Administration

- Sprache ?
  - zu verhandeln (Englisch, Hochdeutsch, Bayerisch)
- Selection of topics
  - Proteins / DNA / RNA
- Two themes
  - Torda: larger molecules, proteins
  - Rarey: Chemoinformatics, Wirkstoff Entwurf
  - Woche (12 Termine)
    - 1 – 3 Torda
    - 4 – 6 Rarey
    - 7 – 9 Torda
    - 10 – 12 Rarey

Administration

- Who are we ? (Torda parts)
    - Andrew Torda
    - + Gundolf Schenk
    - + Thomas Margraf
- Where am I
    - 42838 7331
    - ZBH 1st floor (Bundesstr. 43)
- Background
    - numerical simulations
- Administrative helper
    - Annette Schade

Course Themes

- What we omit
  - genomics, numerical simulations, gene finding, proteomics,…
- What we will do
  - Similarities in sequences
    - finding and assessing similarities
- Different kinds of predictions

Predictions

- what shape is this molecule ?
- will this small molecule inhibit some enzyme ?
- will this molecule be broken down in the body quickly ?

…

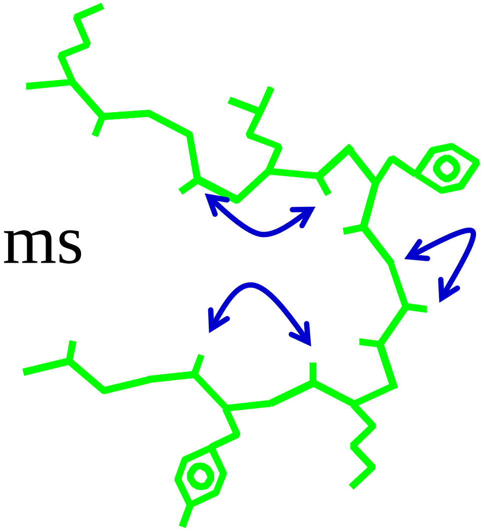
Predictions – different approaches

- First principles (physics, chemistry)
- Finding patterns (underlying principles not known)
- Similarity

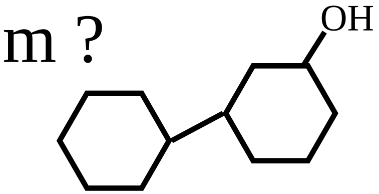
… explanation

First principles prediction

- protein structure example
  - a protein molecule = set of atoms in space
  - I know all the interactions between the atoms
  - should be able to predict the 3D structure



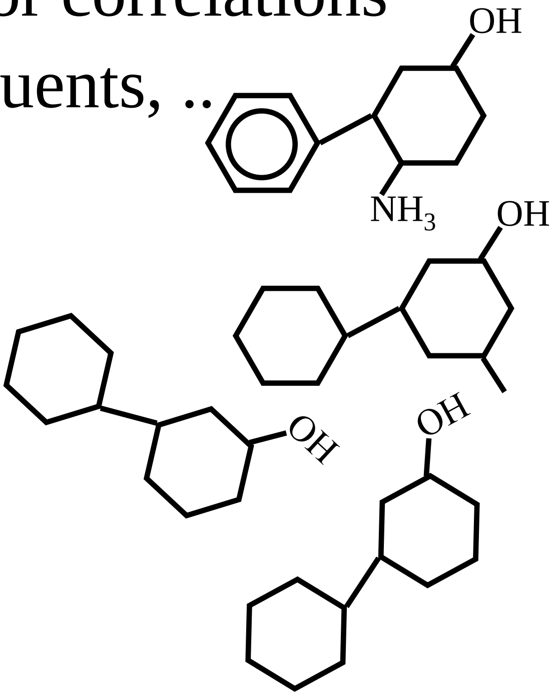
- quantum chemistry
  - I have a model for electron wave functions
  - can I predict electron density around each atom ?
  - predict pK_a for this molecule ?
  - …



- Maybe best method
  - elegant, expensive, needs good models

Finding patterns

- Take known data – collect properties, look for correlations
  - look at mol wt, aromatic/aliphatic, substituents, ..
  - for each molecule collect pK_a
  - hope patterns can be found



- gene regulator recognition
  - take known examples
    - look at GC content
    - proximity to protein
    - sizes …

- field of "data mining", machine learning
- often little understanding of problem / chemistry
- often works

Similarity

- Answer to many questions..
  - DNA
    - is this region coding ?
    - where does the reading frame start ?
    - is this region involved in regulator binding ?
  - protein sequence
    - can one guess the structure ?
    - is this membrane bound ?
    - does it have a certain activity (kinase, transferase, ..) ?
  - protein structure (maybe from structural genomics)
    - what is a likely function ?
  - from proteomics, we know the N-terminal 6 residues
    - what protein could it be ?

Prediction by similarity

- For some examples
  - solve structure of a protein
  - find DNA which binds to regulators
  - measure that RNA has enzymatic activity

} slow, expensive
must be done

- For some queries / your sequence
  - is your protein sequence similar to a known structure ?
  - is your stretch of DNA similar to a known regulatory region ?
  - is your RNA similar to some RNAzyme ?

- why is experiment it so slow and expensive ?

Real experiments

- very problem specific
- DNA – to find function ? make knockouts
    - essential (bad news)
    - involved in regulation – still more measurements
    - involved in some pathway
- Protein – usually has to be cloned, expressed, ..
    - function *in vitro*, *in vivo*
    - structure from NMR, crystallography
- RNA
    - how do you show it is involved in regulation (assays ?)
    - how can you show it is a riboswitch ?
    - structures difficult

Similarity in sequences

- Protein / nucleotide
  - same ideas, differences later
- Questions
  - are two sequences similar ?
  - suspected similarity
    - how reliable is it ?
  - detailed alignments (modelling, important residues, ..)
- Plan
  - generalities
  - alignment methods
  - DNA versions
  - Protein versions
  - differences

Alignments and Similarities

- Problem

```
. . . A C A C T G A C T A . . .
. . . . . A T T G A G T A . . .
. . . . . 1 0 1 1 1 0 1 1 . . .
```

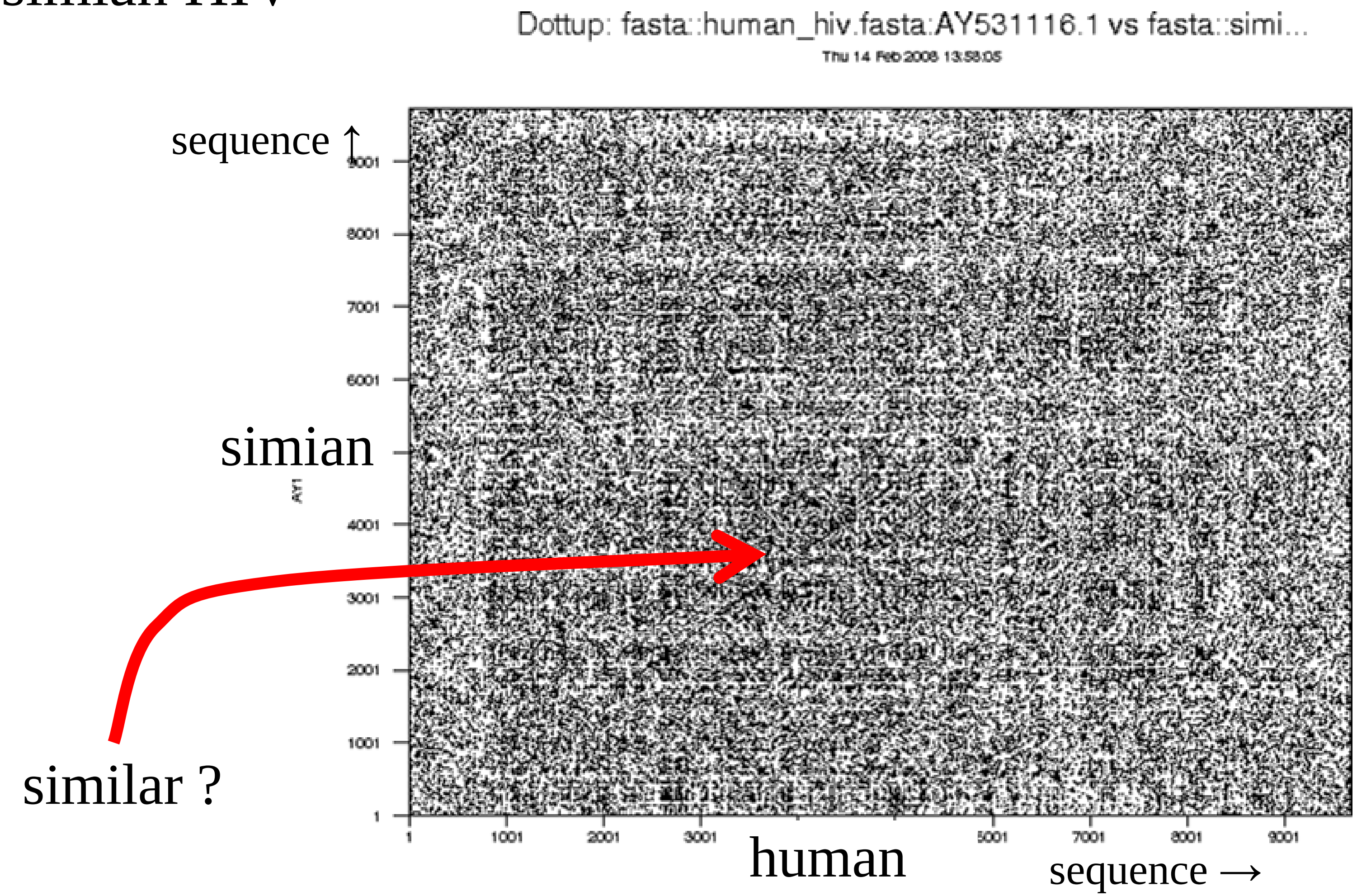
- 4 of 8 positions match
- implicit
  - I have already moved second sequence over the first
- gaps

```
. . . A C A C T T G A C T A . . .
. . . . . A T T - G A G T A . . .
. . . . . 1 0 1 1 1 0 1 1 . . .
```

