

Data Mining: Concepts and Techniques

— Chapter 8 —

8.4. Mining sequence patterns in biological data

Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign

www.cs.uiuc.edu/~hanj

©2006 Jiawei Han and Micheline Kamber. All rights reserved.

11/26/2007

Data Mining: Principles and Algorithms

1

Chapter 8. Mining Stream, Time-Series, and Sequence Data

- Mining data streams
- Mining time-series data
- Mining sequence patterns in transactional databases
- **Mining sequence patterns in biological data** ←

11/26/2007

Data Mining: Principles and Algorithms

2

Mining Sequence Patterns in Biological Data

- A brief introduction to biology and bioinformatics ←
- Alignment of biological sequences
- Hidden Markov model for biological sequence analysis
- Summary

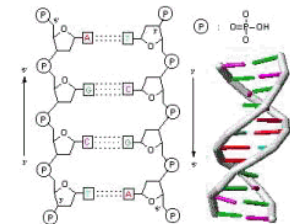
11/26/2007

Data Mining: Principles and Algorithms

3

Biology Fundamentals (1): DNA Structure

- DNA: helix-shaped molecule whose constituents are two parallel strands of nucleotides
- DNA is usually represented by sequences of these four nucleotides
- This assumes only one strand is considered; the second strand is always derivable from the first by pairing A's with T's and C's with G's and vice-versa



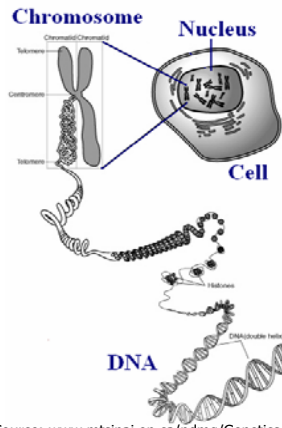
- Nucleotides (bases)
 - Adenine (A)
 - Cytosine (C)
 - Guanine (G)
 - Thymine (T)

11/26/2007

Data Mining: Principles and Algorithms

4

Biology Fundamentals (2): Genes



- **Gene:** Contiguous subparts of single strand DNA that are templates for producing *proteins*. Genes can appear in either of the DNA strand.
 - **Chromosomes:** compact chains of coiled DNA
- **Genome:** The *set of all genes* in a given organism.
- **Noncoding part:** The function of DNA material between genes is largely unknown. Certain intergenic regions of DNA are known to play a major role in *cell regulation* (controls the production of proteins and their possible interactions with DNA).

Source: www.mtsinai.on.ca/pdmg/Genetics/basic.htm
11/26/2007 Data Mining: Principles and Algorithms

5

Biology Fundamentals (3): Transcription

- **Proteins:** Produced from DNA using 3 operations or transformations: *transcription*, *splicing* and *translation*
 - In *eukaryotes* (cells with nucleus): genes are only a minute part of the total DNA
 - In *prokaryotes* (cells without nucleus): the phase of splicing does not occur (no pre-RNA generated)
- DNA is capable of replicating itself (*DNA-polymerase*)
- **Center dogma:** The capability of DNA for replication and undergoing the three (or two) transformations
- Genes are *transcribed* into pre-RNA by a complex ensemble of molecules (*RNA-polymerase*). During transcription T is substituted by the letter U (for *uracil*).
- Pre-RNA can be represented by alternations off sequence segments called *exons* and *introns*. The exons represents the parts of pre-RNA that will be *expressed*, i.e., translated into proteins.

11/26/2007

Data Mining: Principles and Algorithms

6

Biology Fundamentals (4): Proteins

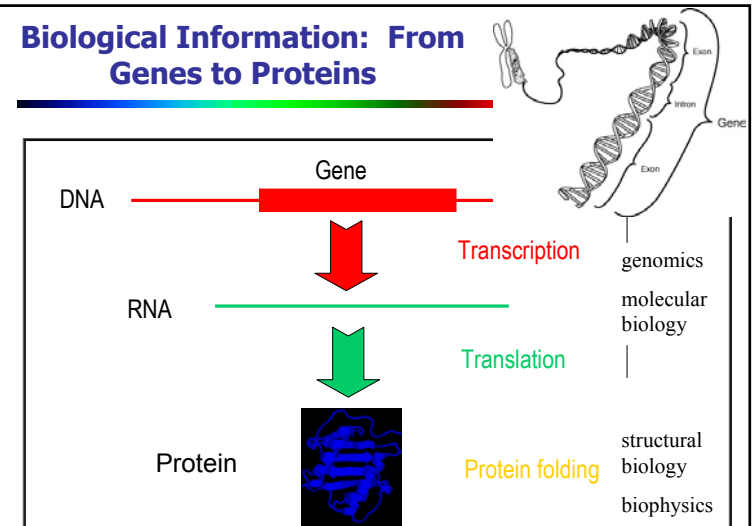
- **Splicing** (by spliceosome—an ensemble of proteins): concatenates the exons and excises introns to form mRNA (or simply RNA)
- **Translation** (by ribosomes—an ensemble of RNA and proteins)
 - Repeatedly considers a *triplet* of consecutive nucleotides (called *codon*) in RNA and produces one corresponding amino acid
 - In RNA, there is one special codon called *start codon* and a few others called *stop codons*
- An **Open Reading Frame (ORF)**: a sequence of codons starting with a start codon and ending with an end codon. The ORF is thus a sequence of nucleotides that is used by the ribosome to produce the sequence of amino acid that makes up a protein.
- There are basically **20 amino acids** (A, L, V, S, ...) but in certain rare situations, others can be added to that list.

11/26/2007

Data Mining: Principles and Algorithms

7

Biological Information: From Genes to Proteins



11/26/2007

Data Mining: Principles and Algorithms

8

Biology Fundamentals (5): 3D Structure

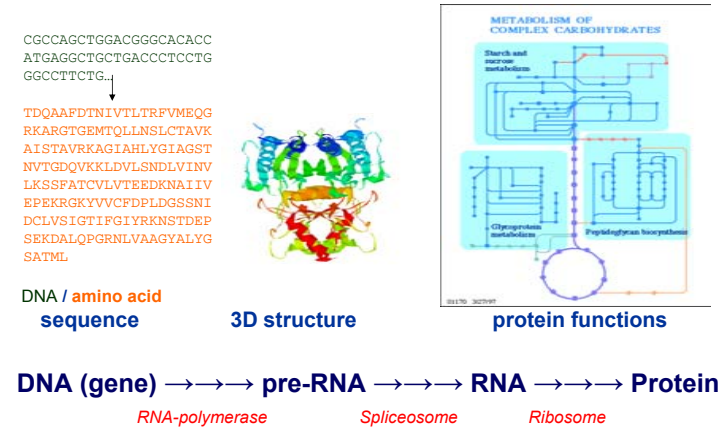
- Since there are 64 different codons and 20 amino acids, the “table look-up” for translating each codon into an amino acid is redundant: multiple codons can produce the same amino acid
- The table used by nature to perform translation is called the *genetic code*
- Due to the *redundancy* of the genetic code, certain nucleotide changes in DNA may not alter the resulting protein
- Once a protein is produced, it folds into a unique structure in 3D space, with 3 types of components: α -*helices*, β -*sheets* and *coils*.
- The *secondary* structure of a protein is its sequence of amino acids, annotated to distinguish the boundary of each component
- The *tertiary* structure is its 3D representation

11/26/2007

Data Mining: Principles and Algorithms

9

From Amino Acids to Proteins Functions

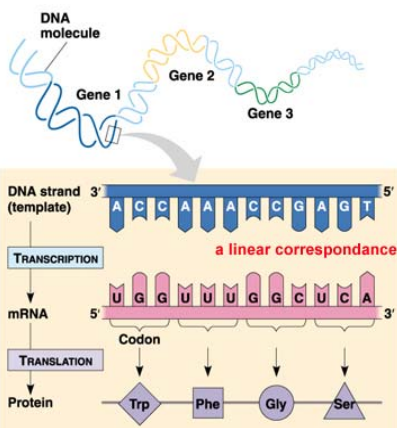


11/26/2007

Data Mining: Principles and Algorithms

10

Biology Fundamentals (6): Functional Genomics

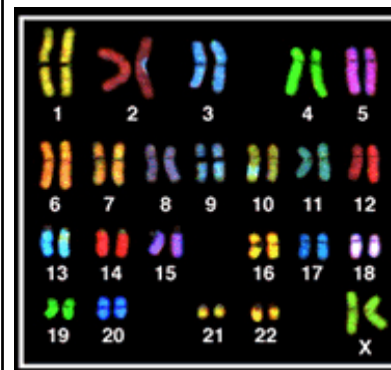


- The *function* of a protein is the way it participates with other proteins and molecules in keeping the cell alive and interacting with its environment
- Function is closely related to tertiary structure
- Functional genomics*: studies the function of all the proteins of a genome

Source: fajercp.magnet.fsu.edu/Education/2010/Lectures/26_DNA_Transcription.htm
nd Algorithms

11

Biology Fundamentals (7): Cell Biology



Human Genome—23 pairs of chromosomes

Source: www.mtsinai.on.ca/pdmg/images/pairscolour.jpg

11/26/2007

Data Mining: Principles and Algorithms

12

- A cell is made up of molecular components that can be viewed as 3D-structures of various shapes
- In a living cell, the molecules interact with each other (w. shape and location). An important type of interaction involve catalysis (*enzyme*) that facilitate interaction.
- A *metabolic pathway* is a chain of molecular interactions involving enzymes
- Signaling pathways* are molecular interactions that enable communication through the cell's membrane

