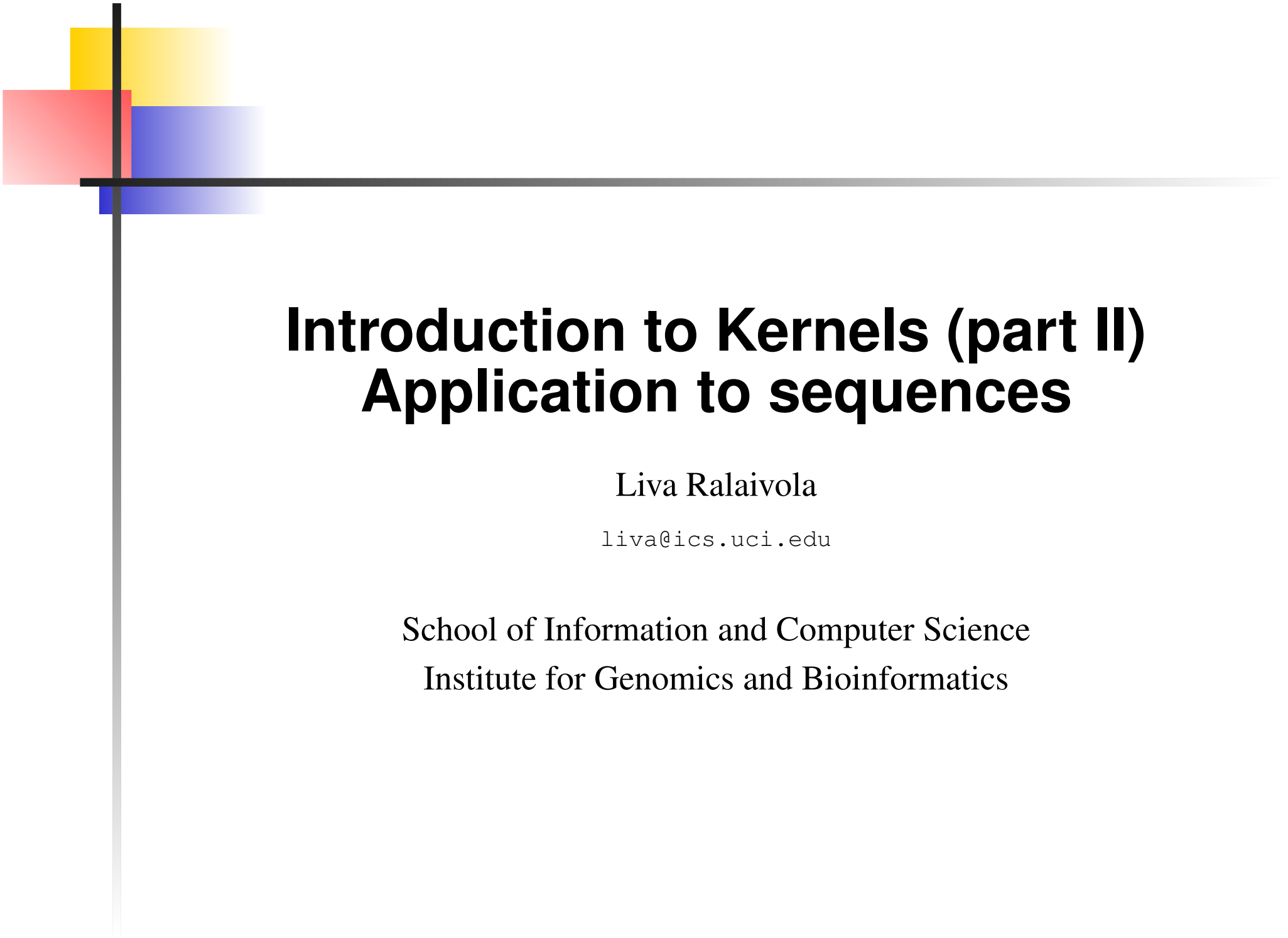


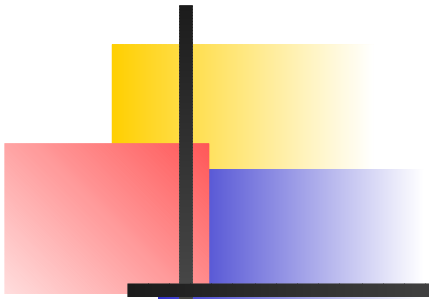
# Introduction to Kernels (part II) Application to sequences