- alignment not so obvious (gaps anywhere)
  - quick look

dot plot

- human and simian HIV



sequence ↑

simian

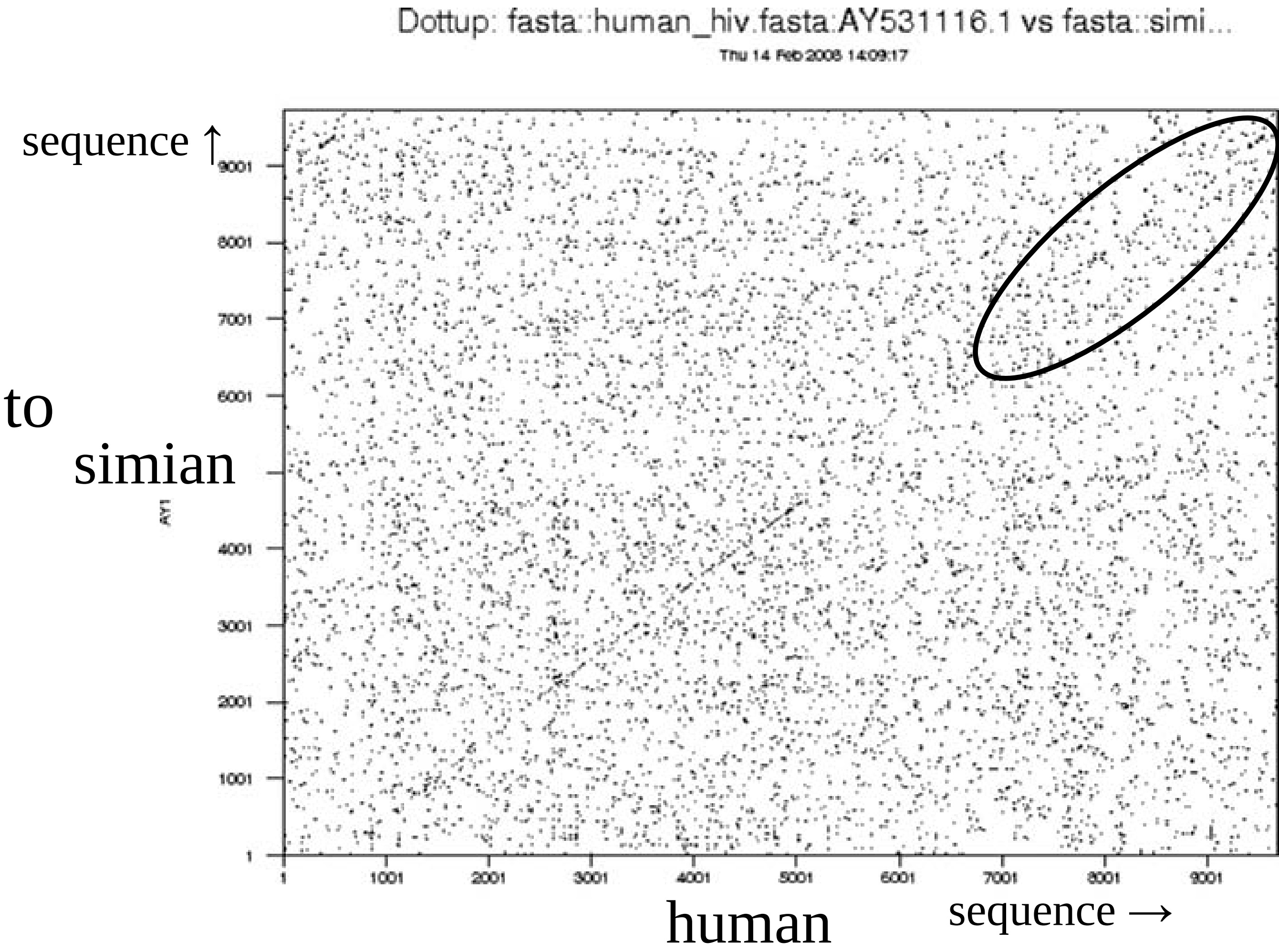
similar ?

human

sequence →

dot plot filtered

- similarity up to about 5200
- circled region ?
  - not so clear
- easy for a human to recognise
- not so easy to automate
- worse case …
  - two protein sequences



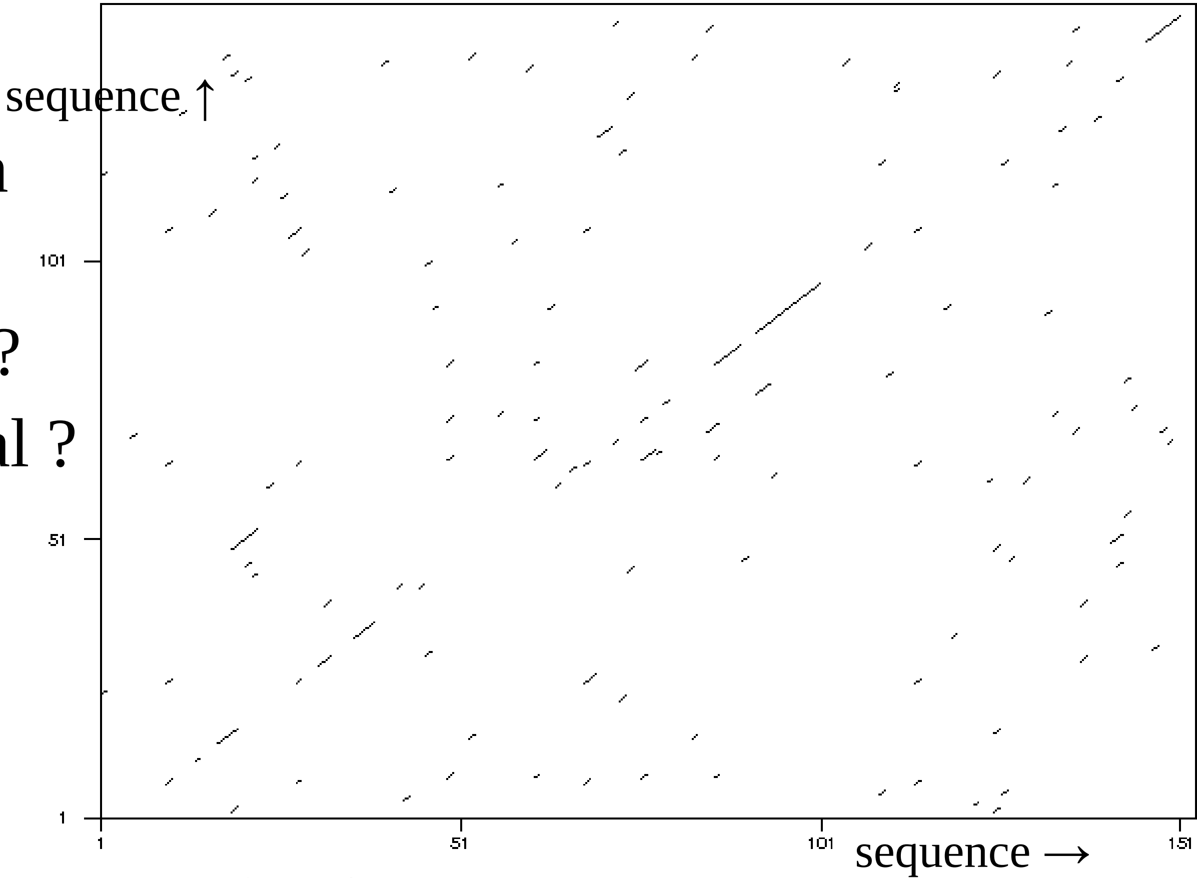
protein dot plot

2 proteins

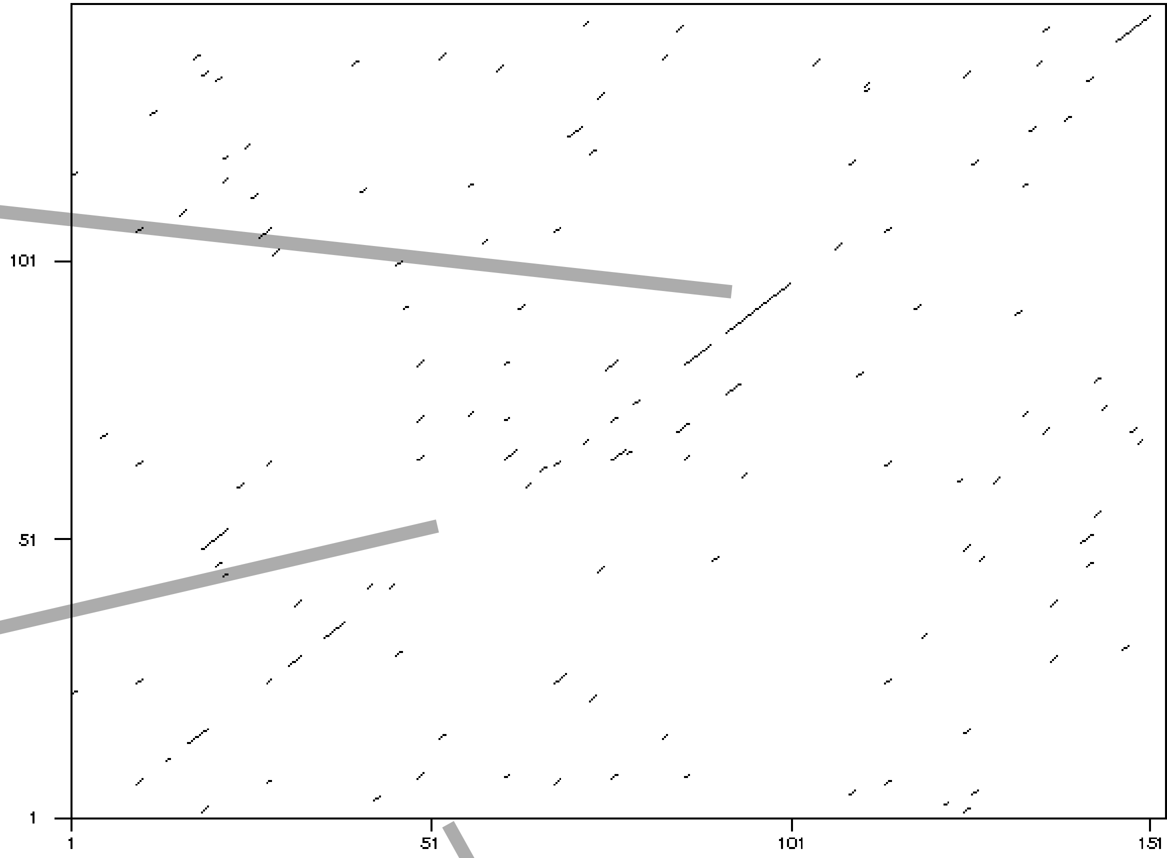
- 2nrl, 2o58
- tuna / horse myoglobin

- are they really similar ?
- how real is the diagonal ?

