7. Mining Biological Data

Contents of this Chapter

7.1 Introduction

DNA and Proteins

Data Mining Challenges

7.2 Mining Frequent Sequential Petterns

7.3 Sequence Classification

7.4 Sequence Clustering

7.1 Introduction

Motivation

Many biological processes are not well-understood

Biological knowledge is

- Highly complex
- Descriptive and experimental
  → Different from physics / chemistry

Wide availability of biological data

- Genome sequencing
- Protein sequencing
- Microarray expression data

Data mining methods to gain biological insights

7.1 Proteins

Function

Structural Proteins
- Building blocks of various tissues

Enzymes
- Catalyze chemical reactions

Transporters
- Carry chemical elements from one part of organism to another

Antibody Proteins
- Part of the immune system

7.1 Proteins

Structure

1D Structure
- Chains of amino-acids: AVFAMLCNFQDMAQSWKKKAVFAAGDE . . .
- 20 different amino-acids (one / three letter codes)
- Typical length of proteins: 3 to 400 amino-acids

Physico-chemical properties
- Hydrophic / hydrophile
- Charged / uncharged
- Polar / non-polar

same properties imply similarity of proteins

Amino-acid	**Three-Letter Code**	**One-Letter Code**	**Physico-chemical Properties**
Alanine	Ala	A	Hydrophobic
Lusine	Lys	K	Charged
Glutamine	Gln	Q	Polar
. . .	. . .	. . .	. . .

7.1 Proteins

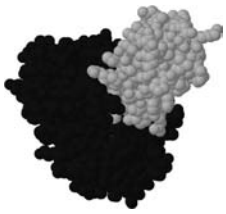
Structure

2D Structure

- Subsequences of the 1D structure form 2D structures such as sheets, strands, . . .

3D Structure

- Coordinates of the atoms in 3D space
- Known only for small subset of all sequenced proteins
- Protein surface important for many biological processes



Protein-Protein Docking

7.1 Proteins

Databases

Swiss-Prot (http://www.ebi.ac.uk/swissprot/)

- Proteins with their 1D structure
- Entries have been checked for sequencing errors
- Entries have a textual description (annotation): organism, function, references to publications, other related information
- Currently, 120‘960 entries

Protein Data Bank (PDB) (http://www.rcsb.org/pdb/)

- Proteins with their 1D, 2D and 3D structure
- Plus annotations
- Currently, 17‘828 protein entries

7.1 DNA (Desoxyribonucleic Acid)

Basics

Function

- Genetic information (genome)
- Codes proteins

Structure

- Chain of nucleotides (bases)
- Four different nucleotides:
  Adenine (A), Guanine (G), Cytosine (C), Thymine (T)
- Two interconnected parallel strands of nucleotides (double helix)
- Second strand is redundant

  DNA can be represented as sequence of nucleotides

7.1 DNA

DNA and Proteins

Structure

- Triplets of nucleotides code one amino-acid
- Genetic code: 64 different nucleotide triplets → 20 different amino-acids

  redundancy
- genome → chromosomes → genes → triplets
  = =
  protein amino-acid

Genes

- Protein coding region (exon / expressed region)
- Delimited by start / stop codons
- Largest part of genome is non-coding (introns)

  95% of human genome is non-coding

7.1 DNA

Mutations

Types

- Substitutions
  one base ←→ another base
- Insertions
  of one or more bases
- Deletions
  of one or more consecutive bases

At gene level or (less frequently) at protein level
Protein mutations may destroy function / create new function

7.1 Sequence Alignment

Goal

- Given two or more input sequences
- Identify similar sequences with long conserved subsequences

Method

- Use substitution matrices (probabilities of substitutions of amino-acids / bases) and probabilities of insertions and deletions
- *Optimal* alignment problem: NP-hard
- Heuristic method to find *good* alignments
- Many algorithms, e.g. BLAST
- Result:
  Alignment of the input sequences
  Similarity measures, score and % sequence identity (for two input sequences)

7.1 Sequence Alignment

Example

Input: ABFGRP, BDFLRP, AFRP

```
AB-FGRTP
-BDFLR-P          - gap
A--F-R-P
```

Output: abdFlR-P (capital letters conserved in all sequences)

