Master Course

Algorithms in Sequence Alignment

Lecture 8

Homology searching (2)

Content

- PHI-BLAST method
- Profile-profile homology searching
- Statistical scoring the hits
  - Scrambling sequences and Z-score
  - Extreme Value Distribution and E-value calculation
- Philosophical points about the meaning of homology searches
- Validating homology searching methods
- Issues when running Blast or other methods
  - Low complexity sequences

PHI-BLAST (Pattern Hit Initiated)

- Method to find database sequences based on a given sequence pattern
- Input is a sequence ***S*** and a sequence pattern ***P***
- PHI-BLAST helps answer the question: What other protein sequences both contain an occurrence of ***P*** and are homologous to ***S*** in the vicinity of the pattern occurrences?
- PHI-BLAST may be preferable to just searching for pattern occurrences because it filters out those cases where the pattern occurrence is probably random and not indicative of homology.

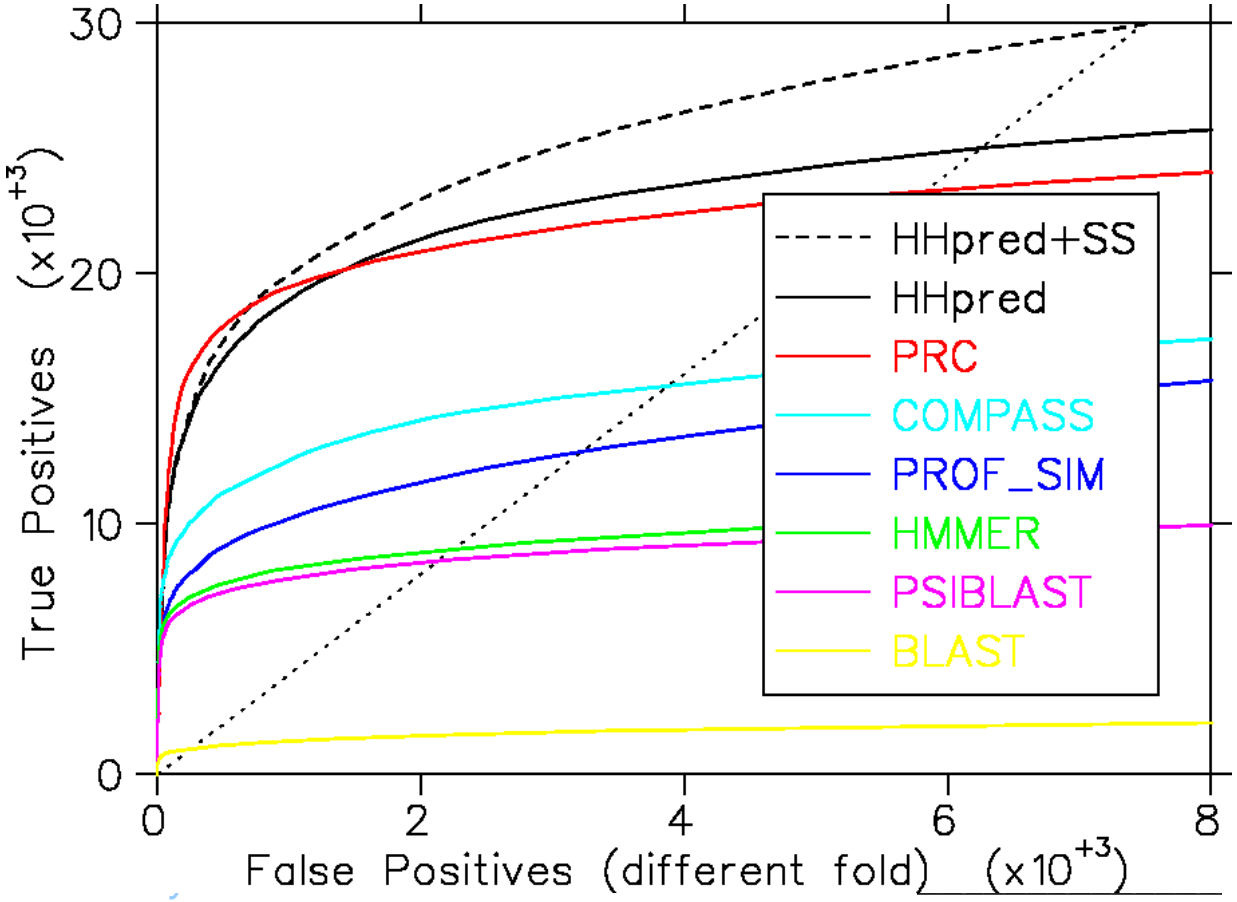
PHI-BLAST (Pattern Hit Initiated) (cont.)

- Sequence patterns are found using regular expressions (later in this lecture) in PROSITE format

  (PROSITE is a database of functional sequence motifs expressed as regular expressions or profiles)
- Pattern occurrences need not exceed threshold *T*
- Extension and alignment are the same as in Gapped (or PSI-)Blast
- Scoring scheme is stricter than (PSI-)Blast due to the requirement of pattern occurrence

State of the art is profile-profile comp.

- Profile-profile comparisons using HMM (later lectures)
- A query sequence or query MSA is aligned against family databases such as PFAM, SMART, PANTHER, TIGRFAM, PIRSF or COG/KOG, where each entry (family) is abstracted in a profile HMM.



[5] A

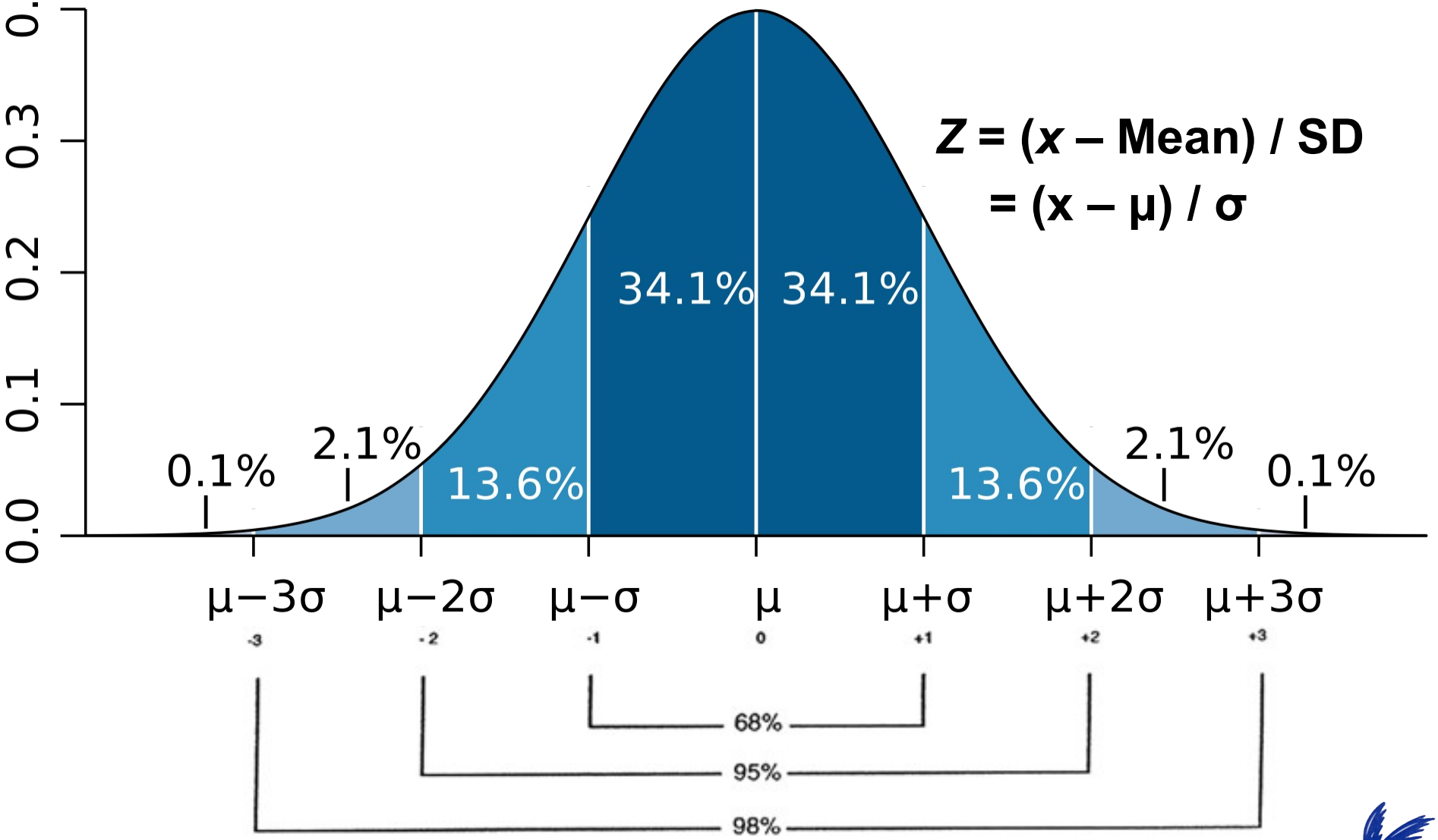
What is the statistical significance of an alignment: Z-scores

- To get a null model: extract *local* alignments from random sequences
- Use sequences in original alignment (with score *x*) to make 'random' sequences by scrambling and then align these and compare resulting scores
- Using multiple randomisations:
  - Get a series of random alignment scores, calculate Mean and SD
  - $Z = (x - \text{Mean}) / \text{SD}$ X: the alignment score
  - In practice: Z should be >4 to >6 SD (significance threshold)

What is the statistical significance of an alignment: Z-scores

- To get a null model: extract *local* alignments from random sequences
- Use sequences in original alignment (with score *x*) to make 'random' sequences by scrambling and then align these and compare resulting scores
- Using multiple randomisations: This is slow…
  - Get a series of random alignment scores, calculate Mean and SD
  - $Z = (x - \text{Mean}) / \text{SD}$
  - In practice: Z should be >4 to >6 SD (significance threshold)

What is the statistical significance of an alignment: Z-scores



$$Z = (x - \text{Mean}) / \text{SD}$$
$$= (x - \mu) / \sigma$$

Scoring BLAST alignments

- Score should optimise the chance to select proper hits (True Positives)
- Scoring alignments is dependent on
  - The scoring system used (residue exchange matrix and gap penalty regime)
  - Characteristics of the sequence database (size, residue composition)
- The BLAST way of scoring has been adopted by other methods as well; e.g., some recent implementations of FASTA, etc.
  - Bit-score
  - E-value

Alignment Bit Score

$$B = (\lambda S - \ln K) / \ln 2$$

- S is the raw alignment score
- The bit score ('bits') B has a standard set of units
- The bit score B is calculated from the number of gaps and substitutions associated with each aligned sequence. The higher the score, the more significant the alignment
- λ and K and are statistical parameters associated with a given scoring system (e.g. BLOSUM62 in Blast)
  - See Altschul and Gish (1996) for a collection of values for λ and K over a set of widely used scoring matrices.
- Because bit scores are normalized with respect to the scoring system, they can be used to compare alignment scores from different searches based on different scoring schemes (a.a. exchange matrices)

What is the statistical significance of an alignment

- Using a null model based on *local* alignments from random sequences
- P-value
  - The probability of obtaining the result by pure chance
  - An alignment giving a lower P-value than a threshold value set by the user is considered *a hit*.

Normalised sequence similarity

The p-value is defined as the probability of seeing at least one unrelated score *S* greater than or equal to a given score *x* in a database search over *n* sequences.

This probability follows the Poisson distribution (Waterman and Vingron, 1994):

$$P(x, n) = 1 - e^{-n \cdot P(S \geq x)},$$

where *n* is the number of sequences in the database

Depending on *x* and *n* (fixed)

Normalised sequence similarity

Statistical significance

The E-value is defined as the expected number of non-homologous sequences with score greater than or equal to a score x in a database of n sequences:

$$E(x, n) = n \cdot P(S \geq x)$$

For example, if E-value = 0.01, then the expected number of random hits with score $S \geq x$ is 0.01, which means that this E-value is expected by chance only once in 100 independent searches over the database.

