Sequence alignments
CONTACT: Lothar Richter richter@rostlab.org

Videos: YouTube / www.rostlab.org

THANKS:

Special lectures:
- TBD - Thomas Hopf
- TBD - Jonas Reeb

No lecture:
- 05/10 Student assembly (SVV)
- 05/17 Ascension day
- 05/26 Whitsun holiday
- 06/04 Corpus Christi

LAST lecture: June 28 (followed by 2 wrap-up sessions)

Examен:
- Makeup: June 30, 2016: lecture time room TBD
- TBC: Oct 18 & 20, 2016 - lecture time

EXERCISES:

Dmitrij Nechaev

Lothar Richter

© Burkhard Rost
Science is communication
questions are often the first step
Central dogma

DNA Polymerase

replication (DNA -> DNA)

DNA

transcription (DNA -> RNA)

RNA Polymerase

RNA

translation (RNA -> Protein)

Ribosome

Protein

Function

Structure

slide: Andrea Schafferhans

dhorspool@en.wikipedia
Why compare 3D shapes, why not function?
Why 3D not function compared?

- because structure is related to function
- because we CAN compare 3D structures
- sometimes:
  similar in structure -> similar in function
Alcohol dehydrogenase (ADH)


http://www.proteopedia.org/wiki/images/7/7b/1htb2.png

http://upload.wikimedia.org/wikipedia/commons/thumb/a/a5/Protein_ADH5_PDB_1m6h.png/800px-Protein_ADH5_PDB_1m6h.png
How to get to shapes?
How to get to shapes?

- primarily by experiment (most accurate)
- then by computational biology (most inferences)

- how that is done?
  lectures in 2 weeks/rest of the entire lecture
Recap:
3D comparison
Notation: protein structure 1D, 2D, 3D
3D classification: SCOP

SCOP

http://scop.mrc-lmb.cam.ac.uk/scop/

[Murzin et al. J. Mol. Biol. 247, 536-540]

Alexei Murzin
3D classification: CATH

CATH

http://www.cathdb.info/

superfamily 3.40.640.10
Type I PLP-dependent aspartate aminotransferase-like (Major domain)

Christine Orengo

Janet Thornton
CATH: 50 structures - 1 superfamily

superfamily 3.40.640.10

Type I PLP-dependent aspartate aminotransferase-like (Major domain)
RMSD is not a metric

A similar B
B similar C
NOT implying:
A similar C

cRMSD = 2.8 Å = 0.28 nm

cRMSD = 2.85 Å = 0.285 nm
3D classification - COPS/TopSearch

COPS
http://cops.services.came.sbg.ac.at
[SJ Suhrer et al. (2009) NAR 37, W539-W44. ]

PDB id 1z6t-A with 2a5y-B

Manfred Sippl
Salzburg
COPS metric

protein A
protein B
protein C

Triangle inequality/transitivity:

\[ A \approx B \]  (A similar to B)
\[ B \approx C \]  (B similar to C)

\[ \Rightarrow B \approx C \]  (does not imply: A similar to C)
COPS metric

Axioms / Definitions:

\[ S_{a,a} = L_a \]
\[ S_{a,b} \geq 0 \]
\[ S_{a,b} = S_{b,a} \]
\[ S_{b,c} \geq S_{a,b} + S_{a,c} - L_a \]
\[ D_{a,b} = L_a + L_b - 2S_{a,b} \]

- Alignment method not so important! for COPS: TopMatch
- Metric can reveal alignment problems (e.g. via triangle inequality)

PDB updates 2008/08/19-2009/04/14

SJ Suhrer et al. (2009) NAR 37:W539-W544
Sequence comparisons: pairwise methods
Alignments answer: How similar are proteins?
Alignments

OPA
PAPI
Alignments

OPA..
PAPI
Alignments

OPA..
.PAPI
.PAPA
Alignments

.PAPI
OPA..
.PAPA
.PAPU
.PIPI
Alignments

.PAPI
OPA..
.PAPA
.PAPU
.PIPI

father
Alignments

.PAPI
OPA..
.PAPA
.PAPU
.PIPI

father
something else
“Right” alignment would have been

grandfather → .PAPI → OPA.. → .PAPA → .PAPU → .PIPI

father → 3rd

grandfather → .PAPI → OPA.. → .PAPU → .PAPA → .PIPI

father
“Right” is defined by objective function
“Correct” alignment: objective function