- what is the identity ?
  - ≈ 45 %
- how similar are these two proteins ?
- is there a "correct alignment" ? Physical interpretation ?



Properties of alignment ?



Is the alignment above diagonal ?

What is happening here ?

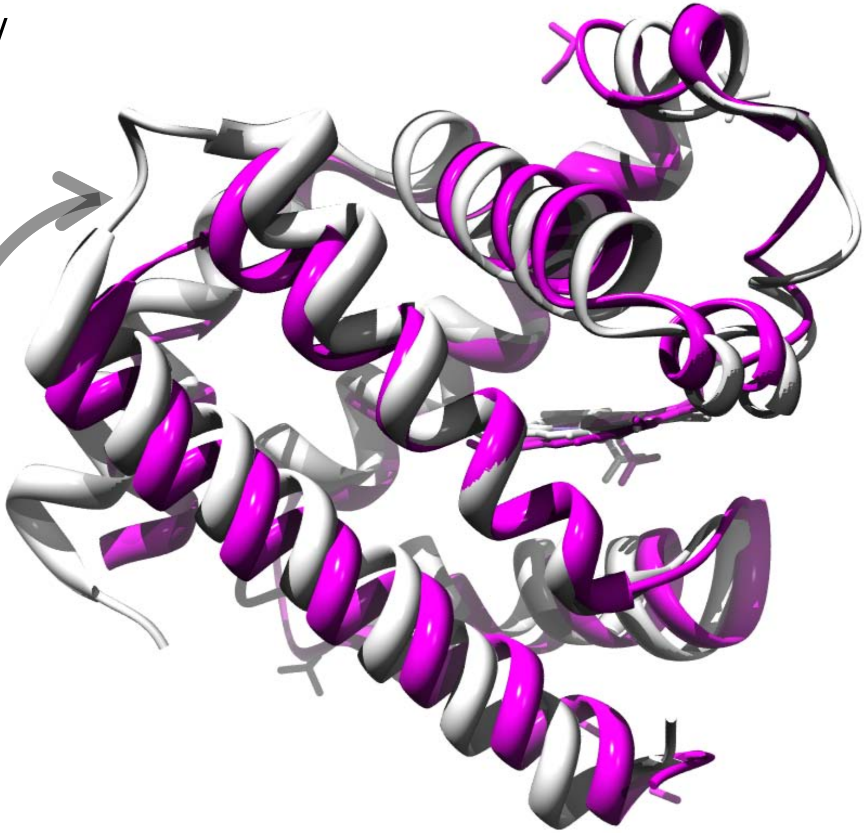
What is aligned to residue 51 ?

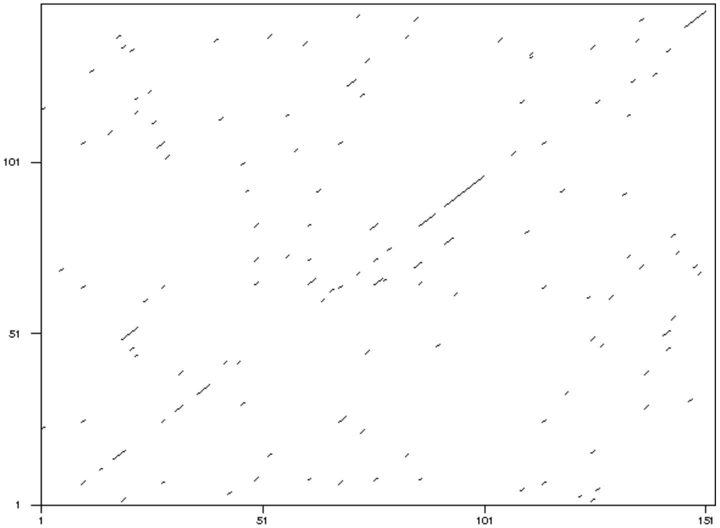
- here we know the answer
  - look at structures

correctness of alignment

- The same proteins as before tuna / horse myoglobin

- there are no holes ?
  - there are some differences
    - some bits are longer

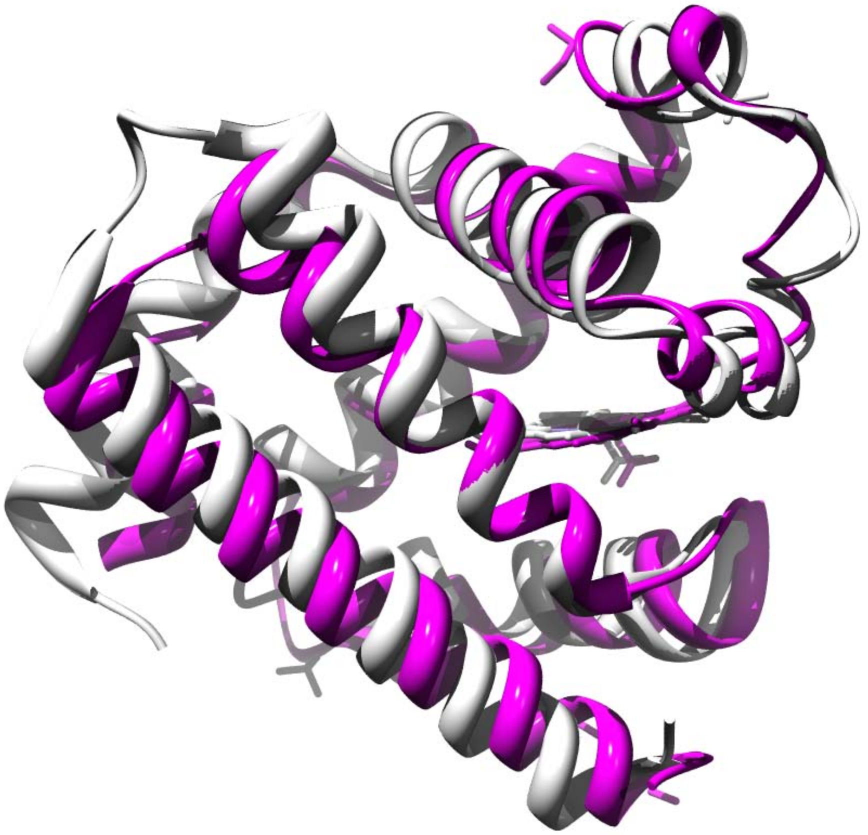


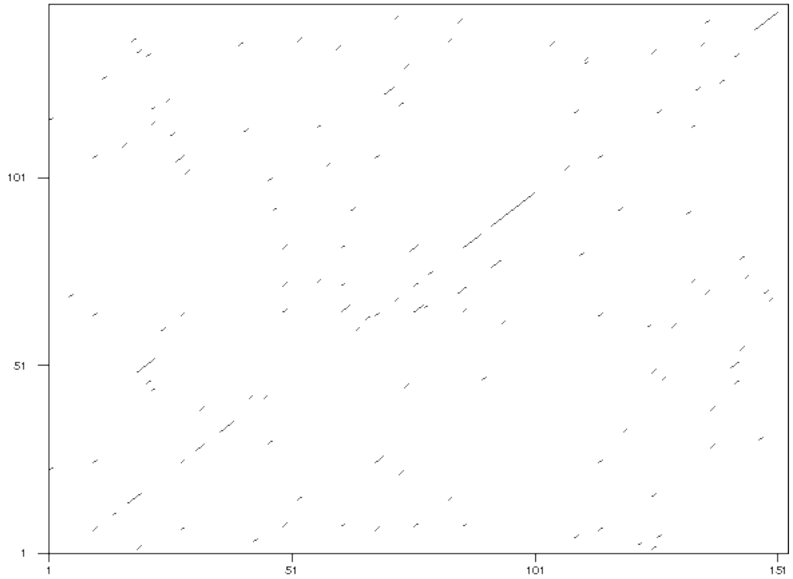


- for almost every pink residue, there is a corresponding grey residue

If one knew the structure..

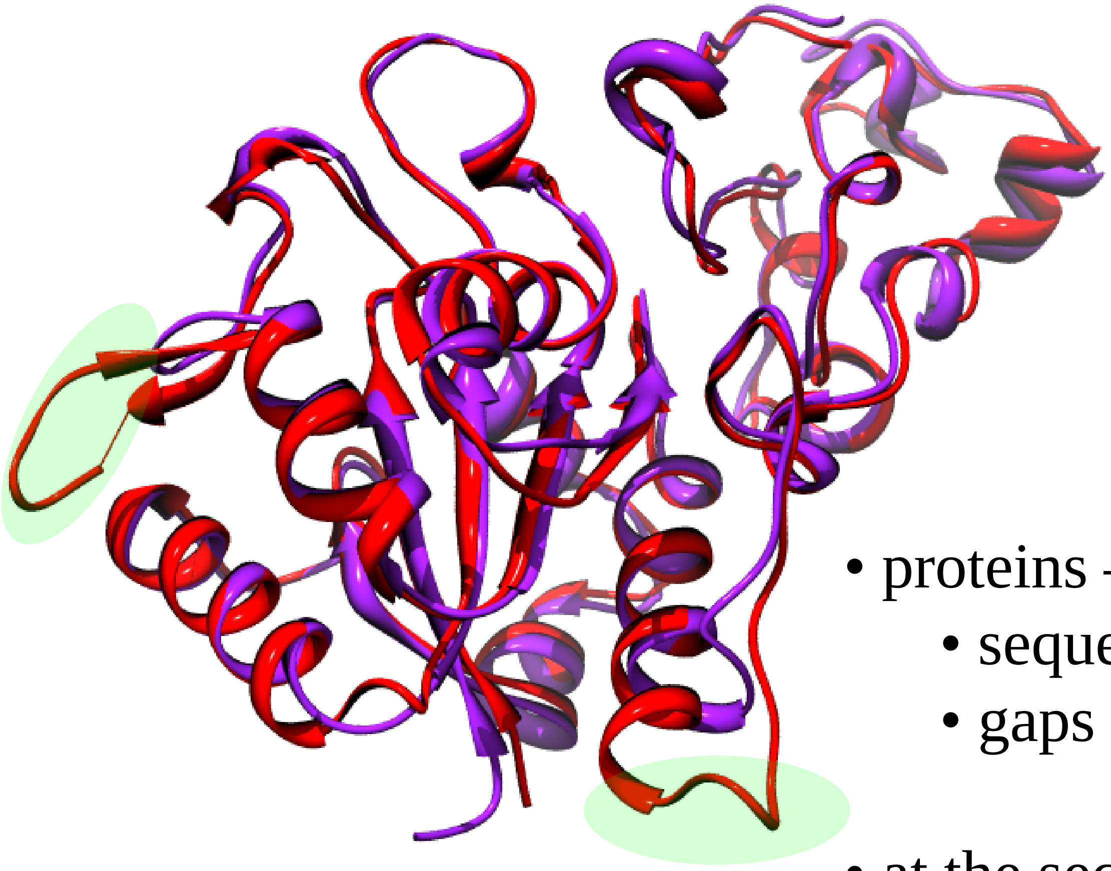
- would you have recognised this from dotplot ?