Liva Ralaivola

liva@ics.uci.edu

School of Information and Computer Science

Institute for Genomics and Bioinformatics



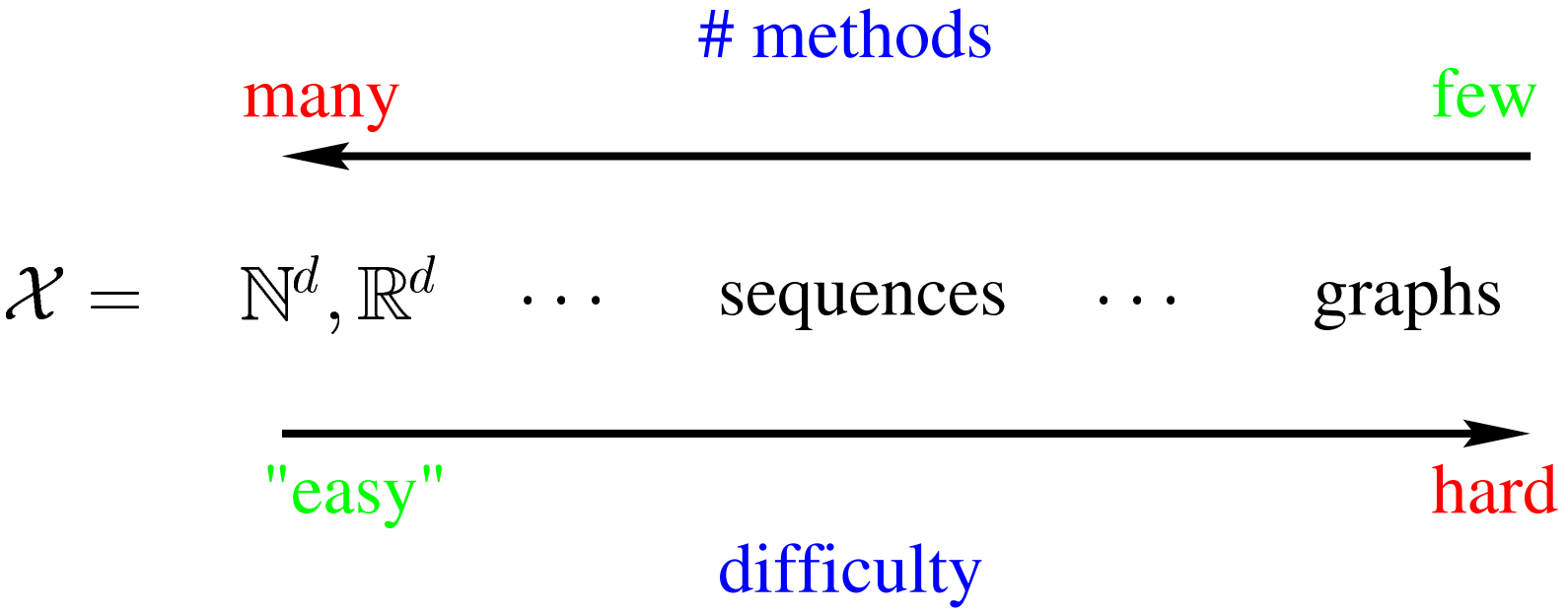
# Outline

- Classifying sequences
- A rough scale on the difficulty to classify objects
- Gram matrices
- Combining Gram matrices to classify proteins
- Building sequence kernels

# Classifying sequences

- Problem of general interest
  - classification of texts
  - classification of music/sound/speech
  - classification of web logs (user modeling)
  - ...
- and of particular interest in bioinformatics
  - remote homology detection between proteins from their sequences of amino acids
  - proteins structure prediction
  - prediction of DNA splice sites
  - ...