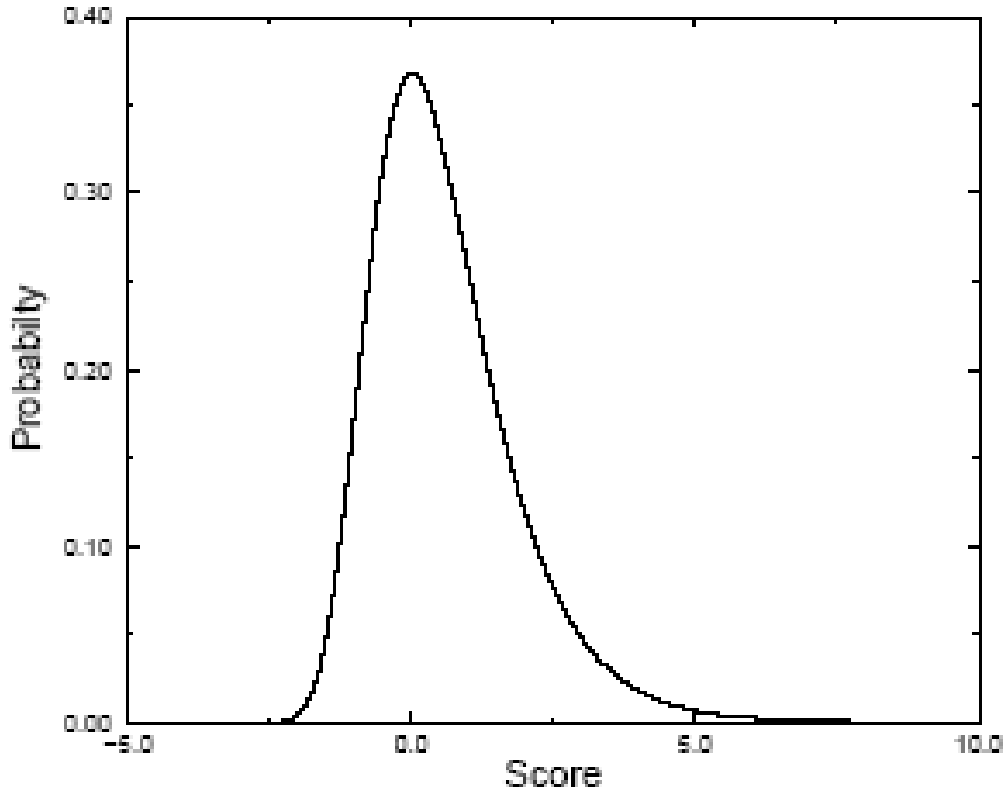
if the E-value of a hit is 5, then five fortuitous hits with $S \geq x$ are expected within a single database search, which renders the hit not significant.

- NCBI's NR database (used in BLAST) contains > 100 million sequences.

A model for database searching score probabilities

- Scores resulting from searching with a query sequence against a database follow the Extreme Value Distribution (EVD) (Gumbel, 1955).
- Using the EVD, the raw alignment scores are converted to a statistical score (E value) that keeps track of the database amino acid composition and the scoring scheme (a.a. exchange matrix)

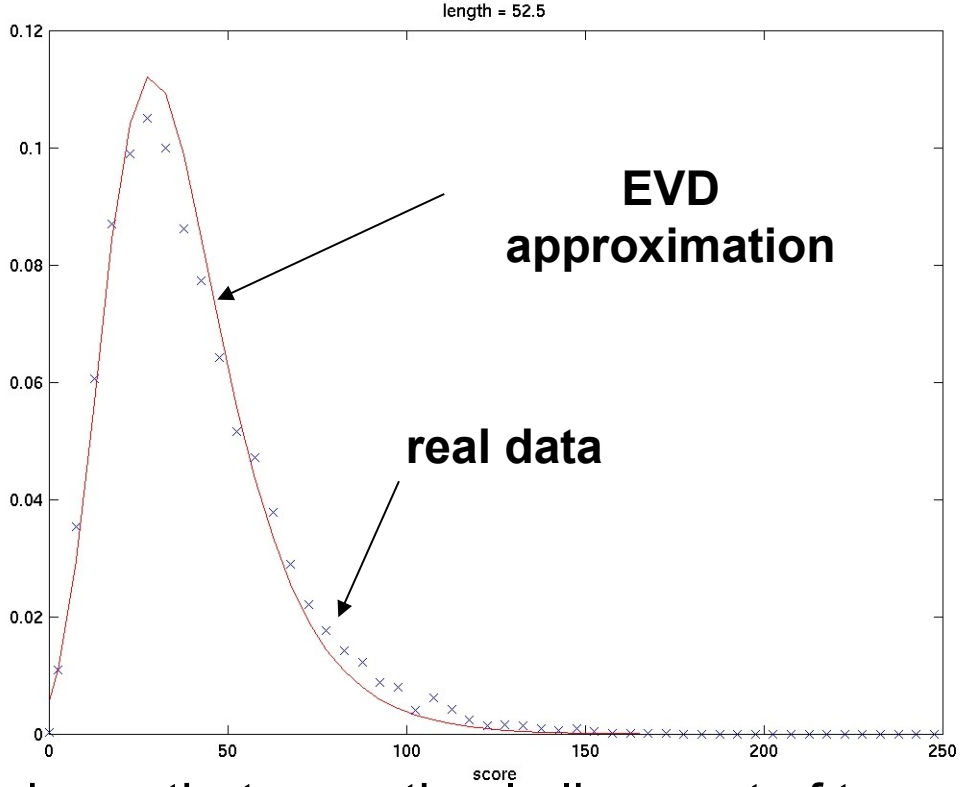
Extreme Value Distribution



$$y = 1 - \exp(-e^{-\lambda(x-\mu)})$$

Probability density function for the extreme value distribution resulting from parameter values μ = 0 and λ = 1, [$y = 1 - \exp(-e^{-x})$], where μ is the characteristic value (where the EVD peaks) and λ is the decay constant.

Extreme Value Distribution (EVD)



Not a normal (Gaussian) distribution

You know that an optimal alignment of two sequences is selected out of many suboptimal alignments, and that a database search is also about selecting the best alignment(s) out of many database sequences. This double selection bodes well with the EVD which has a right tail that falls off more slowly than the left tail. Compared to using the normal distribution, when using the EVD an alignment has to score further away from the expected mean value to become a significant hit.

Extreme Value Distribution

The probability of a **unrelated** score *S* to be larger than a given value *x* can be calculated following the EVD as:

E-value: $P(S \geq x) = 1 - \exp(-e^{-\lambda(x-\mu)})$,

where $\mu = (\ln Kmn)/\lambda$, and *K* a constant that can be estimated from the background amino acid distribution and scoring matrix (see Altschul and Gish, 1996, for a collection of values for λ and *K* over a set of widely used scoring matrices). Variables *m* and *n* are the length of the query sequence and the size of the search database, resp.

Extreme Value Distribution

Using the equation for μ (preceding slide), the probability for the raw alignment score S becomes

$$P(S \geq x) = 1 - \exp(-Kmne^{-\lambda x}).$$

In practice, the probability $P(S \geq x)$ is estimated using the approximation $1 - \exp(-e^{-x}) \propto e^{-x}$, valid for large values of x. This leads to a simplification of the equation for $P(S \geq x)$:

$$P(S \geq x) \propto e^{-\lambda(x-\mu)} = Kmne^{-\lambda x}.$$

The lower the probability (E value) for a given threshold value x, the more significant the score S.

Normalised sequence similarity

Statistical significance

- Database searching is commonly performed using an E-value in between 0.1 and 0.001.
- Lower E-value threshold settings decrease the number of false positives in a database search, but increase the number of false negatives, thereby lowering the sensitivity of the search (see later slides).

Approximating statistical significance

- Scrambling sequences allows Z-score calculations that are slow but independent of the database size and composition
- E-value calculations based upon the EVD are much faster but do depend upon the size of the database: an E-value score for a given query and DB sequence can change upon a next release of the sequence database.

What do errors mean for alignment?

- Alignments need to be able to match distantly related sequences, skip secondary structural elements to complete domains (i.e. putting gaps opposite these motifs in the shorter sequence).
- Depending on the residue exchange matrix and gap penalties chosen, the algorithm might have difficulty with aligning distant homologs or inserting long gaps (for example when using high affine gap penalties), resulting in incorrect alignment.

What do errors mean for homology searching?

- Database searching algorithms just need to decide if the alignment score is good enough for inferring homology
- Sometimes, alignments can be incorrect but the score can be close enough for the database searching method to correctly identify the DB sequence as a homolog (or not)
- However, for more distant hits alignment becomes crucial as alignment scores are becoming more different (relatively)

How do we represent and formalise genome information

Human breast cancer susceptibility (BRCA2) mRNA, com

GenBank: U43746.1

GenBank Graphics

```
>U43746.1 Human breast cancer susceptibility (BRCA2) mRNA, complete cds
GGTGGCGCGAGCTTCTGAAACTAGGCGGCAGAGGCGGAGCCGCTGTGGCACTGCTGCGCCTCTGCTGCGC
CTCGGGTGTCTTTTGCGGCGGTGGGTCGCCGCCGGGAGAAGCGTGAGGGGACAGATTTGTGACCGGCGCG
GTTTTTGTCAGCTTACTCCGGCCAAAAAAGAACTGCACCTCTGGAGCGGACTTATTTACCAAGCATTGGA
GGAATATCGTAGGTAAAAATGCCTATTGGATCCAAAGAGAGGCCAACATTTTTTGAAATTTTTAAGACAC
GCTGCAACAAAGCAGATTTAGGACCAATAAGTCTTAATTGGTTTGAAGAACTTTCTTCAGAAGCTCCACC
CTATAATTCTGAACCTGCAGAAGAATCTGAACATAAAAACAACAATTACGAACCAAACCTATTTAAAACT
CCACAAAGGAAACCATCTTATAATCAGCTGGCTTCAACTCCAATAATATTCAAAGAGCAAGGGCTGACTC
```

DNA

```
>ENSANGP00000000001 Gene:ENSANGG00000000001 Status:novel
LDGSAVHPESYPVVERILAKLEQTVDSLLGNSNLLRTLKPADYTDEQFGVPTVTDIIGEL
DKPGRDPRPEFKTATFKEGVEKISDLVPEMVLEGVVTNVTNFGAFVDIGVHQDGLVHISS
LTDRFVKDPREVVKAGDIVRVKVLEVDVPRKRISLTMRLDEKAGQPARKPAEPRHTGNAK
```

Protein

We represent polymeric molecular structures such as nucleotide and amino acid sequences as character strings

Why do we formalise genome information?

- The cellular machinery is exceedingly complex
- The transformation of genomic information in the cell to text sequences (character strings) is a reduction in complexity
- This formalisation makes genomic information accessible and tractable

Compare this to some other formalisations:

- Carl Linnaeus (1707-1778): Systematic classification of species
- Charles Darwin (1809-1882): Evolution
- Alan Turing (1912-1954): Turing machine

Reductionism

"*Ceci n'est pas une pipe.*"
René Magritte (1998-1967)

The treachery of images

- The treachery of protein representations:

```
>ENSANGP00000000001 Gene:ENSANGG00000000001 Status:novel
LDGSAVHPESYPVVERILAKLEQTVDSLLGNSNLLRTLKPADYTDEQFGVPTVTDIIGEL
DKPGRDPRPEFKTATFKEGVEKISDLVPEMVLEGVVTNVTNFGAFVDIGVHQDGLVHISS
LTDRFVKDPREVVKAGDIVRVKVLEVDVPRKRISLTMRLDEKAGQPARKPAEPRHTGNAK
```

This is not a protein.

