BIOINFORMATICS

What you should know

2016. DECEMBER 11.

PPKE ITK
Kajtsa Dóra, Halász Olivér, Kolozsvári Bernadett, Grizner Gyula

Órai diasorok és egyéb jegyzetek felhasználásával. A jegyzetben még nincs minden ellenőrizve, és egyéb javításokra is szorul.

Tartalomjegyzék

1. General intro, core data types, multiple data types

Definition of bioinformatics (narrow sense, broad sense).

Narrow definition:

Mostly molecular biology data. Storage (management), analysis and interpretation (visualization) of data. Mostly static.

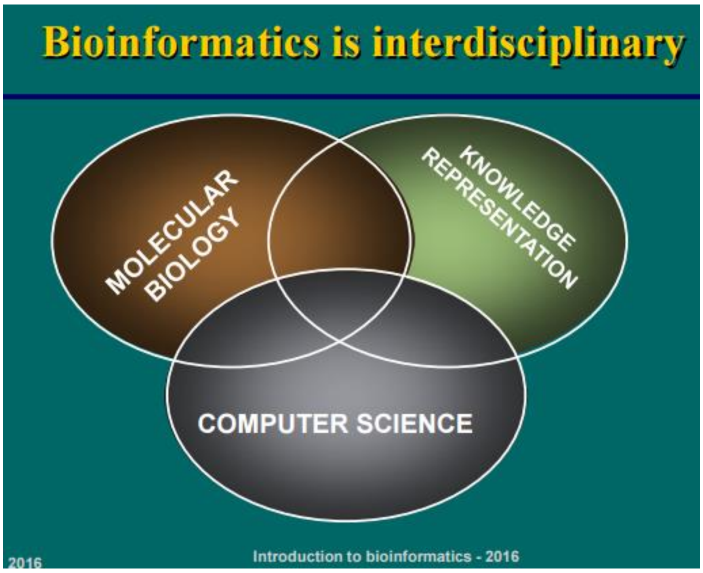
Includes: Data management (Acquisition, Storage, annotation – *service*; Interpretation, analysis, data-mining – *Research, Biocomputing)*

Broad definition:

Science of biological knowledge. All computer application in (molecular) biology including modeling (simulation of behavior). Also includes dynamics.

Includes: Data management (Acquisition, Storage, annotation – *service*; Interpretation, analysis, data-mining – *Research, Biocomputing)*; Modelling, simulation – *research, Biocomputing*

Example for the uses of bioinformatics: comparing sequences with databases; predicting functions of genes, based on comparison (or genomes, proteins' 3D structures etc).


Bioinformatics is interdisciplinary
MOLECULAR BIOLOGY
KNOWLEDGE REPRESENTATION
COMPUTER SCIENCE
2016
Introduction to bioinformatics - 2016

What is particular in bioinformatics?

The objects: molecular structures, metabolic pathways, regulatory networks AND their databases.

The methods: analysis and use of similarity;

Complexity of biological knowledge (and NOT so much the quantity of data...)

Concepts of system, structure, function. Structure is an ensemble of elements and relations.

Structure:

is a set of elements connected with relationships, named according to conventions. (constant space-time arrangement of elements or properties.)

Function:

is a role played within a system.

Molecules have structure and function: Structure and function are concepts of systems theory.

Structural data in brief: Structure definitions are hierarchical (atom – amino acid – protein – pathway – cell – tissue etc.). For a given problem it is convenient to choose a standard description or "core structural level". E.g. DNA sequences are the standard level for molecular biology problems. For a standard or core description, we always have an underlying logical structure, plus various additional, simplified and annotated views. (annotation means extending with external information).

Systems:

Any part of reality that can be separated from the environment (by a boundary). A community in an environment. Consist of interacting parts. Interact with the environment (inputs, outputs). System models are generalizations of reality. Have a structure that is defined by parts and processes. Parts have functional as well as structural relationships between each other.

Systems biology deals with parallel characterization (or modeling) of many objects (genes, molecules, cells), hoping to understand large, complex systems. Systems approach to bioinformatics and modeling.

Transition from simple objects to large systems:

Modern experimental approaches can collect data from a large number of objects at the same time → "systems biology"

Molecular biology/traditional bioinformatics studies single or a few objects.

Biological systems: a cell, a tissue, an organ, an organism…

Typical examples: genomes

Single entities (Molecular biology):

Bioinformatics started as computational support to molecular biology, i.e. the molecular studies of simple systems (1-2 genes, 1-2 proteins, etc).

Example: Predicting gene function via database searching.

Modeling:

Modeling the movement of single molecules in vacuo or in water (molecular modeling, molecular dynamics).

Docking (e,g, pharmacons to their receptor proteins).

Large systems (system biology):

As new measuring methods allow the parallel study of many genes and proteins, systems biology emerged as a new field (measuring technique + specific computational approaches).

Example: Studying gene expression in a whole genome using next generation sequencing.

Modeling:

Modeling large molecular assemblies.

Modeling biological communities (bacteria, animals, human crowds)

Traditional (or standard) bioinformatics deals with 4 core data-types: sequence, 3D, network and text. Each has an underlying logical structure, a standard or core description, plus various simplified and/or extended (annotated) descriptions.

Sequences:

Standard description: Series of characters (denoting amino acids or nucleotides). Sequences can have simplified and/or extended visualization.

Sequences are: language, codes…

Sequence formtas: FASTA, Concatenated FASTA

3D structures:

Standard description: 3D coordinates + subunit descriptions (connectivities);

(atomic, amino acid, nucleotide);

Simplified and/or extended (annotated) visualization

Networks (and genomes):

A genome is more than a sequence: We want to add regulatory links (what regulates what). We want to add functional links (e.g. substrates passed between enzymes in a pathway). All these are links that define a network of genes, proteins substrates etc.

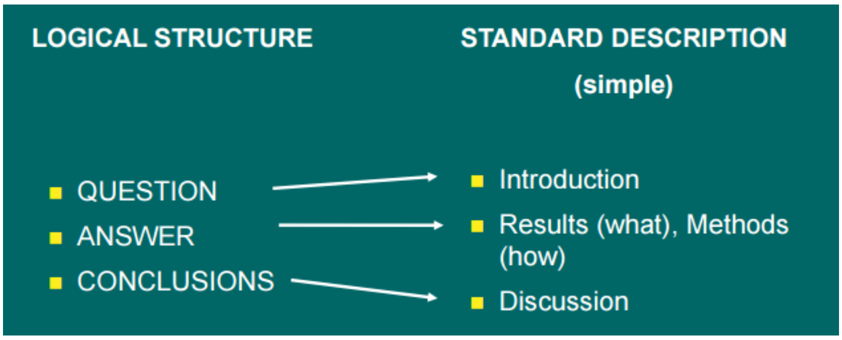
Networks:

Sandard description: Graphs of entities (nodes) and relationships (edges);

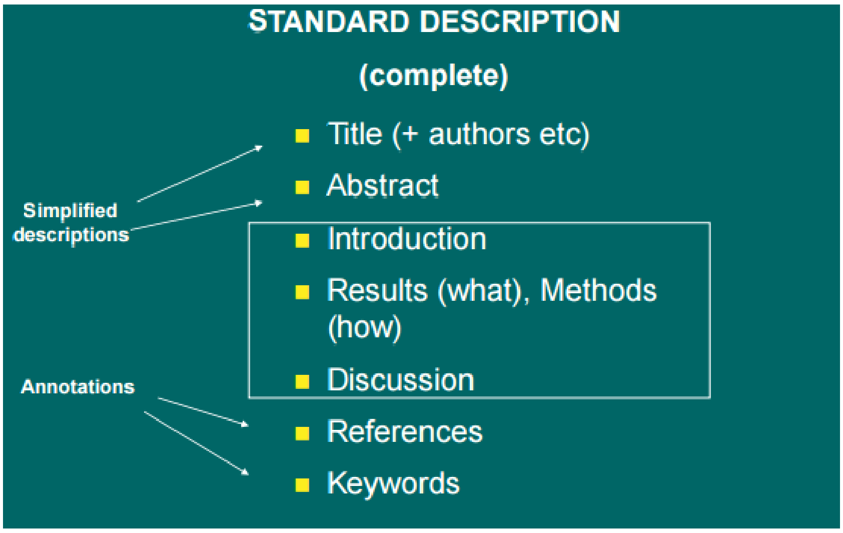
Simplified and/or extended (annotated) visualization.

Texts: Scientific publications

A human message written in scientific language ("special English", ~fixed vocabulary). Like other data, they have logical structure, standard, simplified and extended descriptions and databases BUT: messages have an emitter (author) and an audience (reader, reviewer). In other words, they are context dependent (unlike, say, sequences or atoms). Loosely structured (not as well as molecules). There are ontologies for the language but not for the articles themselves!



LOGICAL STRUCTURE	STANDARD DESCRIPTION (simple)
■ QUESTION	■ Introduction
■ ANSWER	■ Results (what), Methods (how)
■ CONCLUSIONS	■ Discussion



STANDARD DESCRIPTION

(complete)

Simplified descriptions

- ■ Title (+ authors etc)
- ■ Abstract
- ■ Introduction
- ■ Results (what), Methods (how)
- ■ Discussion

Annotations

- ■ References
- ■ Keywords

Example: PubMed

Annotation is adding (textual, sometimes numerical) descriptors to structures or their parts.

Data (e.g. sequence)

Data on data (annotation, meta-data)

Data on annotations (ontologies, meta-meta-data: defining the language of annotations)

Anything added to the “standard description” is annotation

Building a database from raw data + annotations: Put raw data into database records. Add basic annotations (project name, date etc.). Add annotations by similarity. This is called database searching (gives results as: 95% similarity to trypsin → probably trypsin. But only probably!!). Add further information based on human knowledge (analysis programs, literature search).

Annotation and the World Wide Web: Traditionally, annotations to a structure are validated and added by humans: authors trying to suggest a function for a new gene, database developers trying to add structural or functional descriptions to molecular data, etc. WWW is the biggest annotation system: millions of nonvalidated links are added to data. Important types include databases (bioinformatics and bibliographic), Wikipedia (community based encyclopedia), specialist wikis, blogs, discussion lists. Google search is a first step… Today, database annotation means generating standard language descriptions for data, validated via Internet links and specialized programs. Relies on human intervention.

Database records of a molecule (e.g. protein) have a "structural part" that contains the core-description (say sequence), and an "annotation part" that is mostly human readable (e.g. bibliography) but may include references to other structural descriptions (secondary structure, domain architecture, computed quantities etc.)

Annotation (added human knowledge) is crucial, better if machine readable.

2. Core operations I (Comparison)

Understanding data: grouping and classifying, organizing into knowledge items, matching to other knowledge items. Humans operate on “logical structures”. Computers operate on descriptions (which are given to them by humans).

Database records contain data, metadata (annotations, data on data). Rules of data representation and metadata descriptions are in ontologies (definition of concepts = meta-metadata, data on metadata).

Representations (unstructured, structured, mixed).

Sequence descriptions:
ACAACTGG (the sequence itself, structured)
A3C2G2T (composition, unstructured)
(AC)2 (CA)(AA)(CT)(TG)(GG) (word composition, “hybrid”)

Words are structured in themselves, so word composition is partly structured (because the relation between words is not included). Words are “substructures” so this is substructure composition.

Unstructured representations:
We know nothing about internal structure. Only the properties are known (global descriptors), can be discreet or continuous. Best described as vectors (each dimension is an attribute, the contents are the value). Sometimes a large number of dimensions. Vector operations are fast.

Vector types
Binary vectors consist of 0 or 1 values, e.g. 0,1,0,0,1,0. Indicate the presence or absence of attributes.
Non-binary vectors can contain real or integer-valued components, e.g., 0.5, 0.9, 1.0.

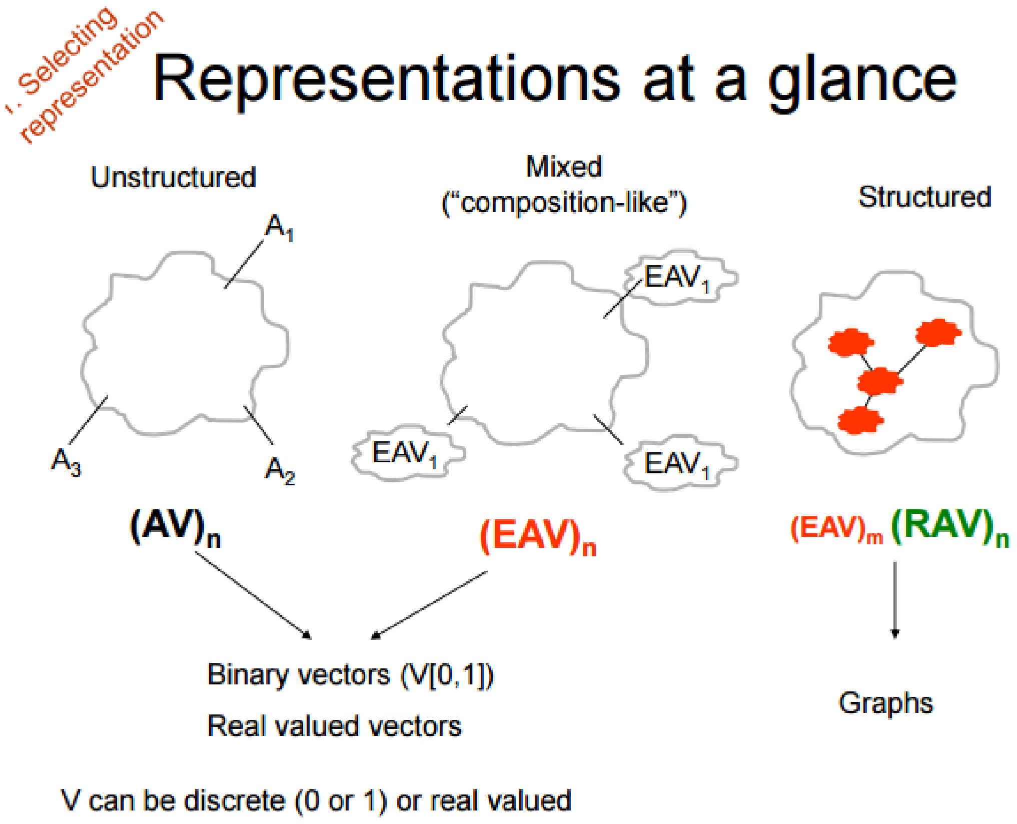
Structured representations:
We know the internal structure in terms or Entities and Relationships (both described in terms of attributes and values → EAV and “RAV”). Information-rich, allows detailed comparisons.

Need alignment (matching) for comparison.

Examples: character strings (sequences), graphs (most molecular structures are like this).

Mixed or composition-like descriptions:

We decompose an object to parts of known structure, and count the parts (atomic composition, H2O, or amino acid composition of proteins). The result is a vector, fast operations, alignment (matching) is not necessary. The information content of the vector depends on the granularity of the parts. Atomic composition of proteins or of people is not informative.



Representations at a glance

V can be discrete (0 or 1) or real valued

Comparison: 1) Proximity measures (similarities, distances) 2) Motifs (from pairwise and multiple alignment of sequences)

Input: Two descriptions

Output:

- For unstructured: a score (similarity, distance), is mandatory

- For structured: a score (like above, mandatory) AND a common pattern (result of matching=alignment), optional

Proximity measures (scores):

Similarity measures (zero for different objects, large for identical objects)

Distances (large different objects, zero for identical objects)

Exist both for vectors and for structures

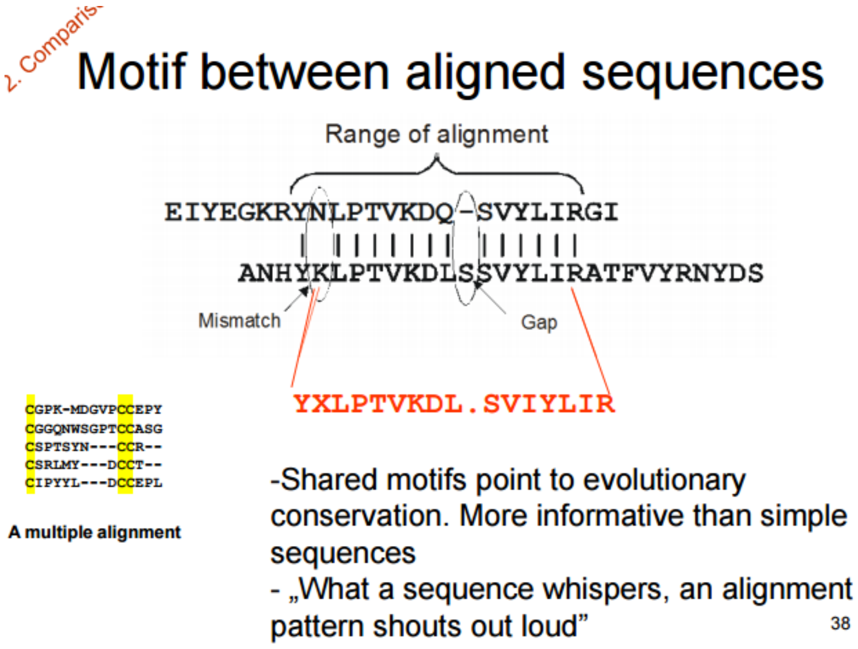
“Well behaved”: if bounded, e.g. [0,1]

Don’t expect linearity in any sense (“twice as similar” makes no sense)

Similarity S~1/D or 1-k*D

Patterns, motifs:

simplified logical structures associated from parts.