- simplest objective function: percentage of letters identical
- more complicated functions describing ‘match’

- but all this ignores “macroscopic similarity” (from level of strings to biology)
  objectives could be:
  - match structure
    (identify all pairs A,B with RMSD(A,B)<2.5Å)
  - match function
    (identify all pairs A,B with same function)
  - evolutionary relation
    (e.g. all A,B with common ancestor)
For time being: let us fix the ‘string matching’ problem
Alignments: Steps to do

- goal: align two sequences

  - (1) Find optimal superposition of the two
  - (2) Define “optimal”

GGQLAKEEAL
EGQPVEVLP
Alignment: Local vs. Global

- compare:

  - global: all residues aligned
    - GGQLAKEEAL
    - EGQPVEVLP

  - local: best matches
    - GGQLAKEEAL
    - EGQPVEVLP

© Burkhard Rost
Alignment: Assumptions about meaning

- Biological meaning of insertion (".")?

- Historical interpretation (reason to start working on the problem):
  evolutionary relation

GGQLAKEEAL
GQ..PVEVL
Alignment: Assumptions about meaning

- Biological meaning of insertion (".")?

- Historical interpretation
  (reason to start working on the problem):
  evolutionary relation

- also implied:
  - similar structure
  - similar function/role
Meaning of “homology”?
phylogeny:
phylo: race/tribe/kind
genesis: birth
genos: origin
tree of life - Darwin 1859 Origin of Species
Phylogenetic Tree Of Life

© Wikipedia
Phylogenetic Tree Of Life

© Wikipedia
Phylogenetic Tree Of Life
Phylogenetic Tree Of Life

Phylogenetic Tree of Life

© Wikipedia
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

typically NOT as simple
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/
homology
homology definition

etymology: Greek:

ομόλογος (homologos):

homo (agreeing/same) + logos (word/reason/opinion)

wikipedia: relation
Homology definition

- **Etymology:** Greek: ομόλογος (homologos): homo (agreeing/same) + logos (word/reason/opinion)

- Originally (1843 Richard Owen):
  
  “same organ in different animals under every variety of form and function”

1847 diagram by Richard Owen shows his conceptual archetype for all vertebrates

© Wikipedia
homology definition

- **etymology:** Greek: ομόλογος (homologos):
  homo (agreeing/same) + logos (word/reason/opinion)

- **originally (1843 Richard Owen):**
  “same organ in different animals under every variety of form and function”

- **Genes:**
  homologs originate from common ancestor
  - orthologs: speciation event
  - paralogs: duplication event
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

species=?
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

species=mating
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

happily munching
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

disaster strikes
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

populations diverge
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

rejoined - yet separated
Evolution: speciation


Biology defines species how?
two species mate

→

offspring not passing on genetic information
Evolution: speciation

© http://evolution.berkeley.edu/evosite/evo101/

Happy Face Spider *Theridion grallator*  
(same species-interbreed)

Carrion/Hooded crow  
(same species?)
homology definition

- **etymology**: Greek: ομόλογος (homologos): homo (agreeing/same) + logos (word/reason/opinion) (wikipedia: relation)

- **originally** (1843 Richard Owen): “same organ in different animals under every variety of form and function”

- **Genes**:
  - homologs originate from common ancestor
    - orthologs: speciation event
    - paralogs: duplication event

- **misuse of term “homolog structure”**
  => similar structure
“Homology” frequently and incorrectly used for genes/proteins
... but, it reminds us of the connection in evolution
Alignment: Assumptions about meaning

☐ Biological meaning of insertion (".")?

☐ Historical interpretation (reason to start working on the problem):

- evolutionary relation
- common ancestor
- homology
Alignment: Assumptions about meaning

- Biological meaning of insertion (".")?
  - GGQLAKEEAL
  - GQ..PVEVL

- Historical interpretation (reason to start working on the problem):
  - evolutionary relation

- what we also WANT:
  - similar structure
  - similar function
we compare 1D sequences to find similarity in 3D structure / function
we compare 1D sequences to find similarity in 3D structure / function
Dynamic programming: brute force
How to align sequences?