How to assess homology search methods

- We need an annotated database, so we know which sequences belong to what homologous (super)families
- Examples of databases of homologous families are PFAM, Homstrad or Astral
- The idea is to take a protein sequence from a given homologous family, then run the search method, and then assess how well the method has carried out the search (i.e. recognised the family members)
- This should be repeated for many query sequences and then the overall performance can be measured

HOMSTRAD

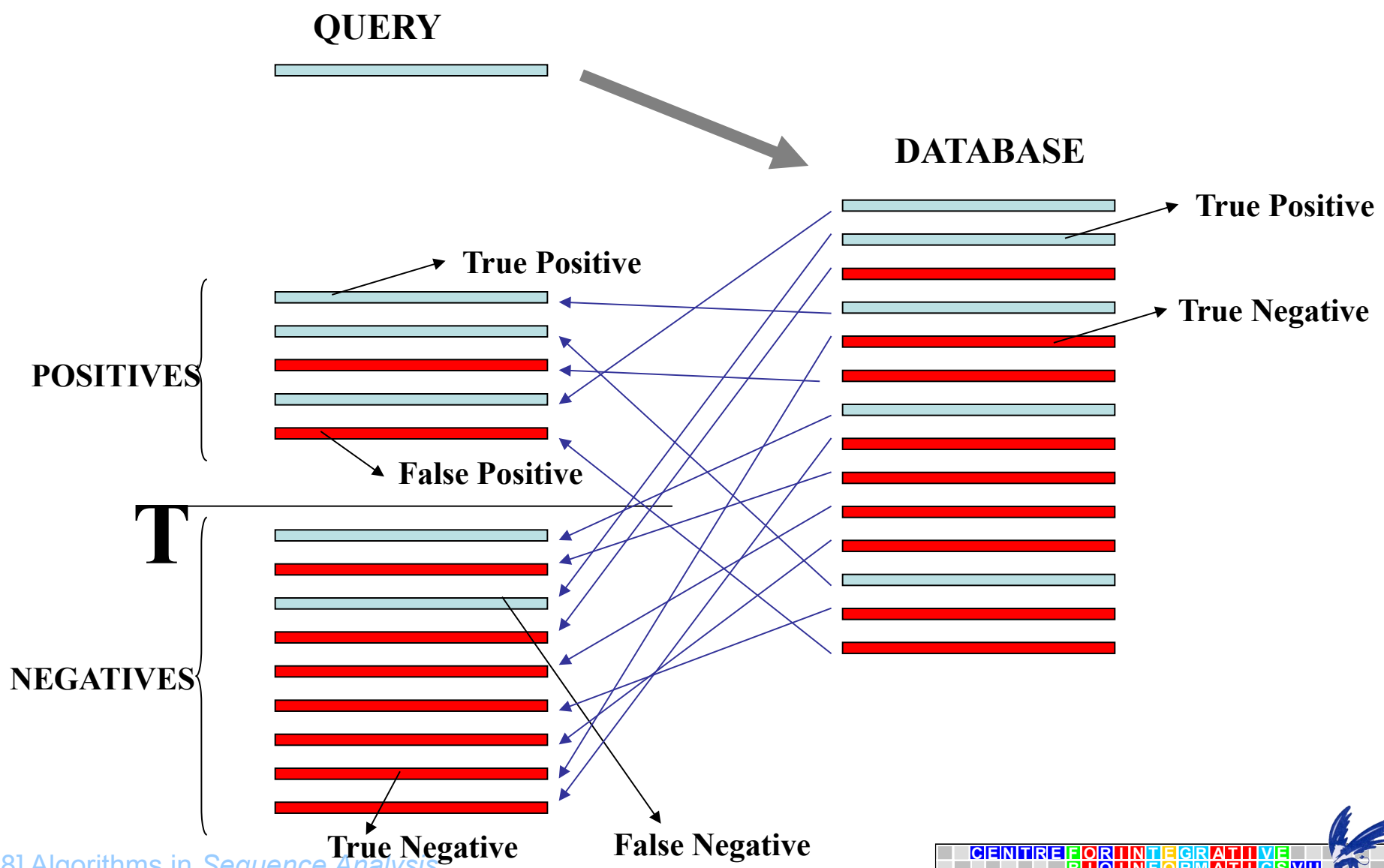
Homologous Structure Alignment Database

Example

```
C; family: zinc finger -- CCHH-type
C; class: small C; reordered by kitschorder 1.0a
C; reordered by kitschorder 1.0a
C; last update 7/9/98
>P1;1zaa1 structureX:1zaa: 3 :C: 33 :C:zinc-finger (ZIF268, domain 1):Mus musculus:2.10:18.20
------RPYACPVESCDRRFSRSDELTRHI-RI-HTGQK*
>P1;1zaa2 structureX:1zaa: 34 :C: 61 :C:zinc-finger (ZIF268, domain 2):Mus musculus:2.10:18.20
-------PFQCRI--CMRNFSRSDHLTTHI-RT-HTGEK*
>P1;1zaa3 structureX:1zaa: 62 :C: 87 :C:zinc-finger (ZIF268, domain 3):Mus musculus:2.10:18.20
-------PFACDI--CGRKFARSDERKRHT-KI-HLR--*
>P1;1ard structureN:1ard: 102 : : 130 : :zinc-finger (transcription factor ADR1):Saccharomyces cerevisiae:-1.00:-1.00
------RSFVCEV--CTRAFARQEHLKRHY-RS-HTNEK*
>P1;1znf structureN:1znf: 1 : : 25 : :zinc-finger (XFIN, 31st domain):Xenopus laevis:-1.00:-1.00
--------YKCGL--CERSFVEKSALSRHQ-RV-HKN--*
>P1;2drp2 structureX:2drp: 137 :A: 165:A:zinc-finger (tramtrack, domain 2):Drosophila melanogaster:2.80:19.30
----NVKVYPCPF--CFKEFTRKDNMTAHV-KIIHK---*
>P1;3znf structureN:3znf: 1 : : 30 : :zinc-finger (enhancer binding protein):Homo sapiens:-1.00:-1.00
------RPYHCSY--CNFSFKTKGNLTKHMKSKAHSKK-*
>P1;5znf structureN:5znf: 1 : : 30 : :zinc-finger (ZFY-6T):Homo sapiens:-1.00:-1.00
------KTYQCQY--CEYRSADSSNLKTHIKTK-HSKEK*
```

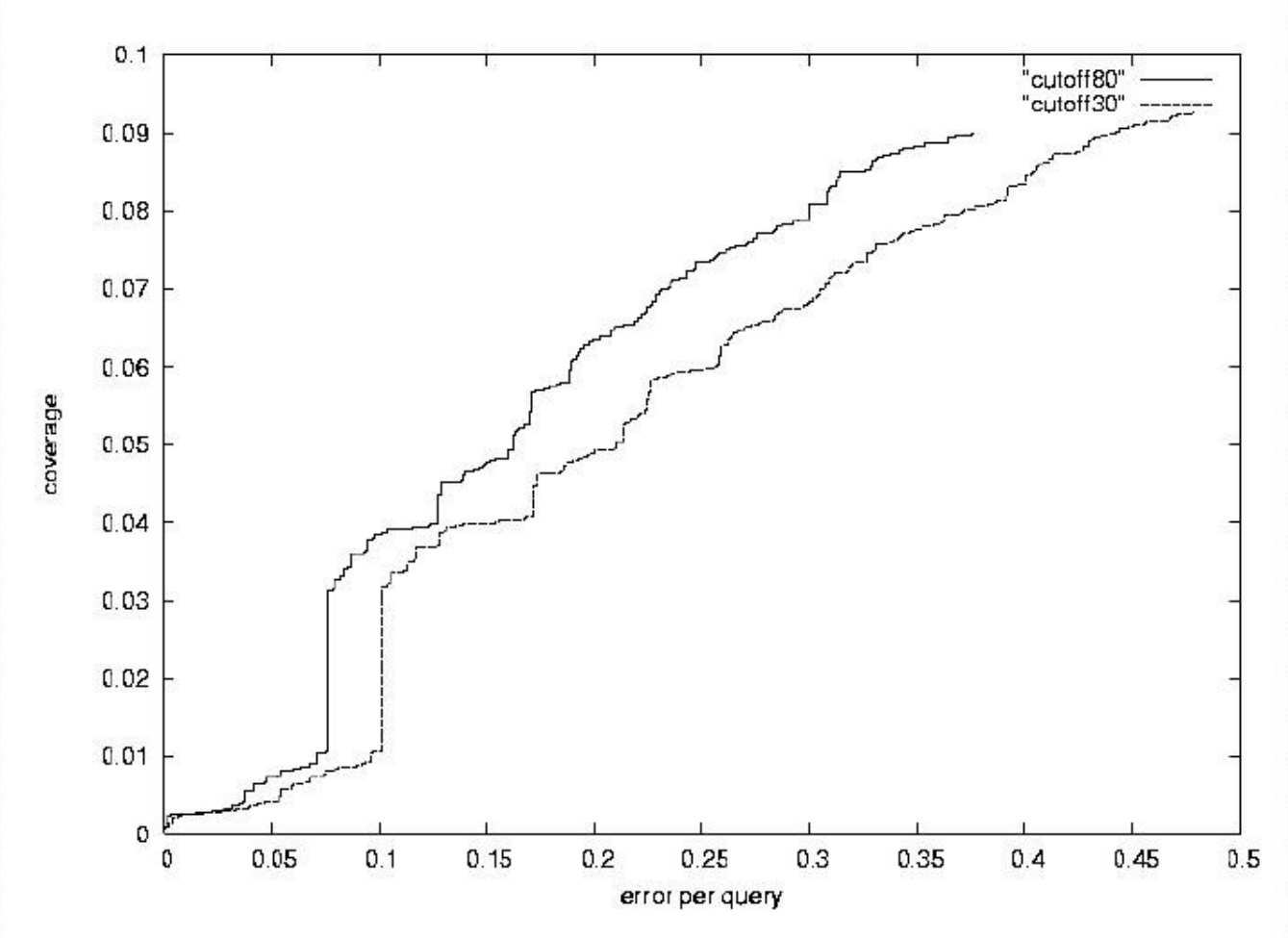
You can also look at superposed structures..

Sequence searching



Comparing methods based on ROC curves

A benchmark is a performance test of a method on a representative set of examples, for which the sequence relationships are known. Therefore, each hit can be judged as true/false. The performance of the search program is reflected in the benchmark curve.



Database Search Algorithms: Sensitivity, Selectivity

- Sensitivity – the ability to detect weak similarities between sequences (often due to long evolutionary separation). Increasing sensitivity reduces false negatives, i.e. those database sequences similar to the query, but rejected.

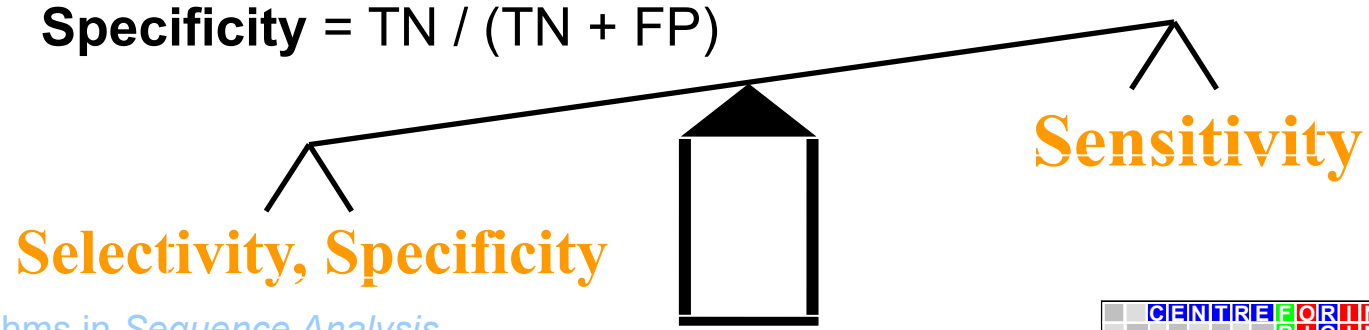
  Sensitivity (or Coverage) = TP / (TP+FN)

- Selectivity – the ability to screen out similarities due to chance. Increasing selectivity reduces false positives, those sequences recognized as similar when they are not.