- look at residue 51 in dot plot
    - aligned residue not clear
- look in structure
    - aligned residues clear

Clearer Example



- hydrogenases
  - 40 % sequence identity
  - 2frvG & 1cc1S

- proteins – obviously similar
  - sequence identity OK
  - gaps and insertions

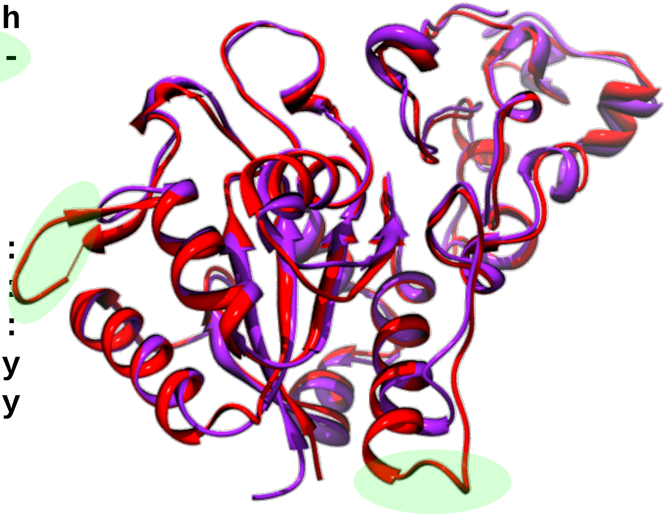
- at the sequence level ?

```
Seq ID 40.6 % (103 / 254) in 280 total including gaps
    :    1    :    2    :    3    :    4    :    5    :    6
    :    0    :    0    :    0    :    0    :    0    :    0
kkapviwvqgqgctgcsvsllnavhprikeilldvislefhptvmasegemalahmyeia
krpsvvylhnaectgcsesvlrtvdpyvdelildvismdyhetlmagaghaveea-l-he
   :    1    :    2    :    3    :    4    :    5    :
   :    0    :    0    :    0    :    0    :    0    :

    :    0    :    0    :    0    :    1    :    1    :    1
    :    7    :    8    :    9    :    0    :    1    :    2
    :    0    :    0    :    0    :    0    :    0    :    0
ekfngnffllvegaiptakegrycivgeakahhhevtmmelirdlapkslatvavgtcsa
aikg-dfvcvieggipmgdggywgk-------vggrnmydicaevapkakaviaigtcat
0     :    0    :    0           :    0    :    1    :    1
6     :    7    :    8           :    9    :    0    :    1
0     :    0    :    0           :    0    :    0    :    0

    :    1    :    1    :    1    :    1    :    1    :    1
    :    3    :    4    :    5    :    6    :    7    :    8
    :    0    :    0    :    0    :    0    :    0    :    0
yggipaaegnvtgsksvrdffadekiekllvnvpgcpphpdwmvgtlvaawshvlnpteh
yggvqaakpnptgtvgvnealgklgvkai--niagcppnpmnfvgtv--vhllltk-----
   :    1    :    1    :    1      :    1    :      1
   :    2    :    3    :    4      :    5    :      6
   :    0    :    0    :    0      :    0    :      0

    :    1    :    2    :    2    :    2         :    2    :
    :    9    :    0    :    1    :    2         :    3    :
    :    0    :    0    :    0    :    0         :    0    :
plpeldddgrpllffgdnihencpyldkydnsefaetftkpg-----ckaelgckgpsty
gmpeldkqgrpvmffgetvhdncprlkhfeagefatsfgspeakkgyclyelgckgpdty
  :    1    :    1    :    1    :    2    :    2    :    2
  :    7    :    8    :    9    :    0    :    1    :    2
  :    0    :    0    :    0    :    0    :    0    :    0
```



Sequence versus structure

- If we know the structure
  - easier to find a "correct" alignment

- do we always know the structure ?
  - if so, we would not do these lectures
  - sequences are cheap
  - structures are expensive

- how bad can alignments be ? (and still sensible)
- mission for today ?
  - how does one find the best alignment based on sequence

Why ?

Where this is going to

- how to exploit sequence information
- how to get alignments
    - easy – hard
- aim
    - find similarities / get information about a new protein

Alignment methods

- best alignment not obvious

```
. . . . . . . C C A T C C G C . .
. . . C G A T C C – T C C T C . . .
```

- 6 matches or

```
. . . . . . . C C A T C C G C . . . .
. . . . . . . C G A T C C T C C T C . .
```

- also 6 matches

- can we invent some rules to say which is best ?

Simple scoring

- For two sequences of length 10, how many alignments could I generate ?

```
. . . . . . . A B C D E F G H I J . . . .
              Q R S T U V W X Y Z
. . . . . . Q R S T U V W X Y Z + more
```

with gaps

```
            Q R S T U V W X Y – Z
            Q R S T U V W X – Y Z  then with gap 2
          Q R S T U V W X Y - - Z
              ...
```

- then with multiple gaps … combinatorial explosion
- do not tackle the problem directly

Mission

- For DNA, protein, RNA
  - develop some scoring scheme
  - maximize matches and similarities
- algorithm
  - allow some gaps, not too many
  - must be much faster than brute force
  - these methods apply to proteins and nucleotides

- What is coming
  - simple scoring –DNA
  - full alignment algorithm (Needleman and Wunsch)
  - better scoring – proteins

Scoring for DNA

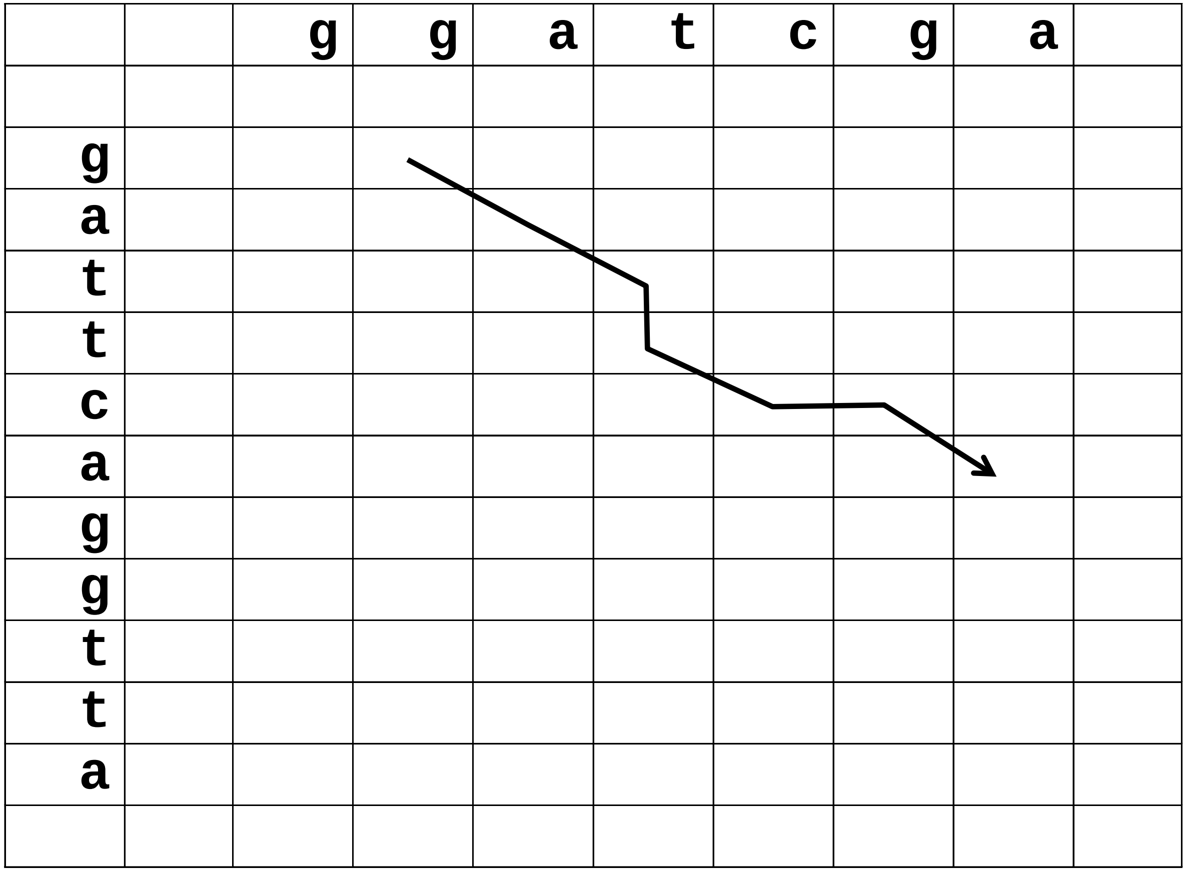
- Sensible scheme
  - matched pairs 2
  - mismatch −3
  - gaps −2

```
A C T G  - A  T T C G A
A C - G  C A  - T C T A
2 2-2 2 -2 2 -2 2 2-3 2
```

- more sophisticated..
  - gap opening costs − 2
  - gap widening costs − 1
  - so $cost = cost_{open} + (n_{gap} - 1)cost_{widen}$