→ Consensus sequence

7.1 Sequence Alignment

Example

Input: AAAAABBBBB, BBBBBAAAAA

Solution 1:

```
-----AAAAABBBBB
BBBBBAAAAA-----
```

Output: AAAAA

Solution 2:

```
-----BBBBBAAAAA
AAAAABBBBB-----
```

Output: BBBBB

→ One of the two domains (AAAAA, BBBBB) will always be missed

7.1 Mining Biological Data

Data Mining Tasks

Tissue classification from micro-array data

- Input: micro-array data for a small number of tissues from two classes (e.g., cancer and normal)
- Goals: (1) accurate classification and (2) discovery of responsible genes

Protein subcellular localization prediction

- Input: protein sequences with their subcellular localization types (e.g., cytoplasmic, periplasmic and extracellular)
- Goals: (1) accurate classification and (2) insight into the determining factors

7.1 Mining Biological Data

Data Mining Tasks

Protein secondary structure prediction

- Input: set of proteins with sequences and 3D structures
- Goal: accurate prediction of the (unknown) 3D structure based on the (known) sequence
  → Structure is a strong indicator of function

Detection of protein families

- Input: set of protein sequences
- Goal: hierarchical structure of protein superfamilies, families, . . .
  → Clustering problem

7.1 Mining Biological Data

Challenges

- Ambiguity of genetic and protein sequences
  Same sequence can have different functions, different sequences same function
- High percentage of noise
  Large portions of genetic data seem to carry no information
- Representation of sequence and 3D data
  No straightforward mapping to a feature space
- Integration of different datatypes
  Sequence, 3D, textual, micro-arrays, . . .
- High precision
  Required for biological applications
- Understandability of discovered knowledge (biological insights!)

7.2 Mining Frequent Sequential Patterns

Motivation

- Similar sequences have the same or similar function with a high probability
- Typically large portions of DNA or protein sequences are considered to be noise
- Sequential patterns determining the function are expected to be relatively short and to occur much more frequently than (random) noise patterns

Find *frequent* (sub)sequences / patterns

- Many frequent sequences
- Find only interesting frequent sequences

E.g., find *maximal* frequent patterns,
(all of its superpatterns are infrequent)

7.2 Mining Frequent Sequential Patterns

Approaches

Bottom-Up Enumeration

- Begin with empty pattern
- Extend in all possible ways
- If extension has minimum support, then continue extending it, else discard the extended pattern

A	B	D	K
AA	AD	DA	DD
	AAD		ADD

Top-Down Alignment

- Align all pairs of sequences
- Continue aligning the alignments until their support reaches minimum support

ABDDKA BADDKDFF BBADD

ADDK BADD

ADD

7.2 Frequent Sequential Patterns in Transactional Data

Overview

Data

- *Sequences* of transactions
- Transaction: *set* of items or attribute-value pairs (with timestamp)

Patterns

time t1 Later time t2

((T = 90F) → (H=60%, P=1.1atm))

attribute value item item item

item

itemset

- k-sequence: sequence /pattern with k items

  Examples: T1→H2P1T3→P2 and P1T2→H4P2T5 are 5-sequences

- S1 is subsequence of S2 (S1 ≤ S2)

  T1→P1T2 ≤ H1T1→P2→H2P1T2 (T1 ⊆ H1T1 , P1T2 ⊆ H2P1T2)

7.2 Frequent Sequential Patterns in Transactional Data

DATABASE

Lid	Time	Event	Lid	Time	Event	Lid	Time	Event	Lid	Time	Event
0	0	H2	4	0	H2	8	0	P1H2	12	0	T1
0	1	T2	4	1	T2	8	1	T2	12	1	H3P1
0	2	P3T3	4	2	P3T3	8	2	P3T3	12	2	P2
1	0	H2T3	5	0	T1	9	0	H2T3	13	0	T1
1	1	T2	5	1	H3P1	9	1	T2	13	1	H3P1
1	2	T3	5	2	P2	9	2	H1T3	13	2	P2
2	0	P1H2	6	0	T1	10	0	T1	14	0	H2
2	1	T2	6	1	H3P1	10	1	H3P1	14	1	T2
2	2	H1T3	6	2	P2	10	2	P2	14	2	T3
3	0	H2	7	0	T1	11	0	T1	15	0	T1
3	1	T2	7	1	H3P1	11	1	H3P1	15	1	H3P1
3	2	H1T3	7	2	P2	11	2	P2	15	2	P2