# A rough scale on classification problems difficulty




- Typical difficulties encountered with structured objects
  - can be of various sizes
  - no straightforward ways to do calculus
  - no all-purpose similarities
  - ...

# (Kernel) Gram matrices (1/2)

- Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel ($\mathcal{X}$ may be a space of sequences)
  - for a set of patterns $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$

$$K_{\mathcal{S}} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_\ell) \\ k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_\ell) \\ & \cdots & \cdots & \\ k(\mathbf{x}_1, \mathbf{x}_\ell) & k(\mathbf{x}_2, \mathbf{x}_\ell) & \cdots & k(\mathbf{x}_\ell, \mathbf{x}_\ell) \end{bmatrix}$$

    is the Gram matrix of $k$ with respect to $\mathcal{S}$
  - if corresponding targets $y_1, \ldots, y_\ell$ are available

    $\Rightarrow K_{\mathcal{S}}$ is sufficient for any Kernel Machine to be trained

# (Kernel) Gram matrices (2/2)

- A property of the Gram matrix (Mercer's property)

  **Proposition 1 (Semi-Positiveness of the Gram matrix).** *Let* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *be a symmetric function.*
  $k$ *is a Mercer kernel* $\Leftrightarrow$

  $$\forall \mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}, \mathbf{x}_i \in \mathcal{X}, \mathbf{v} K_{\mathcal{S}} \mathbf{v} \geq 0, \forall \mathbf{v} \in \mathbb{R}^\ell$$

- This means that for any Mercer kernel $k$ and any set of patterns $\mathcal{S}$, the Gram matrix $K_{\mathcal{S}}$ has only nonnegative eigenvalues
- This gives, in particular, $k_1$ and $k_2$ being Mercer kernels
  - $k_1^p,\ p \in \mathbb{N}$ is a Mercer kernel
  - $\lambda k_1 + \gamma k_2,\ \lambda, \gamma > 0$ is a Mercer kernel
  - $k_1 k_2$ is a Mercer kernel

# Combining Gram matrices to classify proteins (1/2)

- Reference: [Lanckriet et al., 2004]
- Problems addressed (a few thousands of yeast proteins)
  - ribosomal protein classification
  - membrane protein classification
- Method
  - use of genome-wide data sets coded as similarity/Gram matrices
  - convex combination of these kernel functions
  - Semi-definite programming and Support Vector Learning

# Combining Gram matrices to classify proteins (2/2)

- Seven kernels used (all coming from some biological knowledge)
  - $K_{SW}, K_B, K_{Pfam}$: based on amino acids sequence similarity measures provided by state-of-the-art algorithms (Smith-Waterman, BLAST and Pfam HMM)
  - $K_{FFT}$: a Fast Fourier Transform-based kernel matrix using hydropathy profiles of the proteins
  - $K_{LI}, K_D$: kernel matrices based on protein interaction information
  - $K_E$: a similarity matrix based on microarray gene expression measurements
- Omitting the technical details (SDP)
  - a kernel $K = \sum \mu_i K_i,\ \mu_i \geq 0, i = 1, \ldots, 7$ is looked for
  - at the same time an SVM classifier based on $K$ is learned
  - the results obtained are the best ever