Motif between aligned sequences

```
CGPK-MDGVPCCEPY
CGGQNWSGPTCCASG
CSPTSYN---CCR--
CSRLMY---DCCT--
CIPYYL---DCCEPL
```

A multiple alignment

-Shared motifs point to evolutionary conservation. More informative than simple sequences

- „What a sequence whispers, an alignment pattern shouts out loud”

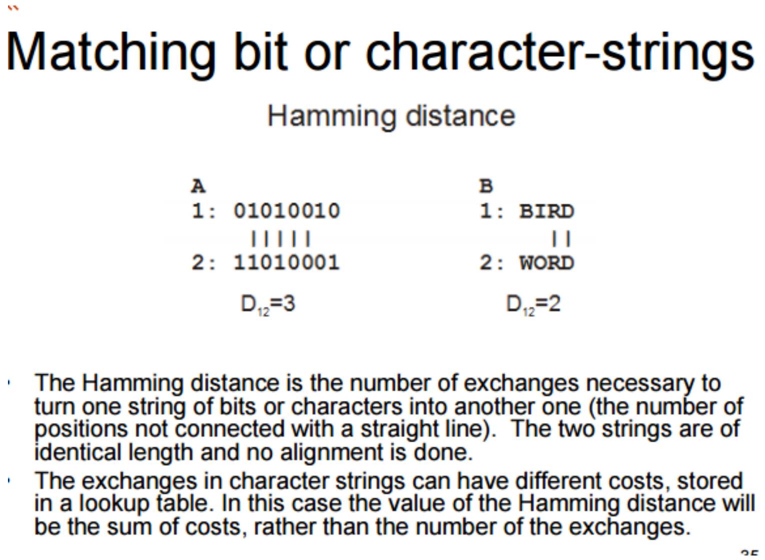
38

There is a very large and ever growing number of proximity measures. For easy problems, many of them work equally well. For difficult problems none of them do. (So do not get scared if you see unknown proximity measures – neither should you trust them)

Comparing structured descriptions: Input: 2 structured descriptions (say, sequences). Output: 1) a proximity measure (score) and 2) a shared pattern (motif). You can use proximity measures if you can turn the description into a vector (see composition type description). In addition, you can match (align) structures that gives a shared pattern.

Main distance and similarity measures

Distance measures:



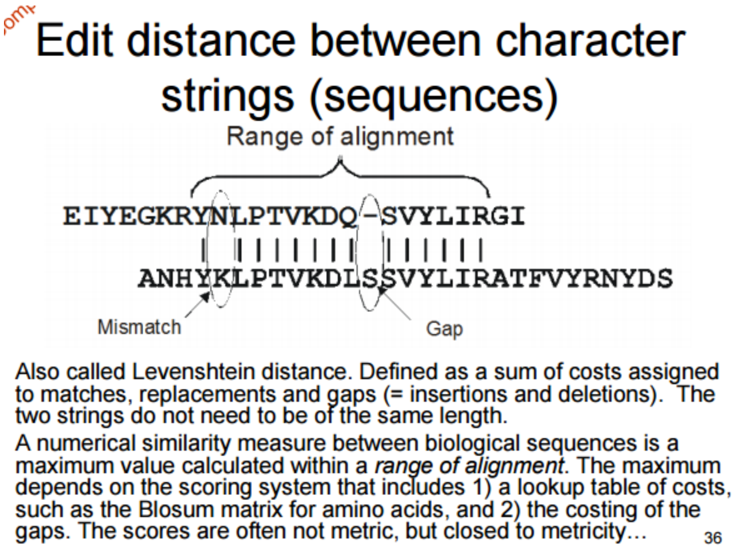
Matching bit or character-strings

Hamming distance

```
A                      B
1: 01010010            1: BIRD
      |||||                  ||
2: 11010001            2: WORD
```

$D_{12}=3$ $D_{12}=2$

- The Hamming distance is the number of exchanges necessary to turn one string of bits or characters into another one (the number of positions not connected with a straight line). The two strings are of identical length and no alignment is done.
- The exchanges in character strings can have different costs, stored in a lookup table. In this case the value of the Hamming distance will be the sum of costs, rather than the number of the exchanges.



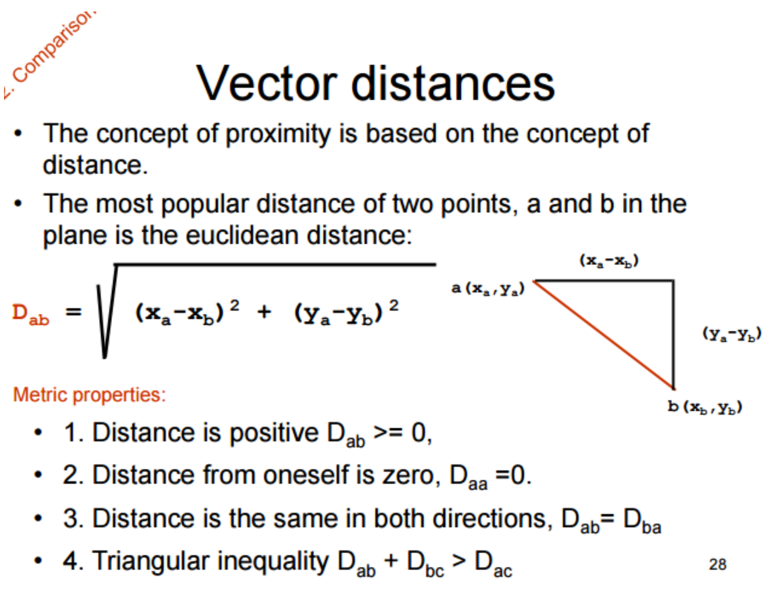
Edit distance between character strings (sequences)

Range of alignment
Mismatch
Gap

```
EIYEGKRYNLPTVKDQ-SVYLIRGI
    |  |||||||  ||||||
   ANHYKLPTVKDLSSVYLIRATFVYRNYDS
```

Also called Levenshtein distance. Defined as a sum of costs assigned to matches, replacements and gaps (= insertions and deletions). The two strings do not need to be of the same length.

A numerical similarity measure between biological sequences is a maximum value calculated within a *range of alignment*. The maximum depends on the scoring system that includes 1) a lookup table of costs, such as the Blosum matrix for amino acids, and 2) the costing of the gaps. The scores are often not metric, but closed to metricity… 36



2. Compariso

Vector distances

- The concept of proximity is based on the concept of distance.
- The most popular distance of two points, a and b in the plane is the euclidean distance:

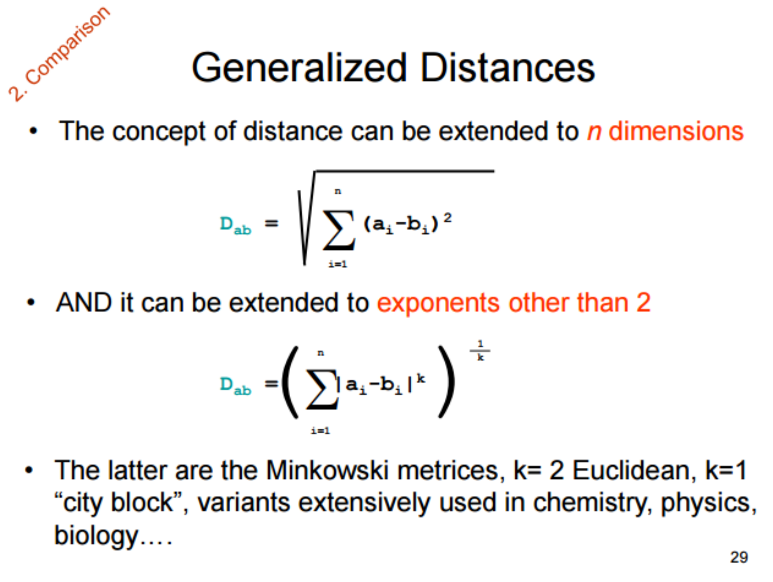
$$D_{ab} = \sqrt{(x_a-x_b)^2 + (y_a-y_b)^2}$$

(x_a-x_b)
$a\ (x_a, y_a)$
(y_a-y_b)
$b\ (x_b, y_b)$

Metric properties:

- 1. Distance is positive $D_{ab} >= 0$,
- 2. Distance from oneself is zero, $D_{aa} = 0$.
- 3. Distance is the same in both directions, $D_{ab} = D_{ba}$
- 4. Triangular inequality $D_{ab} + D_{bc} > D_{ac}$

28



2. Comparison

Generalized Distances

- The concept of distance can be extended to *n* dimensions

$$D_{ab} = \sqrt{\sum_{i=1}^{n} (a_i-b_i)^2}$$

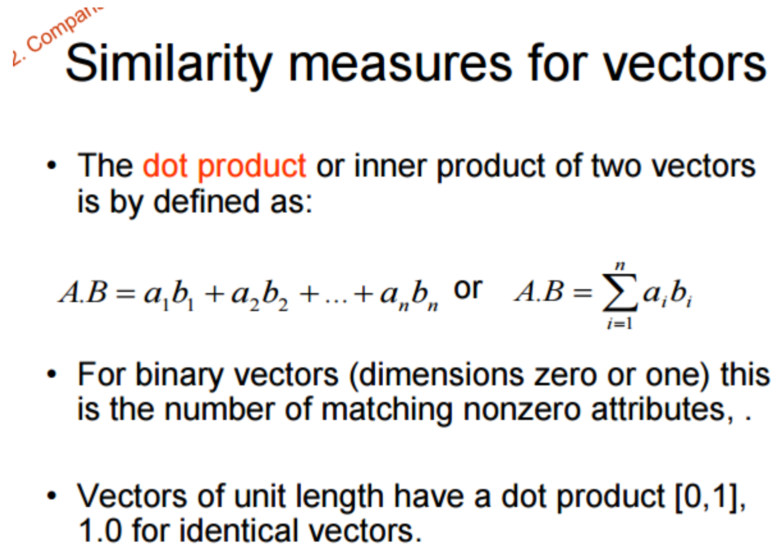
- AND it can be extended to exponents other than 2

$$D_{ab} = \left(\sum_{i=1}^{n} |a_i-b_i|^k \right)^{\frac{1}{k}}$$

- The latter are the Minkowski metrices, k= 2 Euclidean, k=1 “city block”, variants extensively used in chemistry, physics, biology….

29

Similarity measures:



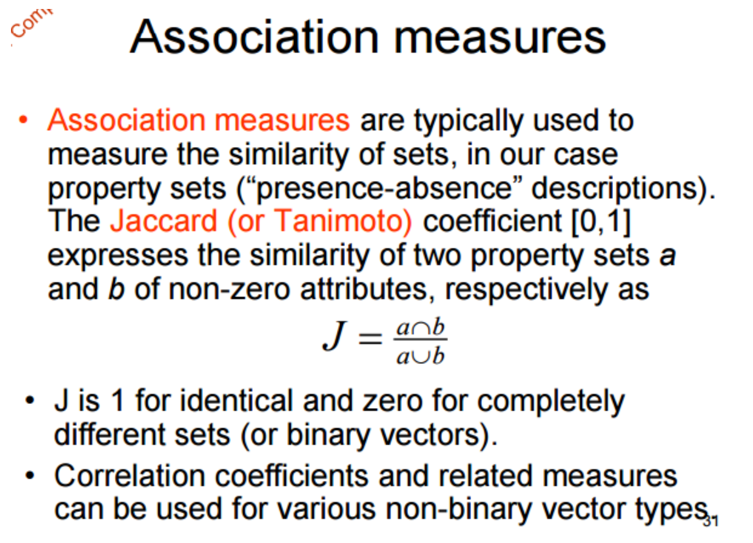
2. Compar

Similarity measures for vectors

- The dot product or inner product of two vectors is by defined as:

$$A.B = a_1b_1 + a_2b_2 + \ldots + a_nb_n \quad \text{or} \quad A.B = \sum_{i=1}^{n} a_ib_i$$

- For binary vectors (dimensions zero or one) this is the number of matching nonzero attributes, .
- Vectors of unit length have a dot product [0,1], 1.0 for identical vectors.



Com

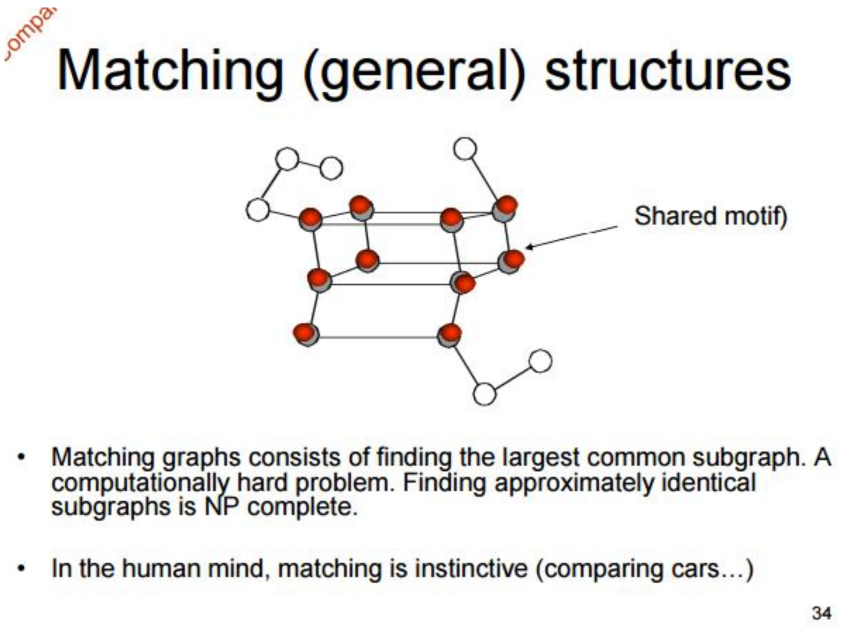
Association measures

- Association measures are typically used to measure the similarity of sets, in our case property sets (“presence-absence” descriptions). The Jaccard (or Tanimoto) coefficient [0,1] expresses the similarity of two property sets *a* and *b* of non-zero attributes, respectively as

$$J = \frac{a \cap b}{a \cup b}$$

- J is 1 for identical and zero for completely different sets (or binary vectors).
- Correlation coefficients and related measures can be used for various non-binary vector types.

31



Matching (general) structures

- Matching graphs consists of finding the largest common subgraph. A computationally hard problem. Finding approximately identical subgraphs is NP complete.
- In the human mind, matching is instinctive (comparing cars…)

34

Granularity problems (in word descriptions)

Granularity is the resolution of a description (e.g. of a vector).

Too high resolution: all objects seem to be different, no similarity between members of the same group...

Too low resolution: All objects are equal, no difference between different groups.

Finding the right amount of detail is hard. This is part of „the curse of dimensionality”

Is there an optimum?

At low dimensions, all sequences are similar. At very high ones all are different…

For a given problem, you can find an optimal resolution. Say you want to find the optimal granularity to separate two sequence groups, using word vectors (one of the groups can be very large a random selected group). Vector example Dbetween Dwithin

You can calculate all distances within the groups and between the groups.

You can compare the two distance sets with standard statistics, such as a t-test etc.

You do this for all resolutions, and see - from a plot - if there is one which is more significant than the other.

This is true of course only for normal distributions, but we can suppose that the bias will be the same for all points on the plot.