1. visually

GGQLAKEEAL
GQ..PVEVL
How to align sequences?

☐ 1. visually

A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins

SAUL B. NEEDLEMAN AND CHRISTIAN D. WUNSCH
Department of Biochemistry, Northwestern University, and Nuclear Medicine Service, V. A. Research Hospital Chicago, Ill. 60611, U.S.A.
(Received 21 July 1969)

A computer adaptable method for finding similarities in the amino acid sequences of two proteins has been developed. From these findings it is possible to determine whether significant homology exists between the proteins. This information is used to trace their possible evolutionary development.

The maximum match is a number dependent upon the similarity of the sequences. One of its definitions is the largest number of amino acids of one protein that can be matched with those of a second protein allowing for all possible interruptions in either of the sequences. While the interruptions give rise to a very large number of comparisons, the method efficiently excludes from consideration those comparisons that cannot contribute to the maximum match.

Comparisons are made from the smallest unit of significance, a pair of amino acids, one from each protein. All possible pairs are represented by a two-dimensional array, and all possible comparisons are represented by pathways through the array. For this maximum match only certain of the possible pathways must be evaluated. A numerical value, one in this case, is assigned to every cell in the array representing like amino acids. The maximum match is the largest number that would result from summing the cell values of every pathway.

1. Introduction

© Burkhard Rost ROSTLAB. 74/165
Dynamic programming: Global

GGQLAKEEAL
EGQPVEVL ..
Dynamic programming: Global

GGQLAKEEAL
EGQPVEVL

GGQLAKEEAL
. EGQPVEVL .
Dynamic programming: Global

GGQLAKEEAL
EGQPVEVL

GGQLAKEEAL
EGQPVEVL

GGQLAKEEAL
..EGQPVEVL
## Dynamic programming concept

<table>
<thead>
<tr>
<th>First letter</th>
<th>G</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Burkhard Rost
Dynamic programming concept

<table>
<thead>
<tr>
<th>2nd letter:</th>
<th>E</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

move on diagonal
Dynamic programming concept

<table>
<thead>
<tr>
<th>2nd letter:</th>
<th>G</th>
<th>G</th>
<th>Q</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Q</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>P</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>L</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2nd letter: GQP
3rd letter: GGQ
Dynamic programming concept

<table>
<thead>
<tr>
<th>2nd letter:</th>
<th>G</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GQP</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GGQ</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>better:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GQP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GQL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Burkhard Rost
Dynamic programming concept no ins

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
allowing for gaps

no gap

GGQLAKEEAL
GQPVEVL

with gap

GGQLAKEEAL
GQ..PVEVL
Dynamic programming concept ins
Dynamic programming dot plots

© Wikipedia

Dot plot of a human zinc-finger transcription factor (GenBank NM_002383) against itself to show self-similarity
Linear gap penalty:

N gaps cost N

EGQ......PVEVLP  <->  E.G.Q.P.V.EVLP
Gap/Insertions: penalty

- linear gap penalty:

  N gaps cost N

  EGQ....PVEVLP  <->  E.G.Q.P.V.EVLP

- idea of BLOCKS:

How to enforce blocks in alignments?

regular alignment score = matches - gaps

\[ NW = \sum_{k=l}^{L_{ali}} M_{U_kT_k} - G_0 \cdot N_{gap} \]
Gap/Insertions: penalty

- **linear gap penalty:**
  - N gaps cost N
  - EGQ......PVEVLP  <->  E.G.Q.P.V.EVLP

- **affine gap penalty:**
  - gap open
  - gap extension (elongation)
  - typical go=10 x ge
Dynamic programming: optimal alignment

Pair of protein sequences
U  GGQLAKEEAL
T  EGQPVEVL

Optimal alignment (no gaps)
U  GGQLAKEEAL
T1  EVL
T2  EGQPVEVL

Optimal alignment (with gaps)
U  GGQLAKEEAL
T  EGQPVEVL

- Global/no gap:
  SB Needleman and CD Wunsch 1970 J Mol Biol 48, 443-53
- Local/Gap:
  TF Smith and MS Waterman 1981 J Mol Biol 147, 195-197

\[
SW = \sum_{k=l}^{L_{ali}} M_{U_kT_k} - Go \cdot N_{gap} - Ge \cdot (L_{gap} - N_{gap})
\]
What is best: local or global?
global clearly worse?

