B13134 - BIOINFORMATICS

What is bioinformatics?

NIH definitions of bioinfo and comp. bio.

In any phenomenological study, after data collection, a model is built that explains the data and can be used to make predictions.

The model is a tool to understand and analyze the data. One of the first models was built by Johannes Kepler for the data collected by Tycho Brahe.

The algorithms are general ways of solving some kind of problems - that is, making sense of the data.

List of topics we'll cover.

Its important to think of the logarithms from a programmer's perspective.

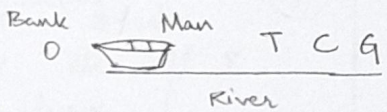
Problem solving recipe

Representation

Scoring (automated)

Sampling (Optimization)

Problem:



M, T, C, G

0 0 0 0 [A]

↓ ↓ ↓ ↓

1 1 1 1 [A]

Bank 1

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Representation of states

→ Allowed [A] Disallowed [D]

The shortest path of allowed states from [0 0 0 0] to [1 1 1 1] would be the solution to the problem

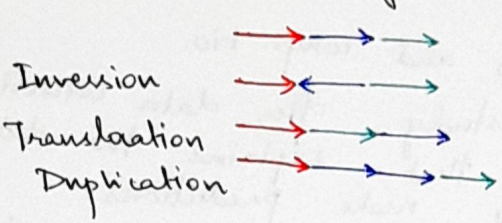
If the problem had more animals and other complex constraints, it can't be worked out by hand.

The man's position should go from $0 \to 1 \to 0 \to 1$

Sequence Alignments

Based on central dogma : DNA → RNA → Protein, we can align sequences to the DNA. Changes in DNA sequences, it creates mutations → variations which is necessary for evolution.

Seq. align important in —
- Deducing evolutionary relationships
- Function annotation
- Identifying important regions



The sequences need not be of equal length — there could be deletions or insertions.

Given 2 sequences, we need to figure out where the insertions / deletions / mutations are. In other words, we're trying to maximise the matching of identical equivalences.

HW : * Given 2 sequences of length N and M, if all enumerations of possible alignments were made, then what's total no. of possible ways?

(Pg 5)

* If all possible enumerations were made, then what would the scoring method be?

Ans : (i) Percentage of matches — but it creates equivalent seq.

(ii) Scoring based on matches —

A – A	: 5		Scoring system.
A – T	: 3	(purine - purine)	
A – G/C	: 1	(purine - pyrimidine)	
A – _	: -1		

23/8

Lecture 02

Aligning pairs of sequences

Given 2 strings : $X = x_1, x_2 \dots x_m$

$Y = y_1, y_2 \dots y_n$

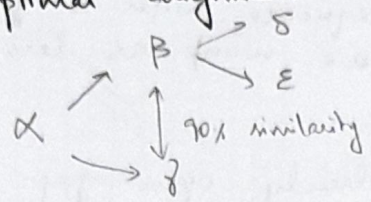
An alignment is an assignment of gaps to positions 0 … M in X and 0 … N in Y such that each letter in one sequence is lined up with either a letter or a gap in the other sequence

$q_a = \frac{1}{\text{No. of a's in the seq}}$ $q_b = \frac{1}{\text{No. of b's in the seq}}$

Sources
Durbin
Lesk

Aligning pairs of sequences

Optimal alignment - maximise identical residue pairs



Why sequence similarity?

- Sequence similarity often implies structural & functional relationship. Over the course of evolution, structure has been conserved more than sequence.
- New sequences can arise by: Insertion/Deletion/Substitution
- Similarity b/w 2 sequences is assessed by an alignment. By assessing how similar sequences are to one-another, for many sequences, we can build a tree to ascertain ancestry.