Representing alignments

- sequences GATTCAGGTTA and GGATCGA



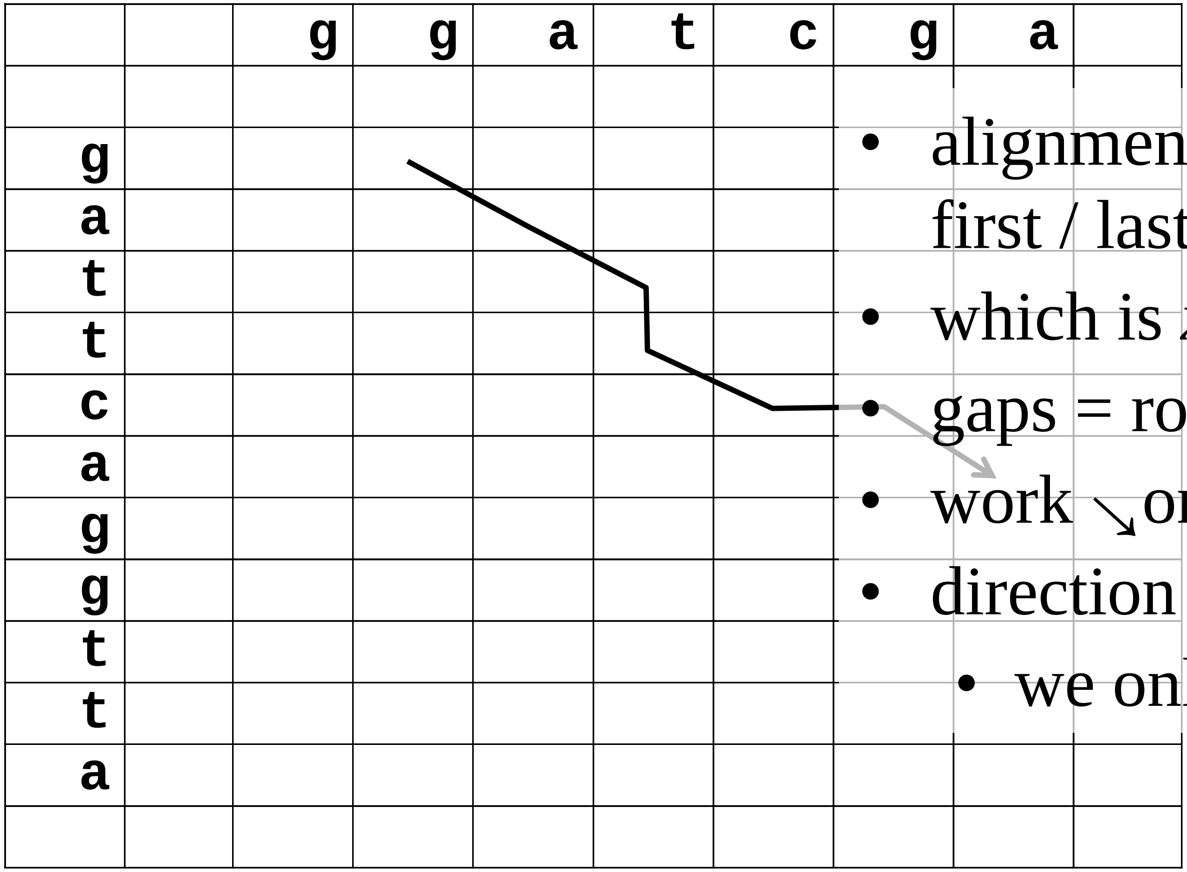
		g	g	a	t	c	g	a	
g									
a									
t									
t									
c									
a									
g									
g									
t									
t									
a									

- would mean
  ```
  GGAT-CGA-----
  -GATTC-AGGTTA
  ```

- notes…

Representing alignments

```
GGAT-CGA-----
-GATTC-AGGTTA
```

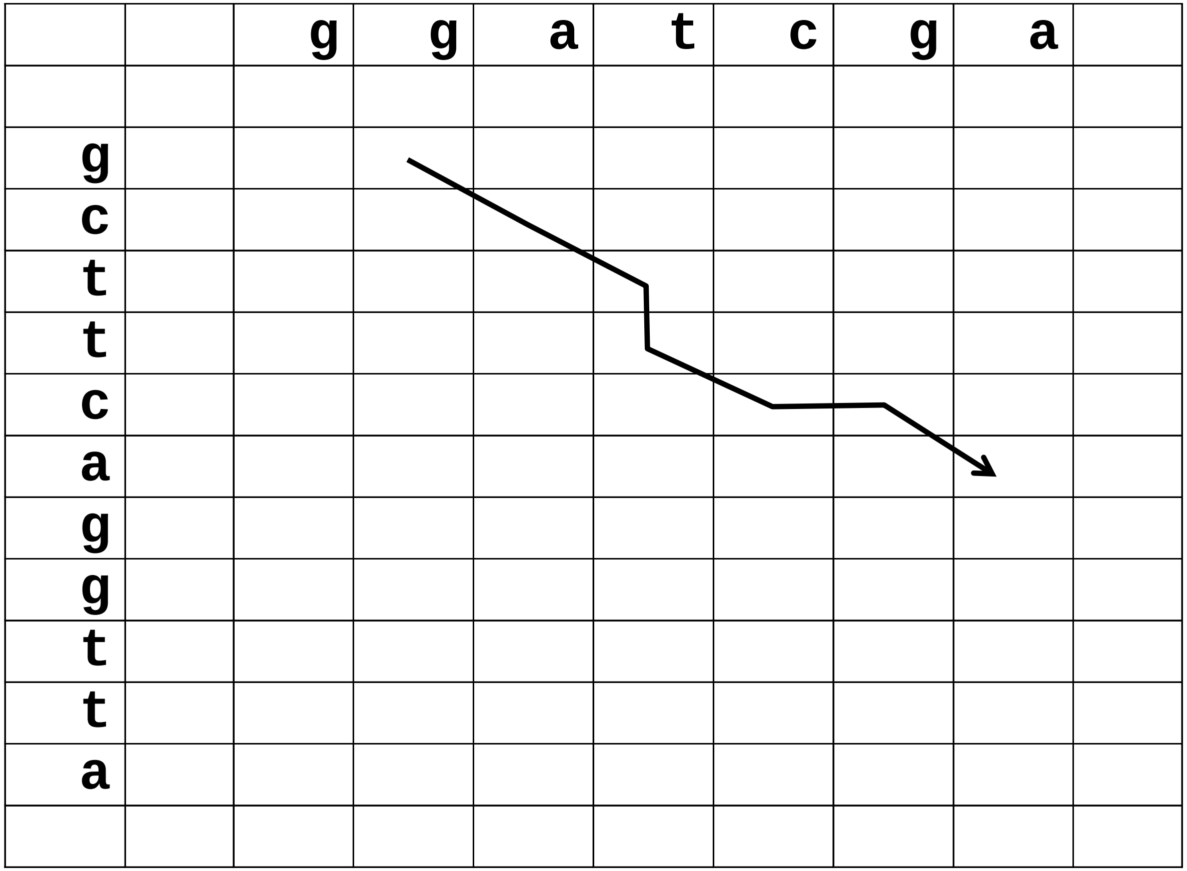


- alignment does not have to go to first / last row or column
- which is *x* and *y* is arbitrary
- gaps = row or column is skipped
- work ↘ or ↖ does not matter
- direction must be consistent
  - we only go → ↓ ↘

- make sure this is clear

Representing alignments with a mismatch

- sequences `GCTTCAGGTTA` and `GGATCGA`



		g	g	a	t	c	g	a	
g									
c									
t									
t									
c									
a									
g									
g									
t									
t									
a									

- would mean

```
GGAT-CGA-----
-GCTTC-AGGTTA
```

Calculating alignment - steps

Needleman and Wunsch algorithm

1. fill score matrix
2. find best score possible in each cell
3. traceback

fill score matrix

- For convenience, add some zeroes to the ends
- Add in match, mismatch scores

		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0								0
a	0								0
t	0								0
t	0								0
c	0								0
a	0								0
g	0								0
g	0								0
t	0								0
t	0								0
a	0								0
	0	0	0	0	0	0	0	0	0

Mission

- find path through this matrix with best score
- account for gaps

		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-3	2	-3	-3	-3	2	0
t	0	-3	-3	-3	2	-3	-3	-3	0
t	0	-3	-3	-3	2	-3	-3	-3	0
c	0	-3	-3	-3	-3	2	-3	-3	0
a	0	-3	-3	2	-3	-3	2	2	0
g	0	2	2	-3	-3	-3	2	-3	0
g	0	2	2	-3	-3	-3	2	-3	0
t	0	-3	-3	-3	2	-3	-3	2	0
t	0	-3	-3	-3	2	-3	-3	-3	0
a	0	-3	-3	2	-3	-3	-3	2	0
	0	0	0	0	0	0	0	0	0

Summing the elements

- start at top left
- move right, then next line
- at each cell
  - find best score it could possibly have

		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-1	4	-3	-4	-5	4	0
t	0	-3	-3	-3	6	-1	-2	-3	4
t	0	-3	-4	-4	4	3	1	0	2
c	0	-3	-5	-5	-2	6	0	-2	1
a	0	-3	-5	-6	-3	0	3	6	3
g	0	2	0	-6	-4	-1	6	0	6
g	0	2	4	-3	-4	-2	5	3	4
t	0	-3	-1	1	4	-2	-1	2	3
t	0	-3	-3	-1	3	1	-1	0	2
a	0	-3	-4	3	-4	0	-2	4	0
	0	0	-2	0	3	1	0	1	4

Diagonal (no gaps)

for each cell, 3 possible scores

1. **diagonal (no gap)**

2. best from preceding column

3 best from preceding row

		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-1	4	-3	-4	-5	4	0
t	0	-3	-3	-3	6	-1	-2	-3	4
t	0	-3	-4	-4	4	3	1	0	2
c	0	-3	-5	-5	-2	6	0	-2	1
a	0	-3	-5	-6	-3	0	3	6	3
g	0	2	0	-6	-4	-1	6	0	6
g	0	2	4	-3	-4	-2	5	3	4
t	0	-3	-1	1	4	-2	-1	2	3
t	0	-3	-3	-1	3	1	-1	0	2
a	0	-3	-4	3	-4	0	-2	4	0
	0	0	-2	0	3	1	0	1	4

GAT
GAT

GG
GG

preceding row (gap)

for each cell, 3 possible scores

1. diagonal (no gap)
2. **best from preceding row**
3. best from preceding column

		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-1	4	-3	-4	-5	4	0
t	0	-3	-3	-3	6	-1	-2	-3	4
t	0	-3	-4	-4	4	3	1	0	2
c	0	-3	-5	-5	-2	6	0	-2	1
a	0	-3	-5	-6	-3	0	3	6	3
g	0	2	0	-6	-4	-1	6	0	6
g	0	2	4	-3	-4	-2	5	3	4
t	0	-3	-1	1	4	-2	-1	2	3
t	0	-3	-3	-1	3	1	-1	0	2
a	0	-3	-4	3	-4	0	-2	4	0
	0	0	-2	0	3	1	0	1	4

```
GAT
G-T
```

preceding column (gap)

for each cell, 3 possible scores

1. diagonal (no gap)