# Building sequence kernels (1/4)

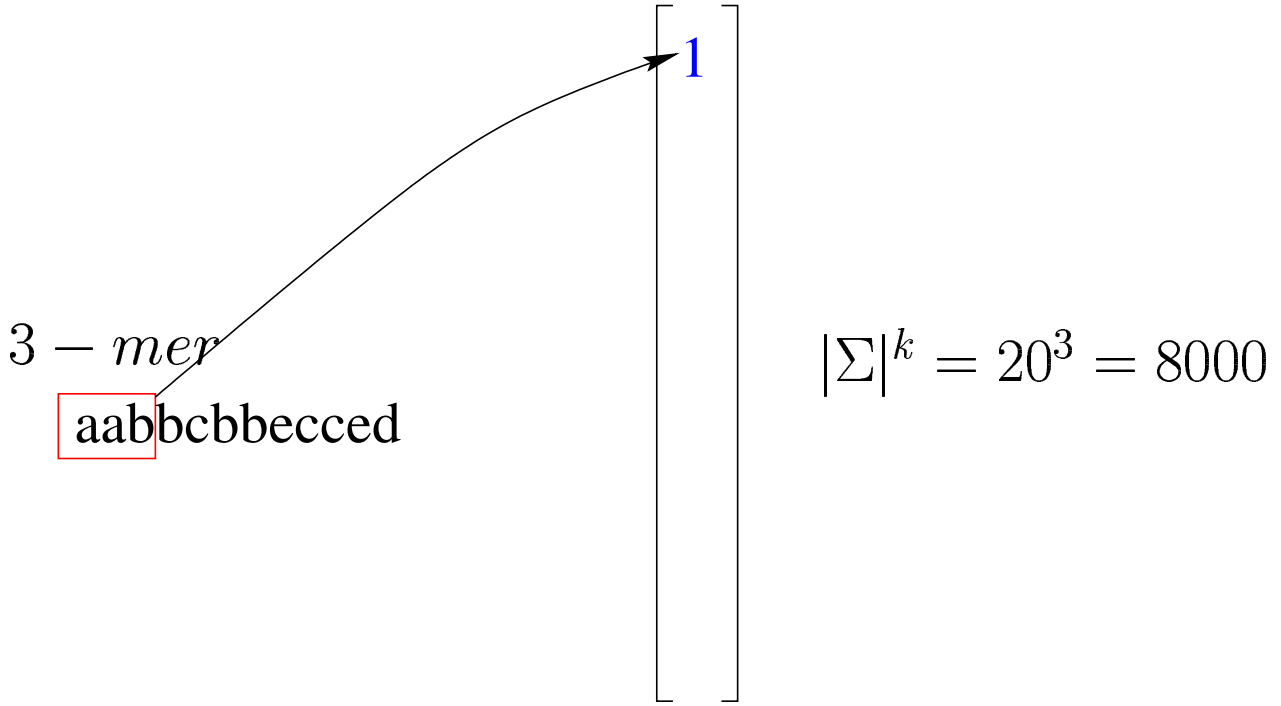
- Question: what if no such an 'exhaustive' or relevant knowledge is available?
- Answer: build you own sequence kernel
- How to do that?
  - strong idea: comparing subsequences of sequences (cf. convolution kernels [Haussler, 1999])
  - another idea: Fisher kernels [Jaakkola and Haussler, 1998]

# Building sequence kernels (2/4)

- The spectrum kernel [Leslie et al., 2002]
  - counts the number of common $k$-mers in two sequences
  - computes a value from this counts
  - for proteins: alphabet $\Sigma$ of $20$ symbols