Basic applications of comparison: database searching, consensus building for groups (multiple alignment, tree)

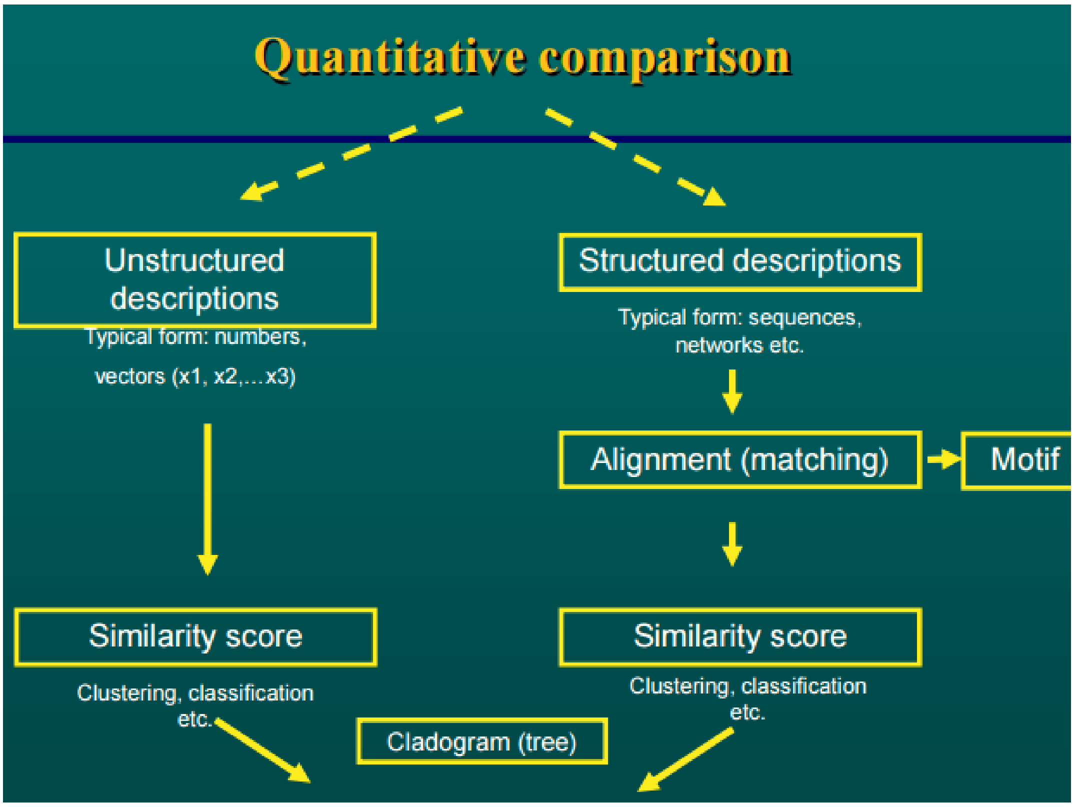
We always compare two sequences/motifs. This is pairwise comparison or pairwise alignment. This gives a score and a motif (pattern).

Two fundamental tasks

1) one sequence compared with each member of a database. Ranking hits by score, pick most similar. This is database searching.

2) Members of a group compared with each other in an all-against-all fashion. Here again we have two tasks: 2A find a common motif for the group. This is done by multiple alignment. Gives a common description for the group; 2B Build a cladogram, or tree from the similarity scores. Shows the structure for the group with implications for evolution.

Similarity, identity: Identity as a mathematical relation is symmetrical i.e. A~B → B~A, and transitive i.e. A~B~C → A~C. Similarity is symmetrical and non-transitive A~B → B~A, but A~B~C does not mean A~C. Group membership by motif is partial identity (shared substructure). This is transitive i.e. it is an identity relation. Group membership by simple score thresholding can be nontransitive.



Quantitative comparison

Unstructured descriptions

Typical form: numbers, vectors (x1, x2,…x3)

Structured descriptions

Typical form: sequences, networks etc.

Alignment (matching)

Motif

Similarity score

Clustering, classification etc.

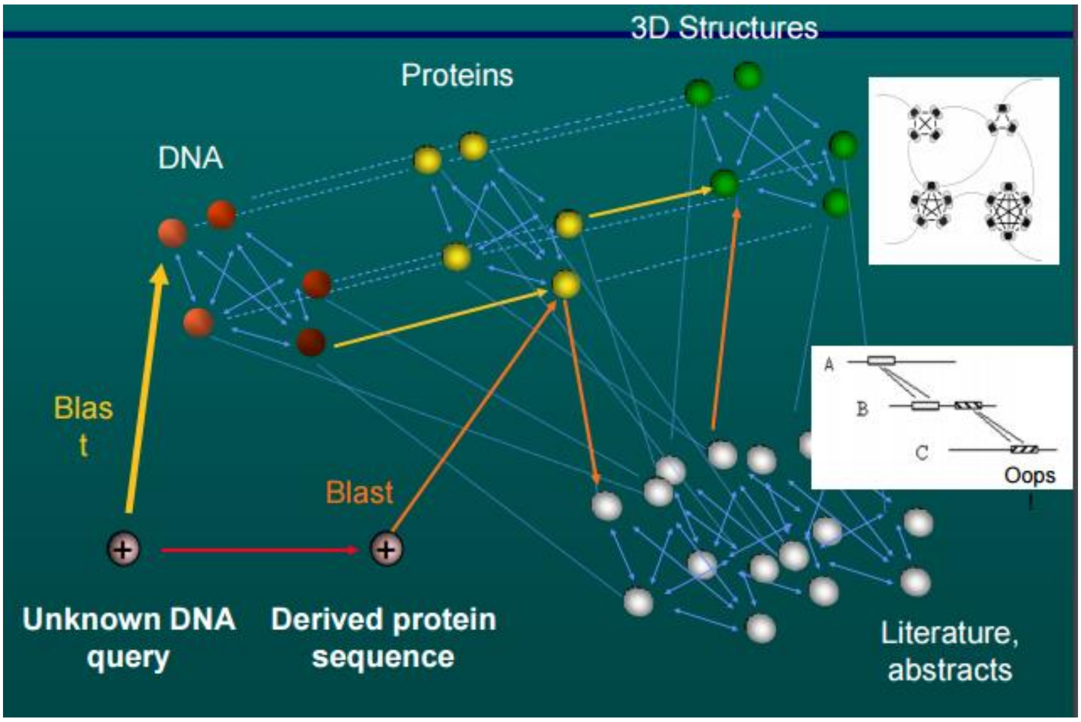
Similarity score

Clustering, classification etc.

Cladogram (tree)

A bioinformatics resource: linked, integrated, searchable databases

Search on a preprocessed, integrated database: the importance if a good neighborhood.



3. Core operations II (Aggregation, projections)

Aggregation of numbers and vectors

A Numerical aggregation operations:

Sum, Average, Median, Minimum, Maximum, Stdev (When we calculate standard deviation, we automatically suppose that the distribution is normal (Gaussian, bell curve). This is often not true, but we still do this as a first approximation…)

B Diagrammatic aggregation:

Histograms and distributions

Aggregation of sequences by functional and similarity links

Aggregation by links:

By similarity/distance links → distance matrix, heat map → similarity group → cladogram, tree

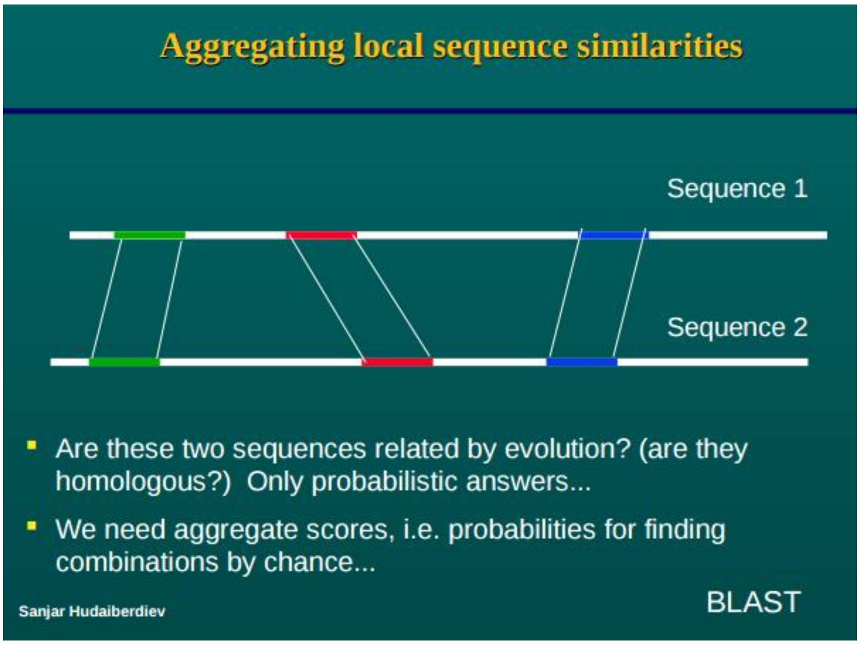
By functional (context) links → pathways

Aggregation by common motif:

Multiple alignment

Consensus descriptions of multiple alignment

Agregation in analysis and identification: Biological objects are large and complex (genomes, proteomes, metagenomes, pathway data, etc.) Often, measuring instruments can only collect data on small pieces (next generation sequencing reads, peptide spectra in proteomics). Computational analysis of small fragments is accurate. There is one general trick: We divide a complex object into simple parts (like characteristic motifs), identify individual parts by simple numerical means, and then AGGREGATE the results. Not elegant, but works, even with very complex problems (forensic fingerprints).



Aggregating local sequence similarities

Sequence 1

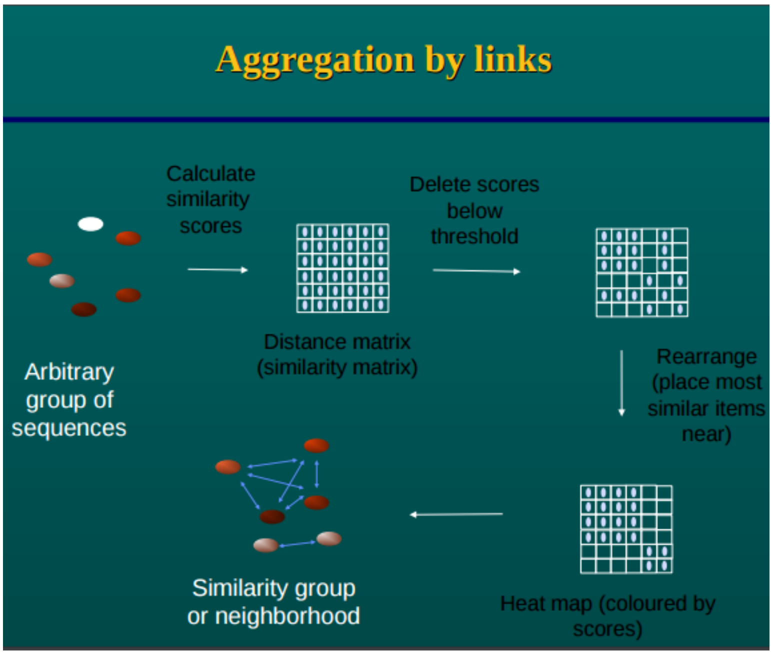
Sequence 2

- Are these two sequences related by evolution? (are they homologous?) Only probabilistic answers...
- We need aggregate scores, i.e. probabilities for finding combinations by chance...

Sanjar Hudaiberdiev

BLAST

Aggregation by similarity into distance matrices, heat maps, trees



Aggregation by links

Calculate similarity scores

Delete scores below threshold

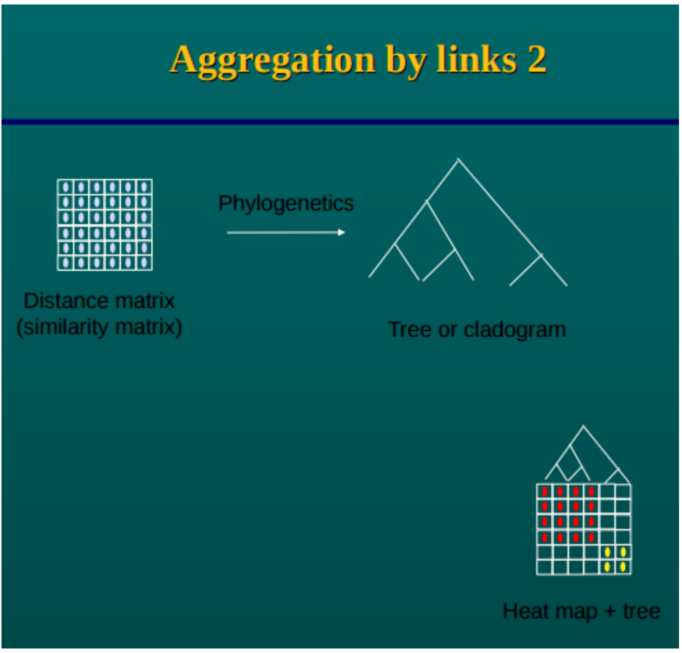
Arbitrary group of sequences

Distance matrix (similarity matrix)

Rearrange (place most similar items near)

Similarity group or neighborhood

Heat map (coloured by scores)



Aggregation by links 2

Phylogenetics

Distance matrix (similarity matrix)

Tree or cladogram

Heat map + tree

Projection: Numerical annotation of sequences, window sliding

What numbers do we plot:

- A property of an amino acid/nucleotide. I.e. a value stored in a lookup-table.
- A value calculated from the sequence or from the associated 3D structure (a „window”)
- A value determined by experiment: The sequencing quality of the position. Number of reads “hitting” a position

Sliding Window Approach

- Calculate property for first sub-sequence
- Use the result (plot/print/store)
- Move to the next position in the sequence

I L I K E I R

$$\underbrace{4.50 + 3.80 + 4.50 - 3.90 - 3.50 + 4.50 - 4.50}_{5.4 / 7 = 0.77}$$

Projection: Hydrophobicity plots

Prediction of hydrophobic and hydrophilic regions in a protein.

Hydrophobicity Plot:

- Sum amino acid hydrophobicity values in a given window
- Plot the value in the middle of the window
- Shift the window one position

$$\langle H_i \rangle = \frac{1}{2k+1} \sum_{n=i-k}^{i+k} H_n$$

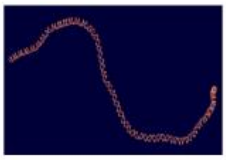
Large H → hydrophobic, e.g. membrane bound segments

Projection: DNA bending plots

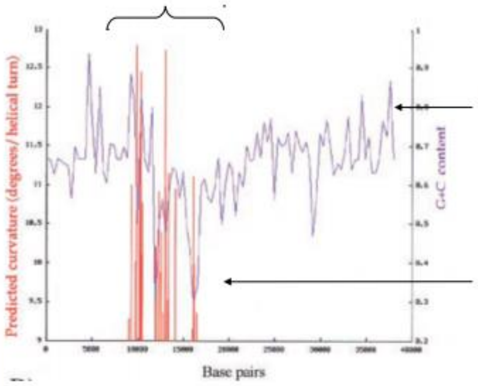
Prediction bent regions in DNA.

Prediction of bent regions in DNA

A strange chromosomal region in Leishmania – analyzed with a www server



Bent region!



Flexibility

Bending

http://pongor.itk.ppke.hu/?q=bioinfoservices

4. Sequence alignment

Sequence scoring matrices (PAM, BLOSUM, unitary, and how to make one's own…)

The substitution matrix (also called scoring matrix) contains cost for amino acid or nucleotide identities and substitutions in an alignment. For amino acids it is a 20×20, for nucleotides it is a 4×4 matrix that can be constructed from pairwise alignments of related sequences. Re-lated means either

- evolutionary relatedness described by an 'approved' evolutionary tree (PAM)
- any sequence similarity as described in the PROSITE database (BLOSUM)

Groups of related sequences can be organized into a multiple alignment for calculation of the matrix elements.

Calculation of scoring matrices from multiple alignment

Matrix elements are calculated from the observed and expected frequencies using the 'log odds' principle. For A and B amino acids the matrix element in the intersection of row (column) A and column (row) B will be $M(A/B)$=log($f(A/B)$ / $f(A)\cdot f(B)$) where $f(X)$ is the number of occurrences (frequency) of X, and $f(X/Y)$ is the number of oc-currences where X is aligned with Y in the multiple alignment.
The log odds values in the matrix are normalized to a given range depending on the application (but the range does not matter much).

Nucleic acid matrices

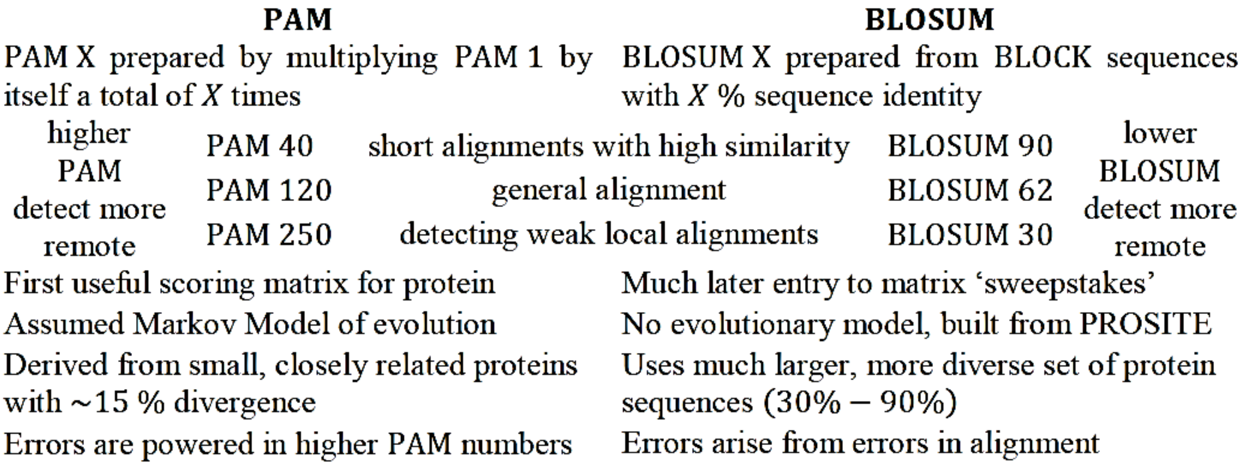
The magnitudes of the elements are relative and can be scaled. Heuristic matrices can be easily constructed: the identity matrix contains '1'-s in the diagonal and '0'-s everywhere else. One can penalize certain associations assigning a large negative value to them, etc.