2. best from preceding row

3 **best from preceding column**

		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-1	4	-3	-4	-5	4	0
t	0	-3	-3	-3	6	-1	-2	-3	4
t	0	-3	-4	-4	4	3	1	0	2
c	0	-3	-5	-5	-2	6	0	-2	1
a	0	-3	-5	-6	-3	0	3	6	3
g	0	2	0	-6	-4	-1	6	0	6
g	0	2	4	-3	-4	-2	5	3	4
t	0	-3	-1	1	4	-2	-1	2	3
t	0	-3	-3	-1	3	1	-1	0	2
a	0	-3	-4	3	-4	0	-2	4	0
	0	0	-2	0	3	1	0	1	4

```
T-C
TTC
```

The order of cells

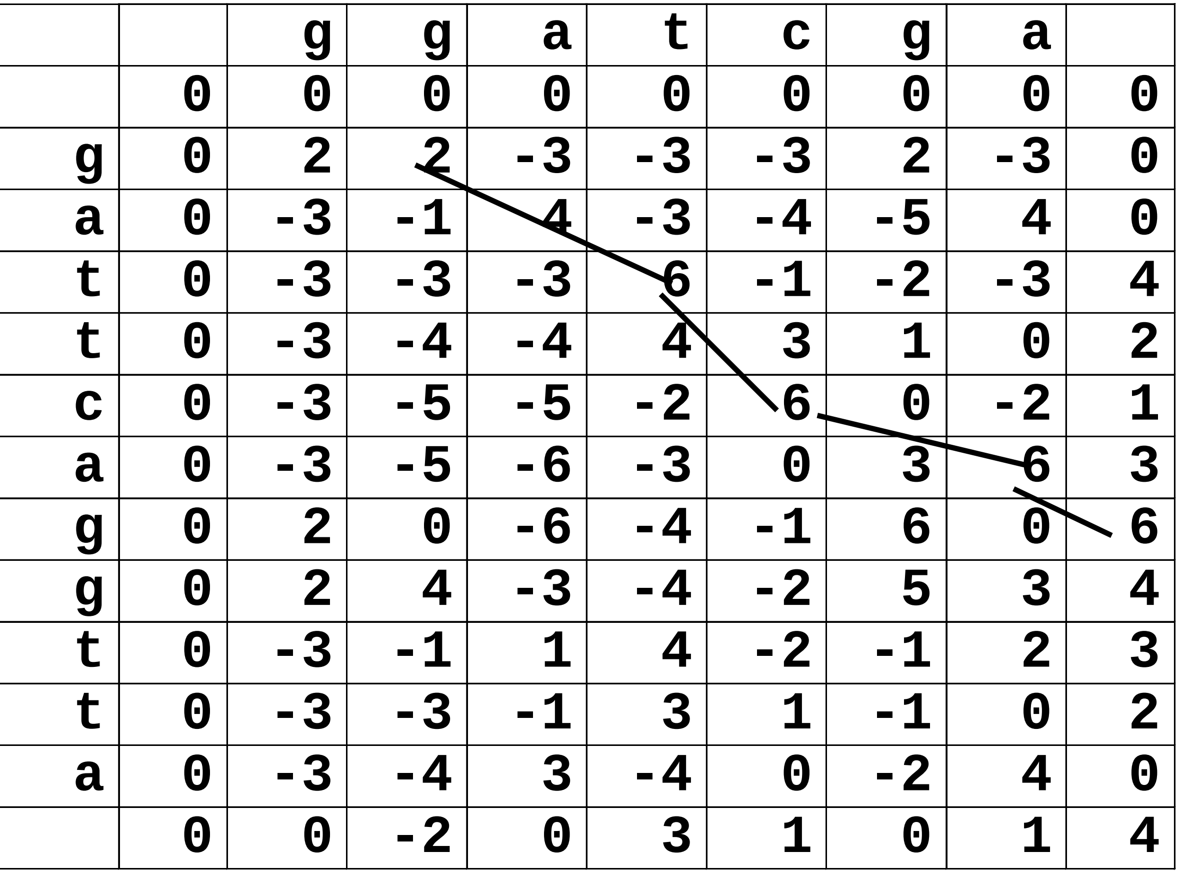
- start at top left
- every cell has best score considering all possible routes
- at end, highest score is best path

		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-1	4	-3	-4	-5	4	0
t	0	-3	-3	-3	6	-1	-2	-3	4
t	0	→							
c	0								
a	0								
g	0								
g	0								
t	0								
t	0								
a	0								
	0								

- would also work if we went left and up

Reading the alignment

- find highest scoring cell (last row or column)
- how did we reach this cell ?
  - how did we reach preceding cell ?
  - …

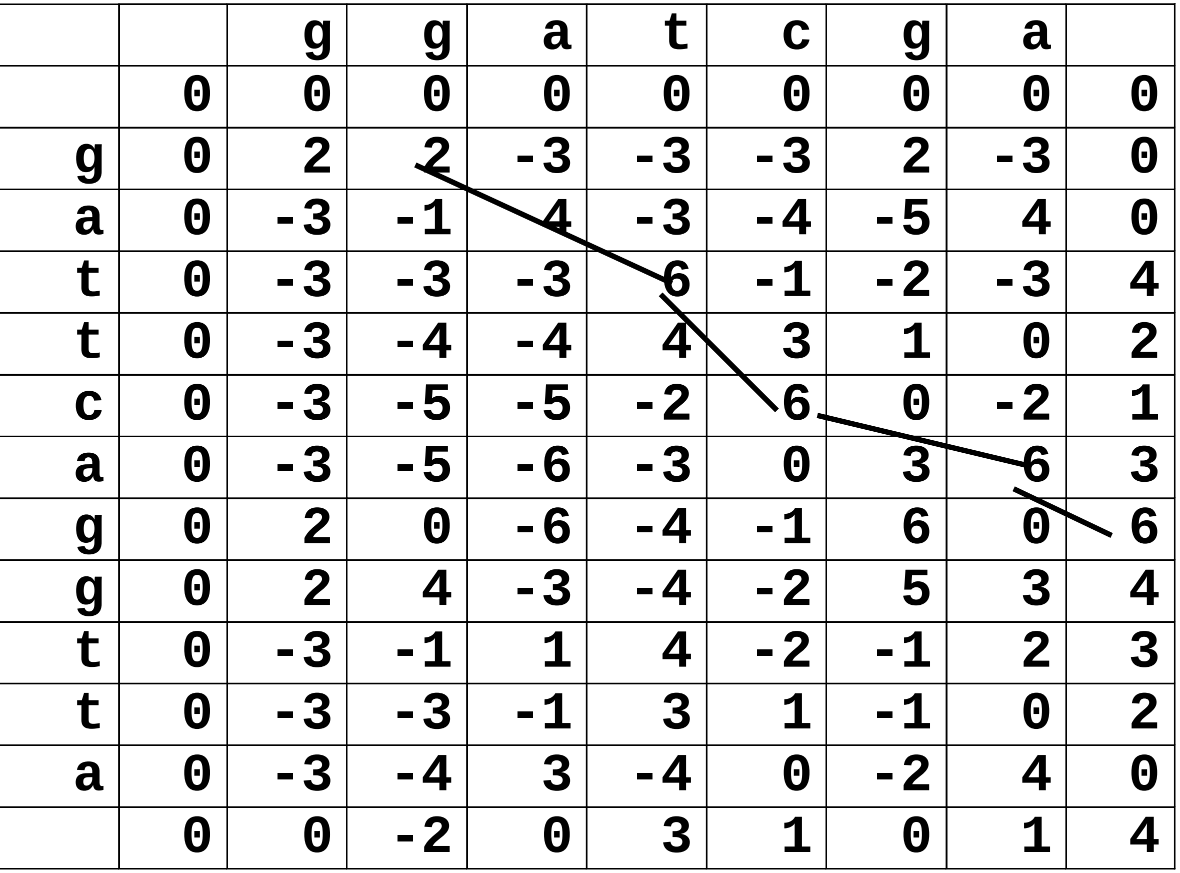


		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-1	4	-3	-4	-5	4	0
t	0	-3	-3	-3	6	-1	-2	-3	4
t	0	-3	-4	-4	4	3	1	0	2
c	0	-3	-5	-5	-2	6	0	-2	1
a	0	-3	-5	-6	-3	0	3	6	3
g	0	2	0	-6	-4	-1	6	0	6
g	0	2	4	-3	-4	-2	5	3	4
t	0	-3	-1	1	4	-2	-1	2	3
t	0	-3	-3	-1	3	1	-1	0	2
a	0	-3	-4	3	-4	0	-2	4	0
	0	0	-2	0	3	1	0	1	4

```
GGAT-CGA
-GATTC-AGGTTA
```

Trick with traceback

- for each cell
  - how did we reach it ? What was the preceding cell ?



		g	g	a	t	c	g	a	
	0	0	0	0	0	0	0	0	0
g	0	2	2	-3	-3	-3	2	-3	0
a	0	-3	-1	4	-3	-4	-5	4	0
t	0	-3	-3	-3	6	-1	-2	-3	4
t	0	-3	-4	-4	4	3	1	0	2
c	0	-3	-5	-5	-2	6	0	-2	1
a	0	-3	-5	-6	-3	0	3	6	3
g	0	2	0	-6	-4	-1	6	0	6
g	0	2	4	-3	-4	-2	5	3	4
t	0	-3	-1	1	4	-2	-1	2	3
t	0	-3	-3	-1	3	1	-1	0	2
a	0	-3	-4	3	-4	0	-2	4	0
	0	0	-2	0	3	1	0	1	4

```
GGAT-CGA
-GATTC-AGGTTA
```

Summary (Needleman and Wunsch)

- Alignments are paths through the matrix
- There is an astronomical number of possibilities (with gaps)
- This algorithm has visited all of them and found best
- allows for gap costs of form $cost = cost_{open} + (n_{gap} - 1)cost_{widen}$
- best or only method ? wait..