Example

FREQUENT SEQUENCES

Frequent 1-sequences		Frequent 2-sequences		Frequent 3-sequences	
H2	8	H2→T2	8	H2→T2→T3	8
H3	8	H2→T3	8	T1→H3P1	8
P1	8	T2→T3	8	T1→H3→P2	8
P2	8	T1→H3	8	T1→P1→P2	8
T1	8	T1→P1	8	H3P1→P2	8
T2	8	T1→P2	8		
T3	8	H3P1	8		
		H3→P2	8		
		P1→P2	8		

Frequent 4-sequences	
T1→H3P1→P2	8

7.2 Frequent Sequential Patterns in Transactional Data

GSP [Srikant & Agrawal 1996]

Problem Specification

- Sliding window model with maximum gap / minimum gap
- Item taxonomy / graph

Bottom-Up Enumeration

- Candidate generation

  Generate (join) *k*-candidates from two *k*-1 frequent patterns

  (a) (b) (c) and (b) (c) (d) → (a) (b) (c) (d)

- Support counting

  Hash-tree for storing candidates

  Transform data sequences using item taxonomy

7.2 Frequent Sequential Patterns in Transactional Data

Discussion

Breadth-first search

- Generate all *k*-1-patterns before starting with the *k*-candidates
- Number of patterns may become very large
  - → Not all candidates fit into memory

Support counting

- Requires one DB scan for each level / length of patterns
- Very expensive operation
  - → Need more efficient method

7.2 Frequent Sequential Patterns in Transactional Data

SPADE [Zaki 2001]

Depth-first search

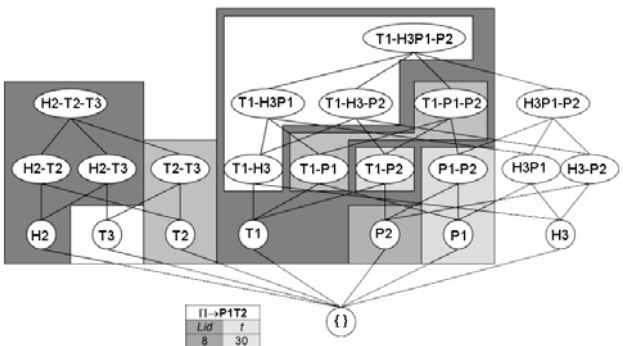
- Extend a frequent *k*-sequence until it becomes infrequent before considering another *k*-sequence
- Need only the path from the root of the lattice (of all patterns) to the current sequence in main memory

Less pruning possible

Vertical representation

- For a candidate sequence, store list of occurrences (sequence, position) or (lid, time) resp.
- Initially, representation of the database:
  For each 1-sequence, store its occurrences
- Support counting: merge of two lists → no DB access

7.2 Frequent Sequential Patterns in Transactional Data



Example

T1→H3P1→P2	
Lid	t
5	2
6	2
7	2
10	2
11	2
12	2
13	2
15	2

T1→H3P1	
Lid	t
5	1
6	1
7	1
10	1
11	1
12	1
13	1
15	1

T1→H3→P2	
Lid	t
5	2
6	2
7	2
10	2
11	2
12	2
13	2
15	2

T1→H3	
Lid	t
5	1
6	1
7	1
10	1
11	1
12	1
13	1
15	1

T1→P1	
Lid	t
5	1
6	1
7	1
10	1
11	1
12	1
13	1
15	1

T1	
Lid	t
5	0
6	0
7	0
10	0
11	0
12	0
13	0
15	0

H3	
Lid	t
5	1
6	1
7	1
10	1
11	1
12	1
13	1
15	1

P1	
Lid	t
2	0
5	1
6	1
7	1
8	0
10	1
11	1
12	1
13	1
15	1

P2	
Lid	T
5	2
6	2
7	2
10	2
11	2
12	2
13	2
15	2

Π→P1T2	
Lid	t
8	30
8	50
8	80

Π→P1	
Lid	t
1	20
1	30
1	40
4	60
7	40
8	10
8	30
8	50
8	80
13	50
13	70
15	60
17	20
20	10

Π→T2	
Lid	t
1	70
1	80
3	10
5	70
8	30
8	40
8	50
8	80
11	30
13	10
16	80
20	20

Π→T2→P1	
Lid	t
8	50
8	80
13	50
13	70

Π→P1→T2	
Lid	T
1	70
1	80
8	30
8	40
8	50
8	80

7.2 Frequent Sequential Patterns in Transactional Data

Comparison with Mining Biological Data

- Data is sequences of *sets* of items instead of sequences of single symbols.
- Sequential order represents *temporal*, not spatial relationship.
- *Many short* data sequences, instead of few long ones.
- *No explicit gaps* in patterns (gaps do not matter).
- Typically, these methods discover *all* frequent patterns.