PAM matrix

PAM stands for **P**ercent **A**ccepted **M**utation, which is a unit of evolutionary change for protein sequences. The PAM matrix is calculated from related sequences organized into 'accepted' evolutionary trees. This 20×20 matrix – where the columns add up to the number of cases observed – is converted into scoring matrix by log odds and scaling. In PAM 1 1% of amino acids mutate. One times the PAM matrix means 1 million years of divergence.

BLOSUM matrix

BLOSUM stands for **Blo**ck **Su**bstitution **M**atrix. No evolutionary model is assumed; it is built from PROSITE derived sequence blocks and uses much larger, more diverse set of protein sequences. These matrices are sensitive to structural and functional substitution.



PAM			BLOSUM	
PAM X prepared by multiplying PAM 1 by itself a total of X times			BLOSUM X prepared from BLOCK sequences with X % sequence identity	
higher PAM detect more remote	PAM 40	short alignments with high similarity	BLOSUM 90	lower BLOSUM detect more remote
	PAM 120	general alignment	BLOSUM 62	
	PAM 250	detecting weak local alignments	BLOSUM 30	
First useful scoring matrix for protein			Much later entry to matrix ‘sweepstakes’	
Assumed Markov Model of evolution			No evolutionary model, built from PROSITE	
Derived from small, closely related proteins with ~15 % divergence			Uses much larger, more diverse set of protein sequences (30% – 90%)	
Errors are powered in higher PAM numbers			Errors arise from errors in alignment	

Dot plots

Dot plot method is to visualize sequence identities along two n-length nucleotide or amino acid sequences. It is based on an $n \times n$ matrix with the two sequences written on the two ax-es. If we put a dot on those positions where the i-th row and the j-th column has the same nucleotide(group) we get a diagonal line for identical sequences and some other pictures for other sequences.

Algorithm

1. Select a word size and a scoring scheme.

2. For every pair of words :

- compute a word match score in the normal way
- if the score reaches the cut-off score, draw a dot in the intersection of the two lines containing the two words. Else, no dots are placed.

Parameters

Word size

Word size defines the ‘windowing function’ applied on the sequence. Large word size can miss small matches. Smaller words pick up smaller features. The drawback is that the smallest features are often just noise. For sequence with regions of small matching features- small words pick up small features individually. Larger words show matching regions more clearly. The lack of detail can be an advantage.

Scoring scheme

Scoring scheme is used in the calculation of the score of each word. There are different scoring schemes depending on the type of the sequences (nucleotide or amino acid).

DNA: Usually, DNA Scoring schemes award a fixed reward for each matched pair of bases and a fixed penalty for each mismatched pair of bases. Choosing between such scor-ing schemes will affect only the choice of a sensible cut-off score and the way ambi-guity codes are treated.

Protein: Protein scoring schemes differ in the evolution distance assumed between the proteins being compared. The choice is rarely crucial for dotplot programs.

Cut-off score

Cut-off score decides whether the dot at a given position is to be shown on the dot plot or not. The higher the cut-off score the fewer dots will be plotted but, each dot is more likely to be significant. The lower the cut-off score the more dots will be plotted, but dots are more likely to indicate a chance match (noise).

Usages

Detection of deletion, insertion

Plot two sequences retrieved from different sources but supposed to be identical. The dot plot will show insertions and deletions with the broken diagonal line.

Detection of repeats

Put the same sequence on both axes. The dot plot will show repeating sequences beside the main diagonal line.

Detection of stem loops

Put the same sequence on both axes, but in reverse direction on one of them. Stem loops will appear as perpendicular lines to the main diagonal.

The two basic algorithms Global alignment (Needleman Wunsch), local alignment (Smith-Waterman)

Global alignment (Needleman–Wunsch)

Align two sequences from ‘head to toe’, i.e. from 5’ ends to 3’ ends. For this problem an exhaustive algorithm was published by Needleman and Wunsch in 1970. The rules of the algo-rithm are

- Match at position (i,j) is $M(i,j)=W(i,j)$ where W is a weighting matrix
- Gap is described by a gap function $G=G_{\text{initiation}}+G_{\text{elongation}}$
- Score at position (i,j) is $S(i,j)$ which is a maximum of
  $S(i+1,j+1)+M(i,j)$ (diagonal movement),
  $S(i+1,j)+M(i,j)-G$ (horizontal movement),
  $S(i,j+1)+M(i,j)-G$ (vertical movement).

To align two sequences we write them along two axis of a matrix. We initialize the matrix by adding a ‘gap’ character to the end of the sequences and calculate the scores in this last row (column). To align the sequences, we fill the matrix step-by step, decreasing i and j values. Also the score and the path of the calculation are stored in a list. To generate the alignment, we trace back the path choosing the maximum of the neighboring cells position-by-position.

Local alignment (Smith–Waterman)

Locate regions with sigh degree of similarity in two sequences. For this problem an exhaus-tive algorithm was published by Smith and Waterman in 1981. The rules of the algorithm are

- Match at position (i,j) is $M(i,j)=W(i,j)$ where W is a weighting matrix
- Gap is described by a gap function $G=G_{\text{initiation}}+G_{\text{elongation}}$
- Score at position (i,j) is $S(i,j)$ which is a maximum of
  $S(i+1,j+1)+M(i,j)$ (diagonal movement),
  $S(i+1,j)+M(i,j)-G$ (horizontal movement),
  $S(i,j+1)+M(i,j)-G$ (vertical movement),
  0 (so we cannot have negative score values).

We align sequences like in global alignment, except that the score values will always be at least zero.

Gap handling

In the simplest case, every gap has a constant G value, e.g. 1. In this case the length of the gap does not matter.
In the linear case, the gap penalty is a linear function of the gap length, e.g. $G=G_{\text{length}}\cdot K$ where K is a constant, e.g. 1.
In the affine case, gap penalty is a sum of a gap initiation cost and a gap elongation cost:
$G=G_{\text{init}}+G_{\text{extension penality}}\cdot G_{\text{length}}$
Also other gap functions are exists like convex gap penalty functions.

5. Multiple alignments

What is a multiple alignment?

The goal of multiple alignment to present conservation conserved sequence motifs within a group of related proteins, protein sites, nucleic acid sites.

The multiple alignment is a method to visualize a group of sequences. An aggregated group description with severa views.

A group of similar sequences has two main representations: tree and the multiple alignment.

We can build up a multiple alignment in case of protein sequences (single domain sequences, short sub-sequences) and DNA/RNA sequences (mostly short sites).

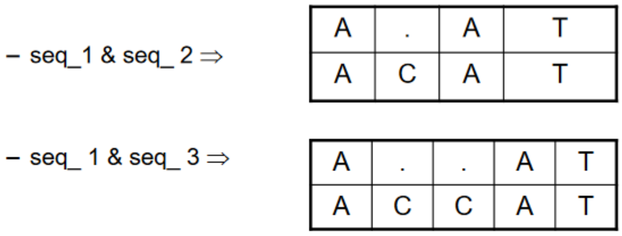
How do we construct multiple alignments?

Early approach: writin sequences on top of each other.

A multiple alignment can be decomposed into to pairwise alignments.

Additive approach:

1. Pairwise comparsion:
   Compare every single sequence to every other sequence.



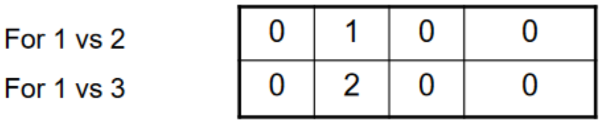
– seq_1 & seq_ 2 ⇒

A	.	A	T
A	C	A	T

– seq_ 1 & seq_ 3 ⇒

A	.	.	A	T
A	C	C	A	T

2. Represent alignments as vectors:
   Record the resulting gaps in vectors.

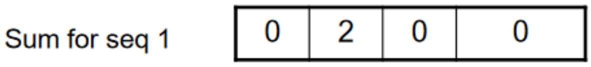


For 1 vs 2	0	1	0	0
For 1 vs 3	0	2	0	0

3. Combine vectors for each pair:

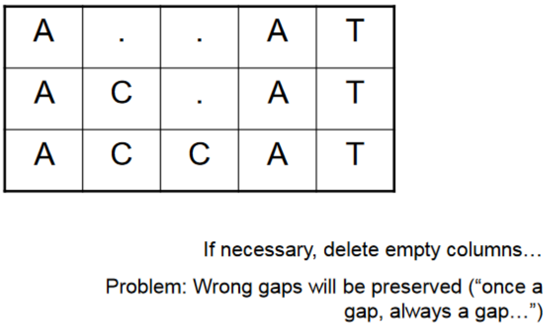
   Vectors to be added.

   Rule of addition: sum(x,y)=max(x,y).



Sum for seq 1	0	2	0	0

4. Construct multiple alignment:



A	.	.	A	T
A	C	.	A	T
A	C	C	A	T

If necessary, delete empty columns…

Problem: Wrong gaps will be preserved ("once a gap, always a gap…")

Iterative approach:

Do a pairwise comparison of all sequences.

From this, calculate how sequences are related to each other (the more similar are easier to align).

Construct a guide tree from the pairwise comparison values, using a clustering algorithm.

Perform multiple alignment in order; the most similar are aligned first, the others are saved for later.

Rewriting multiple alignment as…

regular expression:

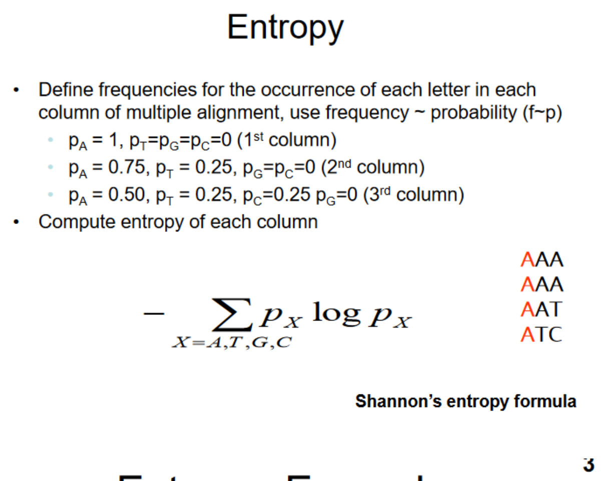
You can use a regular expression to search a large database, to find all occurrences in very short time.

If you construct a good regular expression for your multiple alignment, you can find members of the group.

Regular expressions are better than consensus sequences but still do not capture all details of a multiple alignment.

Pairwise comparison also gives a motif. Motives can be constructed into a consensus motif. This motif (e.g. a regular expression) is also a representation of the group.

conservation plot (???):



Entropy

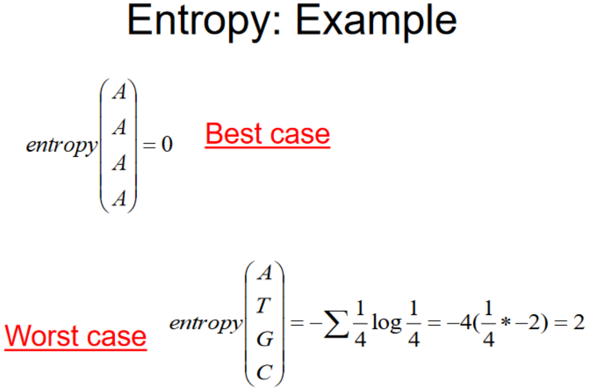
- Define frequencies for the occurrence of each letter in each column of multiple alignment, use frequency ~ probability (f~p)
  - $p_A = 1$, $p_T=p_G=p_C=0$ (1st column)
  - $p_A = 0.75$, $p_T = 0.25$, $p_G=p_C=0$ (2nd column)
  - $p_A = 0.50$, $p_T = 0.25$, $p_C=0.25$ $p_G=0$ (3rd column)
- Compute entropy of each column

$$-\sum_{X=A,T,G,C} p_X \log p_X$$

AAA
AAA
AAT
ATC

Shannon's entropy formula

3



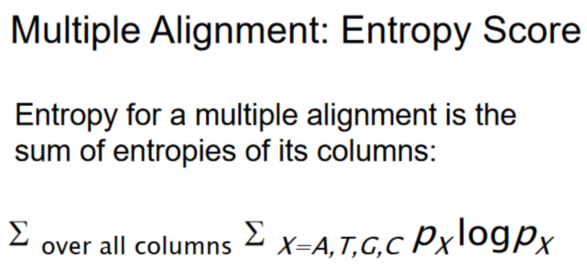
Entropy: Example

$$entropy\begin{pmatrix}A\\A\\A\\A\end{pmatrix}=0$$

Best case

Worst case

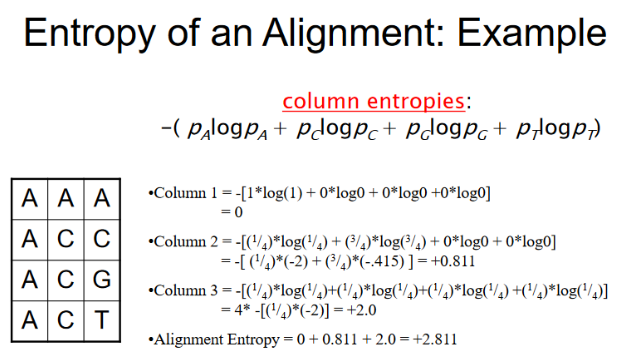
$$entropy\begin{pmatrix}A\\T\\G\\C\end{pmatrix}=-\sum\frac{1}{4}\log\frac{1}{4}=-4(\frac{1}{4}*-2)=2$$



Multiple Alignment: Entropy Score

Entropy for a multiple alignment is the sum of entropies of its columns:

$$\Sigma_{\text{over all columns}}\ \Sigma_{X=A,T,G,C}\ p_X \log p_X$$



Entropy of an Alignment: Example

column entropies:

$$-(\ p_A \log p_A + p_C \log p_C + p_G \log p_G + p_T \log p_T)$$

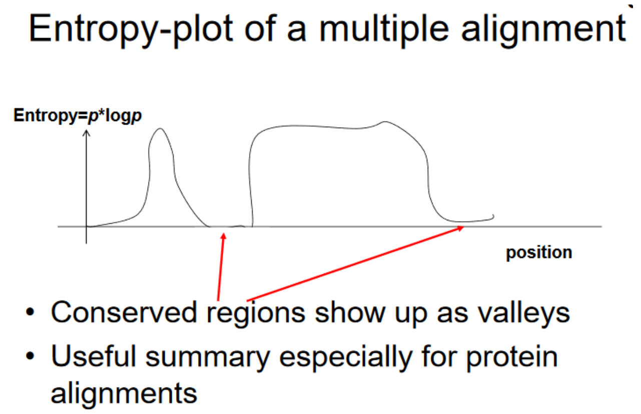
A	A	A
A	C	C
A	C	G
A	C	T

•Column 1 = -[1*log(1) + 0*log0 + 0*log0 +0*log0]
= 0

•Column 2 = -[($^1/_4$)*log($^1/_4$) + ($^3/_4$)*log($^3/_4$) + 0*log0 + 0*log0]
= -[($^1/_4$)*(-2) + ($^3/_4$)*(-.415)] = +0.811

•Column 3 = -[($^1/_4$)*log($^1/_4$)+($^1/_4$)*log($^1/_4$)+($^1/_4$)*log($^1/_4$) +($^1/_4$)*log($^1/_4$)]
= 4* -[($^1/_4$)*(-2)] = +2.0

•Alignment Entropy = 0 + 0.811 + 2.0 = +2.811



Entropy-plot of a multiple alignment

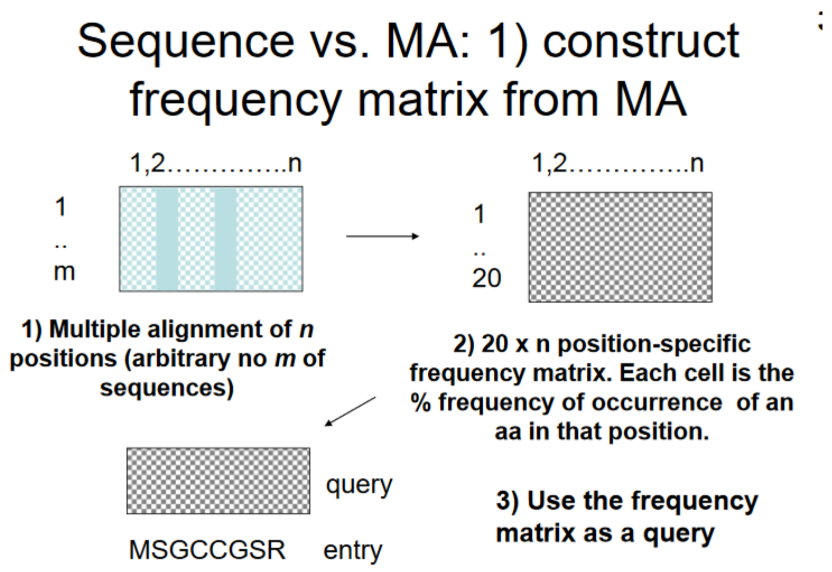
- Conserved regions show up as valleys
- Useful summary especially for protein alignments

frequency matrix and sequence profile:

We count the frequencies of each amino acid in the alignment. If we have many sequences, this will give a good approximation of the probability of finding an amino acid or gap in a given position. This is the sequence profile.