Cost

- pretend both sequences are length n
- we have to visit n^2 cells in matrix
  - each time we have to look at a row or column of length $\approx n$
- total cost n^3 or worst cost $O(n^3)$
  - remember this for later

Smith and Waterman version

- So far: global alignments
  - best match, covers as much as possible
- Imagine 3 domain proteins..

```
ABCDEABCDEABCDE
QRSTUVBCDEQRSTU
```

- Want to see …

```
ABCDEABCDEABCDE
      ||||
QRSTUVBCDEQRSTU
```

not worth trying to align everything

- Use “Smith and Waterman” method
  - scoring scheme: matches positive, mismatches negative
  - during traceback
    - do not just look for max score
    - start with positive score
    - stop if score goes negative
- result: “local alignments” – often most useful

Other alignment algorithms

- Needleman and Wunsch / Smith Waterman
  - for given problem – optimal results
  - allow fancy gap penalties
  - cost $O(n^3)$

Other methods

- $O(n^2)$ – very small limitation on gaps

Faster

- …

Faster Seeded Methods

blast, fasta, more

- seeded
  - idea: use seeds / fragments of length *k*
    - 11-28 for DNA
    - 2 to 3 for protein
  - look for exact matches of query words in database
  - extend if found
  - time depends mainly length $O(n)$ – most of the time no matches
  - slow extension when a match is found
- seed size
  - very small = lots of unimportant matches (slow)
  - too big – may miss a match if there are too many changes

Fast versus slow

- 2 sequences (protein or DNA)
    - time not an issue
    - 1000 alignments ? Time still not an issue
    - 10^3 10^3 alignments ? Your decision
- Databases
    - non-redundant protein sequence database
        - ≈ 8 ½ 10^6 sequences
        - ≈ 3 10^9 residues
    - must be fast
    - maybe occasionally miss a word
    - alignments may not be optimal

Problems so far

- We can align DNA sequences – maybe proteins
- how biological are the alignments, gaps and costs ?
- Coding versus non-coding DNA
    - 3 base pairs →1 residue

  `ACAG`… 100's bases … `CGA`…

  `AC-G`… 100's bases … `CGA` …   one base deletion

    - 100's bases are shifted – amino acids in protein all wrong
  - non-coding region (binding / regulation / tRNA / rRNA..
    - may not be so bad
- General problem – degeneracy ..

Degeneracy and Scoring

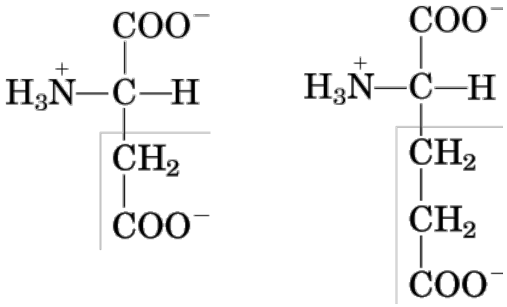
- CCU, CCC, CCA, CCG are all proline (3rd position degenerate)
- CCC→CCA no problem
- CCC→ACC pro → ala (you die)
    - exactly the same mutation at DNA level (C→A)
- our scoring scheme does not know about this
- rule
    - some mutations will have no effect
    - some are drastic
    - usually the third base in each codon is least important
- can we do better ?

Scoring protein alignments

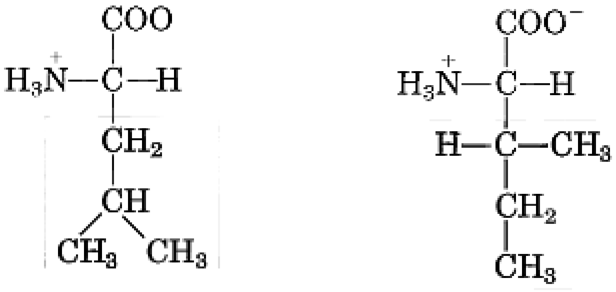
- two aspects
  - forget DNA
  - account for amino acid similarity
- instead of DNA – work directly with protein sequences
- if our DNA is coding – easy to say
  - CCUUCUUAU.. is   pro-ser-tyr…
  - immediate gain
    - CCC→CCA or similar will not be seen
  - more subtle gain

Amino acid similarities

- asp and glu



- think of leu and ile



- many more similar amino acids
- glu →asp mutation, does it matter ? sometimes not
- trp →asp, big hydrophobic to small polar ? usually bad news
- relevance to alignments

Why we need better protein scoring

- ANDREWANDRWANDRWW aligned to QNDRDW

```
ANDREWANDRWANDRWW
QNDRDW-----------
```

```
ANDREWANDR-WANDRWW
------QNDRDW------
```

```
ANDREWANDRWANDRWW
-----------QNDRDW
```

- one of which is biologically more likely (E→D)
- how would we do it numerically ?

Substitution matrices

- Earlier in DNA
  - match = 2
  - mismatch = -3

	A	C	G	T
A	2	-3	-3	-3
C	-3	2	-3	-3
G	-3	-3	2	-3
T	-3	-3	-3	2

- We want a matrix that says

	D	E	W	...
D	10	5	-5	
E	5	10	-5	
W	-5	-5	15	
...				

- A full matrix..

A serious protein similarity matrix

- blosum62:

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

- some features
  - diagonal
  - similar
  - different

Using the score matrix

- Algorithm (global alignment, local, fast, …)
  - unchanged
  - only scoring changes
  - appropriate gap penalties
- If possible use the protein sequence rather than DNA
  - not all DNA codes for proteins
  - regulators, tRNA, catalytic RNA, sRNA, ..
  - not possible for genomic comparisons

- automatically includes codons, amino acid similarity, ..

- where does this kind of matrix come from ?

Substitution Matrices

- Lots exist
  - PAM point accepted mutations
  - BLOSUM blocks substitution matrix
- Philosophy
  - if two amino acids are similar, we will see mutations often
- To quantify this..
- Take some very similar proteins (lots)

parts of some haemoglobins

```
HAHKLRVGPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDAVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSK
HAHKLRVDPVNFKLLSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSK
HAHKLRVDPVNFKLLSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSK
HAHKLRVDPVNFKLLSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSK
HAHKLRVDPVNFKLLSHCLLVTLAAHHPDDFNPSVHASLDKFLANVSTVLTSK
HAHKLRVNPVNFKLLSHSLLVTLASHLPTNFTPAVHANLNKFLANDSTVLTSK
HAYKLRVDPVNFKLLSHCLLVTLACHHPTEFTPAVHASLDKFFTAVSTVLTSK
HAQKLRVDPVNFKFLGHCFLVVVAIHHPSALTPEVHASLDKFLCAVGTVLTAK
HAQKLRVDPVNFKFLGHCFLVVVAIHHPSALTAEVHASLDKFLCAVGTVLTAK
HAQKLRVDPVNFKFLGHCFLVVVAIHHPSALTAEVHASLDKFLCAVGTVLTAK
HAQKLRVDPVNFKLLGQCFLVVVAIHNPSALTPEAHASLDKFLCAVGLVLTAK
HAYNLRVDPVNFKLLSQCIQVVLAVHMGKDYTPEVHAAFDKFLSAVSAVLAEK
HAYNLRVDPVNFKLLSHCFQVVLGAHLGREYTPQVQVAYDKFLAAVSAVLAEK
HAYILRVDPVNFKLLSHCLLVTLAARFPADFTAEAHAAWDKFLSVVSSVLTEK
```

parts of some haemoglobins

```
HAHKLRVGPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSK
HAHKLRVDPVNFKLLSHCLLSTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVDAVNFKLLSHCLLVTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVDPVNFKLLSHCLLSTLA
HAHKLRVDPVNFKLLSHCLLSTLA
HAHKLRVDPVNFKLLSHCLLSTLA
HAHKLRVDPVNFKLLSHCLLSTLA
HAHKLRVDPVNFKLLSHCLLVTLA
HAHKLRVNPVNFKLLSHSLLVTLA
HAYKLRVDPVNFKLLSHCLLVTLA
HAQKLRVDPVNFKFLGHCFLVVVA
HAQKLRVDPVNFKFLGHCFLVVVA
HAQKLRVDPVNFKFLGHCFLVVVA
HAQKLRVDPVNFKLLGQCFLVVVAIHNPSALTPEAHASLDKFLCAVGLVLTAK
HAYNLRVDPVNFKLLSQCIQVVLAVHMGKDYTPEVHAAFDKFLSAVSAVLAEK
HAYNLRVDPVNFKLLSHCFQVVLGAHLGREYTPQVQVAYDKFLAAVSAVLAEK
HAYILRVDPVNFKLLSHCLLVTLAARFPADFTAEAHAAWDKFLSVVSSVLTEK
```

- consider an example column
  - how many pairs do we have ? 1-2, 1-3, … 2-3, 2-4, … get n_{total}
  - count n_{HH}, n_{HY}, ..
  - $p_{\mathrm{HH}}=n_{\mathrm{HH}}/n_{total}$ would be probability that H is conserved (or another amino acid)
  - $p_{\mathrm{AB}}=n_{\mathrm{AB}}/n_{total}$ would be probability that A and B mutate to another

Calculating a substitution matrix

- We have all the probabilities p_{AB} and p_{AA}
- next step matrix element AB is $\log_2(p_{\mathrm{AB}})$ why $\log_2$?
- is my example enough ?
  - needs much more data so as to get good probabilities

Different matrices

- Lots of details PAM vs BLOSUM vs … (not important)
- Degree of homology
  - if two sequences are very similar most residues not changed
  - longer evolutionary time – many things change

Longer evolutionary times

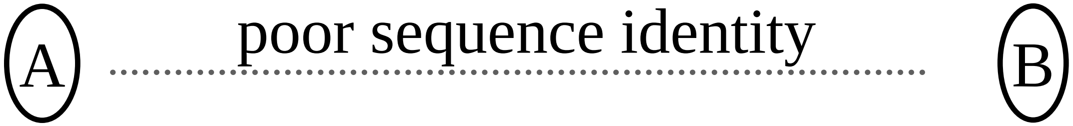
- so far, probability of one mutation A→B
- longer evolutionary time
- D→E→D→W→D…
  - multiple mutations
  - our matrix should reflect this
  - probability of conservation is lower (diagonal elements)
  - all off-diagonal elements will be bigger
- more formally - long time *p* is *p*   *p*   *p* …
- account for this ?
  - take matrix (like blosum) and do matrix multiplication
    - M   M   M   …
  - result: a set of matrices
    - PAM10, PAM20, …
    - Blosum62, blosum80, …

Are these matrices useful ?