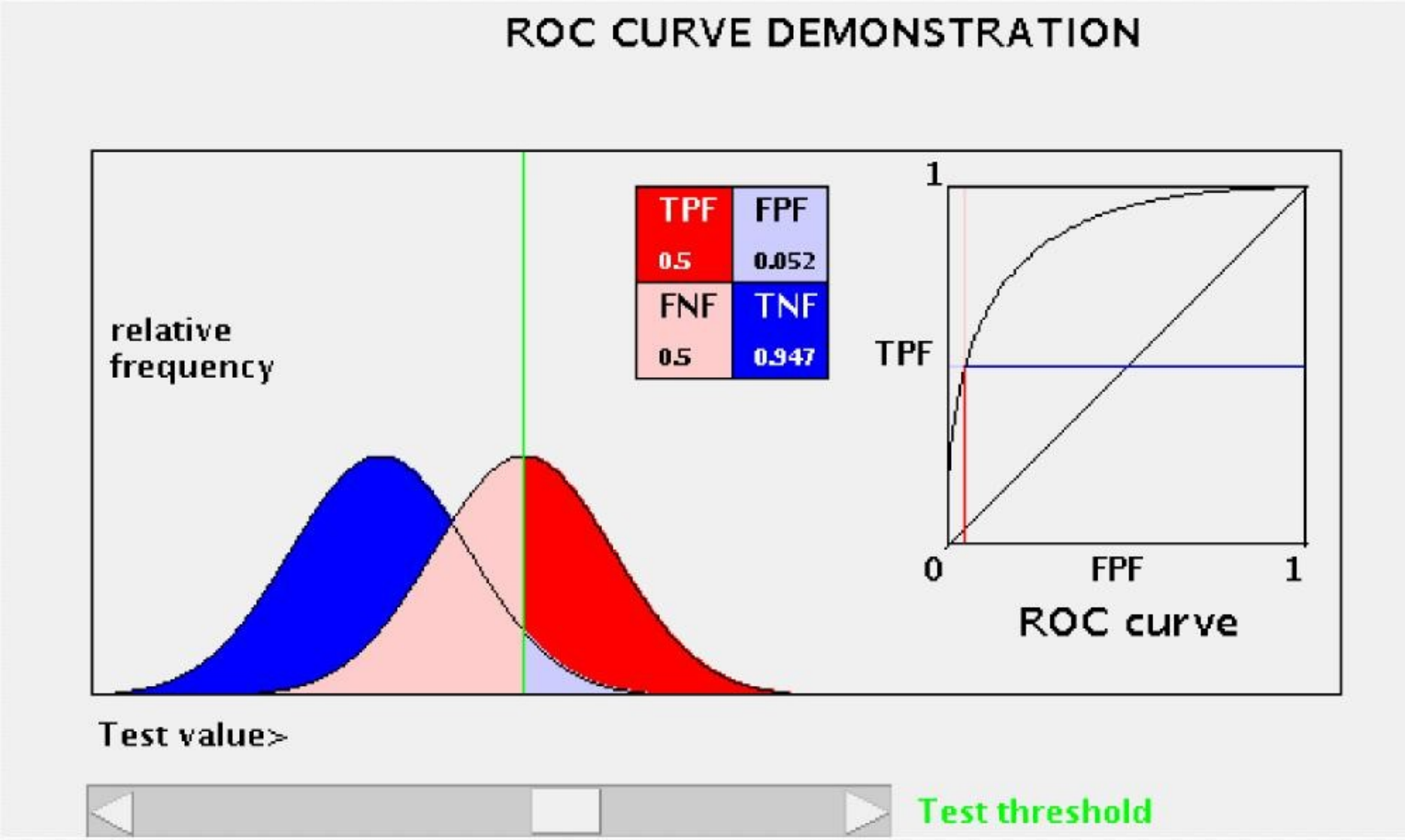
  Selectivity (or Positive Predictive Value) = TP / (TP + FP)

- Specificity also describes the ability of the method to select proper hits. Increasing selectivity reduces false positives.

  Specificity = TN / (TN + FP)



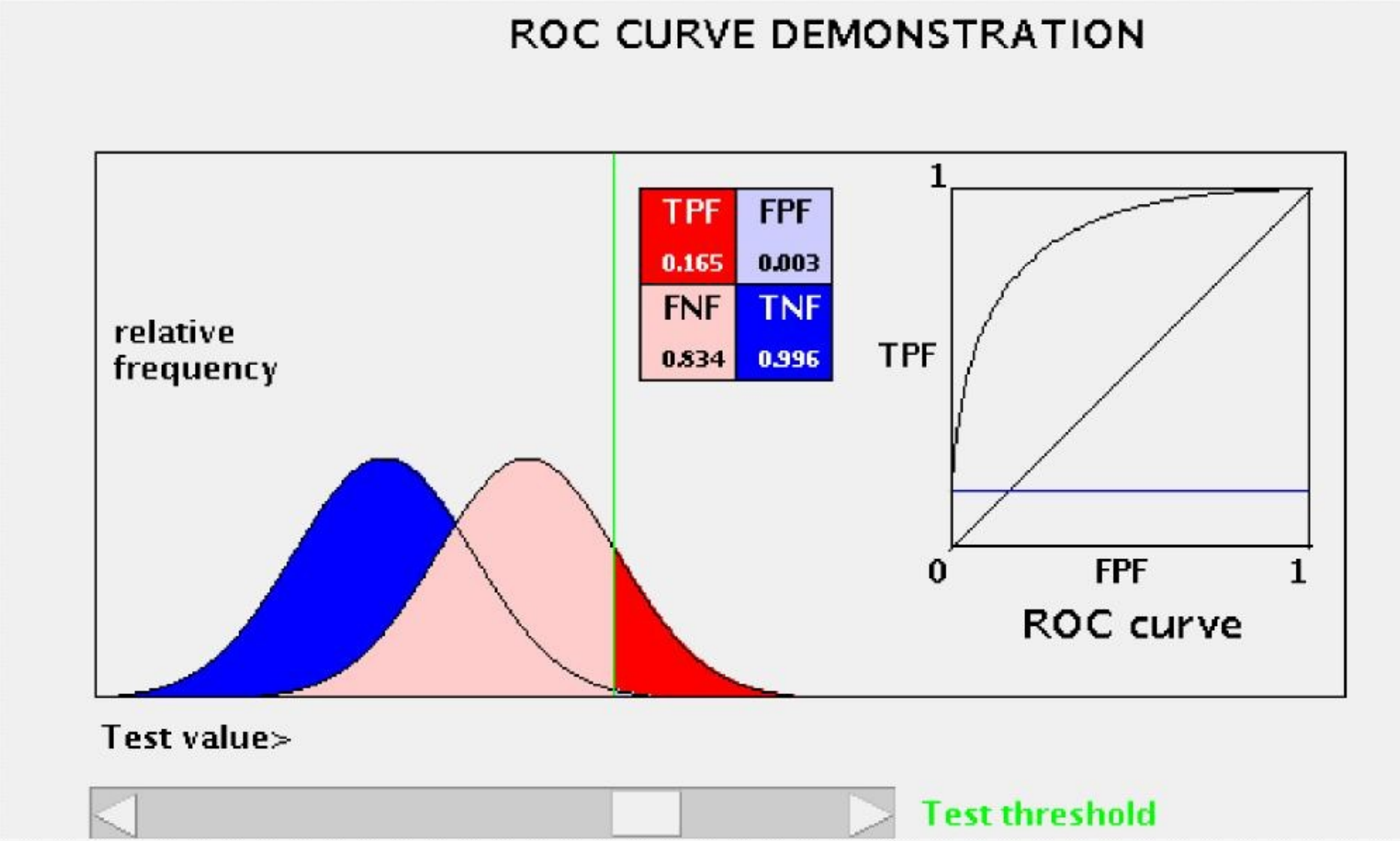
Sequence Searching



http://www.anaesthetist.com/mnm/stats/roc/

https://arogozhnikov.github.io/2015/10/05/roc-curve.html

Sequence Searching



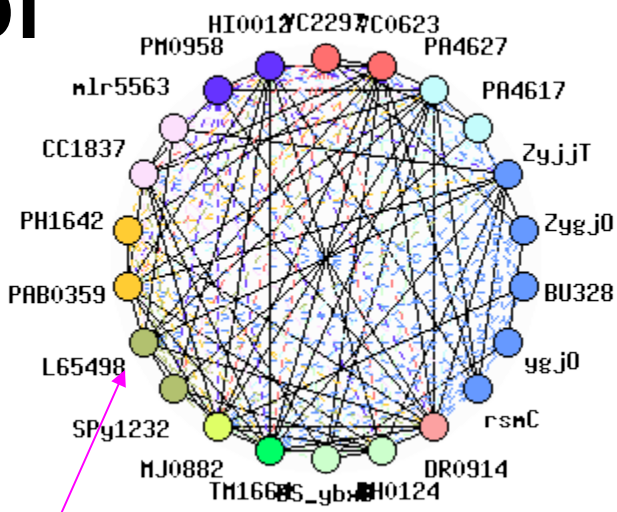
BELOW SLIDES ARE FOR REFERENCE

NOT COVERED IN LECTURE

NOT EXAM MATERIAL
– note that some of the topics below may be covered also in other lectures!

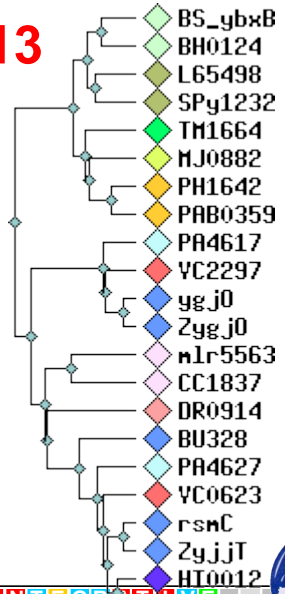
Extending homology searching using COG – Cluster of Orthologous Groups

Tatusov et al, 1997



- Orthologues found using ***bi-directional best hit*** searching with PSI-BLAST
- All COG family members are supposed to have the same function
- Searching with an unknown sequence only needs to hit a single member of a COG family, annotation can then be transferred

COG2813



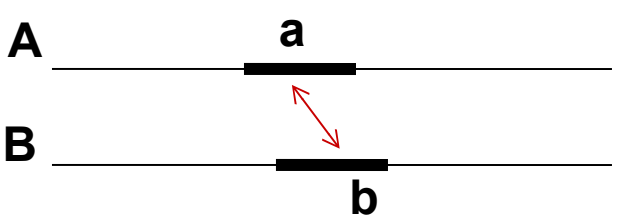
http://www.ncbi.nlm.nih.gov/COG/

Bidirectional best hit

an *operational definition* of orthology

- For a gene *a* in genome *A* (i.e. database with gene sequences of *A*) and a gene *b* in genome *B*:
  - Run query seq *a* against db *B*
  - Take best scoring hit sequence (say seq *b*)
  - Run query seq *b* against db *A*
  - If seq *a* is now best hit

  Then *a* is an ortholog of *b*



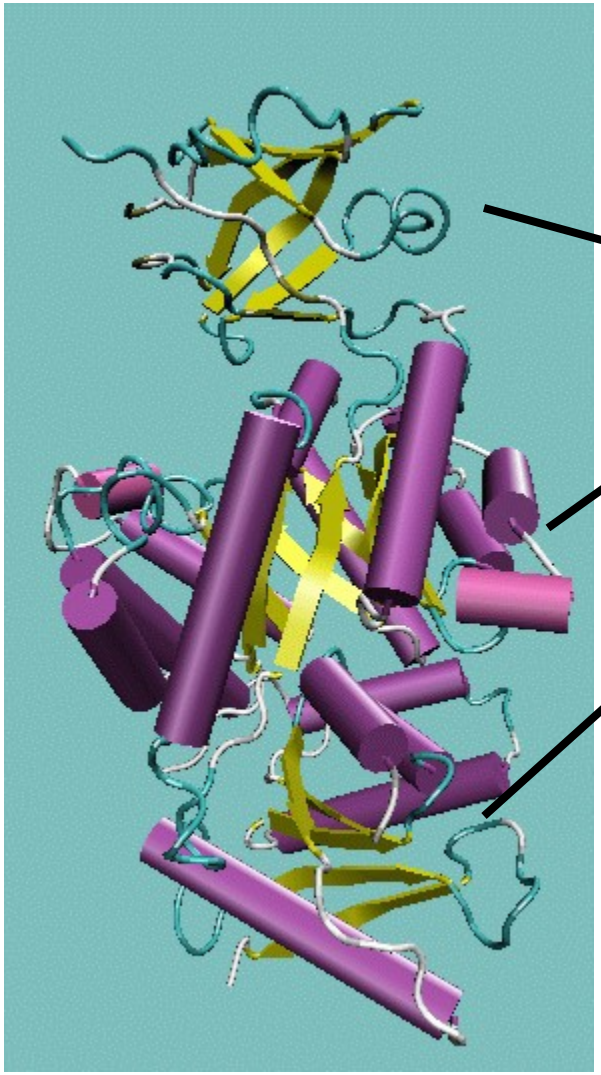
Question: how can you find paralogs using this method?

Some tricky problems when searching for homology

- Multi-domain proteins
- Low-complexity regions
- Redundancy
- Short query sequences
- Distant sequences
- Un-annotated or vaguely annotated sequences
- Profile wander using iterative methods

Protein structural domain organisation

Multi-domain proteins



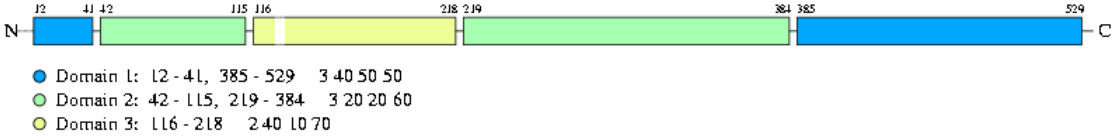
Pyruvate kinase

Phosphotransferase

β barrel regulatory domain

α/β barrel catalytic substrate binding domain

α/β nucleotide binding domain



1 continuous

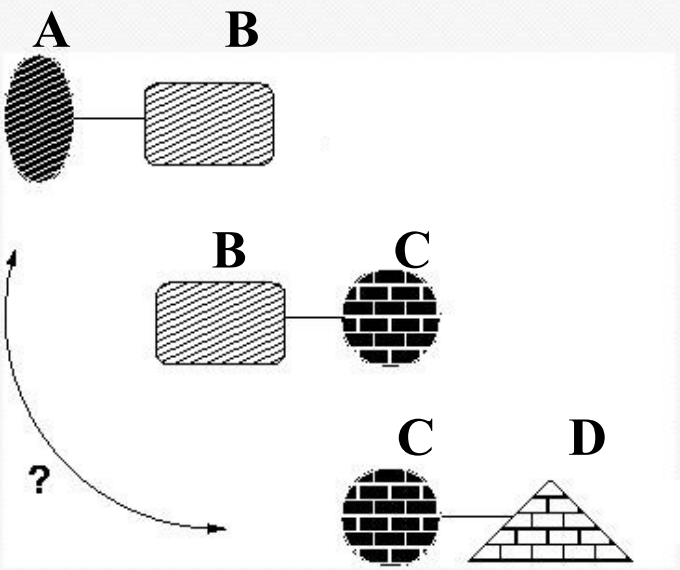
+ 2 discontinuous domains

Multi-domain Proteins

It can be disadvantageous to search with multi-domain proteins. If a multi-domain protein contains domains AB, and B is a common domain, many other (multi-domain) proteins containing this domain will be matched. The interesting domain could be A, but the majority of reported hit matches B.