It is used to find all new members of the group: profile libraries are standard tools for genome annotation.

We can compare two profiles. This will give a similarity between two groups of sequences. This is the most sensitive sequence comparison method today.



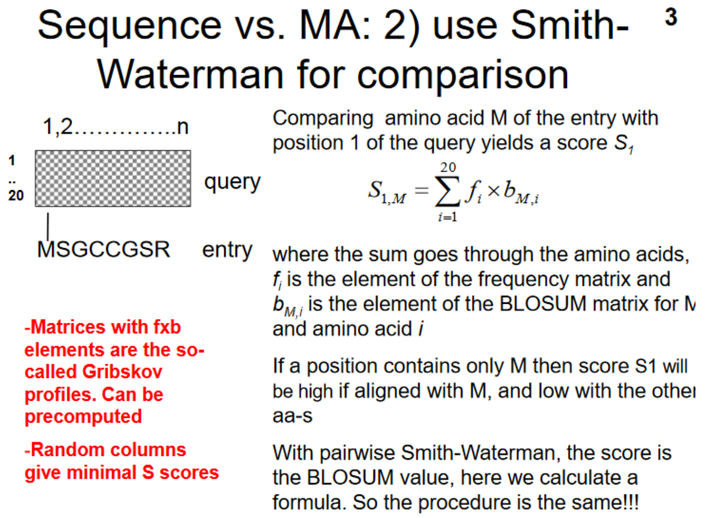
Sequence vs. MA: 1) construct frequency matrix from MA

1) Multiple alignment of *n* positions (arbitrary no *m* of sequences)

2) 20 x n position-specific frequency matrix. Each cell is the % frequency of occurrence of an aa in that position.

3) Use the frequency matrix as a query

Aligning a sequence to a multiple alignment.



Sequence vs. MA: 2) use Smith-Waterman for comparison

3

1,2…………..n
1
..
20
query
MSGCCGSR
entry

Comparing amino acid M of the entry with position 1 of the query yields a score S_1

$$S_{1,M} = \sum_{i=1}^{20} f_i \times b_{M,i}$$

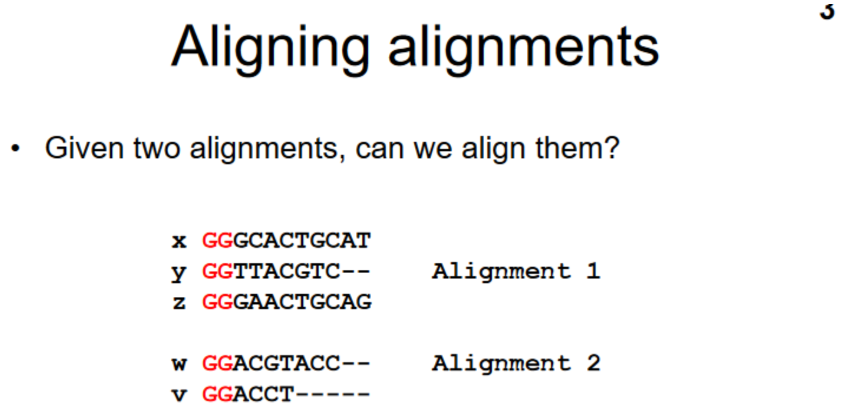
where the sum goes through the amino acids, f_i is the element of the frequency matrix and $b_{M,i}$ is the element of the BLOSUM matrix for M and amino acid i

If a position contains only M then score S1 will be high if aligned with M, and low with the other aa-s

With pairwise Smith-Waterman, the score is the BLOSUM value, here we calculate a formula. So the procedure is the same!!!

-Matrices with fxb elements are the so-called Gribskov profiles. Can be precomputed

-Random columns give minimal S scores



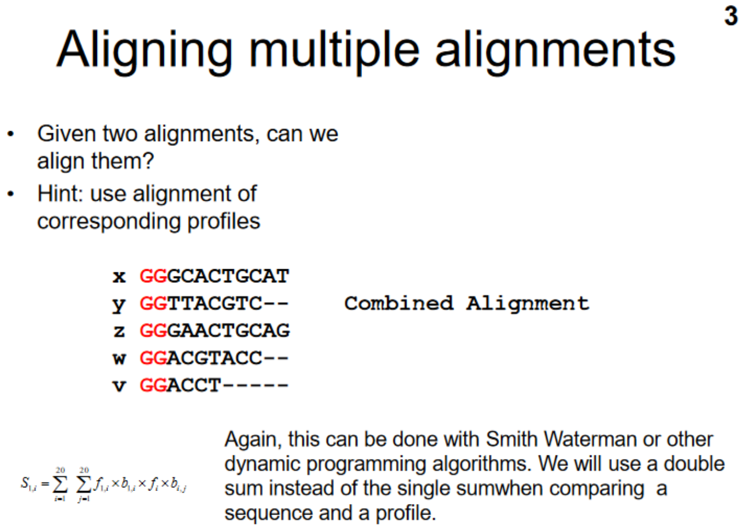
3

Aligning alignments

- Given two alignments, can we align them?

```
x  GGGCACTGCAT
y  GGTTACGTC--     Alignment 1
z  GGGAACTGCAG

w  GGACGTACC--     Alignment 2
v  GGACCT-----
```



Aligning multiple alignments

3

- Given two alignments, can we align them?
- Hint: use alignment of corresponding profiles

```
x  GGGCACTGCAT
y  GGTTACGTC--     Combined Alignment
z  GGGAACTGCAG
w  GGACGTACC--
v  GGACCT-----
```

$$S_{1,i} = \sum_{i=1}^{20} \sum_{j=1}^{20} f_{1,i} \times b_{1,i} \times f_i \times b_{i,j}$$

Again, this can be done with Smith Waterman or other dynamic programming algorithms. We will use a double sum instead of the single sumwhen comparing a sequence and a profile.

Validating a multiple alignment.

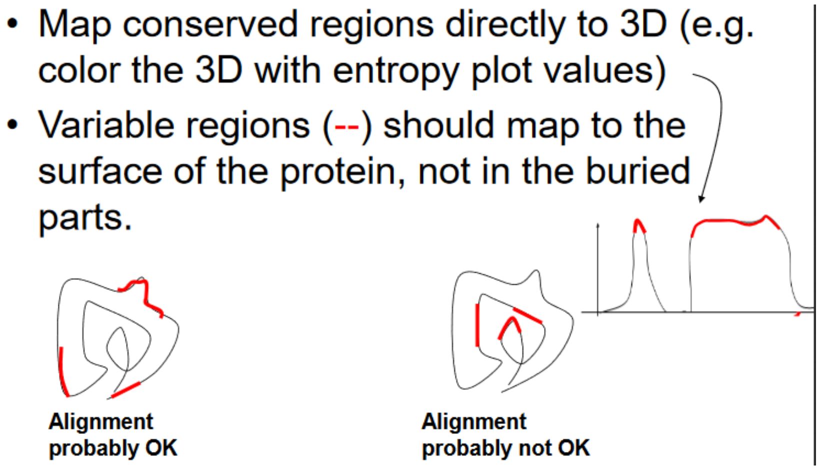
Method 1:

If one of the members of a multiple alignment has a known 3D structure, we can use it for validating the multiple alignment.

Rule: some secondary structures, especially helix, don’t contain gaps.

Convert 3D structure into a series of secondary structure assignments (H,E,T,C) and write on top of the alignment.

Method 2:



- Map conserved regions directly to 3D (e.g. color the 3D with entropy plot values)
- Variable regions (--) should map to the surface of the protein, not in the buried parts.

6. Bioinformatics databases

Data types in bioinformatics:

Sequences, 3D structures, Networks, Texts

Database formats:

- Flat files
- XML
- RDF (Resource Description Framework)
- Relational DBs (SQL <333)
- Accessible forms: webpage, generated

Tasks of DBs

- Searchable, organized, regulaly updated
- Specialized on certain data types (maybe cross-referenced to other DBs)
- Textual info
- Associated with certain computational methods (e.g. Uniprot & BLAST)

Steps of DB constrction

1. Data collection
2. Validation
3. Clustering
4. Annotation
5. Integration, visualization

Structures of Data:

Data: sequences

Metadata (annotations): Locally stored, cross-referenced to other DBs (functions, domains contained, name etc…)

The DB contains records, which contain fields

Sequence DBs

Primary: full sequences

Secondary: domains, modification sites…

Raw DBs

Contains only seq-s and some predicted functions

Construction:

1. Data collection, ID of Open Reading Frames
2. Redundance filtering
3. Comparison with previous release

Annotated DBs

Provides additional info about the seq.

- Human intervention, literature searching
- Literature citation
- Cross-reference to other DBs
- Known/predicted functions
- Variants
- Domains, binding sites…

Ontology

Formal naming/definition of the types/properties of an entity. In DBs standardized concepts and language constitutes the ontology. The Gene Ontology Project (GO project) standardizes the names & functional descriptors of genes /proteins in conceptual hierarchies.

(eg.: Binding -> Ion-binding -> Cation ion-binding)

Protein universe

Clustering used to group similar proteins via alignment scores (same group/function: family; High alignment score)
COG: Clusters of Orotlogous Proteins
Seq.s of common evolutionary origin carrying the same function.

Uniprot sub DBs:

ProteinKb DB, Sequence DB, Proteomes, Sequence clusters

7. Blast.ppt

Similarity searching, main steps.

Given a query and a database, find the entry in the database that is most similar to the query in terms of a numerical similarity measure (distance, similarity score, etc.).

In contrast: retrieval looks for an exact match to the query.

Main steps:

1. Compare query with all entries in the database and register similarity score. Store results above some threshold (cutoff).
2. Calculate significance of the score (compared to chance similarities).
3. Rank entries according to similarity score or significance (top list).
4. Report the best hit (usually after some simple statistics, e.g. if it is higher than a threshold…), add alignment pattern.
5. Assume functionof query, i.e. classify query into a class present in the database. Alignment pattern is an important proof for classification.

Computational and biological heuristics.

Heuristic/hy*oo*-**ris**-tik/ refers to experience-based techniques for problem solving, learning, and discovery.

Computational heuristics:

exact matching:

Exact word matching is fast, use it whenever you can.

Longer words are more informative but few of the possible words occur in a given sequence, and most of them only once.

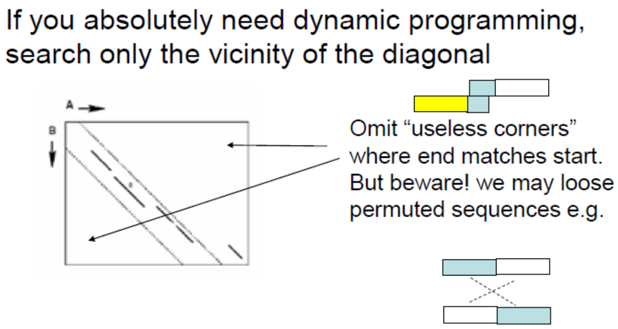
space reduction:

Most sequences in a database are not similar to a given query. It is easy to construct filters that throw away uninteresting sequences.

Threshold based filtering is important, but has drawbacks (you can throw away important hits).

Biological heuristics:

search space reduction:


If you absolutely need dynamic programming, search only the vicinity of the diagonal

Omit “useless corners” where end matches start. But beware! we may loose permuted sequences e.g.

short conserved *sequences*:

Sequence related by evolution always contain conserved regions that are highly similar, contain rows of identical residues.

Identify them during the calculation that do not and omit them from the score.

Danger: difficult to define, what is a conserved region (and what is a good threshold).

repetitive regions:

Biological sequences often contain repetitive of “biased composition”, “low complexity” regions. These can be excluded from the query by masking them in advancewith X-es in order to omit them from the entire calculation.

Such sequences contain a biased composition of n-gram words, they are thus less complex (can be described with fewer words).

BLAST algorithm.

1) Stores dbase in a hash table, with *n*-mer words and occurrences (preprocessing) (n=~11 for nucleotides, 2-4 for amino acids).

2) Records words in a query, includes similar words, based on BLOSUM similarity.

3) Selects dbase entries that share a given no of common words with the query (fast).

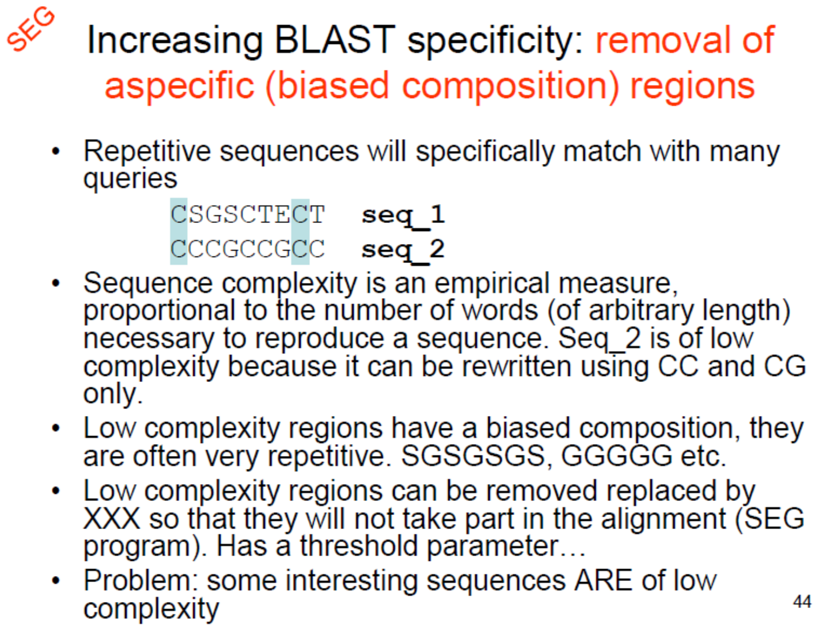
4) From here it selects those where the shared words densely cluster in certain regions ("seed" of "high scoring pair", HSP)

5) Elongates HSPs and splices them together by some (~arbitrary) criterion.

6) Calculates significance of the spliced region(s) using the BLAST formula.

7) Ranks the top scoring entries by significance, and presents a) the scores and b) the pairwise alignments for the toplist. The PA-s are made by Smith Waterman (rigorous local alignment)

Refinements.



SEG

Increasing BLAST specificity: removal of aspecific (biased composition) regions

- Repetitive sequences will specifically match with many queries

```
CSGSCTECT   seq_1
CCCGCCGCC   seq_2
```

- Sequence complexity is an empirical measure, proportional to the number of words (of arbitrary length) necessary to reproduce a sequence. Seq_2 is of low complexity because it can be rewritten using CC and CG only.
- Low complexity regions have a biased composition, they are often very repetitive. SGSGSGS, GGGGG etc.
- Low complexity regions can be removed replaced by XXX so that they will not take part in the alignment (SEG program). Has a threshold parameter…
- Problem: some interesting sequences ARE of low complexity

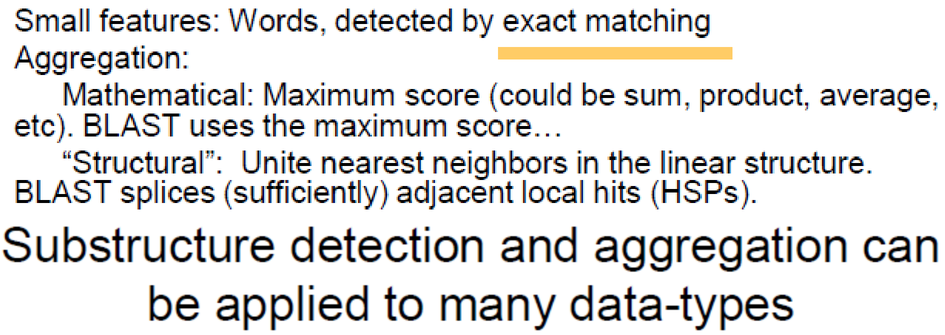
44

Different kind of BLAST programs. (PSI-BLAST)

PSI-BLAST (Position-Specific Iterated BLAST) derives a position-specific scoring matrix (PSSM) or profile from the multiple sequence alignment high scoring sequences and updates with iterated searches.