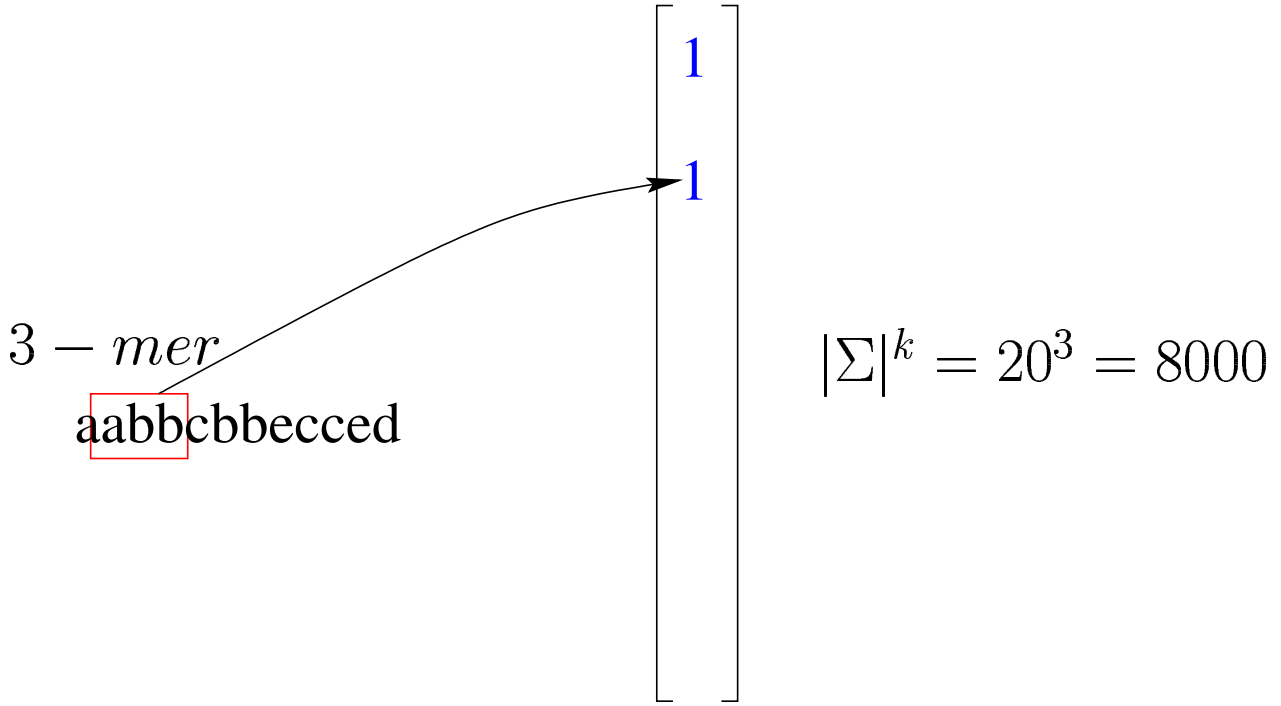
# Building sequence kernels (2/4)

- The spectrum kernel [Leslie et al., 2002]
  - counts the number of common $k$-mers in two sequences
  - computes a value from this counts
  - for proteins: alphabet $\Sigma$ of $20$ symbols



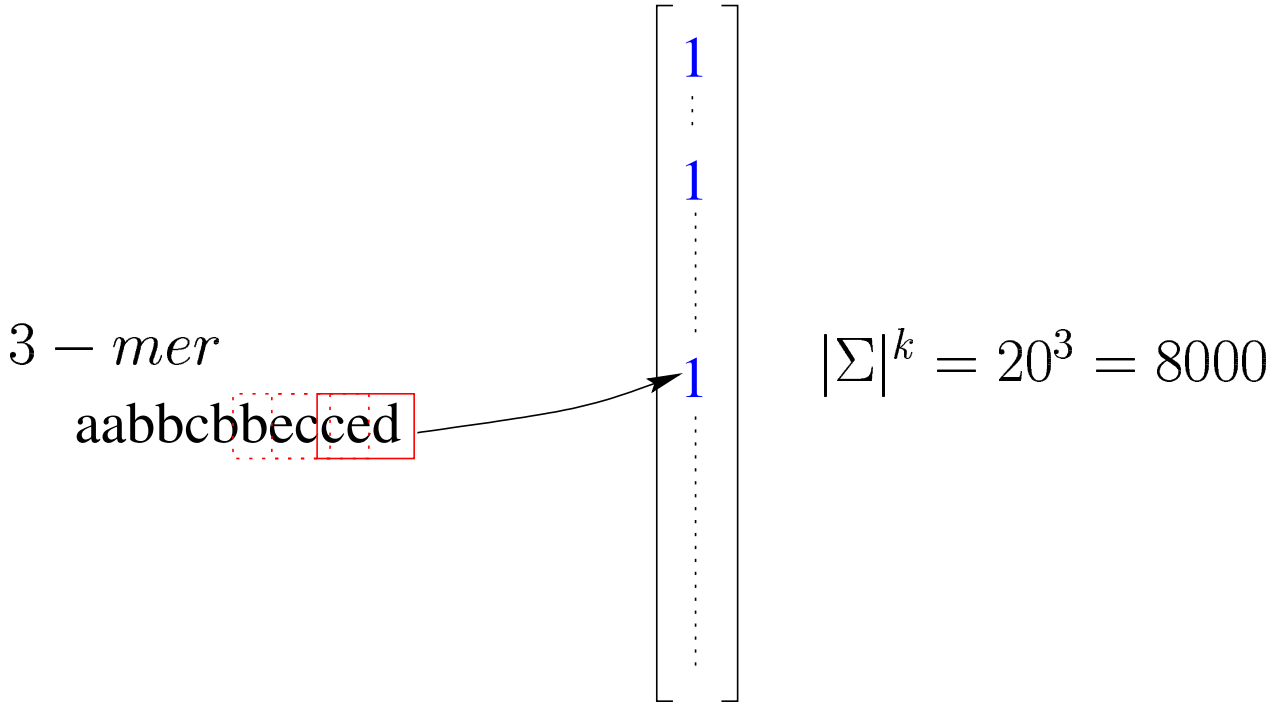
# Building sequence kernels (2/4)