In iterative sequence searching, the multi-domain proteins BC that were detected in the first round will produce hits to other proteins containing CD. The search may drift from AB to CD.

To avoid this problem, (PSI-)Blast prunes hits that extend beyond the query sequence



Multi-domain Proteins (cont.)

- A common conserved protein domain such as the tyrosine kinase domain can obscure weak but relevant matches to other domain types (e.g. only appearing after 5000 kinase hits)
- Sequences containing low-complexity regions, such as coiled coils and transmembrane regions, can cause an explosion of the search rather than convergence because of the absence of any strong sequence signals.
- Conversely, some searches may lead to premature convergence; this occurs when the PSSM is too strict only allowing matches to very similar proteins, i.e., sequences with the same domain organization as the query are detected but no homologues with different domain combinations.

Low-complexity Regions

Some genome sequences contain low-complexity regions.
These can give false-positive hits.

Example:

```
HSGDLPERTCPPCPPPCPPCPPPPCPPPCPCPPCPPPPLWQPSSERTD
         |-  low-complexity region  -|
```

Most sequence searching programs use filters to recognize and skip such low-complexity regions. If such regions are by chance included in the hit, the output looks like

```
HSGDLPERTXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXLWQPSSERTD
         |-  low-complexity region  -|
```

Calculating low-complexity regions using the SEGS program (DNA example)

- A sequence of L residues of N types can have $L!/\Pi_N n_a!$ different sequences of that same composition, where the composition vector = $(n_1,.., n_a,.., n_N)$ and
  $\Pi_N n_a! = n_1! * n_2! * .. * n_N!$
- If R_c is a vector $(r_0, r_1, \ldots, r_L)$ of length L+1, where the vector numbers correspond to the number of residues with a given frequency (e.g. there are 5 amino acid types with 0 abundance, 3 amino acid types with abundance 1, etc., in the sequence), then the total number of distinct sequences corresponding to a particular complexity state-vector is
  $(L! / \Pi_N n_a!)*(N! / \Pi_{L+1} r_c!)$, where $\Pi_{L+1} r_c! = r_0! * r_1! * .. * r_{L-1}! * r_L!$
- Based on this, the final complexity score calculated by the SEG program is
  $P_{SEG} = (1/N^L) * (L! / \Pi_N n_a!) * (N! / \Pi_L r_c!)$

See *Understanding Bioinformatics,* Pp. 151-152

Calculating low-complexity regions using the SEGS program - DNA example

- **S = ATTAT**: L = 5, N = 4, comp = (2_A, 0_C, 0_G, 3_T,),
  R_c = (2_0, 0_1, 1_2, 1_3, 0_4, 0_5)
- $L!/\Pi_N n_a!$ = 5! / (2! * 0! * 0! * 3!) = 10 different sequences of that same composition
  $N! / \Pi_L r_c!$ = 4! / (2! * 0! * 1! * 1! * 0! * 0!) = 12 different compositions giving rise to the same complexity
  - E.g. sequence CGGCG has same complexity as ATTAT
- the total number of distinct sequences corresponding to a particular complexity state-vector is
  $(L! / \Pi_N n_a!) * (N! / \Pi_L r_c!) = 10 * 12 = 120$
- The final complexity score calculated by the SEG program is
  $P_{SEG} = (1/N^L) * (L! / \Pi_N n_a!) * (N! / \Pi_L r_c!) = 1/4^5 * 10 * 12 =$
  $= 1/1024 * 120 = 0.117$

See *Understanding Bioinformatics*, Pp. 151-152

Detecting Low-Complexity using SEGS

For reference

- SEG and PSEG/NSEG algorithms
  - Wootton and Federhen
    - Methods in Enzymology 266:33 (1996)
    - Computers and Chemistry 17:149 (1993)
- SEG
  - UNIX Executable available on ncbi servers
    - Command:

seg FASTAfile Window TriggerComplexity Extension

$K_2(1)$ $K_2(2)$

    - Longer Window lengths define more sustained regions, but overlook short biased subsequences

Redundancy

Some databases, like the non-redundant sequence database, contain large number of nearly identical sequences. A typical example is the fusion peptide of hemagglutinin, a protein of the flu virus. This relatively short sequence returns more than 6000 hits to nearly identical (point mutants) sequences.

If one is interested in distant homologues, these redundant hits obscure the results.

One solution is to filter out all similar hits.

Can you think of any filter mechanisms?

Very Short Sequences

Hit selection works on the basis of a cutoff score or a cutoff probability. Very short sequences will not yield score above the cutoff, even if the similarity to a homologue is very high.

Very short sequences are not suited for sequence database searching.

The alternative is searching with sequence patterns or motifs.

Very Distant Sequences

Very distant sequences (in evolution) will probably fall into the 'twilight zone' of questionable sequence homology.

There are some issues that can give supportive evidence:

1. Select a substitution matrix for distant sequences

2. If there are conserved parts, use patterns as well

3. Use related query sequences or a sequence profile to search

4. Use Markov models to generate a family profile

5. Look for any other source of supporting information (structure, function)

Annotations

Un-annotated sequence

>P000001

Hypothetical protein

>P000001 Hypothetical protein

>P000002 Hypothetical membrane transporter

Tentative classification

>P000001 Putative membrane transporter

'Like' proteins

>P000001 Insulin-like growth factor

>P000002 Insulin receptor-like receptor

Conserved hypotheticals

>P00001 Conserved hypothetical

A substantial fraction of genes in sequenced genomes encodes 'conserved hypothetical' proteins, i.e. those that are found in organisms from several phylogenetic lineages but have not been functionally characterized.

Profile wander (or matrix migration)

- Permissive iterative searching using high E-values can lead to incorrect hits (false positives) that become included into the profile. More incorrect hits can then be added in subsequent iterations, and true homologues can be lost. Also, the search can explode, leading to large numbers of spurious hits.

- A further loss of information can be incurred with PSIBLAST, because PSI-BLAST PSSMs are trimmed to only use the highest scoring region in a search, ignoring less conserved regions

Sequence identity scoring zones

- >25-30%: **putative homology** zone
- 15-25%: **twilight** zone - many cases of homology but hard to detect using sequence analysis methods
- <15%: **midnight** zone (Rost, 1999) – still abundant cases of homology but virtually impossible to detect using sequence-based methods

Recap

- PHI-Blast
- HMM profile-profile methods
- Homology principle
- Alignment score statistical significance
  - Z-scores over scrambled sequences
  - BLAST statistical scheme
    - Extreme value distribution
- Various sequence searching methods
- Low complexity filtering
- Homology searching pitfalls

Sequence motifs and their description

Outline

Profiles / PSSMs
- Pros and cons

Genetical control
- Transcription factors and gene expression

Motifs
- What are they?
- Binding Sites

Combinatoric Approaches
- Regular expressions
- Regulatory motifs

Profiles and PSSMs

- Represent blocks of aligned sequences (local or global alignments)
- Various ways to represent amino acid probabilities at each alignment (or profile) position
  - Schemes range from simple frequencies to elaborate schemes with probability transformations, statistical normalisations, pseudo counts, etc.
- Profiles can include position-specific gap penalties and weighting schemes
- Profiles conform to the i.i.d. (identically independent distributed) model of sequence alignment.

Profiles and PSSMs

- Profiles do not incorporate relationships between columns such as:
  - If there is an L at position 3, there can not be a W at position 5
  - If there is a K at position 15, there should be hydrophobic amino acids at positions 17 and 18, etc.
- If profiles become short (covering only a few residues), then discriminatory capability breaks down
- Reduced signals (e.g. low complexity) can also reduce profile sensitivity

Profiles and PSSMs

- Still, many databases exist with grouped proteins sequences (e.g. homologous (super)families), while (HMM-based) profile-profile comparison methods are state of the art.
  - Profiles can include secondary structure information and other structural features for increased sensitivity
  - Profiles can include elaborate sequence weighting schemes (to unequally weigh contributions from sequences; e.g. based on statistical information content)
  - Powerful are Hidden Markov Method (HMM)-based profiles (end of this lecture)
- Profile-profile comparisons are symmetrical, but often profile-sequence comparisons are not (e.g. gap insertion).

Pattern matching

- Pattern matching can capture the information of a multiple sequence alignment in a complementary way to sequence profiles
  - Profile searches become limited when patterns are short in length, have varying spacing between conserved elements, or have many unconserved positions
- They are suitable for recognising protein function
- Database searching is crucial strategy
  - trypsin has *catalytic triad* (His, Asp, Ser). How to recognize this?

Recap sequence motif introduction

- Many biological mechanisms are associated with local sequence motifs
  - An important example is binding of transcription factor proteins (TFs) to specific DNA motifs called transcription factor binding sites (TFBS)

- Profiles are often not optimal for sequence motif recognition
  - E.g., they cannot express that at sequence position *i* a residue type x must be followed by residue type *y* at position *j*

Nucleotide motifs

- Short sequences of DNA or RNA (or amino acids)
- Often consist of 5- 16 nucleotides
- May contain gaps
- Examples include:
  - Splice sites
  - Start/stop codons
  - Centromeres
  - Transcription factor binding sites (TFBSs)

- Examples of protein motifs:
  - Transmembrane domains
  - Phosphorylation sites
  - Functional sites (e.g. active sites or binding sites)
  - Coiled-coil domains

Degenerate DNA codes

Four bases: A, C, G, T

Two-fold degenerate IUB/IUPAC codes:

- R=[AG]
- Y=[CT]
- K=[GT]
- M=[AC]
- S=[GC]
- W=[AT]

Four-fold degenerate: N=[AGCT]

Degenerate protein codes

20 amino acid types:

ACDEFGHIKLMNPQRSTVWY

Degenerate codes:

- X = unknown, all (20-fold degenerate)
- B = [DE]
- Z = [NQ]

Defining sequence motifs

regular expressions (regex)

- alphabet: set of symbols
  - {A, C, T, G}
- string: sequence of symbols from alphabet
  - AACTG, CATG, GGA, ACFT, ε
- regex: formal method to define (sub)set of strings
  - [^C].AG?T*
  - used for pattern matching
    - check if database sequence ∈ regex

Regular expressions
- rationale -

- "I want to see all sequences that ...
  - ... contain a C"
  - ... contain a C or an F"
  - ... contain a C and an F"
  - ... contain a C immediately followed by an F"
  - ... contain a C later followed by an F"
  - ... begin with a C"
  - ... do not contain a C"
  - ... contain at least three Cs"
  - ... contain exactly three Cs"
  - ... has a C at the seventh position"
  - ... either contain a C, an E, and an F in any order except CFE, unless there are also at most three Ps, or there is a ....

Construction of a regex

- regex contains:
  - symbols from alphabet
    - C → `{C}`
  - operators
    - operations on regex(es) yield new regex
    - concatenation, union, repetition, ...