Generalization: Substructure + aggregation (mathematical and structural). Applicable to most data-types.

Substructure: dafuq?!



Small features: Words, detected by exact matching

Aggregation:

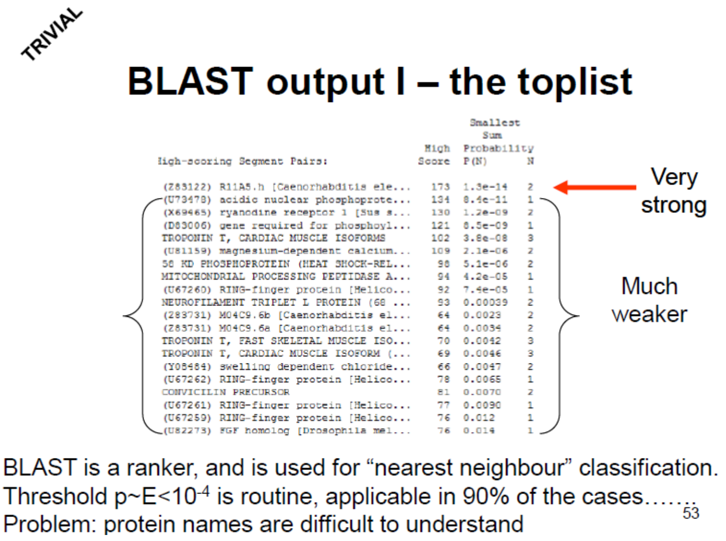
Mathematical: Maximum score (could be sum, product, average, etc). BLAST uses the maximum score…

“Structural”: Unite nearest neighbors in the linear structure. BLAST splices (sufficiently) adjacent local hits (HSPs).

Substructure detection and aggregation can be applied to many data-types

Evaluation of BLAST top lists (score, motif).

Trivial case:



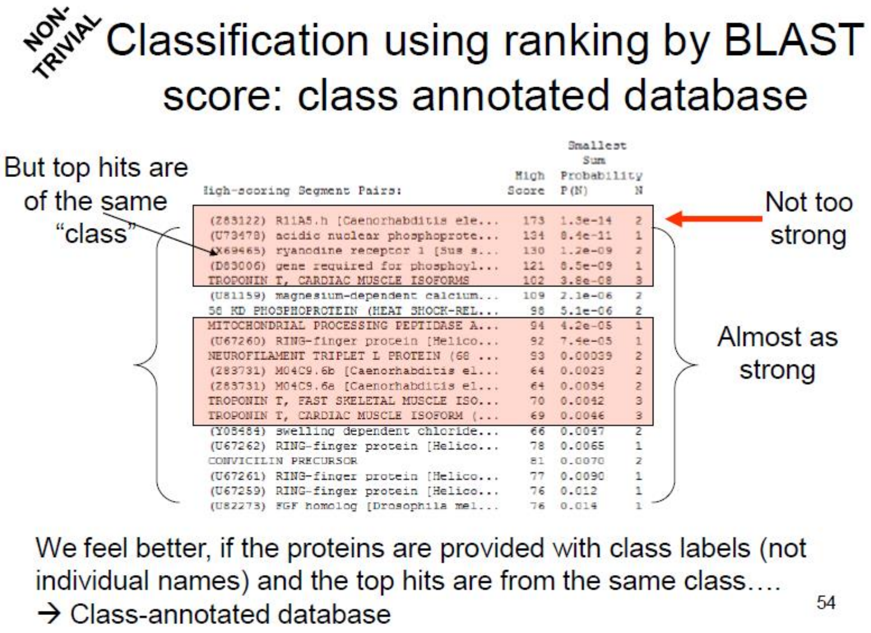
BLAST output I – the toplist

```
                                                      Smallest
                                                        Sum
                                               High  Probability
High-scoring Segment Pairs:                    Score  P(N)      N

 (Z83122) R11A5.h [Caenorhabditis ele...     173  1.3e-14   2
 (U73478) acidic nuclear phosphoprote...     134  8.4e-11   1
 (X69465) ryanodine receptor 1 [Sus s...     130  1.2e-09   2
 (D83006) gene required for phosphoyl...     121  8.5e-09   1
TROPONIN T, CARDIAC MUSCLE ISOFORMS         102  3.8e-08   3
 (U81159) magnesium-dependent calcium...     109  2.1e-06   2
58 KD PHOSPHOPROTEIN (HEAT SHOCK-REL...      98  5.1e-06   2
MITOCHONDRIAL PROCESSING PEPTIDASE A...      94  4.2e-05   1
 (U67260) RING-finger protein [Helico...      92  7.4e-05   1
NEUROFILAMENT TRIPLET L PROTEIN (68 ...      93  0.00039   2
 (Z83731) M04C9.6b [Caenorhabditis el...      64  0.0023    2
 (Z83731) M04C9.6a [Caenorhabditis el...      64  0.0034    2
TROPONIN T, FAST SKELETAL MUSCLE ISO...      70  0.0042    3
TROPONIN T, CARDIAC MUSCLE ISOFORM (...      69  0.0046    3
 (Y08484) swelling dependent chloride...      66  0.0047    2
 (U67262) RING-finger protein [Helico...      78  0.0065    1
CONVICILIN PRECURSOR                          81  0.0070    2
 (U67261) RING-finger protein [Helico...      77  0.0090    1
 (U67259) RING-finger protein [Helico...      76  0.012     1
 (U82273) FGF homolog [Drosophila mel...      76  0.014     1
```

BLAST is a ranker, and is used for “nearest neighbour” classification.
Threshold p~E<10^{-4} is routine, applicable in 90% of the cases……
Problem: protein names are difficult to understand

Non-trival case:



NON-TRIVIAL

Classification using ranking by BLAST score: class annotated database

But top hits are of the same “class”

```
                                                      Smallest
                                                        Sum
                                              High  Probability
High-scoring Segment Pairs:                   Score  P(N)      N

 (Z83122) R11A5.h [Caenorhabditis ele...       173  1.3e-14    2
 (U73478) acidic nuclear phosphoprote...       134  8.4e-11    1
 (X69465) ryanodine receptor 1 [Sus s...       130  1.2e-09    2
 (D83006) gene required for phosphoyl...       121  8.5e-09    1
TROPONIN T, CARDIAC MUSCLE ISOFORMS            102  3.8e-08    3
 (U81159) magnesium-dependent calcium...       109  2.1e-06    2
58 KD PHOSPHOPROTEIN (HEAT SHOCK-REL...         98  5.1e-06    2
MITOCHONDRIAL PROCESSING PEPTIDASE A...         94  4.2e-05    1
 (U67260) RING-finger protein [Helico...        92  7.4e-05    1
NEUROFILAMENT TRIPLET L PROTEIN (68 ...         93  0.00039    2
 (Z83731) M04C9.6b [Caenorhabditis el...        64  0.0023     2
 (Z83731) M04C9.6a [Caenorhabditis el...        64  0.0034     2
TROPONIN T, FAST SKELETAL MUSCLE ISO...         70  0.0042     3
TROPONIN T, CARDIAC MUSCLE ISOFORM (...         69  0.0046     3
 (Y08484) swelling dependent chloride...        66  0.0047     2
 (U67262) RING-finger protein [Helico...        78  0.0065     1
CONVICILIN PRECURSOR                            81  0.0070     2
 (U67261) RING-finger protein [Helico...        77  0.0090     1
 (U67259) RING-finger protein [Helico...        76  0.012      1
 (U82273) FGF homolog [Drosophila mel...        76  0.014      1
```

Not too strong

Almost as strong

We feel better, if the proteins are provided with class labels (not individual names) and the top hits are from the same class….
→ Class-annotated database

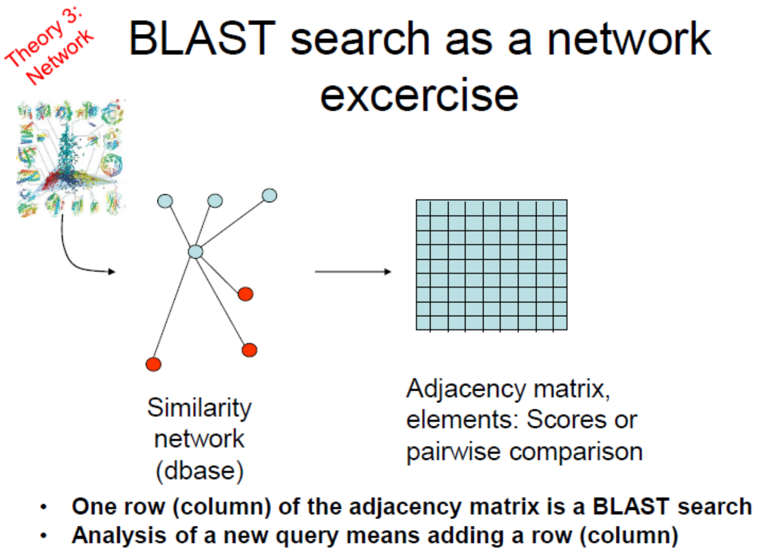
54

motif

Trivial and non-trivial problems.

Lásd előző pont

The protein universe (as a BLAST similarity network).



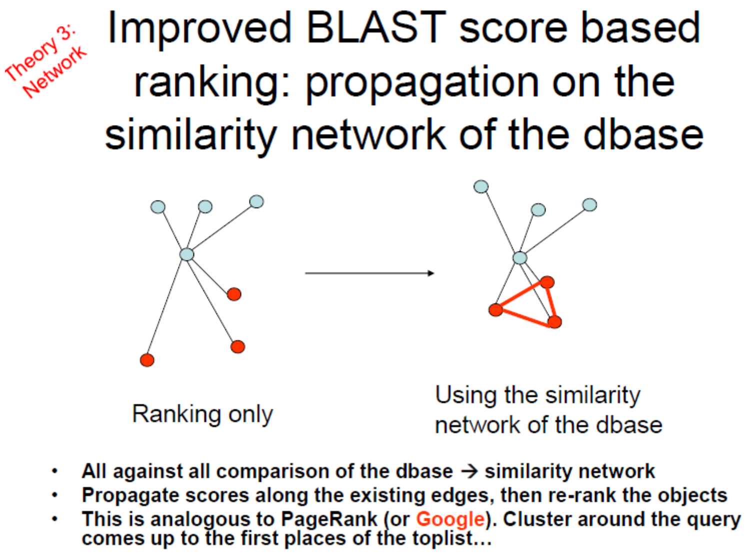
Theory 3: Network

BLAST search as a network excercise

Similarity network (dbase)

Adjacency matrix, elements: Scores or pairwise comparison

- **One row (column) of the adjacency matrix is a BLAST search**
- **Analysis of a new query means adding a row (column)**



Theory 3: Network

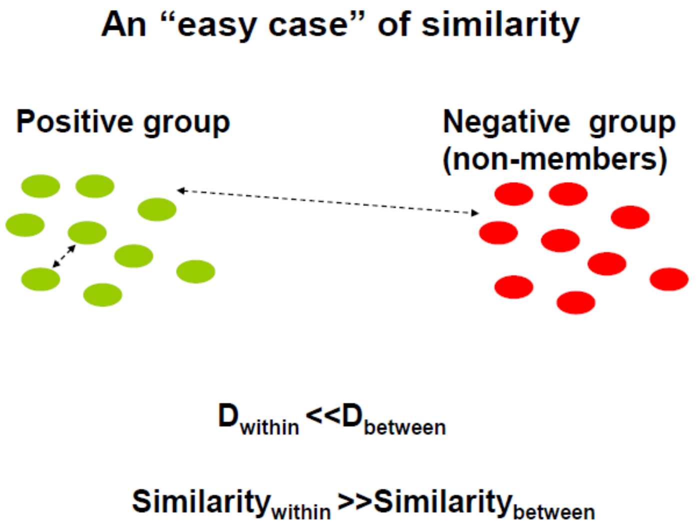
Improved BLAST score based ranking: propagation on the similarity network of the dbase

Ranking only

Using the similarity network of the dbase

- All against all comparison of the dbase → similarity network
- Propagate scores along the existing edges, then re-rank the objects
- This is analogous to PageRank (or Google). Cluster around the query comes up to the first places of the toplist…

Protein groups (easy-difficult, large-small, tight-loose, etc.).



An "easy case" of similarity

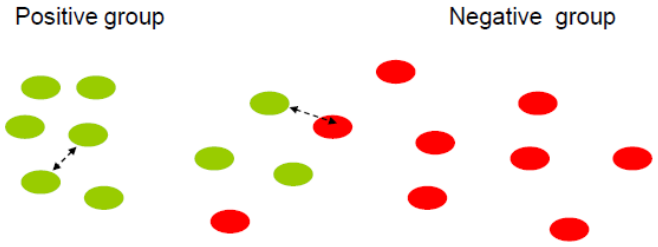
Positive group

Negative group (non-members)

$D_{within} << D_{between}$

$\text{Similarity}_{within} >> \text{Similarity}_{between}$

A “difficult case” of similarity



$\mathbf{D_{within} \sim D_{between}}$

$\mathbf{Similarity_{within} \sim Similarity_{between}}$

The neighborhoods of positive members include negative members…

8. Philogenetics

General steps:

- Data selection
- Aligning sequences
- Generating trees
- Answer scientific question

Sequence could be either nucleotide or protein:

Nucleotide seq.: mutations happen here; Some methods are specific for nucleotide sequences

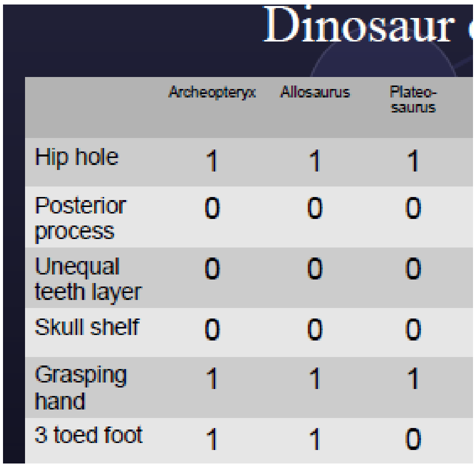
Protein seq.: Easier to align; Some methods are specific for this; In case of Frame Shift, homolgy is not meaningful; Losing info of same-sense mutations

Tree: Root node -> branches -> internal nodes -> branches -> terminal nodes

Algorithms for generating trees

Parsimony:

Popular method, easy to understand, deals with characteristics present at a species (called characters just because…). Character: feature; Character state: present or not (1 or 0)



Dinosaur

	Archeopteryx	Allosaurus	Plateo-saurus
Hip hole	1	1	1
Posterior process	0	0	0
Unequal teeth layer	0	0	0
Skull shelf	0	0	0
Grasping hand	1	1	1
3 toed foot	1	1	0

Characters should be *unique* and have *evolutionary significance* (its absence means evolutanary divergence before the emergence of the character)

Homoplasy*:* similarity that is not homologous, the result of independent evolution. Doesn’t convey evolutionary relation (see convergent evolution).

The goal of parsimony is to minimize homoplasy and maximize congruence (correspondence in character)

Parsimonious trees provide the shortest tree. These trees are not guaranteed to be 100% valid. Different characters can be weighted differently. We can construct multiple trees (and score them based on previous studies) to decude the truest hypothesis.

Parsimony advantages:

- Simple
- Independent of evolutionary model
- Yields trees and hypotheses of character evolution
- Reliable if data is well structured and low chance of homoplasy

Parsimony disadvantages:

- Misleading in case of concentrated homoplasy
- Underestimates branch lengths
- Long branch attraction
- Often purely philosophical

Nucleotide substitution models

Differences between models:

- Nucleotide fequencies (1/4 or measured from data or estimated from models)
- Mutation rates (uniform or transitions/transversions or constant/changing in time)

Jukes-Cantor model: ¼ nucl. Freq. ; single mutation rate (a constant number)

Used to calculate pairwise distances -> distance matrices.
Different methods to create trees from distance matrices.

Neighbor joining

Constructs tree step-wise. In each step, it joins two nodes (taxa) and calculates the sum of branch length, the one with the smallest sum is chosen. Produces an unrooted tree, quick and good guess of true phylogeny.

Model based phylogenics

Maximum likelihood: Probability of seeing the observed data (D) given a model/theory (T): P(D|T) Based on the assesment of probabilities of particular mutations. (Tree that require more mutations, is less probable). Useful if we have few nucleotide sequences.

Bayesian inference: Probability that the model/theory is correct given the observed data. P(T|D)

Probabilty distribution of possible trees that could be generated. Useful if we have more sequnces.

Rooting trees

Outgroup method. A sequence outside of our focus of interest (outgroup) is aligned to the sequences from the ingroup.