7.2 Frequent Sequential Patterns in Biological Data

Pattern Types

Concrete patterns

- Strings from the alphabet Σ (nucleotides or amino-acids)

  ABDAWWF

With rigid gaps

- Introduce "." (matches *one* arbitrary symbol from Σ)

  A. . .BDA..W.WF

With unrestricted gaps

- Introduce "*" (matches *zero or more* arbitrary symbols from Σ)

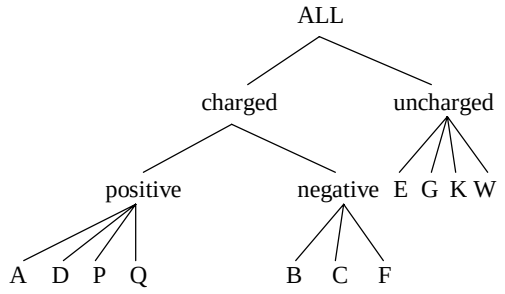
  ABD*AWW*F

7.2 Frequent Sequential Patterns in Biological Data

Pattern Types

With concepts

- Introduce concepts (subsets of Σ), e.g. "unchargedBLLApositivenegative"
- With a tree structure or even a graph structure



7.2 Frequent Sequential Patterns in Biological Data

TEIRESIAS [Rigoutsos & Floratos 1998]

- Pattern class: Σ (Σ U {'.'})* Σ

  e.g. A.CH..E or SA.CH..E

- Restricted to <l,w> patterns with l <= w: every subpattern of length w or more contains at least l symbols from Σ

  patterns must be "dense enough"

- Finds all *maximal* <l,w> patterns with support of at least *min-supp*

7.2 Frequent Sequential Patterns in Biological Data

Method

Scanning phase

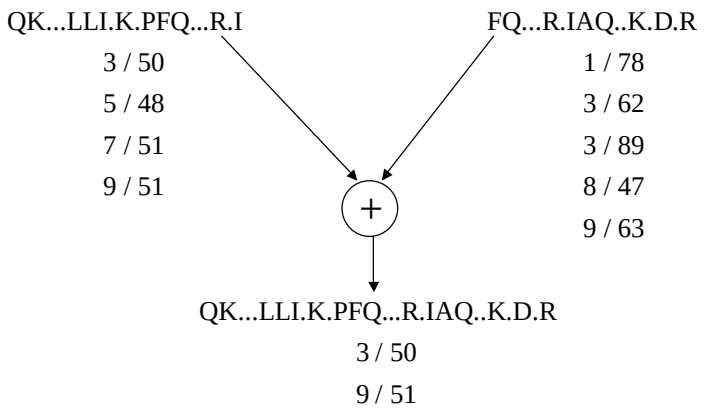
- Determine all elementary patterns: frequent <l,w> patterns with exactly l symbols from Σ
- Ex.: <3,4> patterns F.AS, AST, AS.S, STS, A.TS

Convolution phase

- Join pairs of elementary patterns P1 and P2 where the suffix of P1 is identical to the prefix of P2 (convolution)
- Ex.: F.AS and AST → F.AST and AST
  F.AST and STS → F.ASTS

7.2 Frequent Sequential Patterns in Biological Data

Example



7.2 Frequent Sequential Patterns in Biological Data

Discussion

Properties

- Can extend patterns by more than one symbol at a time
- Maximal patterns generated before non-maximal subpatterns
- Compare new frequent pattern with all frequent patterns already discovered
  - → Use hash table to efficiently locate superpatterns
- Returns only maximal patterns
- Generates all such patterns
- Can also handle concept trees / graphs

7.2 Frequent Sequential Patterns in Biological Data

Top-Down Alignment Method [Martinez 1988]