Lab Tools for Determining Bio. Data (I)

- **Sequencer**: machines capable of reading off a sequence of nucleotides in a strand of DNA in biological samples
 - It can produce 300k base pairs per day at relatively low cost
 - A user can order from biotech companies vials containing short sequences of nucleotides specified by the user
- Since sequences gathered in a wet lab consist of short random segments, one has to use the **shotgun method** (a program) to reassemble them
 - Difficulty: redundancy of seq. and ambiguity of assembly.
- **Mass spectroscopy**: identifies proteins by cutting them into **short** sequences of amino acids (**peptides**) whose molecular weights can be determined by a mass spectrograph, and then computationally infer the constituents of peptides

11/26/2007

Data Mining: Principles and Algorithms

13

Lab Tools for Determining Bio. Data (II)

- The **3D-structure** of proteins is mainly determined (costly) by
 - **X-ray crystallography**: X-ray passing through a **crystallized** sample of that protein, and
 - **nuclear magnetic resonance (NMR)**: obtain a number of matrices that express that fact that two atoms are within a certain distance and then deduce a 3D shape
- **Expressed sequence tags (ESTs)**: RNA chunks that can be gathered from a cell in minute quantities (not containing the materials that would be present in introns), can be used to infer positions of introns
- **Libraries of variants of a given organism**:
 - Each variant may correspond to cells having a single one of its genes knocked out
 - Enable biologists to perform experiments and deduce information about cell behavior and fault tolerance
 - **RNA-i**: (the **i** denoting interference): chunks of the RNA of a given gene are inserted in the nucleus of a cell, that may prevent the production of that gene

11/26/2007

Data Mining: Principles and Algorithms

14

Lab Tools for Determining Bio. Data (III)

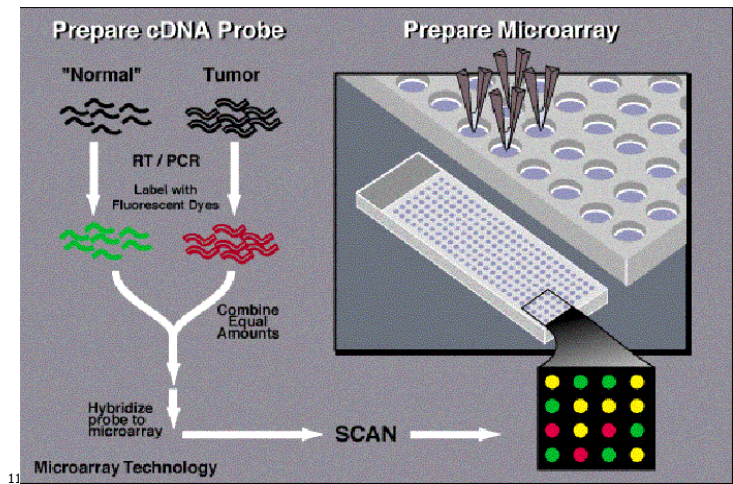
- **Microarrays**: determine simultaneously the amount of mRNA production (gene expression) of thousands of genes. It has 3 phases:
 - Place thousands of different one-strand chunks of RNA in minuscule wells on the surface of a small glass chip
 - Spread genetic material obtained by a cell experiment one wishes to perform
 - Use a laser scanner and computer to measure the amount of combined material and determine the degree (a real number) of gene expression for each gene on the chip
- **Protein-arrays**: chips whose wells contain molecules that can be bound to particular proteins (for study of protein expression)
- Determining protein interaction by **two-hybrid** experiments:
 - Construct huge Boolean matrices, whose rows and columns represent the proteins of a genome
 - If a protein interacts with another, the corresp. position is set to true

11/26/2007

Data Mining: Principles and Algorithms

15

Gene Expression and Microarray



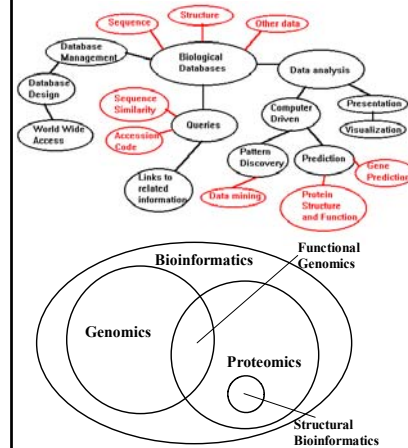
11

16

Biological Data Available

- Vast majority of data are *sequence of symbols* (*nucleotides*—genomic data, but also good amount on *amino acids*).
- Next in volume: *microarray* experiments and also *protein-array* data
- Comparably small: *3D structure of proteins* (PDB)
- *NCBI* (National Center for Biotechnology Information) server:
 - Total 26B bp: 3B bp human genome, then several bacteria (e.g., E. Coli), higher organisms: yeast, worm, fruitfly, mouse, and plants
 - The largest known genes has ~20million bp and the largest protein consists of ~34k amino acids
 - PDB has a catalogue of only 45k proteins, specified by their 3D structure (i.e, need to infer protein shape from sequence data)

Bioinformatics



- Computational management and analysis of biological information
- Interdisciplinary Field (Molecular Biology, Statistics, Computer Science, Genomics, Genetics, Databases, Chemistry, Radiology ...)
- Bioinformatics vs. *computational biology* (more on algorithm correctness, complexity and other themes central to theoretical CS)

Grand Challenges in Genomics Research (I) Genomics to Biology

- Comprehensively identify the *structural and functional* components encoded in *human and other genomes*
 - Catalogue, characterize and comprehend the entire set of functional elements encoded in the human and other genomes
 - Compare genome sequences from evolutionary diverse species
 - Identify and analyze functional genomic elements
- Elucidate the organization of *genetic networks* and *protein pathways* and establish how they contribute to cellular and organismal phenotypes
- Develop a detailed understanding of the *heritable variation* in the human genome
- Understand *evolutionary variation across species* and the mechanisms underlying it

Grand Challenges in Genomics Research (II) Genomics to Health

- Develop robust strategies for identifying the *genetic contributions to disease and drug response*
- Develop strategies to identify *gene variants that contribute to good health and resistance to disease*
- Develop genome-based approach to *prediction of disease susceptibility and drug response*, early detection of illness, and *molecular taxonomy of disease* states
- Use new understanding of genes and pathways to develop powerful *new therapeutic approaches to disease*
- Develop *genome-based tools* that improve the health of all
- Understand the *relationships between genomics, race, and ethnicity*, and the consequences of uncovering these relationships

Data Mining & Bioinformatics : Why?

- Many biological processes are not well-understood
- Biological knowledge is **highly complex, imprecise, descriptive, and experimental**
- Biological data is abundant and information-rich
 - Genomics & proteomics data (sequences), microarray and protein-arrays, protein database (PDB), bio-testing data
 - Huge data banks, rich literature, openly accessible
 - Largest and richest scientific data sets in the world
- Mining: gain biological **insight (data/information → knowledge)**
 - Mining for correlations, linkages between disease and gene sequences, protein networks, classification, clustering, outliers, ...
 - Find correlations among linkages in literature and heterogeneous databases

11/26/2007

Data Mining: Principles and Algorithms

21

Data Mining & Bioinformatics: How (1)

- **Data Integration: Handling heterogeneous, distributed bio-data**
 - Build Web-based, interchangeable, integrated, multi-dimensional genome databases
 - Data cleaning and data integration methods becomes crucial
 - Mining correlated information across multiple databases itself becomes a data mining task
 - Typical studies: mining database structures, information extraction from data, reference reconciliation, document classification, clustering and correlation discovery algorithms, ...

11/26/2007

Data Mining: Principles and Algorithms

22

Data Mining & Bioinformatics: How (2)

- **Master and exploration of existing data mining tools**
 - Genomics, proteomics, and functional genomics (functional networks of genes and proteins)
- What are the current bioinformatics tools aiming for?
 - Inferring a protein's shape and function from a given sequence of amino acids
 - Finding all the genes and proteins in a given genome
 - Determining sites in the protein structure where drug molecules can be attached

11/26/2007

Data Mining: Principles and Algorithms

23

Data Mining & Bioinformatics – How (3)

- **Research and development of new tools for bioinformatics**
 - Similarity search and comparison between classes of genes (e.g., diseased and healthy) by finding and comparing **frequent patterns**
 - Identify **sequential patterns** that play roles in various diseases
 - New **clustering and classification** methods for micro-array data and protein-array data analysis
 - Mining, indexing and similarity search in sequential and structured (e.g., **graph and network**) data sets
 - Path analysis: linking genes/proteins to different disease development stages
 - Develop pharmaceutical interventions that target the different stages separately
 - High-dimensional analysis and OLAP mining
 - Visualization tools and genetic/proteomic data analysis

11/26/2007

Data Mining: Principles and Algorithms

24

Algorithms Used in Bioinformatics

- **Comparing sequences:** Comparing large numbers of long sequences, allow insertion/deletion/mutations of symbols
- **Constructing evolutionary (phylogenetic) trees:** Comparing seq. of diff. organisms, & build trees based on their degree of similarity (evolution)
- **Detecting patterns in sequences**
 - Search for genes in DNA or subcomponents of a seq. of amino acids
- **Determining 3D structures from sequences**
 - E.g., infer RNA shape from seq. & protein shape from amino acid seq.
- **Inferring cell regulation:**
 - Cell modeling from experimental (say, microarray) data
- **Determining protein function and metabolic pathways:** Interpret human annotations for protein function and develop graph db that can be queried
- **Assembling DNA fragments** (provided by sequencing machines)
- **Using script languages:** script on the Web to analyze data and applications