- In principle, yes
  - looking for similar proteins – use blosum80
  - more remote ? – use blosum62
  - …
- in practice ?
- better way to find remote homologues
- huge advance in practical terms

iterated searches (psi-blast)

- You search with protein A and find a very remote protein B

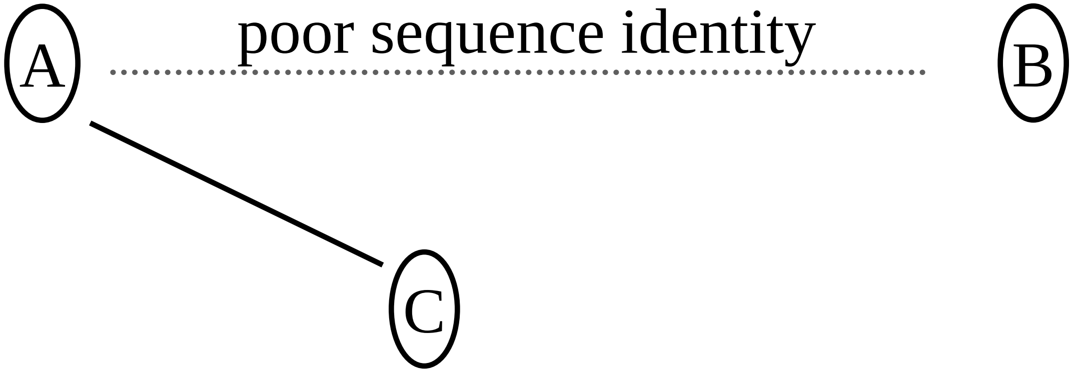


A

poor sequence identity

B

- but there another protein C



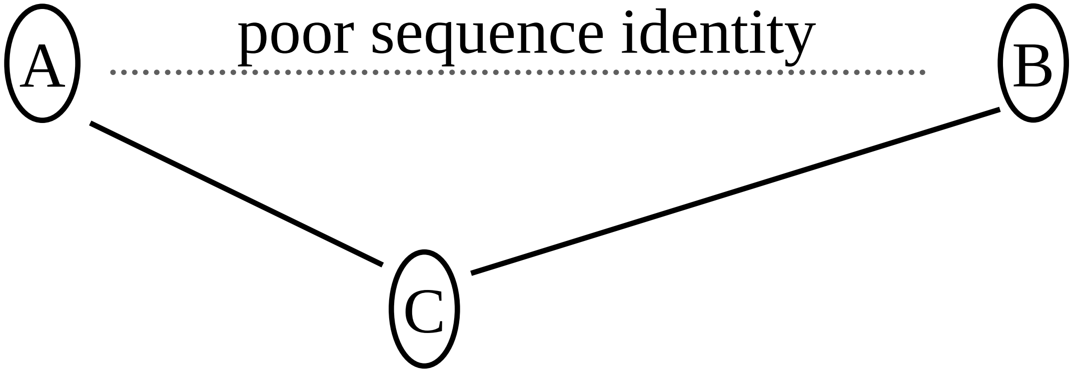
A

poor sequence identity

B

C

- searching with C
- the original AB relation is believable
- how to automate this ?



A

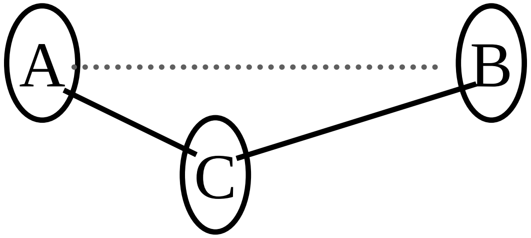
poor sequence identity

B

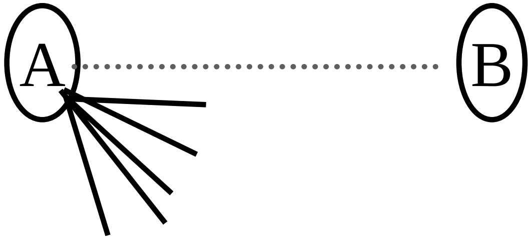
C

iterated searches (psi-blast)

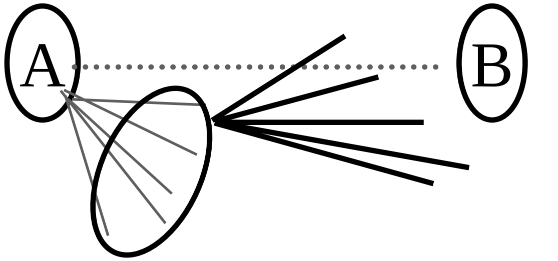
- Searching with "A" finds lots of homologues
  - cannot start a search with each



- alternative
  - find all the homologues to A
  - build an average sequence (profile)
  - from this profile – repeat search
  - build new average / repeat



- result
  - at each step
  - include reliable homologues
  - eventually A→ B may be found



iterated searches (psi-blast)

- in practice
- really only one program (+ web page) ncbi blast / psi-blast
- most significant advance in finding remote homologues in a decade

sequence identity / similarity / significance

Significance

- I find a homologue – is it evolutionarily related or just noise ?
  - probability estimations later
- how important is 10% sequence identity ? 90 % ?
- is 25 % identity in DNA as useful as in a protein ?

- First principles DNA
- what would you expect by chance ?
- `GGATCGA`
  `GATTCAGGTTA`
- At each position ¼ chance of a match
  - average 25 % sequence identity with random DNA
  - wrong

Naïve identity expectation – base usage

- Two problems – uneven character frequency, gaps

Character frequency

- what if I have a two letter alphabet ? `GCGCGC`
  - average sequence identity 50 %

`GCGCGCGCGCGCGCGCGCGCGC`	50 %
`GCGACGCGTCGCGCGTTCGCGC`	< 50 %
`GCGACACGTCGTGAGTTCTTGC`	nearly 25 %

- as the base usage becomes less even
  - random sequence identity becomes bigger
- how significant ?
  - malaria is about ⅓ GC (not ½)
  - GC differs between organisms, coding/non-coding
- even with random DNA, identity will be > 25 %

Naïve identity expectation - gaps

- ungapped: 2 matches from 9 aligned (22 %)

```
GGATCGCAC
GACTGAGGTTA
```

- one gap: 3 matches 8 aligned (38 %)

```
GGATCGCAC
GACT-GAGGTTA
```

- more gaps: 4 matches from 6 positions (50 %)

```
GGATCGCAC
GACT-G-AGGTTA
```

- more gaps: 5 matches from 6 positions (83 %)

```
GGATC-GCAC
G-A-CTG-AGGTTA
```

- the more gaps one allows - the higher the identity
- cheating ? One can make score arbitrarily good

Protein – random matches

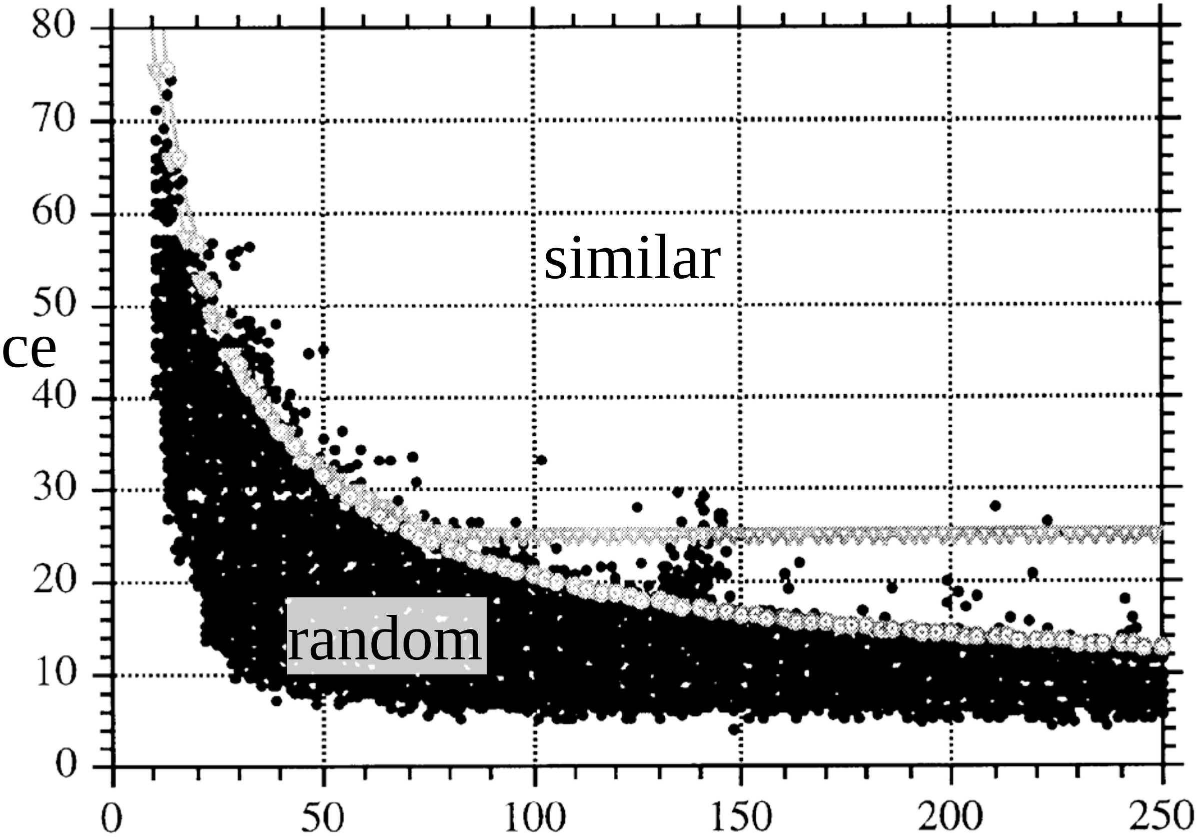
- 20 amino acids
- naïve expectation – 5 %
- proteins are not like a 20 character alphabet:
  - varies between organisms
  - varies between cell compartments, soluble, membrane bound…
- practical result - random sequences, realistic gaps
  - 20 to 25 % identity by chance
  - depends on length..

	%
ala	8.4
leu	8.3
gly	7.8
trp	1.5
cys	1.7

protein size and identity

- small proteins – need 30 % to believe they are related
- big proteins < 20 % , almost certainly related

% sequence ident



chain length (residues)

Rost, B.Prot Eng, 12,85–94, 1999

Summarise problem and steps

Mission

- you have a protein sequence
  - no structure
  - maybe no biochemistry (substrates, binding targets, ..)

- find what you can
  - related proteins of known structure
  - related proteins with known function

- Is there
  - an answer ?
  - one set of steps ?

easy	98 % similar to protein of known function and structure
↕	
hard	weak possible similarity to a poorly characterised family

General Idea

- Try easy steps first
  - simple searches first
  - see if enough information is found

  - gradually go to more sensitive methods (slightly more error prone)

- Use the “least speculative” methods first
  - accurate alignments – not seeded
  - simple blast searches before iterated ones

What are the expectations ?

- for easy sequences
  - very good molecular models
  - no doubt about function

- middle difficult
  - reasonable models
    - enough to guide mutagenesis (which residues can be mutated safely)

- very difficult
  - not even sure what class of proteins or what function
  - may be able to suggest experiments most likely to be useful