- Align all pairs of input sequences
- Pairwise alignments have (at least) a support of two
- Score all pairwise alignments and order them according to decreasing score

  place similar alignments close together
- Iteratively, keep aligning the alignments

  until their support reaches minimum support

7.2 Frequent Sequential Patterns in Biological Data

Other Alignment Methods

- Use pairwise alignments to create dendrogram
  and apply hierarchical clustering algorithm
- Perform multiple sequence alignment
  and create consensus sequence directly

Generates only one pattern (consensus pattern)

7.3 Sequence Classification

Overview

Feature Selection

Support Vector Machines (SVM)

- Application for sequence classification

Markov Models

- Markov chains
- Hidden Markov models

7.3 Sequence Classification

Types of Features

Amino-acid / Nucleotide Composition

- 20 dim. / 4 dim. vectors

Physico-chemical properties

- Hydrophobicity, charge, polarity, size, . . .

Subsequences

- All possible subsequences of length *k*
- All frequent subsequences

7.3 Sequence Classification

Feature Selection

Method

- Measure the *relevance* of features w.r.t. classification:

  T-test for continuous attributes $t = \frac{\mu_1 - \mu_2}{\sigma}$

  Mutual information for categorical attributes

- Consider the *redundancy* of features

  Minimize correlation among selected features

  → Weighted combination of relevance and redundancy

- Greedily, select top *k* features

7.3 Sequence Classification

SVM for Protein Classification [Leslie et al 2002]

- Two sequences are similar when they share many common substrings (subsequences)
- $K(x, x') = \sum_{s \text{ common substring}} \lambda^{|s|}$ where λ is a parameter

  and |s| denotes the length of string s

- Very high classification accuracy for protein sequences
- Variation of the kernel (when allowing gaps)

$$K(x, x') = \sum_{s \text{ common substring}} \lambda^{length(s,x)+length(s,x')}$$

length(s,x): length of the subsequence of x matching s

7.3 Sequence Classification

SVM for Prediction of Translation Initiation Sites [Zien et al 2000]

- Translation initiation site (TIS): starting position of a protein coding region in DNA

  all TIS start with the triplet "ATG"
- Problem: given an "ATG" triplet, does it belong to a TIS?
- Representation of DNA

  window of 200 nucleotides around candidate "ATG"

  encode each nucleotide with a 5 bit word (00001, 00010, . . ., 10000) for

  A, C, G, T and unknown

  → Vectors of 1000 bits

7.3 Sequence Classification

SVM for Prediction of Translation Initiation Sites

- Kernels

  $K(x, x') = (\mathrm{x.x'})^d$ d = 1: number of common bits

  d = 2: number of common pairs of bits

  . . .

  locally improved kernel: compare only small window around "ATG"
- Experimental results

  long range correlations do not improve performance

  locally improved kernel performs best

  outperforms state-of-the-art methods

7.3 Sequence Classification

Markov models

- Markov chains (Markov models)

  Symbol in a sequence depends only on its preceding symbol(s)

  Can be used for classification

  [Deshpande & Karypis 2002]

- Hidden Markov Models

  Symbol in a sequence depends on a hidden state

  State depends on preceding state

7.3 Sequence Classification

1-order Markov Chains

- For each class, determine the conditional probabilities $P(s_i|s_j)$

  → For each pair of symbols s_i and s_j

- For each class c_i, calculate the probability $P(s|\ c_i)$

  of observing the given sequence $s = s_1 s_2 \cdots s_L$

$$P(s \mid c_i) = P(s_L \mid s_{L-1}, c_i) \cdot \cdots \cdot P(s_2 \mid s_1, c_i) \cdot P(s_1 \mid c_i)$$

- Choose the class with the highest likelihood
- Decision function for two classes (+ and -)

$$f(s) = \sum_{i=1}^{L} \log \frac{P(s_i \mid s_{i-1}, +)}{P(s_i \mid s_{i-1}, -)}$$

7.3 Sequence Classification

Higher-order Markov Chains

Idea

- Symbol in a sequence depends on all its *k* preceding symbols

Discussion

- In general: higher classification accuracy than 1-order Markov chains
- But

Exponential number of transition probabilities

Hard to accurately estimate these probabilities

7.3 Sequence Classification

Hidden Markov Models

- Goal: distinguish patterns (motifs) from background in a sequence

  motif: relatively short highly conserved region in a biological sequence
- Hidden Markov Model (HMM)