11/26/2007

Data Mining: Principles and Algorithms

25

Mining Sequence Patterns in Biological Data

- A brief introduction to biology and bioinformatics
- Alignment of biological sequences
- Hidden Markov model for biological sequence analysis
- Summary

11/26/2007

Data Mining: Principles and Algorithms

26

Comparing Sequences

- All living organisms are related to evolution
- **Alignment:** Lining up sequences to achieve the maximal level of identity
- Two sequences are *homologous* if they share a common ancestor
- Sequences to be compared: either nucleotides (DNA/RNA) or amino acids (proteins)
 - Nucleotides: identical
 - Amino acids: identical, or if one can be derived from the other by substitutions that are likely to occur in nature
- **Local vs. global alignments:** Local—only portions of the sequences are aligned. Global—align over the entire length of the sequences
 - Use gap “-” to indicate preferable not to align two symbols
- **Percent identity:** ratio between the number of columns containing identical symbols vs. the number of symbols in the longest sequence
- **Score** of alignment: summing up the matches and counting gaps as negative

11/26/2007

Data Mining: Principles and Algorithms

27

Sequence Alignment: Problem Definition

- **Goal:**
 - Given two or more input sequences
 - Identify similar sequences with long conserved subsequences
- **Method:**
 - Use substitution matrices (probabilities of substitutions of nucleotides or amino-acids and probabilities of insertions and deletions)
 - *Optimal* alignment problem: NP-hard
 - Heuristic method to find *good* alignments

11/26/2007

Data Mining: Principles and Algorithms

28

Pair-Wise Sequence Alignment

■ Example

HEAGAWGHEE
PAWHEAE

HEAGAWGHE-E
| | | |
P-A--W-HEAE

HEAGAWGHE-E
| | | |
--P-AW-HEAE

- Which one is better? → **Scoring alignments**
- To compare two sequence alignments, calculate a score
 - PAM (Percent Accepted Mutation) or BLOSUM (Blocks Substitution Matrix) (*substitution*) matrices: Calculate matches and mismatches, considering amino acid substitution
 - Gap penalty: Initiating a gap
 - Gap extension penalty: Extending a gap

Pair-wise Sequence Alignment: Scoring Matrix

	A	E	G	H	W
A	5	-1	0	-2	-3
E	-1	6	-3	0	-3
H	-2	0	-2	10	-3
P	-1	-1	-2	-2	-4
W	-3	-3	-3	-3	15

■ Gap penalty: -8

■ Gap extension: -8

HEAGAWGHE-E
| | | |
--P-AW-HEAE

$$(-8) + (-8) + (-1) + 5 + 15 + (-8) + 10 + 6 + (-8) + 6 = 9$$

HEAGAWGHE-E
| | | |
P-A--W-HEAE

Exercise: Calculate for

Formal Description

■ Problem: **PairSeqAlign**

■ Input: Two sequences x, y

Scoring matrix s

Gap penalty d

Gap extension penalty e

■ Output: The optimal sequence alignment

■ Difficulty:

If x, y are of size n then the number of possible global alignments is

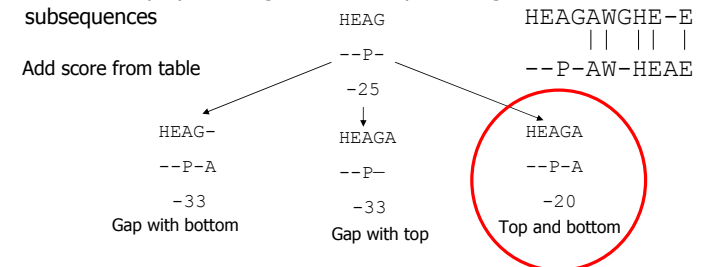
$$\rightarrow \binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

Global Alignment: Needleman-Wunsch

■ Needleman-Wunsch Algorithm (1970)

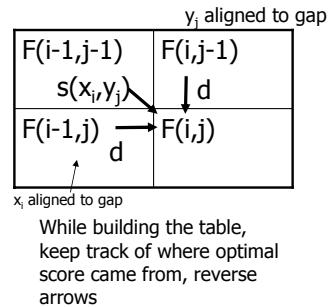
- Uses weights for the outmost edges that encourage the best overall (global) alignment
- An alternative algorithm: Smith-Waterman (favors the contiguity of segments being aligned)

■ Idea: Build up optimal alignment from optimal alignments of subsequences



Global Alignment

- Uses recursion to fill in intermediate results table
- Uses $O(nm)$ space and time
 - $O(n^2)$ algorithm
 - Feasible for moderate sized sequences, but not for aligning whole genomes.



11/26/2007

Data Mining: Principles and Algorithms

33

Pair-Wise Sequence Alignment

Given $s(x_i, y_j), d$
 $F(0, 0) = 0$

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Given $s(x_i, y_j), d$
 $F(0, 0) = 0$

$$F(i, j) = \max \begin{cases} 0 \\ F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Alignment: $F(0, 0) - F(n, m)$

Alignment: $0 - F(i, j)$

We can vary both the model and the alignment strategies

11/26/2007

Data Mining: Principles and Algorithms

34

Mining Sequence Patterns in Biological Data

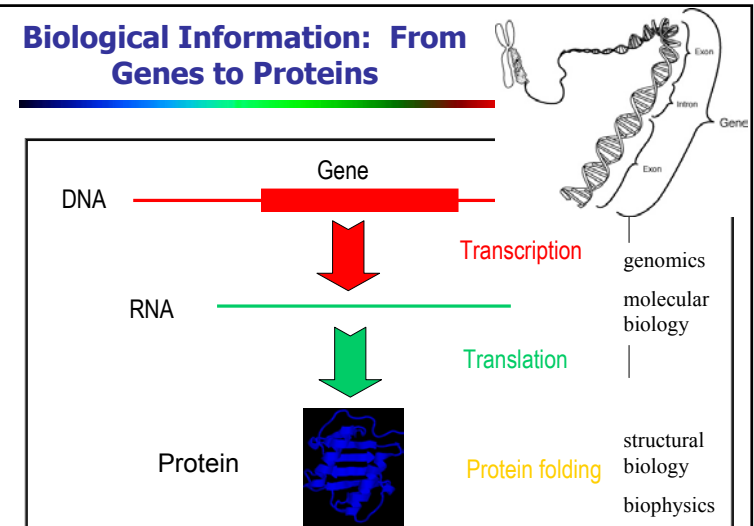
- A brief introduction to biology and bioinformatics
- Alignment of biological sequences
- Hidden Markov model for biological sequence analysis
- Summary

11/26/2007

Data Mining: Principles and Algorithms

35

Biological Information: From Genes to Proteins



11/26/2007

Data Mining: Principles and Algorithms

36

From Amino Acids to Proteins Functions

CGCCAGCTGGACGGGCACACC
ATGAGGCTGCTGACCCCTCTG
GGCCTTCTG...

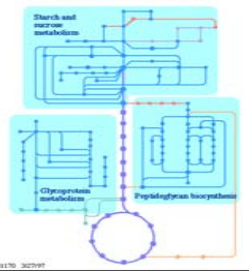
TDQAAFDTNIVTLTRFVMEQG
RKARGTGEMTQLNSLCTAVK
AISTAVRKAGIAHLYGIAGST
NVTGDQVKKLDVLSNDLVINV
LKSSFATCVLVTEDKNIAIV
EPEKRGKYVVCDFPLDGSSNI
DCLVSIPTIFGIYRKNSTDEP
SEKDALQPGRNLVAAGYALYG
SATML



DNA / amino acid
sequence

3D structure

METABOLISM OF COMPLEX CARBOHYDRATES



protein functions

DNA (gene) → → → pre-RNA → → → RNA → → → Protein

RNA-polymerase

Spliceosome

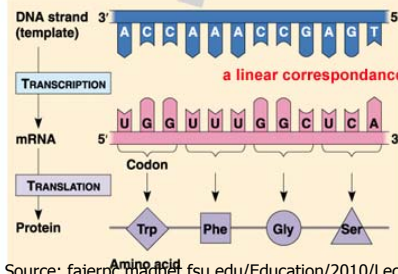
Ribosome

11/26/2007

Data Mining: Principles and Algorithms

37

Biology Fundamentals (6): Functional Genomics



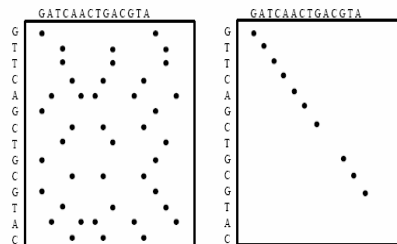
- The *function* of a protein is the way it participates with other proteins and molecules in keeping the cell alive and interacting with its environment
- Function is closely related to tertiary structure
- *Functional genomics*: studies the function of all the proteins of a genome

Source: fajerp.c.magnet.fsu.edu/Education/2010/Lectures/26_DNA_Transcription.htm

38

Dot Matrix Alignment Method

- Dot Matrix Plot: Boolean matrices representing possible alignments that can be detected visually
 - Extremely simple but
 - $O(n^2)$ in time and space
 - Visual inspection