Basic operators

$r_1 r_2$ **concatenation**

AC → { AC }
AAC → { AAC }

[$s_1 s_2$ ... s_n] **union (of symbols)**

[ACG] → { A, C, G }
[AC]G → { AG, CG }

$r_1 | r_2$ **union (of regexes)**

A|CC → { A, CC }
[AC]|AC → { A, C, AC }

r+ **repeat once or more**

C+ → { C, CC, CCC, CCCC, ... }
A[AC]+ → { AA, AC, AAA, AAC, ACA, ACC, AAAA, AAAC, ...}

Derived operators

***r*?** **optional**

C?	→	{ ε, C }
AC?G	→	{ AG, ACG }

r* **repeat zero or more times**

C*	→	{ ε, C, CC, CCC, CCCC, ... }
A*C	→	{ C, AC, AAC, AAAC, ... }
[AC]*	→	{ ε, A, C, AA, AC, CA, CC, AAA, AAC, ACA, ACC, ... }

r^{n-m} **repeat *n* – *m* times**

C^{4}	→	{ CCCC }
C^{2-4}	→	{ CC, CCC, CCCC }
C^{-3}	→	{ ε, C, CC, CCC }
C^{3-}	→	{ CCC, CCCC, CCCCC, ... }

Miscellaneous

. **any symbol**

.	→	{ A, C, G, T }
A.C	→	{ AAC, ACC, AGC, ATC }
.?	→	{ ε, A, C, G, T }
.*	→	{ ε, A, C, G, T, AA, AC, AG, AT, CA, CC, CG, CT, GA, ... }

[^$s_1 s_2$ ... s_n] **exclude symbols**

[^A]	→	{ C, G, T }
[^AC]	→	{ G, T }

(*r*) **grouping**

(AC)?	→	{ ε, AC }
AC?	→	{ A, AC }
(AC)*	→	{ ε, AC, ACAC, ACACAC, ACACACAC, ... }
AC*	→	{ A, AC, ACC, ACCC, ... }

Start and end of string

- ^ matches start-of-string (if outside brackets)
  - ^CG: match everything starting with CG
  - ^[^CG]: match everything not starting with C or G
- $ matches end-of-string
  - AC$: match everything ending with AC

Limitations

- regex cannot remember indeterminate counts !!!
  - "I want to see all sequences with ...
    - ☺ ... six Cs followed by six Ts"
      - C^6T^6
    - ☺ ... any number of Cs followed by any number of Ts"
      - ☆ C*T*
    - ☹ **... ~~Cs followed by an equal number of Ts~~"**
      - ☆ C^nT^n
      - ☆ $(CT|CCTT|CCCTTT|C^4T^4| \ldots)?$
  - use (context-free) grammar

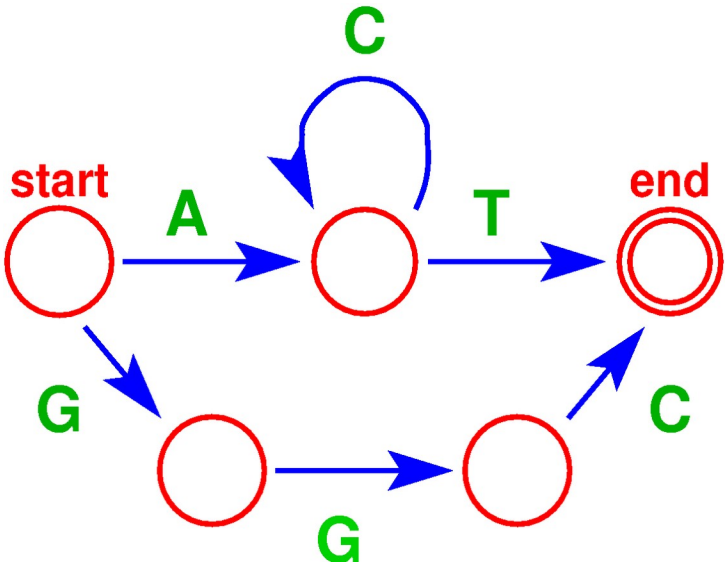
Regexes in pattern matching

- pattern described by regex
- check if sequence $\in$ regex
- matching done <u>*very*</u> efficiently
  - O(n)
  - using state machine

From earlier lecture

State machines

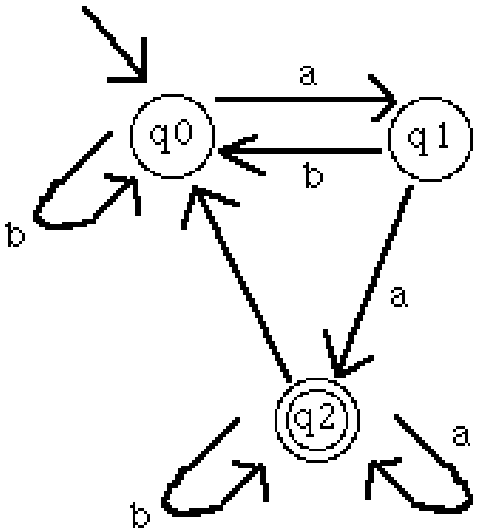
AC*T|GGC



- compile regex to state machine
- match sequence with regex

Example from BLAST: Scanning the Database

- consider a DFA to recognize the query words: QL, QM, ZL
- All that a DFA does is read strings, and output "accept" or "reject."
- use Mealy paradigm (accept on transitions) to save space and time



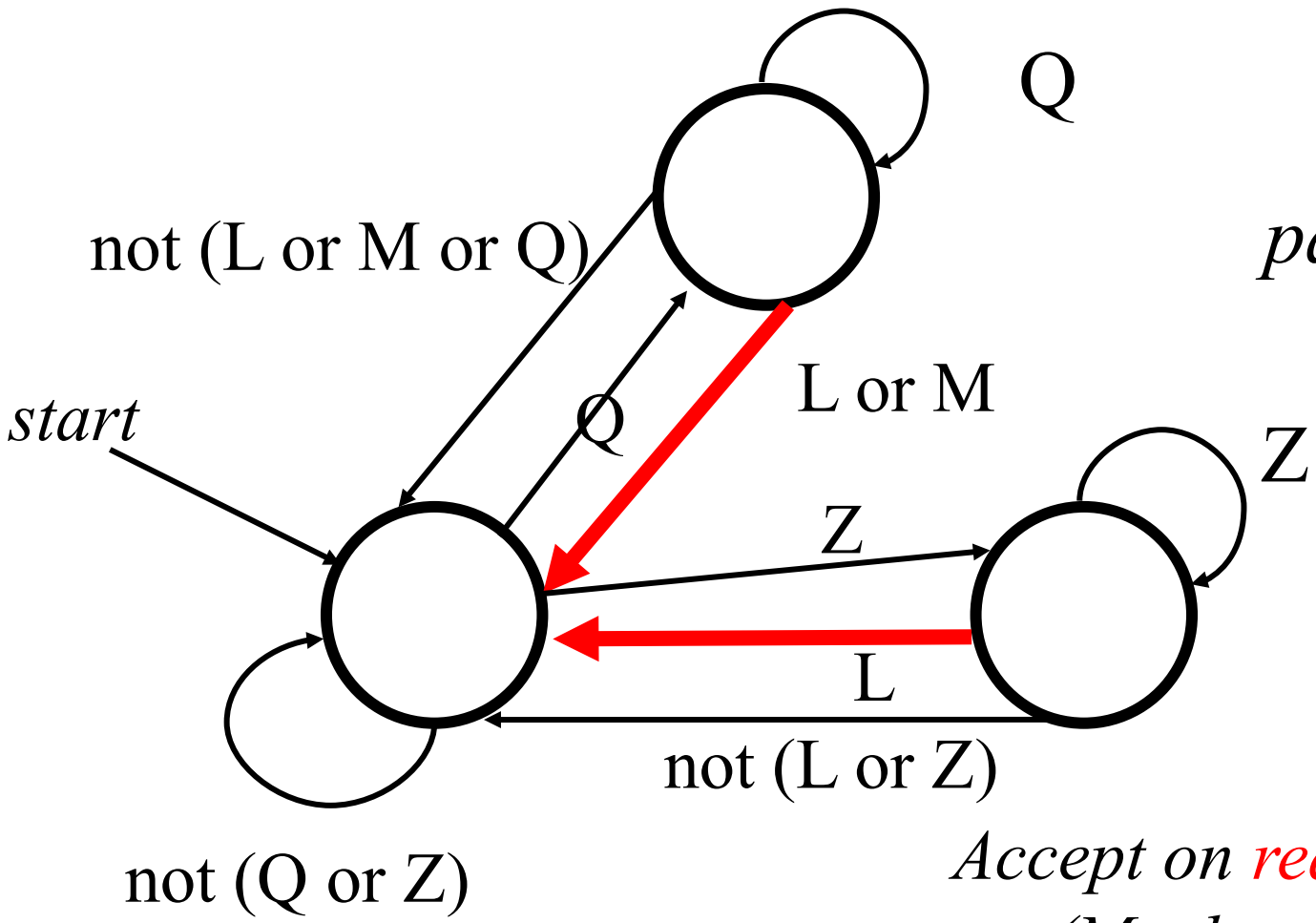
Moore paradigm: the alphabet is (a, b), the states are q0, q1, and q2, the start state is q0 (denoted by the arrow coming from nowhere), the only accepting state is q2 (denoted by the double ring around the state), and the transitions are the arrows. The machine works as follows. Given an input string, we start at the start state, and read in each character one at a time, jumping from state to state as directed by the transitions. When we run out of input, we check to see if we are in an accept state. If we are, then we accept. If not, we reject.

Moore paradigm: accept/reject states

Mealy paradigm: accept/reject transitions

From earlier lecture

Example from BLAST: a DFA to recognize query words: QL, QM, ZL



Mealy paradigm

Accept on red transitions (Mealy paradigm)

From earlier lecture

Other uses of RegEx

- many programs use regular expressions
  - command-line interpreter
    - del *.*
  - editor
    - search
    - replace
  - compilers
  - perl, grep, sed, awk
- ☹ many different syntaxes

Regular expressions

The Prosite way

Alignment

ADLGAVFALCDRYFQ

SDVGPRSCFCERFYQ

ADLGRTQNRCDRYYQ

ADIGQPHSLCERYFQ

For short sequence stretches, regular expressions are often more suitable to describe the information than alignments (or profiles)

Regular expression

```
[AS]-D-[IVL]-G-x4-{PG}-C-[DE]-R-[FY]2-Q
```

```
{PG} = not (P or G)
```

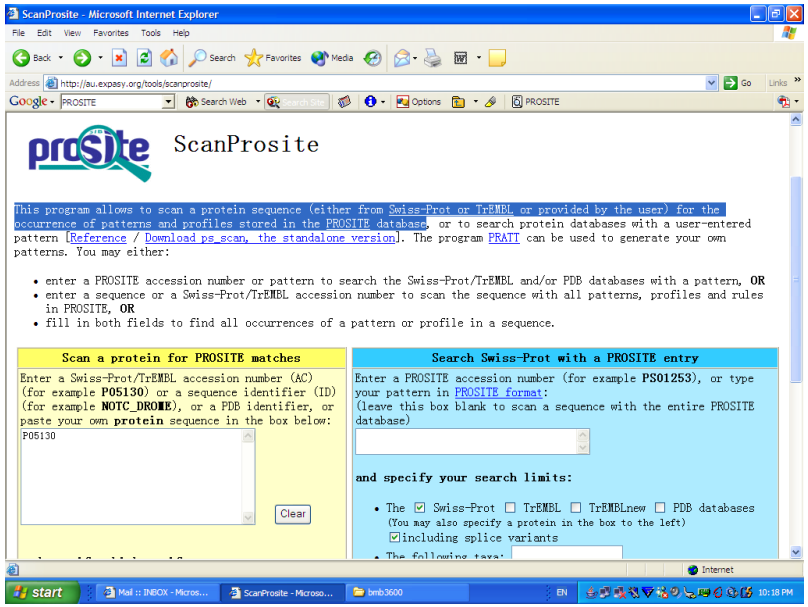
Regular expressions

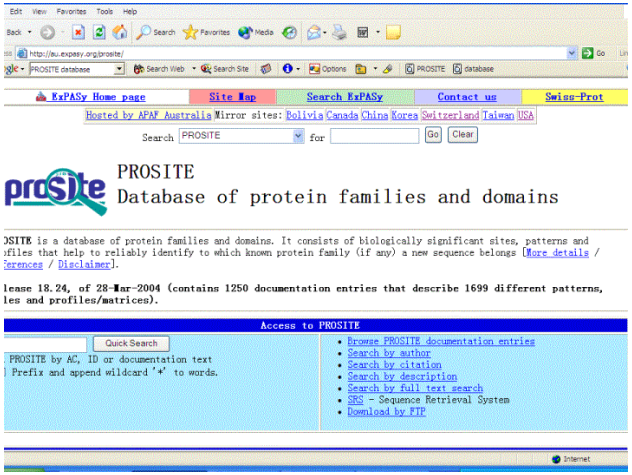
Regular expression	*No. exact matches in DB*
`D-A-V-I-D`	71
`D-A-V-I-[DENQ]`	252
`[DENQ]-A-V-I-[DENQ]`	925
`[DENQ]-A-[VLI]-I-[DENQ]`	2739
`[DENQ]-[AG]-[VLI]2-[DENQ]`	51506
`D-A-V-E`	1088

Motif-based function prediction

Prosite

- Prediction of protein functions based on identified sequence motifs
- PROSITE contains patterns specific for more than a thousand protein families.
  - ScanPROSITE -- allows to scan a protein sequence for occurrence of patterns and profiles stored in PROSITE





http://www.expasy.org/prosite/

Prosite example: RegEx

Post-translational modification

ASN_GLYCOSYLATION, PS00001; **N-glycosylation site** (PATTERN with a high probability of occurrence!)	
Consensus pattern:	**N - {P} - [ST] - {P}** *N is the glycosylation site*

Prosite also contains extended profiles (equivalent to profile HMMs)

Acyl carrier protein phosphopantetheine domain profile.