  Generative process for motifs/patterns of length L with

  consensus pattern (motif)

  noise level ε

  frequency F
- Hidden states: one for each position of the motif, one for the background

  Determines the next symbol to be generated (multinomial distribution)

  Determines the next state (transition probabilities)

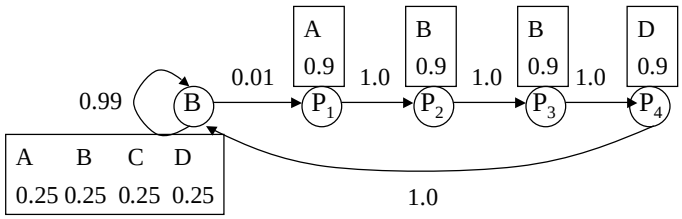
7.3 Sequence Classification

Basics

- Background state: probability of symbols = frequency in background
- Pattern states $P_i, 1 \le i \le L$

  Symbol at position *i* in consensus pattern: probability $1-\varepsilon$

  Other symbols: probability ε
- Example (consensus pattern ABBD, uniform background)



7.4 Sequence Clustering

Overview

Alignment-Based Methods

- Pairwise alignment allows to define similarity / distance
- Hierarchical agglomerative clustering
- Connected components of graph

Frequent-Sequence-Based Methods

- No alignment, but mining of frequent subsequences
- Use vector space model and any applicable algorithm

7.4 Sequence Clustering

Alignment-Based Methods

Hierarchical agglomerative clustering [Barton & Sternberg 1987]

- Perform all pairwise alignments
- Define appropriate similarity measure:

  percentage identity, normalised alignment score (raw score divided by the length of the alignment), etc.
- Apply agglomerative hierarchical clustering

Runtime complexity > $O(n^2)$

7.4 Sequence Clustering

Alignment-Based Methods

Connected components of graph [Bolten et al 2000]

- Homologue proteins: share an ancestor
- Many homologue proteins do not have a significant sequence similarity
- Need to consider transitivity of homology
- Construct a graph: nodes = sequences, edges = significant sequence similarity
- Clusters: connected components of this graph

→ Runtime for clustering SwissProt: 600 CPU days

7.4 Sequence Clustering

Frequent-Sequence-Based Methods

Method [Guralnik & Karypis 2001]

- Determine all frequent subsequences
- Efficiently select relevant subset of these sequences (features)
- Count occurrences of features (vector space model)
- Apply any clustering algorithm for vector spaces

  e.g. *k*-means

  Very efficient

  But feature selection is difficult

References

- Barton G. J., Sternberg M. J. E.: „A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisons“, *J. Mol. Biol*, 1987
- Bolten E., Schliep A., Schneckener S., Schomburg D., Schrader R.: „Clustering Protein Sequences Structure Prediction by transitive homology“, 2000
- Burges C. J. C.: “A Tutorial on Support Vector Machines for Pattern Recognition”, *Knowledge Discovery and Data Mining*, 1998.
- Deshpande M., Karypis G.: “Evaluation of Techniques for Classifying Biological Sequences”, PAKDD 2002
- Guralnik V., Karypis G.: „A scalable algorithm for clustering sequential data“ Proc. of the 1st *IEEE International Conference on Data Mining* (ICDM 2001), 2001
- Leslie C., Eskin E., Noble W.S.: „The Spectrum Kernel: A String Kernel for SVM Protein Classification“, *Proc. Pacific Symposium on Biocomputing*, 2002.

References

- Martinez H.M.: "A Flexible Multiple Sequence Alignment Program", *Nucleic Acids Research*, 1988.
- Rigoutsos I. , Floratos A: "Combinatorial Pattern Discovery In Biological Sequences: The TEIRESIAS Algorithm.", *Bioinformatics*, 1998.
- Srikant R., Agrawal R.: „*Mining sequential patterns: generalizations and performance improvements*", Proc. 5th EDBT, 1996.
- Zaki M.: "SPADE: An Efficient Algorithm for Mining Frequent Sequences", *Machine Learning*, 2001.
- Zien A., Ratsch G., Mika S., Schoelkopf B., Lengauer T., Muller K.-R.: "Engineering Support Vector Machine Kernels that Recognize Translation Initiation Sites", *Bioinformatics*, 2000