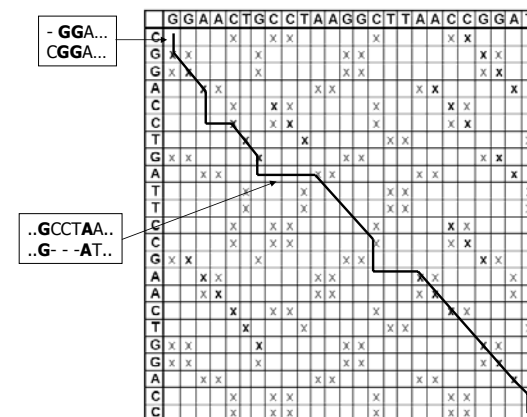


11/26/2007

Data Mining: Principles and Algorithms

39

TGCA Matrix Plot



11/26/2007

Data Mining: Principles and Algorithms

40

Heuristic Alignment Algorithms

- Motivation: Complexity of alignment algorithms: $O(nm)$
 - Current protein DB: 100 million base pairs
 - Matching each sequence with a 1,000 base pair query takes about 3 hours!
- Heuristic algorithms aim at speeding up at the price of possibly missing the best scoring alignment
- Two well known programs
 - BLAST: Basic Local Alignment Search Tool
 - FASTA: Fast Alignment Tool
 - Both find high scoring local alignments between a query sequence and a target database
 - Basic idea: first locate high-scoring short stretches and then extend them

11/26/2007

Data Mining: Principles and Algorithms

41

FASTA (Fast Alignment)

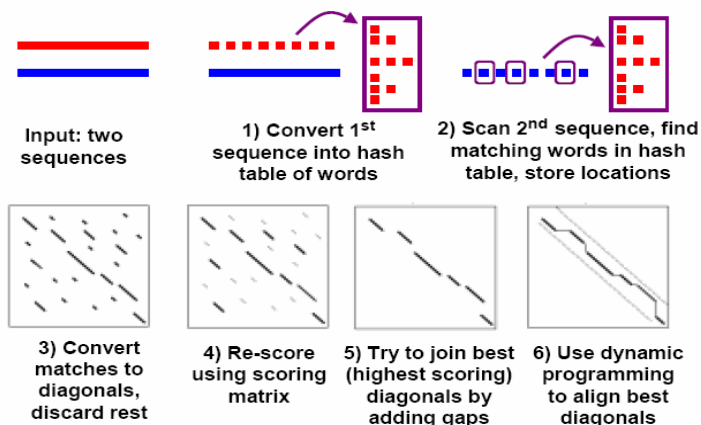
- Approach [Pearson & Lipman 1988]
 - Derived from the logic of the dot matrix method
 - View sequences as sequences of short words (k -tuple)
 - DNA: 6 bases, protein: 1 or 2 amino acids
 - Start from nearby sequences of exact matching words
- Motivation
 - Good alignments should contain many exact matches
 - Hashing can find exact matches in $O(n)$ time
 - Diagonals can be formed from exact matches quickly
 - Sort matches by position ($i - j$)
- Look only at matches near the longest diagonals
- Apply more precise alignment to small search space at the end

11/26/2007

Data Mining: Principles and Algorithms

42

FASTA (Fast Alignment)



11/26/2007

Data Mining: Principles and Algorithms

43

BLAST (Basic Local Alignment Search Tool)

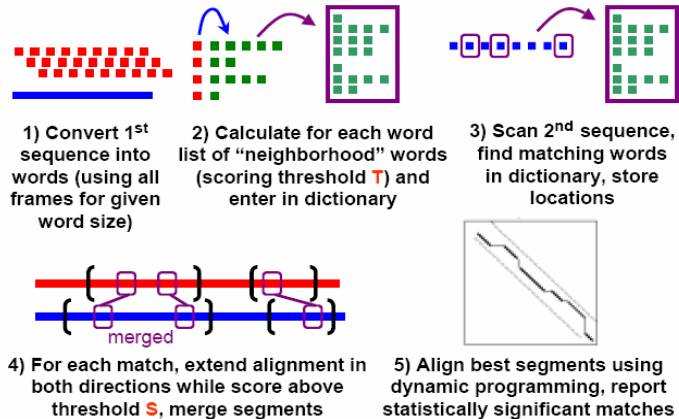
- Approach (BLAST) (Altschul et al. 1990, developed by NCBI)
 - View sequences as sequences of short words (k -tuple)
 - DNA: 11 bases, protein: 3 amino acids
 - Create hash table of neighborhood (closely-matching) words
 - Use statistics to set threshold for "closeness"
 - Start from exact matches to neighborhood words
- Motivation
 - Good alignments should contain many close matches
 - Statistics can determine which matches are significant
 - Much more sensitive than % identity
 - Hashing can find matches in $O(n)$ time
 - Extending matches in both directions finds alignment
 - Yields high-scoring/maximum segment pairs (HSP/MSP)

11/26/2007

Data Mining: Principles and Algorithms

44

BLAST (Basic Local Alignment Search Tool)

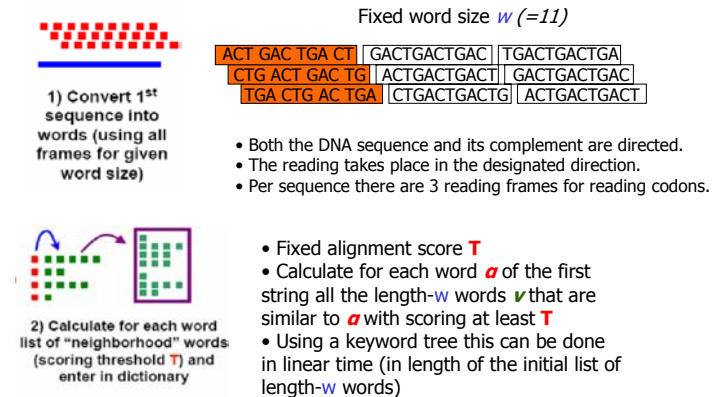


11/26/2007

Data Mining: Principles and Algorithms

45

BLAST

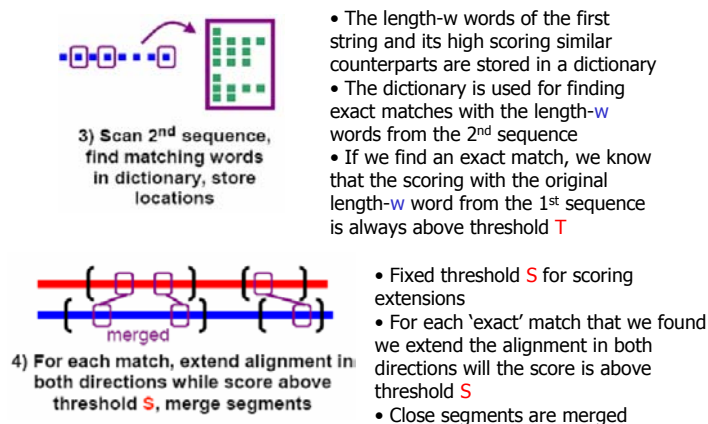


11/26/2007

Data Mining: Principles and Algorithms

46

BLAST

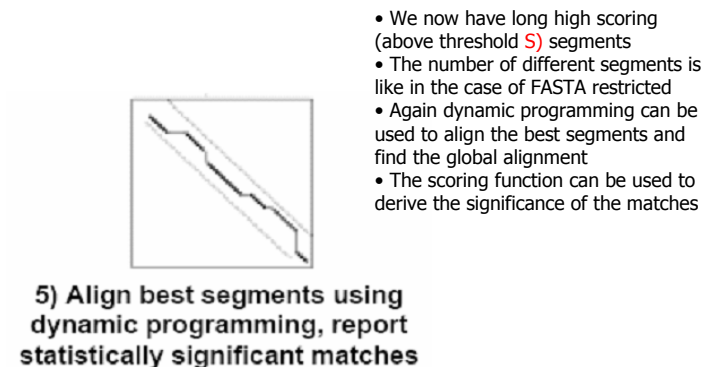


11/26/2007

Data Mining: Principles and Algorithms

47

BLAST

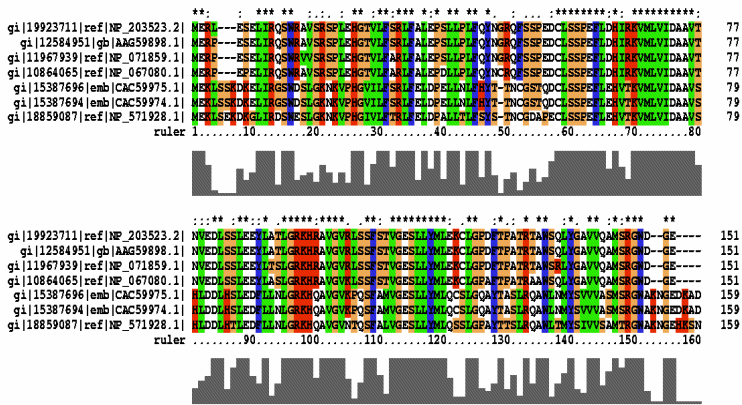


11/26/2007

Data Mining: Principles and Algorithms

48

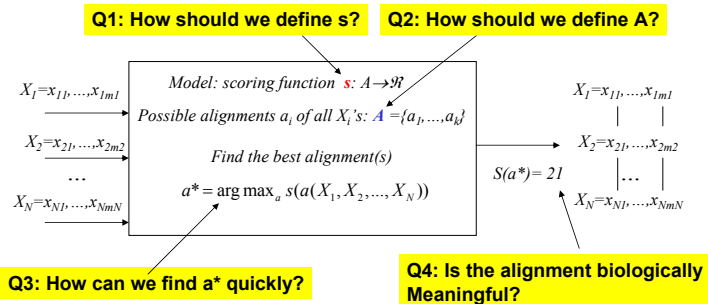
Multiple Sequence Alignment



Multiple Sequence Alignment: Why?