The alignment scheme

1. Representation - each residue is represented by a letter
2. Scoring - substitution scores
   gap penalties
3. Optimization - enumerate all possible alignments?
   dynamic programming

→ Substitution scores

$$s(a,b) = \log\left[\frac{(P_{ab})}{(q_a q_b)}\right]$$

$$= \log\left(\frac{obs}{exp}\right) \quad \frac{\text{observed subs}}{\text{expected subs}}$$

P_{ab} : likelihood of a being substituted by b

q_a, q_b : probabilities of A & B to occur alone

So $q_a q_b$: likelihood of a, b occuring together?

log likelihood eases calculations, easy derivative, keeps it bounded in a range

$S(a,b)$ - score for substituting residue a by b
- for all proteins, S is a 20×20 matrix

How to get the values of this matrix?
By existing available matrices: PAM250, BLOSUM62 ...

→ Gap penalties

- Everytime we align a sequence with a gap we want to penalize it cuz we want as less gaps as possible
- But extension of an already open gap is less of a crime, so it should incur lesser penalty
- Linear gap penalty — all gaps are penalised equally
- Affine gap penalty — distinguishes b/w opening and extending gaps, where penalty for opening is larger than gap extension.

24/8

Lecture 3

Dynamic Programming

Optimization problem — find optimal alignment by maximising score

Enumerating all possible sequences is gonna take a long time

Dynamic programming is easier — we find the optimal solution by optimising the parts.

Say, $X = x_1, x_2 \dots x_i$

$Y = Y_1, Y_2 \dots Y_j$

This alignment can be constructed from 3 sub-alignments

$x_1, x_2 \dots x_{i-1}$ / $Y_1, Y_2 \dots Y_{j-1}$

$x_1, x_2 \dots x_{i-1}$ / $Y_1, Y_2 \dots Y_j$

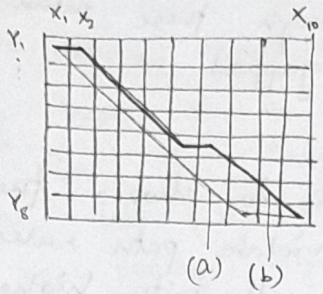
$x_1, x_2 \dots x_i$ / $Y_1, Y_2 \dots Y_{j-1}$

Two types of dynamic programming —

Needleman-Wunsch → * Global alignment — trying to align full sequences

Smith-Waterman → * Local alignment — trying to align some overlapping sequence (might have functional significance) where overhangs (sequences on either side) are not penalised

Compact representation of sequences



X ..... XXX } (a)
Y ..... Y – – }

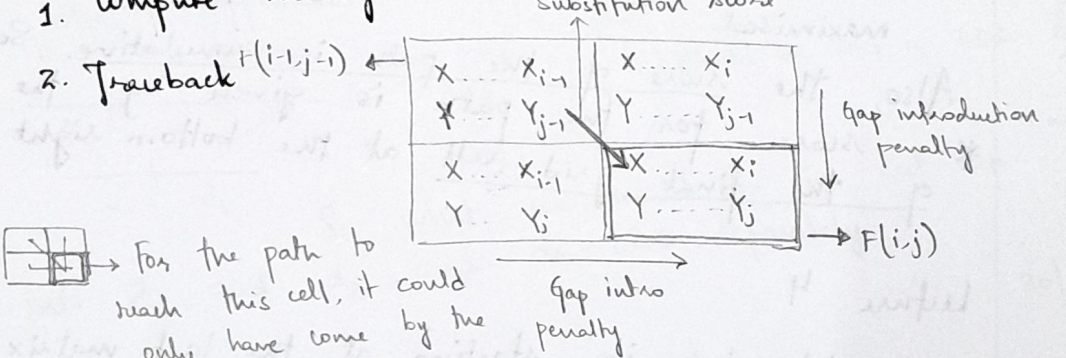
$X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9 X_{10}$ } (b)
$Y_1 - Y_2 Y_3 Y_4 Y_5 - Y_6 Y_7 Y_8$ }

The question: total no. of possible sequences can be calculated by counting the number of paths from one corner to another

But the path can't be haphazard – it should give a one-to-one equivalence. So in our grid, starting from top left corner, it can only go horizontally right or diagonally downward or vertically down also possible

Dynamic programming has three steps –

1. Compute scoring matrix: Recurrence formula
2. Traceback



Since we have scored all the (-+) quadrant cells wrt to the focal cell. Or the (--) quadrant if you're going from bottom right corner.