- The spectrum kernel [Leslie et al., 2002]
  - counts the number of common $k$-mers in two sequences
  - computes a value from this counts
  - for proteins: alphabet $\Sigma$ of $20$ symbols



# Building sequence kernels (2/4)

- The spectrum kernel [Leslie et al., 2002]
  - counts the number of common $k$-mers in two sequences
  - computes a value from this counts
  - for proteins: alphabet $\Sigma$ of $20$ symbols



# Building sequence kernels (3/4)

- Computation of the spectrum kernel
  - explicit construction of the feature space $\mathcal{H} = \mathbb{R}^{|\Sigma|^k}$
  - feature vectors are very sparse (few nonzero elements)
  - using a suffix tree structure [Ukkonen, 1995]
    - makes it possible to compute the kernels efficiently
    - makes it possible to manage space efficiently
- Results on a protein classification problem
  - comparable to other methods (but not better)
  - raises the need of another sequence kernel

# Building sequence kernels (4/4)

- Mismatch String Kernel [Leslie et al., 2003]
  - based on idea of the spectrum kernel
  - allows mismatches in $k$-mers comparisons
    - for $k = 3$
      - spectrum kernel would consider the similarity between $aaa$ and $aab$ as being 0
      - (3,1)-mismatch string kernel would assign a value>0 to this similarity
  - can use the suffix tree data structure for efficient computations
- Results on a protein classification task
  - the use of this kernel with SVM outperformed the results of the spectrum kernel
  - and compared favorably to several state-of-the-art methods

# Conclusion

- Importance of
  - Gram matrices
  - combination of kernels (cf. *Combining Classifiers and Combining Kernels*)
- Building kernels
  - convolution and spectral kernels
  - substructures enumeration
  - complexity of kernel computation
  - Fisher kernels (cf. presentation by a group a students ?)

# References


[Haussler, 1999] Haussler, D. (1999). Convolution Kernels on Discrete Structures. Technical Report UCS-CRL-99-10, UC Santa Cruz.

[Jaakkola and Haussler, 1998] Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *Adv. in Neural Information Processing Systems*, volume 11.

[Lanckriet et al., 2004] Lanckriet, G. R. G., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004). A statistical framework for genomic data fusion. *Bioinformatics*.

[Leslie et al., 2003] Leslie, C., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. (2003). Mismatch string kernels for SVM protein classification. In *Adv. in Neural Information Processing Systems*, volume 15.

[Leslie et al., 2002] Leslie, C., Eskin, E., and Noble, W. S. (2002). The spectrum kernel: A string kernel for svm protein classification. In *Proc. of the Pacific Symposium on Biocomputing, 2002*, pages 564–575.

[Ukkonen, 1995] Ukkonen, E. (1995). On–line construction of suffix trees. *Algorithmica*, 14:249–60.