GGQLAKEEAL
G.Q..PVEVL

GGQLAKEEAL
GQ..PVEVL
Is really the total number of matches the best criterion?
Is really the total number of matches the best criterion?

Remember: protein domains are 60-400 residues
what is best?

<table>
<thead>
<tr>
<th>G</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2 of 2 better than 4 of 8?

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to answer the question which is best?

2 in 2
or
4 in 7

© Wikipedia
Identity the best criterion? Substitution matrices
## Alignments in brief

### Query and Subject Information
- **Query**: 1bl8_A  Protein length: 97  Potassium Channel Protein
- **Subject**: 1orq_C  Protein length: 223  Potassium Channel

### Sequence Alignment

<table>
<thead>
<tr>
<th>Query: 21 GSYIATLAERGAPGAQLITYPRALWWSVETATTGGYLGYDLYPVTLVGRCVAVVVMVAGITS 80</th>
<th>Subject: 147 GAFAIYIVEYDPDNSSIKSVFDALWWAVVTTATTGGYGDVVAPATPIKVGIHAVMTGISA 206</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 G++</td>
<td>147 G++</td>
</tr>
<tr>
<td>22 E</td>
<td>148 P</td>
</tr>
<tr>
<td>23 P</td>
<td>149 +</td>
</tr>
<tr>
<td>24 +</td>
<td>150 +</td>
</tr>
<tr>
<td>25 +</td>
<td>151 +</td>
</tr>
<tr>
<td>26 +</td>
<td>152 +</td>
</tr>
</tbody>
</table>
| 27 AL WW+V | 153 TATTGGYGD+ P T G+ + + VM+ GI+
| 81 | 206 |

### Scoring Matrix

**Algorithm to optimize score**
Alignment: Substitution Matrix: PAM

PAM: Point Accepted Mutations (substitution matrix)

Margaret Oakley Dayhoff (1925-1983)

introduced 1978 based on 1572 observed mutations in 71 families of closely related (85% PIDE) proteins
Protein Atlas (Atlas of Protein Sequence and Structure)
(1st sequence: insulin: 1955 Fred Sanger, 1st structures: 1958: Perutz et al. (hemoglobin) / Kendrew et al. (myoglobin) )


PAM1: 1 point mutation per hundred comparisons
PAMn=(PAM1)n
BLOSUM

- BLOcks of amino acid SUbstitution Matrices
- Align only conserved regions
- Compile log-odd ratios

\[ S_{i,j} = \log \frac{p_i \cdot M_{i,j}}{p_i \cdot p_j} = \log \frac{M_{i,j}}{p_j} = \log \frac{\text{observed frequency}}{\text{expected frequency}} \]

- BLOSUM \( n \) = threshold at \( n \% \) pairwise sequence identity

Steven Henikoff & Jorja Henikoff (1992) PNAS 89:10915-9
### Alignments: scoring matrix

![Scoring Matrix](image)

Alignments: scoring matrix

Scoring Matrix

Alignments: scoring matrix

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>T</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td>-3</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>-3</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>-4</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>Q</td>
<td>-5</td>
<td>-1</td>
</tr>
<tr>
<td>H</td>
<td>-3</td>
<td>-1</td>
</tr>
<tr>
<td>R</td>
<td>-4</td>
<td>0</td>
</tr>
<tr>
<td>K</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>-5</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>L</td>
<td>-6</td>
<td>3</td>
</tr>
<tr>
<td>V</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>-4</td>
<td>3</td>
</tr>
<tr>
<td>Y</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>W</td>
<td>-8</td>
<td>2</td>
</tr>
</tbody>
</table>

Scoring Matrix

Alignments: scoring matrix

Many more substitution matrices exist today
Many more substitution matrices exist today

Most use: BLOSUM62
Many more substitution matrices exist today. Most use: BLOSUM62.
Interactive software tool

Ignacio Ibarra & Francisco Melo:

Interactive software tool to comprehend the calculation of optimal sequence alignments with dynamic programming

Bioinformatics 2010, 26:1664-5

http:/melolab.org/sat
dynamic programming
optimal, but ...
Issues: Dynamic programming

- time used?
  order of length^2 -> long
### Big data - big CPU

<table>
<thead>
<tr>
<th></th>
<th>Proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009/09</td>
<td>$8 \cdot 10^6$</td>
</tr>
<tr>
<td>2014/04</td>
<td>$55 \cdot 10^6$</td>
</tr>
</tbody>
</table>

- Double every 2 years
- 6.9 fold
- 55 months
Big data - big CPU

<table>
<thead>
<tr>
<th></th>
<th>Proteins</th>
<th>comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009/09</td>
<td>$8 \cdot 10^6$</td>
<td></td>
</tr>
<tr>
<td>2014/04</td>
<td>$55 \cdot 10^6$</td>
<td></td>
</tr>
</tbody>
</table>

*Double every 2 years*

**Cytochrome P450 - Mentha piperita - peppermint**
MELQLWSALILVVTYTISLLINQWRKPKPQGFPFPGLPPKLPLIGHHLWLQGKLPOHAL ASVAKEYGPVAHVQGLGEVFSVLSSREATKEAMKLVDPACANRFESIGTRIMWYDNEDII

**Cytochrome P450 - Mentha spicata - spearmint**
MELDSLAILVATYIVLLINQWKLKSSQQLPPPSPPKLPGVGLHLWGLPQHVF R SIAQKYGPVAHVQGLGEVYSVLSSAEEAKQAAMKVLDPNFADRFDGIGSRTMNYDKDIIF
## Big data - big CPU

<table>
<thead>
<tr>
<th></th>
<th>Proteins</th>
<th>comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009/09</td>
<td>$8 \cdot 10^6$</td>
<td>$64 \cdot 10^{12}$</td>
</tr>
<tr>
<td>2014/04</td>
<td>$55 \cdot 10^6$</td>
<td>$3025 \cdot 10^{12}$</td>
</tr>
</tbody>
</table>

- Proteins: double every 2 years
- comparisons: >20-fold every 2 years
- Overall increase: 47 fold in 55 months
### Big data - big CPU

<table>
<thead>
<tr>
<th></th>
<th>Proteins</th>
<th>comparisons</th>
<th>we need</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009/09</td>
<td>$8 \cdot 10^6$</td>
<td>$64 \cdot 10^{12}$</td>
<td>5 TB</td>
</tr>
<tr>
<td>2014/04</td>
<td>$55 \cdot 10^6$</td>
<td>$3025 \cdot 10^{12}$</td>
<td>300 TB</td>
</tr>
</tbody>
</table>

- Double every 2 years
- 20-fold every 2 years
- >5-fold every 2 years

© Burkhard Rost
TUM

115/160
Issues: Dynamic programming

- time used?
  order of length^2 -> long

- how to choose parameters?
How to speed align?
Alignment: hashing
(fast & dirty)
BLAST: fast matching of single ‘words’

TTYKLILNGKTLKGETTTEAVIDAATAEKFQYANDNGVGEWTYDDATKTFTVTEK
TTYKLILLLLLLLLLLLLLLLLLLAWTVEKAFKFTHAAAWTVEKAFKFTHAAA

#1 seed=3

Default “word” size for “seeds” = 3
BLAST: fast matching of single ‘words’

Default “word” size for “seeds” = 3
BLAST: fast matching of single ‘words’

TTYKLI LNLGKTLKGETTTEAVDAATAEKVFKQYANDNGVDGEWTYDDATKFTFVTTEK
TTYKLI LLLLLLLLLLLLLLLLLLLAWTVEKAFKTFAAAAAAAAAAWTVEKAFKTFAAAAA

#1 seed=3

#2 extend

Default “word” size for “seeds” = 3
Word searches: challenge: statistics

the major challenge for word search algorithms is to get the statistics right
Significance of match (e.g. BLAST E-values)
How accurate are pairwise alignments?
Annotation transfer

similar sequence \rightarrow similar function

similar sequence \rightarrow similar structure
Sequence -> Structure

Sequence folds into unique structure

S -> T
Sequence -> Structure

- Sequence folds into unique structure
  \[ S \rightarrow T \]
- Similar sequences fold into similar structures
  \[ S + S' \rightarrow T \]
Sequence -> Structure