/GENERAL_SPEC: ALPHABET='ABCDEFGHIKLMNPQRSTVWYZ'; LENGTH=71;
/DISJOINT: DEFINITION=PROTECT; N1=6; N2=66;
/NORMALIZATION: MODE=1; FUNCTION=LINEAR; R1=2.3; R2=.02281121; TEXT='-LogE';
/CUT_OFF: LEVEL=0; SCORE=271; N_SCORE=8.5; MODE=1; TEXT='!';
/CUT_OFF: LEVEL=-1; SCORE=184; N_SCORE=6.5; MODE=1; TEXT='?';
/DEFAULT: D=-20; I=-20; B1=-80; E1=-80; MI=-105; MD=-105; IM=-105; DM=-105; MM=1; M0=-1; A B C D E F G H I K L M N P Q R S T V W Y Z
/I: B1=0; BI=-105; BD=-105;
/M: SY='T'; M= -5,-15,-20,-17,-12,-10,-22,-18, 2,-13, -1, 0,-13, -6,-10,-13, -5, 4, 1,-23, -9,-12;
/M: SY='E'; M= -6, -6,-22, -6, 9,-13,-21, -9,-11, 0, -8, -7, -7,-13, 1, 1, -4, -3, -8,-24,-10, 4;
/M: SY='E'; M= -5, 9,-24, 11, 15,-24,-12, -3,-23, 3,-20,-15, 6, -9, 5, 1, 4, -2,-19,-29,-16, 9;
/M: SY='E'; M= -5, 2,-26, 4, 8,-22,-13, -7,-21, 7,-17,-12, 0,-13, 3, 7, -2, -6,-16,-22,-12, 5;
/M: SY='L'; M= -6,-27,-19,-30,-23, 4,-30,-23, 26,-25, 28, 17,-25,-27,-21,-20,-19, -5, 23,-23, -3,-23;
/M: SY='R'; M= -3,-10,-10,-11, 2,-16,-19,-11,-13, -1, -8, -7, -8,-17, -1, 3, -5, -6, -9,-26,-13, -1; /M: SY='E'; M= -1, 3,-23, 4, 9,-24,-11, -7,-22, 8,-19,-13, 2,-11, 5, 6, 2, -2,-17,-26,-15, 7; /M: SY='I'; M= -5,-22,-20,-27,-19, -4,-29,-20, 20,-19, 13, 10,-18,-21,-14,-17,-15, -6, 14,-20, -4,-18; /M: SY='I'; M= -8,-30,-24,-33,-27, 8,-29,-26, 19,-24, 15, 9,-28,-27,-23,-21,-22,-10, 17, 9, 4,-25; /M: SY='A'; M= 11, -8, -8,-12, -5,-19,-11,-14,-14, -1,-14, -9, -6,-15, -4, -4, 2, -2, -6,-25,-15, -5; /M: SY='E'; M= -5, 10,-26, 15, 22,-28,-12, -2,-26, 6,-21,-16, 4, -8, 10, 0, 2, -6,-23,-28,-16, 16; /M: SY='V'; M= -5,-14,-15,-16, -6,-11,-23,-14, 4,-11, 0, 1,-13,-19, -5,-12, -7, -5, 6,-24, -8, -6; /M: SY='L'; M= -2,-24,-21,-26,-19, 5,-24,-20, 10,-23, 22, 7,-23,-24,-18,-19,-18, -7, 6, -7, 0,-18; /M: SY='G'; M= 3, -4,-25, -5, -4,-27, 24,-12,-29, -6,-25,-16, 1,-12, -4, -8, 5, -9,-23,-24,-22, -4; /M: SY='V'; M= -1,-12,-19,-14, -8,-11,-20,-14, 4,-12, 0, 1,-11,-18,-10,-13, -5, -2, 7,-25, -9,-10; /I: I=-4; MI=0; MD=-15; IM=0; /M: M= -2, -6,-13, -6, -5, -9,-11,-10, -2, -8, 0, -2, -7, -7, -7, -8, -5, -3, -1,-18, -8, -7; D=-3; /I: DM=-15;

Bucher P, Karplus K, Moeri N and Hofmann K (*1996*) A flexible motif search technique based on generalized *profiles*. Computers and Chemistry 20: 3–23.

Recap

- Many biological mechanisms are associated with local sequence motifs
- Profiles are often not optimal for sequence motif recognition
- Regular expressions are a widely used formalism to express sequence motifs.
  - DFAs are convenient for carrying out RegEx-based searches
- The PROSITE database is a repository of regular expressions or extended profiles to capture functional motifs in protein sequences
  - It comes with program to check, for given query sequence, what PROSITE entries are associated with that sequence

APPENDIX

Sequence notational formalisms and structural features

Sequence Databank Searching - Part 1

Sequence

- a string of characters that represents the chain of building blocks in a heteropolymer
- building blocks are amino acids (proteins) and nucleotides (DNA,RNA)

Backbone - Side chain

- backbone of proteins and poly-nucleotides is invariant!
- sidechains and bases make the difference: they define the sequence

Database Searching

Sequence databank searching is the process of extracting homologues of one or several query sequence(s) from a sequence database.

Nucleotides and Amino Acids

Nucleotides

A	Adenine
T	Thymine
C	Cytosine
G	Guanine

Amino acids

A	Ala	Alanine	P	Pro	Proline
C	Cys	Cysteine	Q	Gln	Glutamine
D	Asp	Aspartatic acid	R	Arg	Arg
E	Glu	Glutamic acid	S	Ser	Serine
F	Phe	Phenylalanine	T	Thr	Threonine
G	Gly	Glycine	V	Val	Valine
H	His	Histidine	W	Trp	Tryptophan
I	Ile	Isoleucine	Y	Tyr	Tyrosine
K	Lys	Lysine			
L	Leu	Leucine			
M	Met	Methionine			
N	Asn	Asparagine			

Sequence notation - Format Conventions

A sequence is composed of a name (often including an accession number) and the residue string. A sequence databank is a formatted (and often sorted) list of sequences (here FASTA format).

```
>1epi epidermal growth factor (Mus musculus)
NSYPGCPSSYDGYCLNGGVCMHIESLDSYTCNCVIGYSGDRCQTRDLRWWELR
>1ixa EGF-like module coagulation factor (Homo sapiens)
VDGDQCESNPCLNGGSCKDDINSYECWCPFGFEGKNCEL
```

Proteins are written from the N-terminus to the C-terminus:

charge

```
('+') H3N-CH(R1)-CO-NH-CH(R2)-CO-.(n).-NH-CH(Rn+2)-COO ('-')
```

Nucleotide sequences are defined within a 'reading frame', because an amino acid is defined by a nucleotide triplet. The notation is from 5' to 3':

```
('-') OPO(O'-')(OR1)-OPO(O'-')(OR2)-.(n).-OPO(O'-')(ORn+2)
```

Sequence Notation - Name Conventions

The bond between two nucleotides is called a 'phosphodiester bond', and the molecule is called a 'dinucleotide'.

The bond between two amino acids is called 'peptide bond', which is chemically an amide bond, and the molecule is called a 'dipeptide'.

2 - dipeptide, e.g. GA or Gly-ALA or glycinyl-alanine
3 - tripeptide, e.g. YGA or Tyr-Gly-ALA or tyrosyl-glycinyl-alanine
4 - tetrapeptide
5 - pentapeptide
...
tens - oligopeptide
...
>~50 - polypeptide, protein

A protein adopts a folded structure with a hydrophobic core.

Sequence Notation - Positions and Chains

Mutation

Y35G-BPTI (bovine pancreatic trypsin inhibitor)
mutation from Tyr to Gly at position 35

K(B29)P-insulin
mutation from Lys to Pro at position 29 in chain B

Des(B27-B30)-insulin-B26-carboxamide
residues 27 to 30 deleted in chain B and C-terminus amidated

Chain notation

Chains are denoted A, B, C, D ... in successive order.

Disulfide bridges

Oxidation of proximous Cys Cys pairs leads to formation of a covalent Cy-Cy disulfide bond **(cysteine bridge)**

Hierarchical (self)organisation

Primary structure: sequence

Secondary structure: repetitive backbone angles -> repetitive 3D configuration
alpha-(3.6,13)helix, beta-sheet (parallel, antiparallel)
pi-(3.0,10)helix, beta-turn, gamma-turn, loop

Super-secondary structure: combinations of secondary structure elements
helix-turn-helix, helix-turn-sheet

Domain: Packing of secondary structure elements around common core
The domain is of central importance to protein evolution.
It is a stable structural/functional entity with a core.
Many proteins are composed of several domains.

Tertiary structure: total structure of one chain

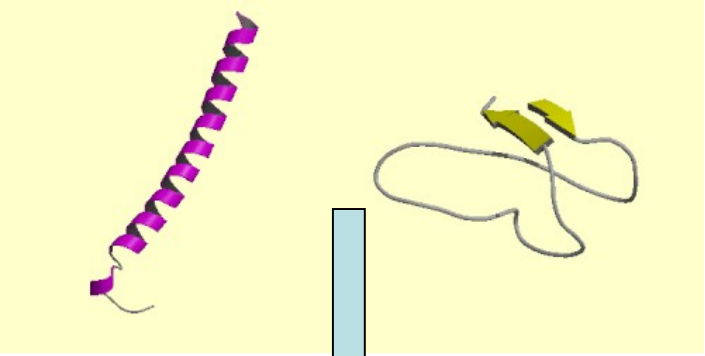
Quaternary structure: association of several chains

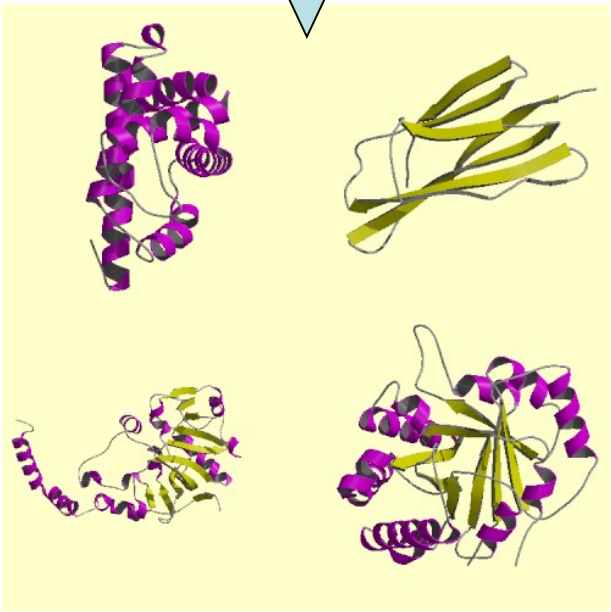
Protein structure hierarchical levels

PRIMARY STRUCTURE (amino acid sequence)

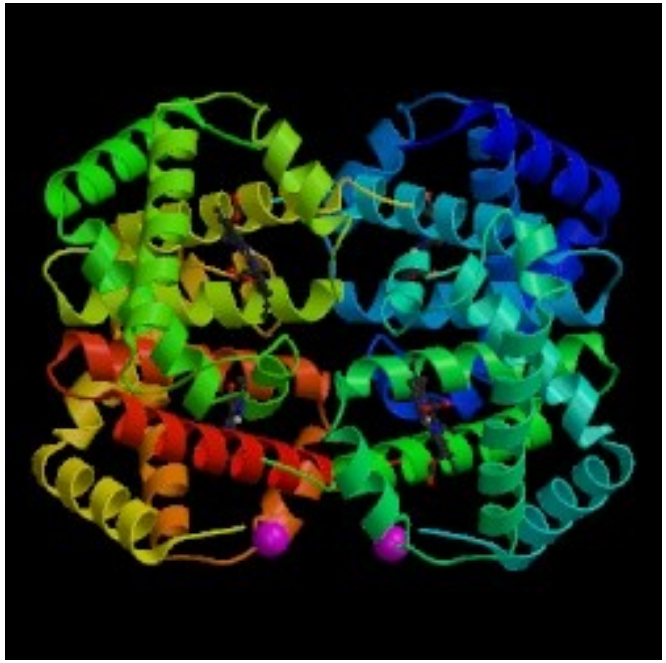
VHLTPEEKSAVTALWGKVNVD
EVGGEALGRLLVVYPWTQRFF
ESFGDLSTPDAVMGNPKVKAH
GKKVLGAFSDGLAHLDNLKGT
FATLSELHCDKLHVDPENFRLL
GNVLVCVLAHHFGKEFTPPVQ
AAYQKVVAGVANALAHKYH