Recurrence relationship –

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

So we're optimising each and every residue with every other

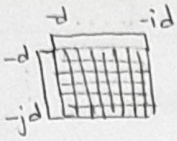
Effective when only parts of sequences have detectable similarity

Global Shortcoming: * Could be inaccurate when we're interested in finding best alignment b/w sequences

* Effective only when 2 sequences have sig. similarity over entire len

Base conditions : $F(i,0) = -i \times d$

$F(0,j) = -j \times d$



An extra row and column is added which is just linear penalty.

Drawing a line through these means that everything is gapped i.e no overlap at all

Traceback

Given a grid cell, we need to choose the next cell (ensuring it doesn't violate path rules) by optimising ie choosing the cell ~~with~~ from highest score.

This way we optimising the sequence by optimising the path at every step.

For the matrix score calculation you start from top left and for each grid cell is marked with the direction from which the score was maximised.

So the traceback goes from bottom right corner back the way in which the score had been maximised

Also, the score of the path is cumulative. So the score for the path is given by the score of the final grid cell at the bottom right corner. *

25/8

Lecture 4

We're interested in starting at the last matrix point and then traceback because its global dynamic programming - the entire sequence has to be optimized while actually writing the aligned sequence, we follow the traceback line forward

In local dynamic programming, only segments of interest are optimised to align. So overhangs are not penalised

Base conditions -

$F(i,0) = 0$

$F(0,j) = 0$

Recurrsive relation -

$$\text{Max of} \begin{cases} 0 \\ F(i-1, j-1) + S(i,j) \\ F(i, j-1) - d \\ F(i-1, j) - d \end{cases}$$

Here, given two sequences, X and Y, we find subsequences α and β whose alignment score is max over all pairs of subsequences.

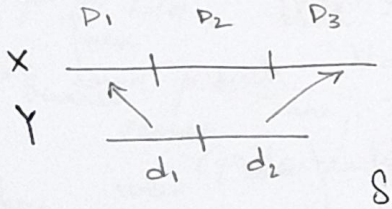
Why have 0 in recursive relationship?

In the traceback, when we come across 0 in the cell (i.e otherwise the grid cell would be negative - ie poorly aligned), we stop there & consider it as no overhang.

All tracebacks start at 0. But where would tracebacks start? We start ~~this~~ with the gridcell with maximum score and go back till we encounter 0. This will give us local alignment

local DS is better at detecting remote similarities whereas Global DS detects sequences with high similarity.

for both, computational complexity : $O[N^2]$ in time & memory



D_1	D_3	Would be lost in LDS
d_1	d_2	
28	26	

So what we could do is cut up Y into d_1 and d_2 and do LDS separately

30/8

lecture 5 - Substitution Matrices (Rec)

Alignment of identical nucleotide > similar protein ("conservative") substitution > non-conservative protein substitution

Most ← Favourable → least

The scoring is used to arrive at the most optimal alignment of two sequences. But how do we arrive at the scores?

Alignment score → log of relative likelihood that 2 sequences are related, compared to being unrelated

$$* \left[\frac{obs}{exp}\right] *$$

P(obs) P(exp)

Scoring system provides a measure for judging the quality of an alignment in relation to other possible alignments

Say there's a bowl of blue & red balls. Prob that an unbiased sampler picks 1 red & 1 blue ball in each hand is —

$$P(R, B) = \frac{n_R}{N} \cdot \frac{n_b}{N} = \text{exp}$$

$$N^{obs}(R, B) = \text{obs}$$

If $\frac{obs}{exp} > 1 \Rightarrow$ I had many more observations than expected by chance

Similarly in case of nucleotides / residues, if obs/exp > 1, then the substitution is favourable in some way.