- Sequence folds into unique structure
  \[ S \rightarrow T \]

- Similar sequences fold into similar structures
  \[ S + S' \rightarrow T \]

- Most sequences don’t fold, at all
  \[ S \rightarrow \text{no } T \]
Zones

Zones

Zones

Schematic: structural similarity in Twilight zone

True Positives = pairs of proteins with similar structure

Percentage: accuracy -or- specificity

not true data: just to illustrate the idea of the twilight zone
Schematic: true and false in Twilight zone

True Positives = pairs of proteins with similar structure
False Positives = pairs of proteins with DIFFERENT structure

Number of pairs:

<table>
<thead>
<tr>
<th>Number of pairs</th>
<th>True Positives</th>
<th>False Positives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Burkhard Rost
How to assess alignment accuracy?
All-vs-all: PDB

3D = structural alignment

1D = sequence alignment

proteins of known 3D structure (PDB)

<0.2nm rmsd — SAME 3D
in between  ____ ignore
>0.5nm rmsd  ____ DIFFER in 3D
PDB all-against-all ok?

proteins of known 3D structure (PDB)
Databases biased: MUST remove bias!
Original HSSP curve

$log_{10}(\frac{PIDE}{1-PIDE})$ vs. number of residues aligned

$PIDE = \text{Percentage pairwise sequence identity}$

saturation

Chris Sander
SKCC New York

Reinhard Schneider
Univ Luxembourg

HSSP curve revised with 10x data

\[ \text{FINAL IDE-chosen: } 480 L^{-0.32}(1 + 1/e^{L/1000}) \]

\[ \text{PIDE = Percentage pairwise sequence identity} \]

\[ \text{Number of residues aligned} \]

B Rost 1999 Prot Engin 12, 85-94
Sequence conservation of protein structure

C Sander & R Schneider 1991 *Proteins* 9:56-69
B Rost 1999 *Prot Engin* 12, 85-94
Structure prediction using homology

C Sander & R Schneider 1991 *Proteins* 9:56-69
Structure prediction using homology

C Sander & R Schneider 1991 *Proteins* 9:56-69

Raw data: density? lesson learned?

C Sander & R Schneider 1991 *Proteins* 9:56-69
B Rost 1999 *Prot Engin* 12, 85-94
How to estimate performance from the curves?
Distance from new HSSP-curve

C Sander & R Schneider 1991 Proteins 9:56-69
B Rost 1999 Prot Engin 12, 85-94
Twilight zone = true positives explode

Number of protein pairs

Distance from HSSP threshold

Percentage sequence identity

Number of residues aligned

Sequences similar

© Burkhard Rost
Twilight zone = false positives bazoom!!

B Rost 1999 Prot Engin 12, 85-94
So far: pairwise - anything more 2 do?
Triangle beyond reach

protein A

protein B

protein C
Sequence Space Hopping

protein A

seq_x
seq_y
sel_x

X

protein B
anb_x
unk_x
unk_y

protein C
cal_x

cal_y

X

X

© Burkhard Rost
Success through sequence space hopping

intermediate sequences

Percentage sequence identity

Cumulative % of true pairs

Distance from threshold

Percentage of cumulative true positives

Distance from threshold

© Burkhard Rost

ROSTLAB
Sequence comparisons: multiple alignment
Multiple alignments

Dynamic programming?
for 3 sequences: $O(N_1 \times N_2 \times N_3)$
NP-complete (L Wang & T Jiang (1994) JCB 1: 337-48)
Multiple alignments