- Identify highly conserved residues
 - Likely to be essential sites for structure/function
 - More precision from multiple sequences
 - Better structure/function prediction, pairwise alignments
- Building gene/protein families
 - Use conserved regions to guide search
- Basis for phylogenetic analysis
 - Infer evolutionary relationships between genes
- Develop primers & probes
 - Use conserved region to develop
 - Primers for PCR
 - Probes for DNA micro-arrays

Multiple Alignment Model



Minimum Entropy Scoring

- Intuition:
 - A perfectly aligned column has one single symbol (least uncertainty)
 - A poorly aligned column has many distinct symbols (high uncertainty)

$$S(m_i) = - \sum_a p_{ia} \log p_{ia}$$

$$p_{ia} = \frac{c_{ia}}{\sum_{a'} c_{ia'}} \quad \text{Count of symbol } a \text{ in column } i$$

m_i column i of the alignment

Example:

12345678

G-T-A-

G-T-C-

G-T-C-

G-A-A--

$$p_{1G} = 1 \Rightarrow S(m_1) = 0$$

$$p_{3T} = 3/4, p_{3A} = 1/4 \Rightarrow S(m_3) = 0.81$$

$$p_{6A} = 2/4, p_{6C} = 2/4 \Rightarrow S(m_6) = 1$$

Multidimensional Dynamic Programming

Assumptions: (1) columns are independent (2) linear gap cost G

$$S(m) = G + \sum_i s(m_i)$$

$$G = \gamma(g) = dg$$

$\alpha_{i1, i2, \dots, iN}$ = Maximum score of an alignment up to the subsequences ending with $x_{i1}^1, x_{i2}^2, \dots, x_{iN}^N$

$$\alpha_{0, 0, \dots, 0} = 0$$

$$\alpha_{i1, i2, \dots, iN} = \max \begin{cases} \alpha_{i1-1, i2-1, \dots, iN-1} + S(x_{i1}^1, x_{i2}^2, \dots, x_{iN}^N) \\ \alpha_{i1, i2-1, \dots, iN-1} + S(-, x_{i2}^2, \dots, x_{iN}^N) \\ \alpha_{i1-1, i2, \dots, iN-1} + S(x_{i1}^1, -, \dots, x_{iN}^N) \\ \dots \\ \alpha_{i1, i2, \dots, iN-1} + S(-, -, \dots, x_{iN}^N) \\ \dots \\ \alpha_{i1-1, i2, \dots, iN} + S(x_{i1}^1, -, \dots, -) \end{cases}$$

Alignments: 0,0,0...,0 - - - |x¹|, ..., |x^N|

We can vary both the model and the alignment strategies

11/26/2007

Data Mining: Principles and Algorithms

53

Complexity of Dynamic Programming

- Complexity: Space: O(LN); Time: O(2NLN)
- One idea for improving the efficiency
 - Define the score as the sum of pair wise alignment scores

$$S(a) = \sum_{k < l} S(a^{kl})$$

Pair wise alignment between sequence k and l

- Derive a lower bound for $S(a^{kl})$, only consider a pair wise alignment scoring better than the bound

$$\sigma(a) \leq S(a^{kl}) - S(\hat{a}^{kl}) + \sum_{k' < l'} S(\hat{a}^{k'l'})$$

$$S(a^{kl}) \geq \beta^{kl}$$

$$\beta^{kl} = \sigma(a) + S(\hat{a}^{kl}) - \sum_{k' < l'} S(\hat{a}^{k'l'})$$

11/26/2007

Data Mining: Principles and Algorithms

54

Approximate Algorithms for Multiple Alignment

- Two major methods (but it remains a worthy research topic)
 - Reduce a multiple alignment to a series of pair wise alignments and then combine the result (e.g., Feng-Doolittle alignment)
 - Using HMMs (Hidden Markov Models)
- Feng-Doolittle alignment (4 steps)
 - Compute all possible pair wise alignments
 - Convert alignment scores to distances
 - Construct a "guide tree" by clustering
 - Progressive alignment based on the guide tree (bottom up)
- Practical aspects of alignments
 - Visual inspection is crucial
 - Variety of input/output formats: need translation

11/26/2007

Data Mining: Principles and Algorithms

55

More on Feng-Doolittle Alignment

- Problems of Feng-Doolittle alignment
 - All alignments are completely determined by pair wise alignment (restricted search space)
 - No backtracking (sub alignment is "frozen")
 - No way to correct an early mistake
 - Non-optimality: Mismatches and gaps at highly conserved region should be penalized more, but we can't tell where is a highly conserved region early in the process
- Iterative Refinement
 - Re-assigning a sequence to a different cluster/profile
 - Repeatedly do this for a fixed number of times or until the score converges
 - Essentially to enlarge the search space

11/26/2007

Data Mining: Principles and Algorithms

56

Clustal W: A Multiple Alignment Tool

- CLUSTAL and its variants are software packages often used to produce multiple alignments
- Essentially following Feng-Doolittle
 - Do pair wise alignment (dynamic programming)
 - Do score conversion/normalization (Kimura's model)
 - Construct a guide tree (neighbour-joining clustering)
 - Progressively align all sequences using profile alignment
- Offer capabilities of using substitution matrices like BLOSUM or PAM
- Many Heuristics

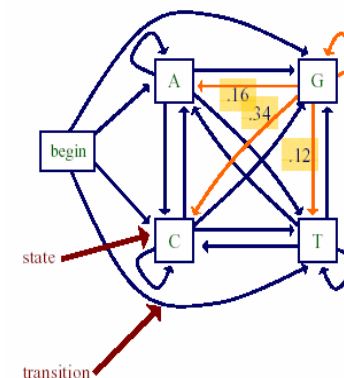
Mining Sequence Patterns in Biological Data

- A brief introduction to biology and bioinformatics
- Alignment of biological sequences
- Hidden Markov model for biological sequence analysis
- Summary

Motivation for Markov Models in Computational Biology

- There are many cases in which we would like to represent the statistical regularities of some class of sequences
 - genes
 - various regulatory sites in DNA (e.g., where RNA polymerase and transcription factors bind)
 - proteins in a given family
- Markov models are well suited to this type of task

A Markov Chain Model



Transition probabilities

- $\Pr(x_i=a|x_{i-1}=g)=0.16$
- $\Pr(x_i=c|x_{i-1}=g)=0.34$
- $\Pr(x_i=g|x_{i-1}=g)=0.38$
- $\Pr(x_i=t|x_{i-1}=g)=0.12$

$$\sum \Pr(x_i | x_{i-1} = g) = 1$$

Definition of Markov Chain Model

- A Markov chain model is defined by
 - a set of states
 - some states emit symbols
 - other states (e.g., the begin state) are silent
 - a set of transitions with associated probabilities
 - the transitions emanating from a given state define a distribution over the possible next states

Markov Chain Models: Properties

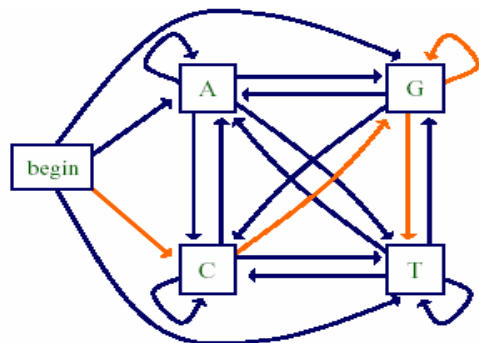
- Given some sequence x of length L , we can ask how probable the sequence is given our model
- For any probabilistic model of sequences, we can write this probability as

$$\begin{aligned}\Pr(x) &= \Pr(x_L, x_{L-1}, \dots, x_1) \\ &= \Pr(x_L | x_{L-1}, \dots, x_1) \Pr(x_{L-1} | x_{L-2}, \dots, x_1) \dots \Pr(x_1)\end{aligned}$$

- key property of a (1st order) Markov chain: the probability of each x_i depends only on the value of x_{i-1}

$$\begin{aligned}\Pr(x) &= \Pr(x_L | x_{L-1}) \Pr(x_{L-1} | x_{L-2}) \dots \Pr(x_2 | x_1) \Pr(x_1) \\ &= \Pr(x_1) \prod_{i=2}^L \Pr(x_i | x_{i-1})\end{aligned}$$

The Probability of a Sequence for a Markov Chain Model



$$\Pr(\text{cggt}) = \Pr(c) \Pr(g|c) \Pr(g|g) \Pr(t|g)$$

Example Application

- CpG islands
 - CG di-nucleotides are rarer in eukaryotic genomes than expected given the marginal probabilities of C and G
 - but the regions upstream of genes are richer in CG di-nucleotides than elsewhere – CpG islands
 - useful evidence for finding genes
- Application: Predict CpG islands with Markov chains
 - one to represent CpG islands
 - one to represent the rest of the genome

Markov Chains for Discrimination

- Suppose we want to distinguish CpG islands from other sequence regions
- Given sequences from CpG islands, and sequences from other regions, we can construct
 - a model to represent CpG islands
 - a null model to represent the other regions
- We can then score a test sequence by:

$$\text{score}(x) = \log \frac{\Pr(x | \text{CpGModel})}{\Pr(x | \text{nullModel})}$$

Markov Chains for Discrimination

- Why can we use

$$\text{score}(x) = \log \frac{\Pr(x | \text{CpGModel})}{\Pr(x | \text{nullModel})}$$

- According to Bayes' rule:

$$\Pr(\text{CpG} | x) = \frac{\Pr(x | \text{CpG}) \Pr(\text{CpG})}{\Pr(x)}$$

$$\Pr(\text{null} | x) = \frac{\Pr(x | \text{null}) \Pr(\text{null})}{\Pr(x)}$$

- If we are not taking into account prior probabilities ($\Pr(\text{CpG})$ and $\Pr(\text{null})$) of the two classes, then from Bayes' rule it is clear that we just need to compare $\Pr(x | \text{CpG})$ and $\Pr(x | \text{null})$ as is done in our scoring function $\text{score}()$.