Data re-sampling

Generates several sub-samples – replications -> Calculates trees for all the sub-samples -> Generate consensus trees

9. NGS, genome assembly

Parts of a genome sequencing project

•DNA sequencing

•Assembly

•Gene prediction/gene annotation

•Data deposition

Traditional (Sanger) sequencing	**Next generation sequencing**
–Accurate	–Less accurate – sometimes much less.
–Also works on few samples	–Shorter reads (in general)
–Expensive for data	–Economical only with many samples
–Small capital investment	–~1000 less expensive for data
–Slow	–Large capital investment
	–Very fast - Uses graph-based methods

Sequence assembly:

1. Find potentially overlapping reads (fragments)
2. Merging reads into larger fragments (contigs)
3. Derive DNA sequence and correct read errors

Assembling genomes (or contigs) from reads is special problem composed of laboratory and computing tricks.

•Sequencing strategiew differ in the length and the accuracy of the reads.

•Early assembly solutions rely on accurate long reads (Sanger), exhaustive comparison (Smith Waterman or similar), and a jigsaw puzzle like assembly.

•Current solutions rely on large numbers of highly redundant and error-laden short reads (NGS)as well as network representations (De Bruijn graphs, overlap graphs) that avoid the need for direct comparisons such as SW.

10. Genome annotation, gene finding

Genome assembly

- The price of sequencing decreases
- The consequence of this decrease is a rush for sequencing new genomes
- However the current assembly methods only provide unfinished draft genomes i.e. pieces in which the genes are not all identified or annotated

Genome annotation

Genome annotation is the process of attaching biological information to sequences.

It consists of two main steps:

1. identifying elements in the genome, a process called gene prediction (syntax)
2. attaching biological information to these elements (semantics)

Automatic annotation tools try to perform all this by computer analysis, as opposed to manual annotation (a.k.a. curation) which involves human expertise. Ideally, these approaches co-exist and complement each other in the same annotation pipeline.

The basic level of annotation is using BLAST for finding similarities, and then annotating genomes based on that. However, nowadays more and more additional information is added to the annotation platform. Other databases (e.g., ENSEMBL) rely on both curated data sources as well as a range of different software tools in their automated genome annotation pipeline.

Structural annotation consists of the identification of genomic elements. (SYNTAX, using only the sequence)

- ORFs (open reading frames) and their localization
- gene structure (intron-exon)
- coding regions
- location of regulatory motifs

Functional annotation consists of attaching biological information to genomic elements. (SEMANTICS, using dbases)

- biochemical function
- biological function
- regulation and interactions involved
- expression
- variants

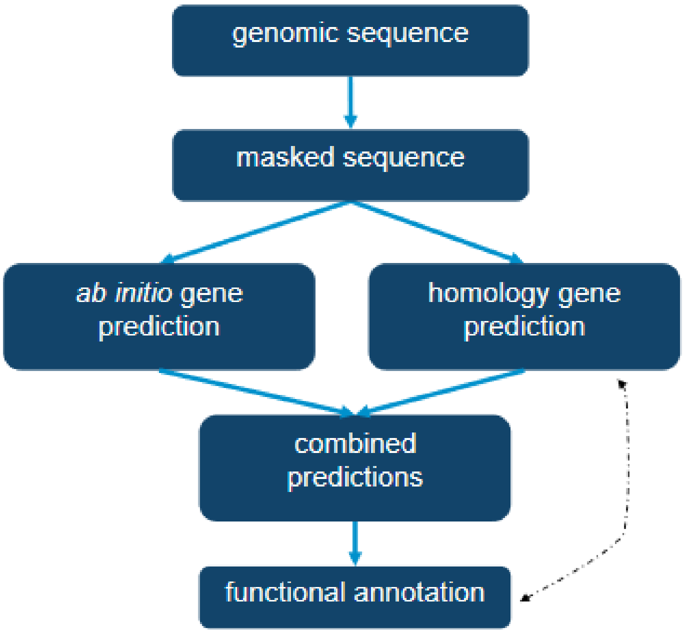
These steps may involve both biological experiments and in silico analysis.

Step by step

Mask repetitive regions (e.g., repeat masker)

Find genes ab initio and if possible by homology (many tools)

Combine both predictions (e.g., MAKER)



Repeats and low complexity regions are bad…

Repeats may confuse ab initio gene finders

- they may call exons or even complete genes in repeat regions
- they may fragment gene predictions

Repeats may confound sequence alignment

- especially in searches for synteny or segmental duplications

Some repetitive elements are found in the human genome

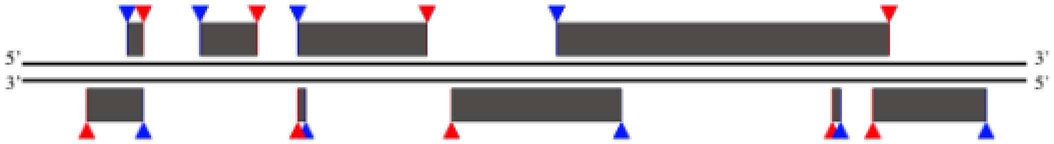
- These repeated elements should be masked (replaced by "N"s)
- Repeat Masker can do it for you
- Some prokaryotes (e.g. bacteria) have less repeats

What is gene finding (or gene prediction)?

- From a genomic DNA sequence we want to predict the regions that will encode for a protein: *the coding genes*
- Gene finding is about detecting these coding regions and infer the gene structure starting from genomic DNA sequences
- We need to distinguish coding from non-coding regions using properties specific to each type of DNA region
- Gene finding is not an easy task!
  - DNA sequence signals have low information content (small alphabet and short sequences);
  - It is difficult to discriminate real signals from noise (degenerated and highly unspecific signals);
  - Gene structure can be complex (sparse exons, alternative splicing, ...);
  - DNA signals may vary in different organisms;
  - Sequencing errors (draft genome, frame shifts, …);

Gene finding in prokaryotes (the simpler case)

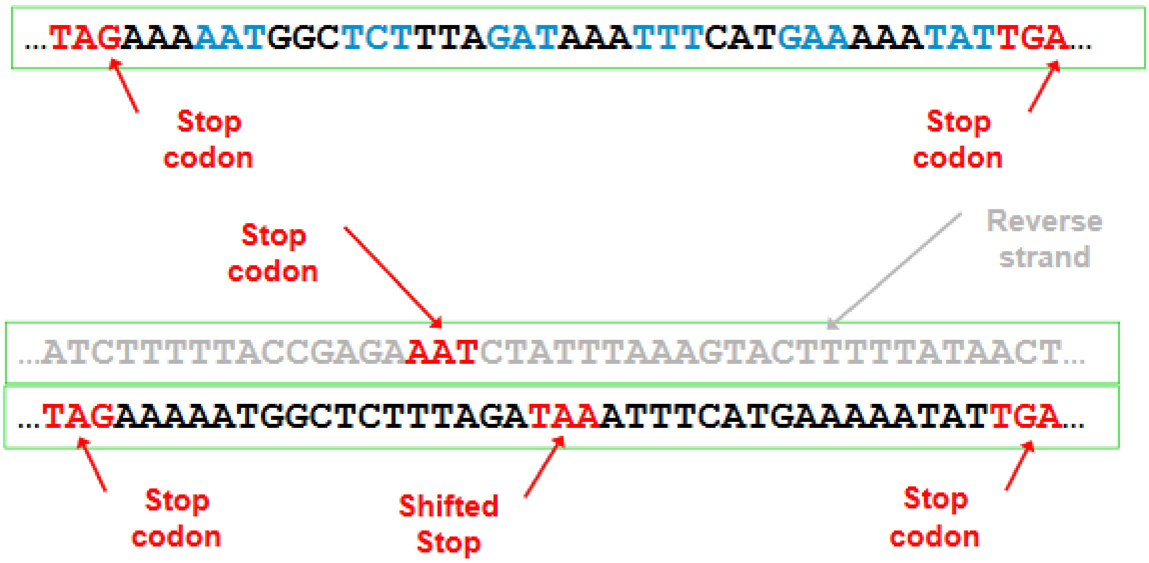
- High gene density and simple gene structure (mostly ORF)
- Short genes have little information
- Overlapping genes



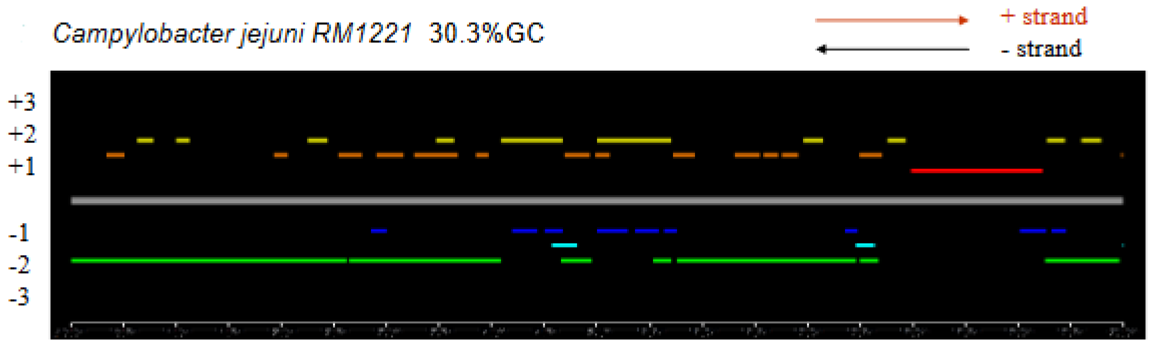
- Syntax is simpler than in eukaryotes
- Example of tools suitable for gene prediction in prokaryotes:
  - Glimmer 3: http://cbcb.umd.edu/software/glimmer/
  - GeneMark: http://opal.biology.gatech.edu/GeneMark/
  - MED2.0: http://ctb.pku.edu.cn/main/SheGroup/Software/MED2.htm
  - many more...
- We will mostly use GLIMMER to illustrate the principles

Steps

- Find open reading frames (ORFs)
  - We use a presumed syntax, the codon table


...TAGAAAAATGGCTCTTTAGATAAATTTCATGAAAAATATTGA...
Stop codon
Stop codon
Stop codon
Reverse strand
...ATCTTTTTACCGAGAAATCTATTTAAAGTACTTTTTATAACT...
...TAGAAAAATGGCTCTTTAGATAAATTTCATGAAAAATATTGA...
Stop codon
Shifted Stop
Stop codon

- » Translation programs give you 6 reading frames
- » But ORFs generally overlap…

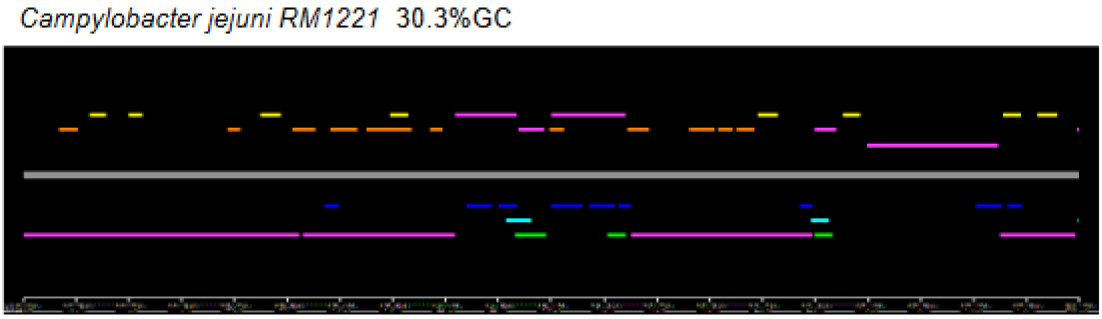


All ORFs on both strands shown

- » color indicates reading frame

Longest ORFs likely to be protein-coding genes

We get this picture at normal or low GC content bacterial genomes



Purple ORFs show annotated ("true") genes

The Problem

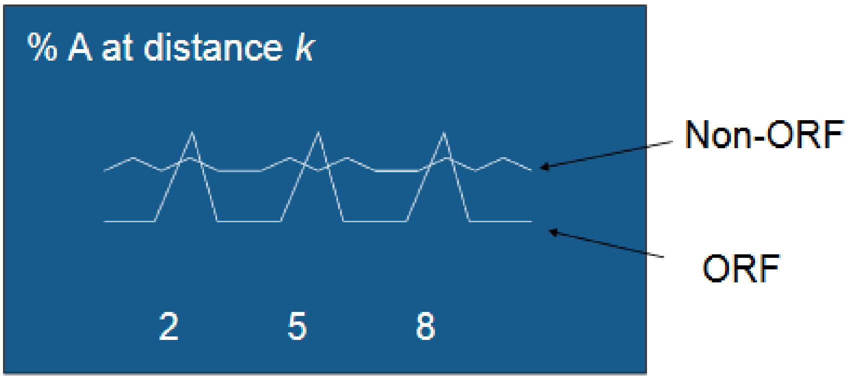
- » Need to decide which ORFs are genes
  - ▪ Then figure out the coding start sites
- » Can do homology searches but that won't find novel genes

- Besides, there are errors in the databases

» Generally can assume that there are some known genes to use as training set

- Or just find the obvious ones

Codon composition

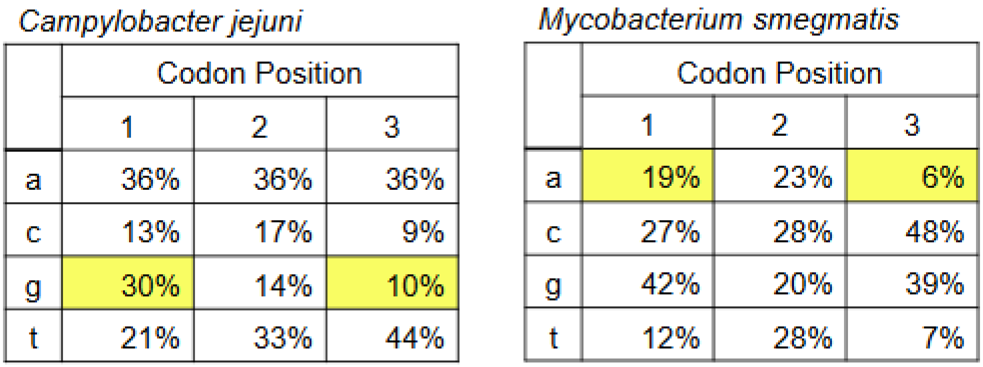


» Early programs used this method

» Codons frequently have A in the third position...

» It is easy to write periodicity recording programs...

Nucleotide variation at codon positions 1-3



Campylobacter jejuni

	Codon Position		
	1	2	3
a	36%	36%	36%
c	13%	17%	9%
g	30%	14%	10%
t	21%	33%	44%

Mycobacterium smegmatis

	Codon Position		
	1	2	3
a	19%	23%	6%
c	27%	28%	48%
g	42%	20%	39%
t	12%	28%	7%

Some bacteria have different compositions but the periodicity may still be there....

Codon-Composition Gene Finders

» ZCURVE

- Guo, Ou & Zhang, NAR 31, 2003
- Based on nucleotide and di-nucleotide frequency in codons
- Uses Z-transform and Fisher linear discriminant

» MED

- Ouyang, Zhu, Wang & She, JBCB 2(2) 2004
- Based on amino-acid frequencies
- Uses nearest-neighbor classification on entropies

» In practice:

- Predict ORFs first, then characterize them by periodicity, length, etc...
- Predict high periodicity regions first and then look for start and stop codons within/around them

Probabilistic Methods

- Create models that have a probability of generating any given sequence. Current programs use Markov style (HMM-like) models.
- Train the models using examples of the types of desired sequence types (like ORF, non-ORF).
- The “score” of an ORF is the probability of the model generating it.
  - Can also use a negative model (i.e., a model of non-ORFs) and make the score be the ratio of the probabilities (i.e., the odds) of the two models.
  - Generally, use logs to avoid underflow

Quick recap of Markov models

- If we know the frequencies of A,C,G,T and the frequencies of all dinucleotides in a genome sequence, we can predict the probability of T following G as

$$p(T|G) = \frac{f(TG)}{f(G)}$$

  where f(GT) is the frequency of GT within the genome.
- This is a first order Markov model. Given a sequence, we can calculate the probability of the sequence by multiplying the probabilities of all dinucleotides. For instance, the probability score of AGGT will be

$$p(AGGT) = p(G|A) \times p(G|G) \times p(T|G)$$

- In practice we use logs, in order to avoid underflow.
- We can easily extend first order Markov models to second order to calculate the probability of say „T” following not only „A” but „AG”:

$$p(T|AG) = \frac{f(AGT)}{f(AG)}$$

- For second order models we need to know the dinucleotide and trinucleotide frequencies. For kth order models we need up to k+1 mer frequencies.
- Markov models allow us to predict the presence of a character (“state”) at position i. DNA is model with 4 observable states.
- We can assign further (“hidden”) states to the sequence, like gene and non-gene. We can calculate separate Markov models for the two hidden states. These are the Hidden Markov Models…

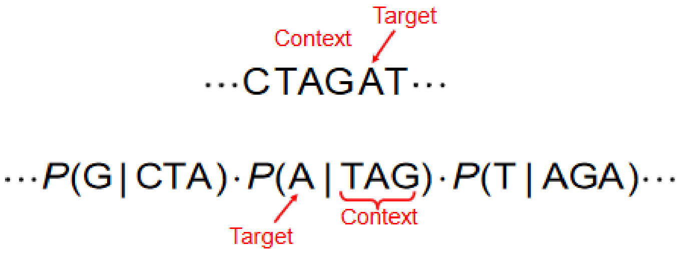
Say we have one model for “genes” and one for “non-genes”.