Dynamic programming?
for 3 sequences: $O(N_1 \times N_2 \times N_3)$
NP-complete (L Wang & T Jiang (1994) JCB 1: 337-48)

claim: computer: up to 6
~60 TB main memory
no quote-> unsure

```
<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>Q</th>
<th>L</th>
<th>A</th>
<th>K</th>
<th>E</th>
<th>E</th>
<th>A</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Multiple alignments

- Dynamic programming?
  for 3 sequences: $O(N_1 \times N_2 \times N_3)$
  NP-complete (L Wang & T Jiang (1994) JCB 1: 337-48)

- hack 1:
  dynamic programming: pairwise, only space in vicinity of intersection searched n-wise
Multiple alignments

☐ Dynamic programming?
  for 3 sequences: $O(N_1 \times N_2 \times N_3)$
  NP-complete (L Wang & T Jiang (1994) JCB 1: 337-48)

☐ hack 1:
  dynamic programming: pairwise, only space
  in vicinity of intersection searched n-wise

☐ hack 2:
  map to tree / pairwise
Russell Doolittle, UCSD
## Multiple alignment: progressive 1

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>90</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alignment:

- A: GGQLAKEEAL
- B: GGQLAKDEAL
- C: GGQIAKDEAL
- D: GGQIAKDEAI
Multiple alignment: progressive

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>90</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 1

GGQLAKEEAL
GGQLAKDEAL
GGQIAKDEAI

Step 1

GGQLAKEEAL
GGQLAKDEAL
ggqlakeeal
## Multiple alignment: progressive

### Table

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>90</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Steps

**Step 1**

- GGQLAKEEAL
- GGQLAKDEAL
- ggqlakeeal

**Step 2**

- GGQIAKDEAL
- GGQIAKDEAI
- ggqiaakdeal
Multiple alignment: progressive 1

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>90</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 1
GGQLAKEEAL
GGQIAKDEAL
ggqlakeeal

Step 2
GGQIAKDEAL
GGQIAKDEAI
ggqiaakdeal

Step 3
GGQIAKDEAL
GGQIAKDEAI
ggqiaakdeal
Multiple alignment: progressive 2

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>90</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 1
GGQLAKEEAL
GGQLAKDEAL
ggqlakeeal

GGQLAKEEAL
GGQLAKDEAL
GGQIAKDEAL
GGQIAKDEAI

Step 1
Multiple alignment: progressive 2

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>90</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 1
GGQLAKEEAL
GGQLAKDEAL
ggqlakeeal

Step 2
GGQIAKDEAL
GGQIAKDEAI
gqqlakeeal
Zones

Multiple alignment: progressive 2

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>90</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 1
- GGQLAKEEAL
- GGQLAKDEAL
- ggqlakeeal

Step 2
- ggqlakeeal
- GGQIAKDEAL
- ggqlakeeal

Step 3
- ggqlakeeal
- GGQIAKDEAI
- GGQIAKDEAI
Lecture plan (CB1 structure)

01: 04/11 Tue: no lecture
02: 04/13 Thu: no lecture
03: 04/19 Tue: Organization of lecture: intro into cells & biology
04: 04/21 Thu: Intro I - acids/structure - domains
05: 04/26 Tue: Alignment 1
06: 04/28 Thu: Alignment 2
07: 05/03 Tue: Alignment 3
08: 05/05 Thu: SKIP: Ascension Day
09: 05/10 Tue: SKIP: student assembly (SVV)
10: 05/12 Thu: Comparative modeling
11: 05/17 Tue: SKIP: Whitsun holiday (05/15-17)
12: 05/19 Thu: Experimental structure determination / 3D -> 1D: Secondary structure assignment
13: 05/24 Tue: 1D: Secondary structure prediction 1
14: 05/26 Thu: SKIP: Corpus Christi
15: 06/31 Tue: 1D: Secondary structure prediction 2
16: 06/02 Thu: 1D: Transmembrane structure prediction 1
17: 06/07 Tue: 1D: Transmembrane structure prediction 2 / Solvent accessibility prediction
18: 06/09 Thu: 2D prediction 1
19: 06/14 Tue: 2D prediction 2 - Thomas Hopf
20: 06/16 Thu: 3D prediction / Nobel prize symposium
21: 06/21 Tue: 1D: Disorder prediction
22: 06/23 Thu: recap 1
23: 06/28 Tue: recap 2
24: 06/30 Thu: examen, no lecture
25: 07/05 Tue: examen alternative, no lecture
26: 07/07 Thu: examen, no lecture
27: 07/12 Tue: wrap up exercises - no lecture
28: 07/14 Thu: wrap up exercises - no lecture