Higher Order Markov Chains

- The Markov property specifies that the probability of a state depends **only** on the probability of the previous state
- But we can build more "memory" into our states by using a **higher order** Markov model
- In an **n-th** order Markov model

$$\Pr(x_i | x_{i-1}, x_{i-2}, \dots, x_1) = \Pr(x_i | x_{i-1}, \dots, x_{i-n})$$

The probability of the current state depends on the previous **n** states.

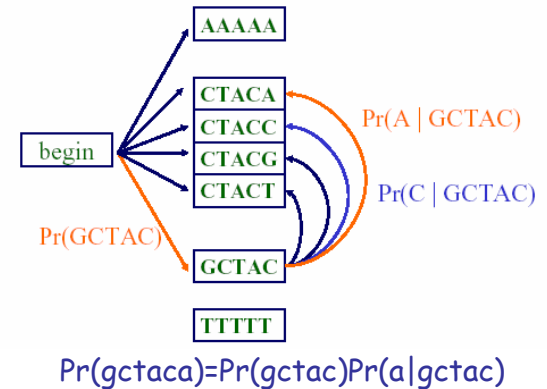
Selecting the Order of a Markov Chain Model

- But the number of parameters we need to estimate grows exponentially with the order
 - for modeling DNA we need $O(4^{n+1})$ parameters for an **n-th** order model
- The higher the order, the less reliable we can expect our parameter estimates to be
 - estimating the parameters of a **2nd** order Markov chain from the complete genome of E. Coli (5.44×10^6 bases), we'd see each word $\sim 85,000$ times on average (divide by 4^3)
 - estimating the parameters of a **9th** order chain, we'd see each word ~ 5 times on average (divide by $4^{10} \sim 10^6$)

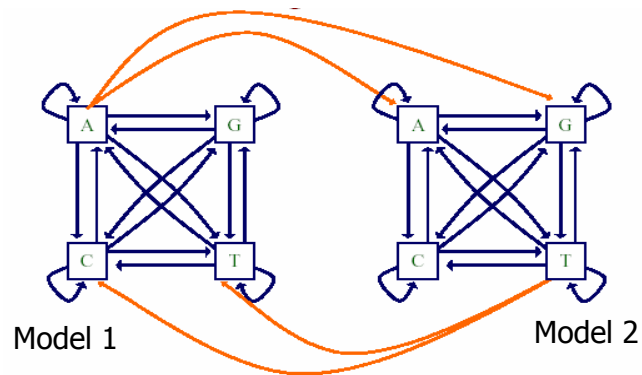
Higher Order Markov Chains

- An n-th order Markov chain over some alphabet A is equivalent to a first order Markov chain over the alphabet of n-tuples: A^n
- Example: A 2nd order Markov model for DNA can be treated as a 1st order Markov model over alphabet
 - AA, AC, AG, AT
 - CA, CC, CG, CT
 - GA, GC, GG, GT
 - TA, TC, TG, TT

A Fifth Order Markov Chain



Hidden Markov Model: A Simple HMM



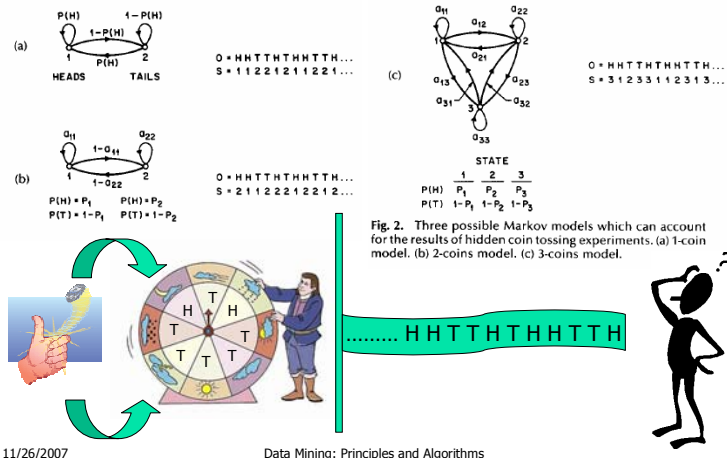
Given observed sequence AGGCT, which state emits every item?

Important Papers on HMM

L.R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceeding of the IEEE, Vol. 77, No. 22, February 1989.

Krogh, I. Saira Mian, D. Haussler, A Hidden Markov Model that finds genes in E. coli DNA, Nucleic Acids Research, Vol. 22 (1994), pp 4768-4778

HMM for Hidden Coin Tossing



11/26/2007

Data Mining: Principles and Algorithms

73

Hidden State

- We'll distinguish between the observed parts of a problem and the hidden parts
- In the Markov models we've considered previously, it is clear which state accounts for each part of the observed sequence
- In the model above, there are multiple states that could account for each part of the observed sequence
 - this is the hidden part of the problem

11/26/2007

Data Mining: Principles and Algorithms

74

Learning and Prediction Tasks

(in general, i.e., applies on both MM as HMM)

- Learning
 - **Given:** a model, a set of training sequences
 - **Do:** find model parameters that explain the training sequences with relatively high probability (goal is to find a model that *generalizes* well to sequences we haven't seen before)
- Classification
 - **Given:** a set of models representing different sequence classes, and given a test sequence
 - **Do:** determine which model/class best explains the sequence
- Segmentation
 - **Given:** a model representing different sequence classes, and given a test sequence
 - **Do:** segment the sequence into subsequences, predicting the class of each subsequence

11/26/2007

Data Mining: Principles and Algorithms

75

Algorithms for Learning & Prediction

- Learning
 - **correct path known for each training sequence** -> simple maximum likelihood or Bayesian estimation
 - **correct path not known** -> Forward-Backward algorithm + ML or Bayesian estimation
- Classification
 - **simple Markov model** -> calculate probability of sequence along single path for each model
 - **hidden Markov model** -> Forward algorithm to calculate probability of sequence along all paths for each model
- Segmentation
 - **hidden Markov model** -> Viterbi algorithm to find most probable path for sequence

11/26/2007

Data Mining: Principles and Algorithms

76

The Parameters of an HMM

- Transition Probabilities

$$a_{kl} = \Pr(\pi_i = l \mid \pi_{i-1} = k)$$

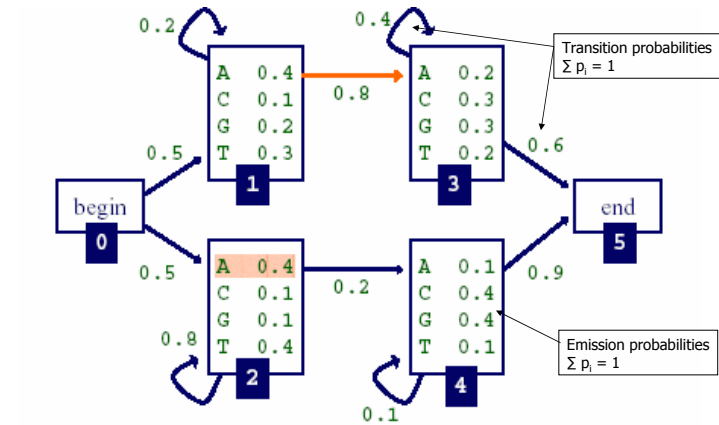
- Probability of transition from state k to state l

- Emission Probabilities

$$e_k(b) = \Pr(x_i = b \mid \pi_i = k)$$

- Probability of emitting character b in state k

An HMM Example



Three Important Questions

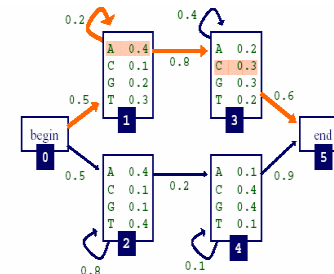
(See also L.R. Rabiner (1989))

- How likely is a given sequence?
 - The Forward algorithm
- What is the most probable "path" for generating a given sequence?
 - The Viterbi algorithm
- How can we learn the HMM parameters given a set of sequences?
 - The Forward-Backward (Baum-Welch) algorithm

How Likely is a Given Sequence?

- The probability that a given path is taken and the sequence is generated:

$$\Pr(x_1 \dots x_L, \pi_0 \dots \pi_N) = a_{0\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$



$$\begin{aligned} \Pr(AAC, \pi) &= a_{01} \times e_1(A) \times a_{13} \times e_3(A) \\ &\times a_{35} \times e_5(C) \times a_{55} \\ &= .5 \times .4 \times .2 \times .4 \times .8 \times .3 \times .6 \end{aligned}$$

How Likely is a Given Sequence?