- The theoretically correct method is to travel along the sequence from left (5’) to right (3’) and calculate the probability of both models at every position, from the nucleotide found at that position. This is a simple dynamic programming procedure, called the Viterbi algorithm.
  - Advantage: exhaustive
  - Disadvantage: compute intensive
- A simpler hack is to cut the sequence into ORFs in advance and score them according to the two models. Then accept those ORFs that are long enough and score higher with the model of “genes”.
  - Advantage: very fast
  - Disadvantage: fails at sequencing errors, so you need perfect data

Fixed-order Markov models

- Early methods used this model
- kth-order Markov model bases the probability of an event on the preceding k events

- Example: With a 3rd-order model the probability of this sequence:

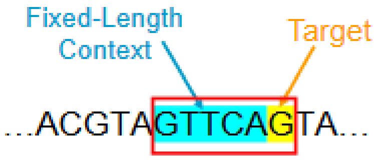


Advantages:

- Easy to train. Count frequencies of (k+1)-mers in training data.
- Easy to compute probability of sequence.

Disadvantages:

- Many (k+1)-mers may be undersampled in training data.
- Models data as fixed-length chunks.



GeneMark

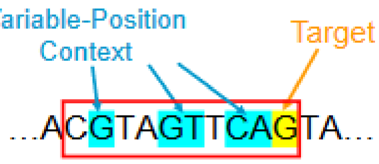
- Borodovsky & McIninch, Comp. Chem 17, 1993.
- Uses 5th-order Markov model.
- Model is 3-periodic, i.e., a separate model for each nucleotide position in the codon.
- DNA region gets 7 scores: 6 reading frames & non-coding — high score wins.
- Lukashin & Borodovsky, Nucl. Acids Res.26, 1998 is the HMM version.

Multiple Markov Models can help if there are not enough data

- Simply put, say we have enough trinucleotides, but the number of tetranucleotides is not sufficient to have a good estimate of the probabilities. So we will mix 4th order and 5th order Markov models.
- E.g., for ggttax the probability of x might depend on previous 3 bases tta. But for context cattax all 5 bases might be used.
- Glimmer 1.0 introduced this as "interpolated Markov models", IMMs, which are a weighted average of the two largest values with statistical support.
    - Salzberg, Delcher, Kasif & White, NAR26, 1998

Further improvements possible by using non-contiguous contexts

- In a classical Markov first, second, etc. models, the probability in position is calculated from the previous, one, two etc. adjacent positions. (Time series analogy)
- But with sequences we can use non-neighboring positions as well....
- So choose set of positions that are most informative about the target position (have the best statistical support)



Microbial gene prediction by GLIMMER

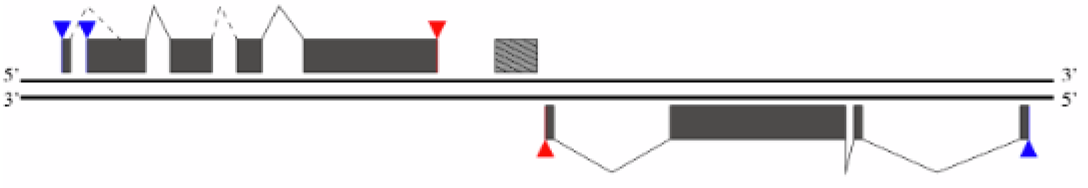
- Current versions of GLIMMER build separate models for all 6 reading frames, both for „genes” and for „non-genes”. These are available as precomputed files, but one might compute them for one’s favorite genome or genomes.
- GLIMMER first identifies all ORFs (hack) then scores them with all models and choses those ORFs that score the highest with the „gene” model. Separate heuristics are used for sorting out overlapping genes.
- Practically 100% accuracy for microbial genes. Used by NCBI for genome annotation.
- Fails is the ORFs are incorrect (i.e. broken by point mutations, indels, etc...) This is not a problem with high coverage sequencing efforts.

Functional prediction of microbial genomes is peanuts...

- We have high quality ORFs that can be translated to protein sequences.
- Protein sequences can be compared with well-annotated protein databases that gives you a function if there is very high similarity (identity>90%. E-value <10^{-4}).
- The rest can be compared to functional database like COGs, or descriptions of structural groups like PFAM.
- Still, about 30% of microbial genomes remain unannotated.
- Lesson: Peanuts are not worth a lot, after all...

Gene finding in eukaryotes

- Low gene density and complex gene structure
- Alternative splicing
- Alternative start and stop
- Pseudo-genes



- Gene-finding is similar, but more complicated than in prokaryotes

Gene finding strategies: ab initio methods

- Based on statistical signals within the DNA:
  - Signals: short DNA motifs (promoters, start/stop codons, splice sites, ...)
  - Coding statistics: nucleotide compositional bias in coding and non-coding regions
- Strengths:

    - easy to run and fast execution time
    - only require the DNA sequence as input
- Weaknesses:
    - prior knowledge is required (training sets)
    - high number of mispredicted gene structures

Methods for signal detection

- Detect short DNA motifs (promoters, start/stop codons, splice sites, intron branching point, …).
- A number of methods are used for signal detection:
    - Consensus string: based on most frequently observed residues at a given position.
    - Pattern recognition: flexible consensus strings.
    - Weight matrices: based on observed frequencies of residues at a given position. Uses standard alignment algorithms.
    - Weight array matrices: weight matrices based on dinucleotides frequencies. Takes into account the non-independence of adjacent positions in the sites.
    - Maximal dependence decomposition (MDD): MDD generates a model which captures significant dependencies between non-adjacent as well adjacent positions, starting from an aligned set of signals. (like GLIMMER)
    - Hidden Markov Models (HMMs): HMMs use a probabilistic framework to infer the probability that a sequence correspond to a real signal.
    - Neural Networks (NNs): NNs are trained with positive and negative examples. NNs “discover” the features that distinguish the two sets.

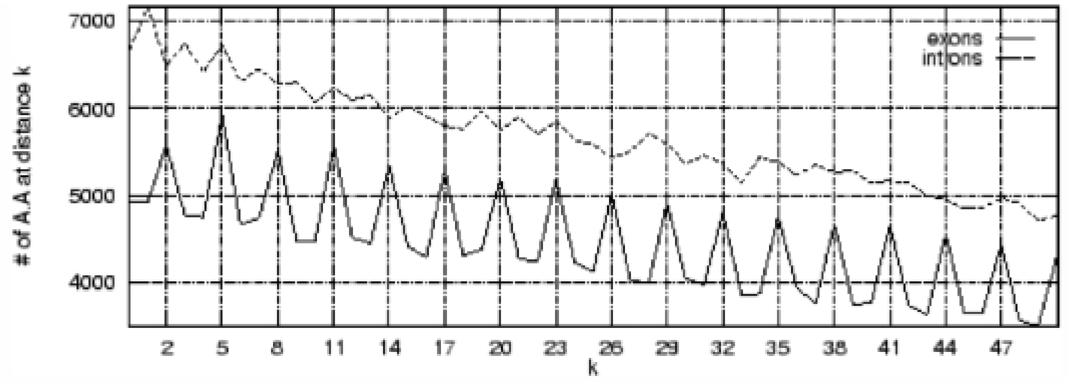
Signal detection limitations

- Problems with signal detection:
    - DNA sequence signals have low information content.
    - Signals are highly unspecific and degenerated.
    - Difficult to distinguish between true and false positive.
- How to improve signal detection:
    - Take context into consideration (ex. acceptor site must be flanked by an intron and an exon).
    - Combine with coding statistics (compositional bias).

Types of coding statistics

- Inter-genic regions, introns, and exons have different nucleotides contents.
- This compositional differences can be used to infer gene structure.
- Examples of coding statistics:
    - ORF length:
        - Assuming a uniform random distribution, stop codons are present every 64/3 codons (≈ 21 codons) in average.
        - In coding regions stop codon average decrease.
        - This measure is sensitive to frame shift errors.
        - Can’t detect short coding regions.
    - Bias in nucleotide content in coding regions:
        - Generally coding regions are G+C rich.

        - There are exceptions! For example coding regions of P. falciparum are A+T rich.
    - Periodicity: The number of residues separating a pair of adenines (A) shows a periodicity in coding regions, but not in non-coding regions. This arise because of the asymmetry in base composition at the third codon position (3rd codon position: 90% are A/T; 10% are G/C).



Coding statistics: codon frequencies

- In practice we use these computations in a search algorithm with a sliding window:
    - Select a window of size n (for example n=30).
    - Slide the window along the sequence and calculate Pi for each start position of the window.
- A variation of the codon frequency method is to use 6-tuple frequencies (n=6) instead of 3-tuple (codon) frequencies. This method was found to be the best single property to predict whether a region of vertebrate genomic sequence was coding or non-coding (Claverie and Bougueleret, 1986).
- The usage of hexamers frequencies has been integrated in a number of gene predictors.

Integrating signal and compositional information for gene structure prediction

- A number of methods exists for gene structure prediction which integrate different techniques to detect signals (splicing sites, promoters, etc.) and coding statistics.
- All these methods are classifiers based on machine learning theory.
- Training sets are required to train the algorithms.

Ab initio methods classification and examples

- Generalized HMM (Hidden Markov Models)
    - GenScan
    - HMMgene
    - Augustus
    - SNAP
- Linear and Quadratic discriminant analysis
    - FGENES and derived versions
    - MZEF
- Decision trees
    - MORGAN

- Neural Networks (NN)
  - GRAIL
- Support Vector Machines (SVM)
  - CONTRAST
  - mGene

Ab initio methods: HMMgene

- Designed to predict complete gene structures
- Uses HMMs with a criterion called Conditional Maximum Likelihood which maximize the probability of correct predictions
- Can return sub-optimal prediction to help identifying alternative splicing
- Regions of the sequence can be locked as coding and non-coding by the user
- Web server: http://genome.cbs.dtu.dk/services/HMMgene
- Training sets: human and worm

Gene finding strategies: homology methods:

- Gene structure is deduced using known homologous sequences (RNAseq, EST, mRNA, protein).
- Very accurate if there are homologous genes with high sequence similarity.
- Strengths:
  - Accurate
- Weaknesses:
  - need of good homologous sequences
  - execution is slow

Gene prediction limits

- Existing predictors are designed for protein coding regions
  - Non-coding areas are not detected (5’ and 3’ UTR)
  - Non-coding RNA genes are missed
- Predictions are for ”typical” genes
  - Partial genes are often missed (beware of draft genomes)
  - Training sets may be biased
  - Atypical genes use other grammars
- The best predictor is highly dependent on the genome (e.g., nGASP results are valid for C. elegans)

Tools for ncRNA

- Genes are not only protein coding...
- Other kinds of RNA
  - ribosomal RNA
  - transfer RNA
  - small nucleolar RNA (snoRNAbase)
  - telomerase RNA
  - miRNA, siRNA (miRBase)
- Blastn 5S,16S, 23S, ...
- Infernal (Rfam database) (HMMs)

- tRNAscan-SE

11. Metagenomics

Metagenomics is the analysis of samples - usually environmental or gut microflora samples - with many thousand species

Metagenomics deals with samples taken directly from the environment.
example: Soil, water, hot spring, oil sands, human gut, stool.
Also called environmental genomics.

Necessarily more complex than genomics:
Mixture of multiple organisms;
Many have never been looked at on the molecular level at all.

Traditional approach uses one reference gene, 16S rRNA, amplified by PCR and NGS sequenced. Bacterial composition is obtained.

Simple analysis using a reference gene 16SrRNA.
Amplification of reference gene with general primers.
Overall microbial community composition (presence-absence).

16S rRNA sequencing:
Early and still common method. Highly conserved yet unique to individual (mostly bacterial) species. Consists of variable and conserved regions. Targeted sequencing using primer pairs.

In whole genome sequencing (WGS) the reads can be mapped to annotated genome sequences, bacterial composition and biological functions are obtained.

Metagenomic sequencing:

- <u>Whole genome sequencing (WGS):</u>
  Sequencing + mapping to known genomes (or to specific marker database)
- Overall microbial community composition (quantitative)
- Dominant functions

- It can also be done with assembly

Whole metagenome with „binning”: Map genes to (annotated) genomic sequences. Count hits by taxa („bins”) – gives taxonomic composition incl. Quantitation of taxa. You can use marker database instead of full genomes. Faster but less sensitive.

Alternatively, WGS reads can be assembled at large computational overheads which makes analysis more accurate.

Whole metagenome with assembly: Assemble reads just like in genome assembly. It is complicated because of multiple unknown sources of metagenomic reads, lower coverage on individual genomes. Requires large computers.

Main programs: MG RAST, Megan, Mothur

Metagenomic Tools (Local):

Qiime (Quantitative Insights Into Microbial Ecology):

Consists of Python scripts. It is used for taxonomy and diversity statistics/visualization

Mothur:

Commands written in C++ programs. Taxonomy and diversity statics/visualization.

MEGAN (Metagenome Analyzer) :

Provides functional as well as taxonomy analysis. GUI with tree-based visualization.

Metagenomic Analysis Related Sites:

MG-RAST (Metagenomic Analysis Server), IMG (Integrated Microbial Genomes), SILVA and GreenGenes (Ribosomal RNA Collection)

12. Clinical utility of sequence-based diagnostics

DNA to sequencing

Steps of preparing DNA for NGS analysis:

1. Sample acquisition: saliva, blood
2. Breaking up the cells: liquid nitrogen, shredder, beads
3. Isolation of DNA
4. Quality control: UV-spectrophotometer
5. DNA fragmentation: physical, chemical, enzymatic

A sequence variation can be a single nucleotide polymorphism (SNP, 1%< in population) or mutation.

Mutation to disease: factor V Leiden

Blood clotting is regulated by multiple pathways, in which Factor V has a critical role.
Mutation in Factor V (=Leiden) leads to increased risk of thrombotic events.
Factor V Leiden can be detected by snake venom, PCR-RFLP and sequencing.

Linking mutations to disease: BRCA I / II

Cancer is a genetic disease originating in a single cell.
BRCA 1⁄2 are tumor suppressor genes (= loss of function).
Germ line mutation in BRCA 1⁄2 leads to early breast cancer.
„A gene is not patent eligible merely because it has been isolated."
Cumulative lifetime risk is not useful for individual decision-making.

Mutation as guide for treatment: KRAS (a prognostic marker)

Oncogene-addiction: tumors depend on a single signal transduction pathway.

The ERBB/KRAS signal transduction pathway has a critical role in solid tumors.

Patients with a KRAS mutation do not respond to a therapy targeting a higher member in the pathway (e.g. panitumumab - EGFR).

Methods for KRAS mutation status detection are either PCR or sequencing based.

Future of NGS

Challenges for NGS: accuracy, interpretation, storage

Recommended: targeted screening

Germ line testing of high risk patients is expanding in targeted panels (e.g. BRCA)

Somatic analysis of tumors will increasingly guide and help to individualize cancer treatment (e.g. KRAS)

Not recommended: NGS for everyone

Risk of discovering untreatable diseases

False positives cause harm

13. Functional Genomics

Genomics vs. functional genomics

Genomics

Sequenation, analyzation and annotation of the whole DNA (nuclear, mitochondrial and chloroplast DNA as well)

Functional genomics

Examination of genes or group of genes with high throughput methods which are designed based on the genome sequence (microarray, SNP) or uses sequenation (ChIP-seq, RNA-seq, etc.). Functional genomics is looking for the connections between genotype and phenotype at the level of the genome.

Microarray vs. NGS based methods

Microarrays are generally considered easier to use with less complicated and less labor-intensive sample preparation than NGS. Microarray technique also require some prior knowledge of the genome. Microarray technique is also more accurate (nowadays).

X-seq techniques

In the above methods the DNA or the RNA is broken into small parts and from these parts 'tags' are sequened. These tags are the aligned with the reference genome.

RNA-seq

Examination of mRNA-s at the level of the genome

Genomic mapping

Map a gene in the genome.

Transcriptome mapping

Questions are the exact location of the promoters, position of the Transcription Start Site and the Transcription Factor Binding Site. The most important question is where, when, why and how many mRNA is produced by which gene? And how it is coded in the DNA?

De Novo assembly

De novo transcriptome assembly is the method of creating a transcriptome without the aid of a reference genome.

Gene expression analysis

RNA-seq can be used to determine the expression profile (abundance measurement of specific transcripts)

ChIP-seq

Examination of DNA-protein interactions at the level of the genome

GRO-seq

Examination of transcription at the level of the genome