SECONDARY STRUCTURE (helices, strands)



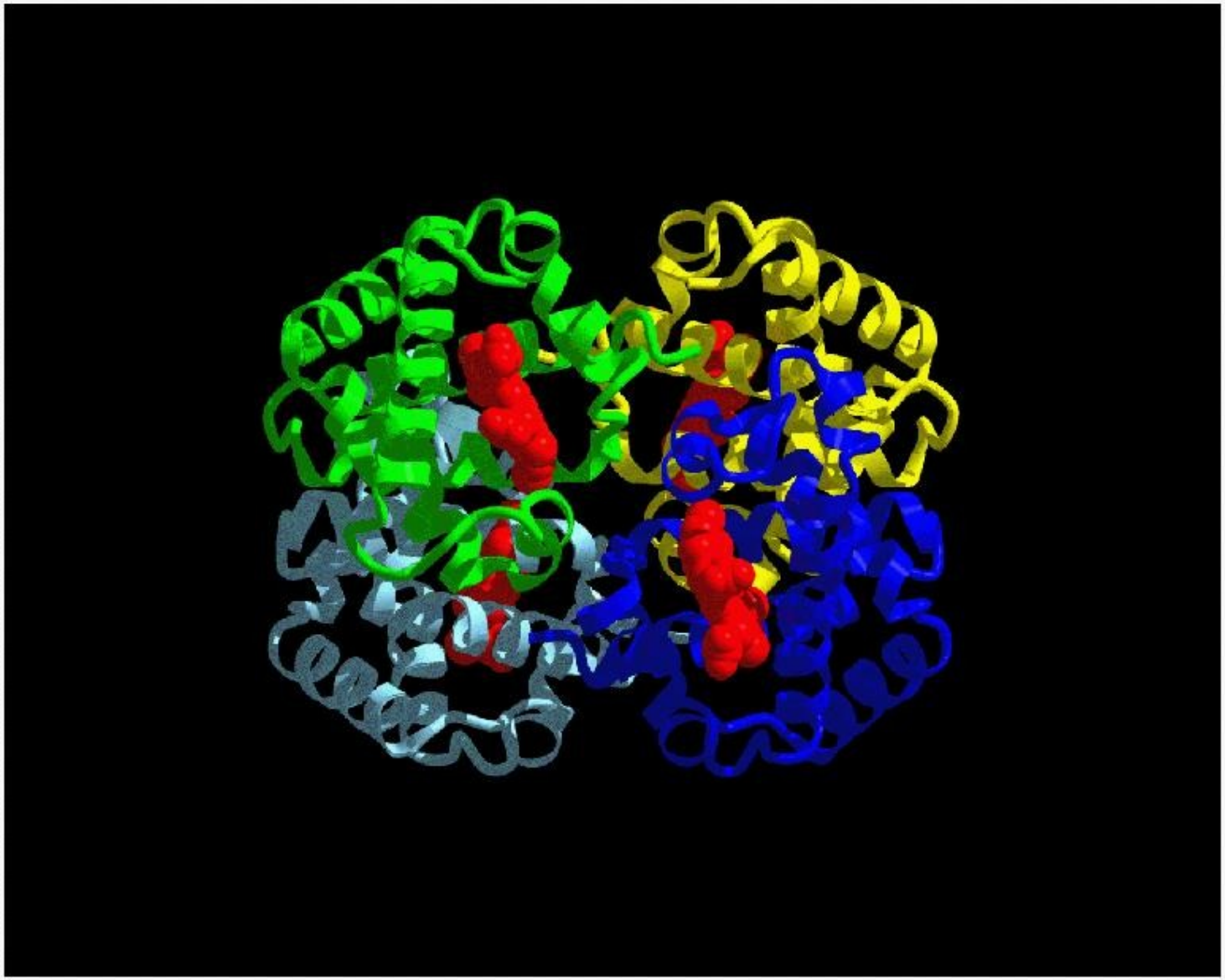


TERTIARY STRUCTURE (fold)



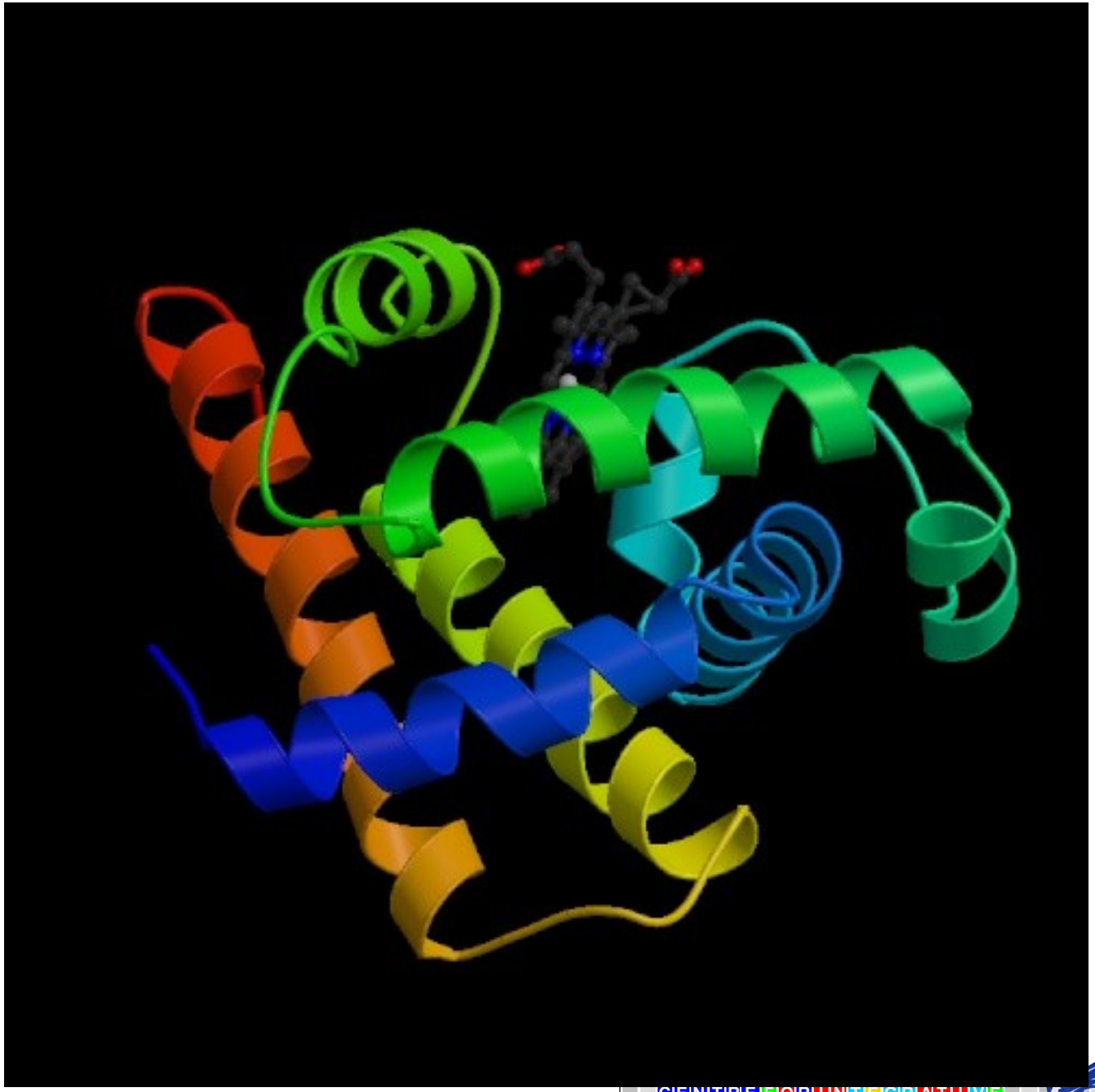
QUATERNARY STRUCTURE (oligomers)

Hemoglobin



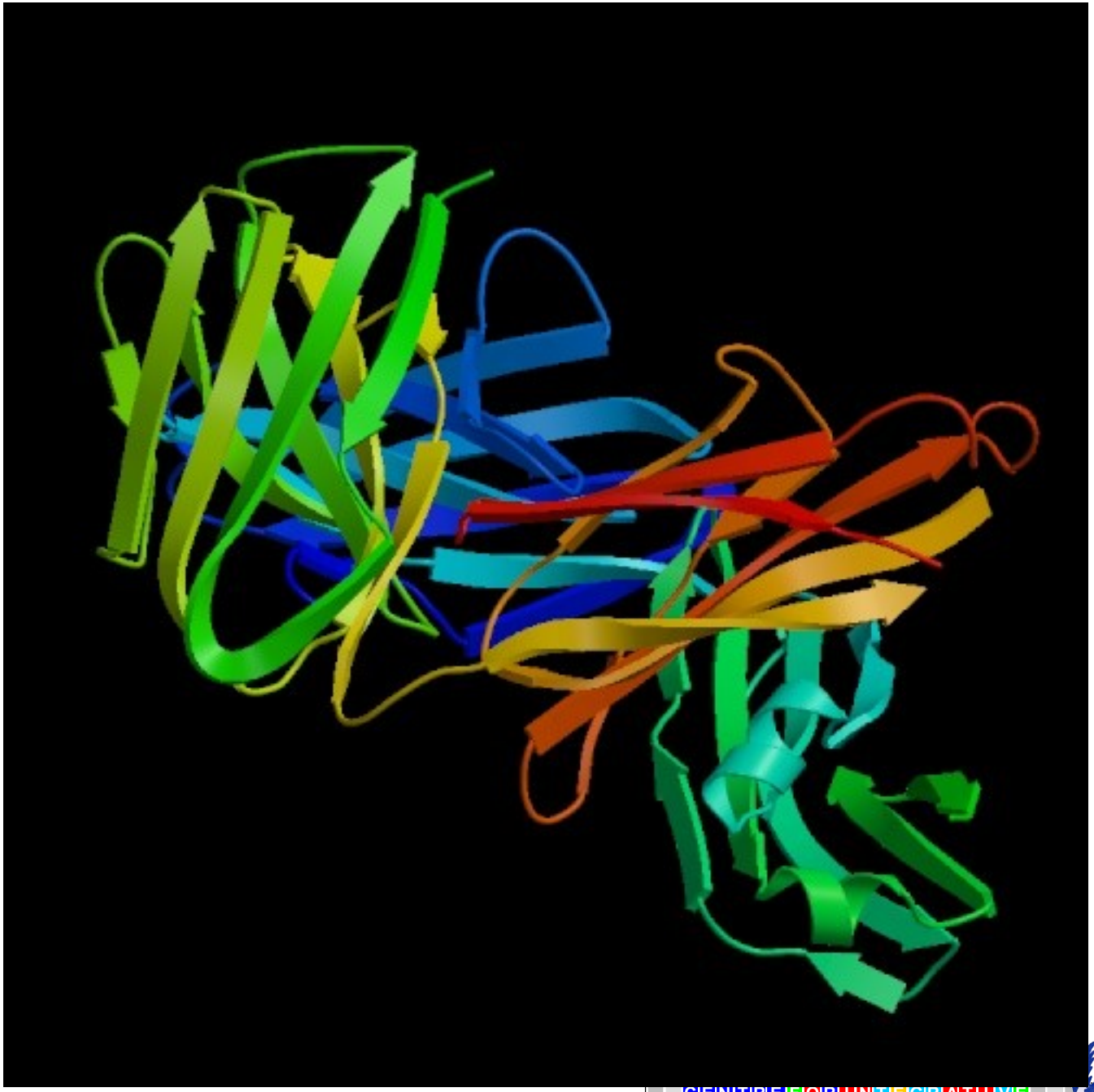
Globin fold α protein myoglobin PDB: 1MBN

Helices are labelled 'A' (blue) to 'H' (red). D helix can be missing in some globins: what happens with the alignment?



β sandwich
β protein

immunoglobulin
PDB: 7FAB



TIM barrel
α/β protein
Triose phosphate IsoMerase
PDB: 1TIM