- The probability **over all paths** is

$$\Pr(x_1 \dots x_L) = \sum_{\pi} \Pr(x_1 \dots x_L, \underbrace{\pi_0 \dots \pi_N}_{\pi})$$

- but the number of paths can be exponential in the length of the sequence...
- the Forward algorithm enables us to compute this efficiently

The Forward Algorithm

- Define $f_k(i)$ to be the probability of being in state k having observed the first i characters of sequence x
- To compute $f_N(L)$, the probability of being in the end state having observed all of sequence x
- Can be defined recursively
- Compute using dynamic programming

The Forward Algorithm

- $f_k(i)$ equal to the probability of being in state k having observed the first i characters of sequence x
- Initialization
 - $f_0(0) = 1$ for start state; $f_i(0) = 0$ for other state
- Recursion
 - For emitting state ($i = 1, \dots, L$)

$$f_i(i) = e_i(i) \sum_k f_k(i-1) a_{ki}$$

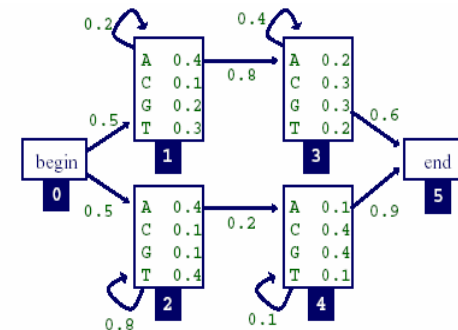
- For silent state

$$f_l(i) = \sum_k f_k(i) a_{kl}$$

- Termination

$$\Pr(x) = \Pr(x_1 \dots x_L) = f_N(L) = \sum_k f_k(L) a_{kN}$$

Forward Algorithm Example



Given the sequence $x = \text{TAGA}$

Forward Algorithm Example

- Initialization
 - $f_0(0)=1, f_1(0)=0 \dots f_5(0)=0$
- Computing other values
 - $f_1(1)=e_1(T)*(f_0(0)a_{01}+f_1(0)a_{11})$
 $=0.3*(1*0.5+0*0.2)=0.15$
 - $f_2(1)=0.4*(1*0.5+0*0.8)$
 - $f_1(2)=e_1(A)*(f_0(1)a_{01}+f_1(1)a_{11})$
 $=0.4*(0*0.5+0.15*0.2)$
 - ...
 - $\Pr(\text{TAGA})= f_5(4)=f_3(4)a_{35}+f_4(4)a_{45}$

Three Important Questions

- How likely is a given sequence?
- What is the most probable "path" for generating a given sequence?
- How can we learn the HMM parameters given a set of sequences?

Finding the Most Probable Path: The Viterbi Algorithm

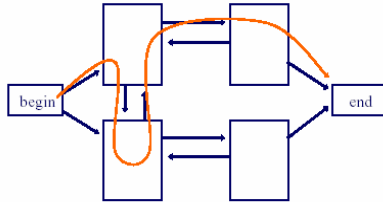
- Define $v_k(i)$ to be the probability of the most probable path accounting for the first i characters of x and ending in state k
- We want to compute $v_M(L)$, the probability of the most probable path accounting for all of the sequence and ending in the end state
- Can be defined recursively
- Again we can use Dynamic Programming to compute $v_M(L)$ and find the most probable path efficiently

Three Important Questions

- How likely is a given sequence?
- What is the most probable "path" for generating a given sequence?
- How can we learn the HMM parameters given a set of sequences?

Learning Without Hidden State

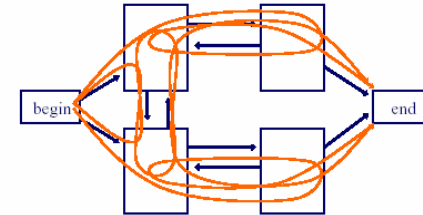
- Learning is simple if we know the correct path for each sequence in our training set



- estimate parameters by counting the number of times each parameter is used across the training set

Learning With Hidden State

- If we don't know the correct path for each sequence in our training set, consider all possible paths for the sequence



- Estimate parameters through a procedure that counts the expected number of times each parameter is used across the training set

Learning Parameters: The Baum-Welch Algorithm

- Also known as the Forward-Backward algorithm
- An Expectation Maximization (EM) algorithm
 - EM is a family of algorithms for learning probabilistic models in problems that involve hidden states
- In this context, the hidden state is the path that best explains each training sequence

Learning Parameters: The Baum-Welch Algorithm

- Algorithm sketch:
 - initialize parameters of model
 - iterate until convergence
 - calculate the *expected* number of times each transition or emission is used
 - adjust the parameters to *maximize* the likelihood of these expected values

Computational Complexity of HMM Algorithms

- Given an HMM with S states and a sequence of length L , the complexity of the Forward, Backward and Viterbi algorithms is

$$O(S^2 L)$$

- This assumes that the states are densely interconnected
- Given M sequences of length L , the complexity of Baum Welch on each iteration is

$$O(MS^2 L)$$

A Hidden Markov Model that finds genes in E. coli DNA.

Krogh, I. Saira Mian, D. Haussler
Nucleic Acids Research, Vol. 22 (1994), pp 4768-4778

Search for

- **Protein coding genes**
 - codons and frequencies
- **Intergenic regions** (basically the rest)
 - Repetitive extragenic palindromic sequences
 - Shine-Delgarno motif (1974)
- **'Noise'** (identified with high probability)
 - Potential sequence errors
 - Frame shifts
 - Insertion and deletions of nucleotides within a codon (very unlikely, but possible)
- **Results**
 - 80% of the known genes found
 - 10% approximate locations and potentially new genes

Hidden Markov Models

HMM's applied to (already in 1994)

- **DNA analysis** (Churchill, 1989)
- **Protein binding site modeling** (Lawrence et al., 1990; Cardon et al., 1992)
- **Protein analysis** (1993)

Applied on

- Directed strand
- Complementary strand
- Protein sequences
- ...

Finding Genes =>

Finding Genes

Two techniques:

1. Locate promotor sequences and splice junctions (NN, statistical methods)
2. Window scoring functions
 - a coding window vs a non-coding window
 - Deviation from the 'average codon'
 - Codon usage scoring (NN, Markov Model)

For example: Based on training data sets one Markov Model is determined for coding windows. A second Markov Model is determined for non coding windows.

Finding Genes

- Both techniques produce probabilistic results
- The results from both techniques have to be analyzed and assembled to produce a coherent 'parse' into genes separated by intergenic regions.
- For this ad hoc/specialized Dynamic Programming techniques are used
- The HMM framework gives a uniform and transparent approach

HMM Organization

- A general looping structure
- Submodels for
 - Each of the 64 codons (with the possibility for very low likelihood single nucleotide insertions/deletions)
 - Gene overlap
 - Frame shift and other programmed recording events (i.e., alternative readings of the genetic code)
 - Intergenic features
 - Repetitive extragenic palindromic sequences (REP's)
 - Shine Delgarno motif
 - Note: these models emerged automatically as a result of the training of more generic HMM's

Codon Frequencies

Codon	Aa	Usage	Random	Codon	Aa	Usage	Random	Codon	Aa	Usage	Random	Codon	Aa	Usage	Random
AAA	Lys	3.5	1.3	GAA	Glu	4.3	1.6	CAA	Gln	1.3	1.4	TAA	*	*	*
AAG	Lys	1.1	1.6	GAG	Glu	1.8	1.8	CAC	Gln	3.0	1.7	TAG	*	*	*
AAC	Asn	2.4	1.4	GAC	Asp	2.2	1.7	CAC	His	1.1	1.5	TAC	Tyr	1.4	1.4
AAT	Asn	1.4	1.3	GAT	Asp	3.2	1.5	CAT	His	1.2	1.4	TAT	Tyr	1.5	1.3
AGA	Arg	0.1	1.6	GGA	Gly	0.6	1.8	CGA	Arg	0.3	1.7	TGA	*	*	*
AGG	Arg	0.1	1.8	GGG	Gly	1.0	2.2	CGG	Arg	0.4	2.0	TGG	Trp	1.4	1.8
AGC	Ser	1.6	1.7	GGC	Gly	3.2	2.0	CGC	Arg	2.4	1.8	TGC	Cys	0.7	1.6
AGT	Ser	0.7	1.5	GGT	Gly	2.8	1.8	CGT	Arg	2.5	1.6	TGT	Cys	0.5	1.5
ACA	Thr	0.5	1.4	GCA	Ala	2.0	1.7	CCA	Pro	0.8	1.3	TCA	Ser	0.6	1.4
ACG	Thr	1.4	1.7	GCG	Ala	3.6	2.0	CCG	Pro	2.6	1.8	TCG	Ser	0.8	1.6
ACC	Thr	2.5	1.5	GCC	Ala	2.5	1.8	CCC	Pro	0.4	1.6	TCC	Ser	0.9	1.5
ACT	Thr	0.9	1.4	GCT	Ala	1.6	1.6	CCT	Pro	0.6	1.5	TCT	Ser	0.9	1.4
ATA	Ile	0.3	1.3	GTA	Val	1.1	1.5	CTA	Leu	0.3	1.4	TTA	Leu	1.1	1.3
ATG	Met	2.5	1.5	GTG	Val	2.7	1.8	CTG	Leu	5.7	1.6	TTG	Leu	1.2	1.5
ATC	Ile	2.7	1.4	GTC	Val	1.5	1.6	CTC	Leu	1.0	1.5	TTT	Phe	1.8	1.4
ATT	Ile	2.8	1.3	GTT	Val	1.9	1.5	CTT	Leu	0.9	1.4	TTT	Phe	1.9	1.2

Table 1: The relative frequencies of the 64 codons (in percent) in the *E. coli* DNA training data used in this study ("Usage"). "Random" gives the corresponding values if codon usage was simply a result of the relative frequencies of the four nucleotides (A, 23.66, G, 27.89, C, 25.30, and T, 23.15). "Aa" and "*" denote amino acid and stop codon respectively.

