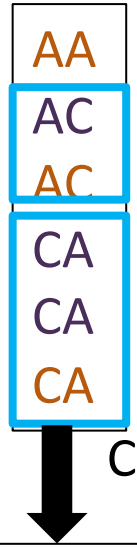
T:AAACA

g=3

m=1

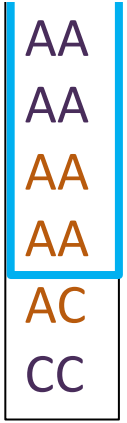


Sort



Count

C_m (S,T)=1+2=3



C_m (S,T)=4

C_m (S,T)=2

Total C_m (S,T)=3+4+2=9

Over Counting = $\binom{3}{1} N_{m=0}$ = 3x 2 = 6

$N_{m=1}$ (S,T)= 9 – 6 =3