Open Reading Frame Probability

codons. The probability of an open reading frame (ORF) consisting of codons c_1, c_2, \dots, c_k and excluding start and stop codons is

$$\text{Prob}(c_1, \dots, c_k) = \prod_{i=1}^k p(c_i), \quad (1)$$

where $p(c_i)$ is the probability of codon c_i given in Table 1 for *E. coli*. We define the gene index of an ORF to be the negative logarithm of this divided by the length of the contig,

$$I(c_1, \dots, c_k) = -\frac{1}{k} \sum_{i=1}^k \log_{64} p(c_i). \quad (2)$$

The average value for a typical *E. coli* gene is equal to the entropy of the *E. coli* codon probability distribution.³ Using an estimate of this distribution obtained from our training set (Table 1) yields

$$\text{average}(I) = 0.935. \quad (3)$$

For genes in the training set, relatively few have a large gene index: roughly 16% have an index greater than 0.96, 7% greater than 0.98, and only about 2.5% have a gene index

³Since logarithm base 64 is used, the entropy of any codon distribution will be at most 1. Therefore, typical genes will have an index less than 1.

Gene Index and ORF's

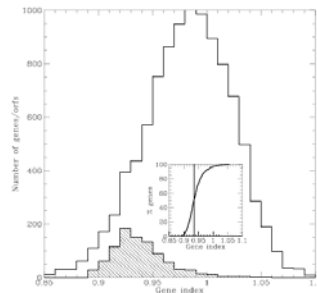


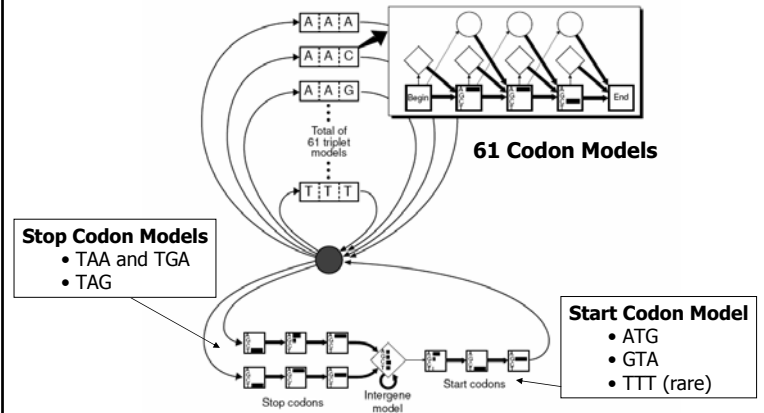
Figure 2: Distribution of gene index for 920 genes in the training set (lower dark histogram). Any genes with a length not divisible by 3 or with unusual start codons (not ATG, GTG and TTG) or stop codons (not TAA, TAG, and TGA) are not counted. The inset shows the cumulative distribution, i.e. the fraction of genes with a gene index below a certain value; the vertical line denotes the average gene index. For comparison the larger histogram shows the gene index for orfs (open reading frames) in the training data. The following criteria were used for selecting orfs: 1) they do not have the same stop codon as a labeled gene, 2) the length is more than 100 base pairs, 3) if several orfs had the same stop codon, only the one with the lowest gene index was included.

11/26/2007

Data Mining: Principles and Algorithms

101

HMM Model



11/26/2007

Data Mining: Principles and Algorithms

102

Intergenic Models

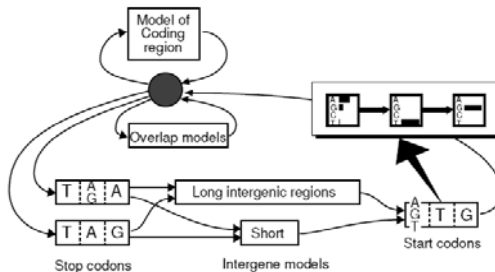


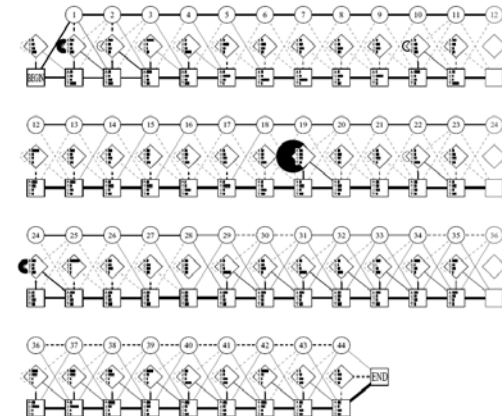
Figure 3: HMM architecture for a parser for *E. coli* DNA with a complex intergenic model. The gene model above the central state that contains the 61 triplet models is identical to the gene model of the simple parser shown in Figure 1. The detailed structure of the long intergenic model is shown in Figure 4.

11/26/2007

Data Mining: Principles and Algorithms

103

Figure 4: The model for long intergenic regions shown in Figure 3. This model was trained by the forward-backward algorithm on 424 intergenic regions of lengths larger than 10.



11/26/2007

Data Mining: Principles and Algorithms

104

Statistics on Data Set EcoSeq6

	Training set	Test set
Total number of contigs	300	129
Total number of characters	1,271,528	324,684
Number of genes	1007	251
Average length (internal genes)	1008	1015
Overlapping genes, length 1	50	7
Overlapping genes, length 4	40	12
Overlapping genes, length > 4	34	1

Table 2: Statistics on the 429 contigs of *E. coli* DNA used in our experiments.

HMM Results

Data Set:

- EcoSeq6 contained about 1/3th of the complete *E. coli* genome (total 5.44×10^6 nucleotides, 5416 genes), and was not fully annotated at that time

HMM Training:

- on $\sim 10^6$ nucleotides from the EcoSeq6 database of labeled genes (K. Rudd, 1991)

HMM Testing

- On the remainder of $\sim 325,000$ nucleotides

Method:

- For each contig in the test the Viterbi algorithm was used to find the most likely path through the hidden states of the HMM
- This path was then used to define a parse of the contig into genes separated by intergenic regions

HMM Results

Type of intergenic model	Post-processing	Data set	EcoSeq6 genes found by parser				Possible false positive
			Perfect	Almost perfect	Partly	Not found	
Complex	None	Training	731 (74.7)	57 (5.8)	141 (14.4)	50 (5.1)	665
		Test	203 (86.0)	12 (5.1)	11 (4.7)	10 (4.2)	191
	After	Training	767 (78.7)	62 (6.4)	88 (9.0)	57 (5.9)	310
		Test	201 (85.2)	13 (5.5)	8 (3.4)	14 (5.9)	82
Simple	None	Training	692 (70.8)	81 (8.3)	163 (16.7)	42 (4.3)	1524
		Test	179 (75.8)	23 (9.7)	25 (10.6)	9 (3.8)	412
	After	Training	694 (71.3)	81 (8.3)	143 (14.7)	55 (5.7)	331
		Test	174 (72.5)	22 (9.3)	23 (9.7)	17 (7.2)	98


HMM Results

- 80% of the labeled protein coding genes were exactly found (i.e. with precisely the same start and end codon)
- 5% found within 10 codons from start codon
- 5% overlap by at least 60 bases or 50%
- 5% missed completely
- Several new genes indicated
- Several insertion and deletion errors were labeled in the contig parse

Markov Models Summary

- We considered models that vary in terms of order, hidden state
- Three DP-based algorithms for HMMs: Forward, Backward and Viterbi
- We discussed three key tasks: learning, classification and segmentation
- The algorithms used for each task depend on whether there is hidden state (correct path known) in the problem or not

Mining Sequence Patterns in Biological Data

- A brief introduction to biology and bioinformatics
- Alignment of biological sequences
- Hidden Markov model for biological sequence analysis
- Summary 

Summary: Mining Biological Data

- Biological sequence analysis compares, aligns, indexes, and analyzes biological sequences (sequence of nucleotides or amino acids)
- Biosequence analysis can be partitioned into two essential tasks:
 - pair-wise sequence alignment and multiple sequence alignment
- Dynamic programming approach (notably, BLAST) has been popularly used for sequence alignments
- Markov chains and hidden Markov models are probabilistic models in which the probability of a state depends only on that of the previous state
 - Given a sequence of symbols, x , the **forward** algorithm finds the probability of obtaining x in the model
 - The **Viterbi** algorithm finds the most probable path (corresponding to x) through the model
 - The **Baum-Welch** learns or adjusts the model parameters (transition and emission probabilities) to best explain a set of training sequences.

References

- Lecture notes@M. Craven's website: www.biostat.wisc.edu/~craven
- A. Baxeavanis and B. F. F. Ouellette. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins (3rd ed.). John Wiley & Sons, 2004
- R.Durbin, S.Eddy, A.Krogh and G.Mitchison. Biological Sequence Analysis: Probability Models of Proteins and Nucleic Acids. Cambridge University Press, 1998
- N. C. Jones and P. A. Pevzner. An Introduction to Bioinformatics Algorithms. MIT Press, 2004
- I. Korf, M. Yandell, and J. Bedell. BLAST. O'Reilly, 2003
- L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE, 77:257–286, 1989
- J. C. Setubal and J. Meidanis. Introduction to Computational Molecular Biology. PWS Pub Co., 1997.
- M. S. Waterman. Introduction to Computational Biology: Maps, Sequences, and Genomes. CRC Press, 1995
- Krogh, I. Saira Mian, D. Haussler, A Hidden Markov Model that finds genes in E. coli DNA, Nucleid Acids Research, Vol. 22, pp 4768-4778, 1994