If ob/exp = 1, then its random i.e occurs by chance

obs/exp < 1, then the substitution is not favourable

To construct substitution scores, we need initial alignments that we completely trust —

Eg: 1. GAH F Q A V
2. G P H F S L V
3. G P H F D A I

} We 3 pairwise alignments

The substitution scores are independent of order i.e. A↔P

$$* \quad S(A,P) = \frac{N_{A,P}}{\sum_i \sum_j N_{ij}} \Big/ \frac{\sum_x N_{A,x}}{\sum_i \sum_j N_{ij}} \cdot \frac{\sum_Y N_{P,Y}}{\sum_i \sum_j N_{ij}}$$

$P(obs) = \frac{\text{No. of obs. A↔P subst}}{\text{Total no. of obs}}$

$P(exp)$ → No of times P↔Y (Y: any nucleotide); → Total substitutions

$$\therefore S(A,P) = \log\left[\frac{P_{A,P}^{obs}}{P_{A,P}^{exp}}\right]$$

Q: Why log? — Pg 3

In the 3 pairwise sequences,

$\sum\sum N_{ij} = 21$

$N_{A,P} = 2 \Rightarrow P_{A,P}^{obs} = \frac{2}{21}$

A↔A also counts

$N_{A,x} = 5$, $N_{P,Y} = 3 \Rightarrow P^{exp} = \frac{5}{21} \cdot \frac{3}{21} = \frac{5}{21} \cdot \frac{1}{7}$

$$\Rightarrow \log\left(\frac{P^{obs}}{P^{exp}}\right) = \log\left(\frac{14}{5}\right) > 1$$

$\therefore$ A↔P is favourable.

Remember: for all this to work out, the given sequence alignments need to be unimpeachable

Problems: 1) Difficult to obtain a good random sample of protein seq.
2) This approach doesn't take into account the effect of evol. distances
Large evol. dist ⇒ $P_{ab} \sim q_a q_b$ ⇒ $S(a,b) \approx 0$

FUGUE
- Given commonly used S(a,b) -
- BLOSUM, PAM, Gonnet
- Risler (derived from structural alignments)
- Environment specific S: Overington,

Usually, the sequences picked are closely related, of the same length, similar sequences. They are done by hand

Also, logically consistent i.e identical substitution would be given max score and so on.

But with small sequences, rare but favourable substitut would be given a v. low score (than they deserve) which would be a misrepresentation

So the chosen standard alignment should be a good representation of real life sequences. Also there sequences are evolutionarily close but we're using them to align significantly divergent sequences.

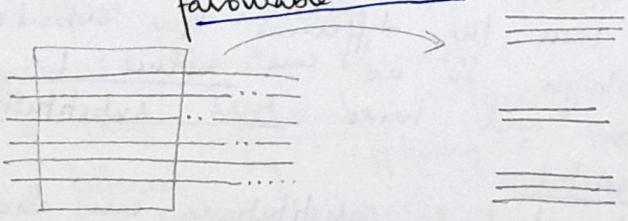
How to resolve this conundrum?

BLOck SUBstitution Matrices (BLOSUM) series of matrices
PAM 250 : another matrix ~500 protein groups — 2000 blocks

Here too we take a set of well-known sequence alignment Then a block of region with no gaps is chosen. Several such blocks are chosen and substitution scores are calculated from those.

They were constructed in early 1990s by Henikoff & Henikoff

Problem: Since the sequences were too similar, there was an overabundance of identical subs. & rare yet favourable subs were given too low a score



Soln: Each block is divided into subsets and in each subset, (any pair) has some ≥ L% sequence identity

For BLOSUM80, any given pair in a subset is L% ie 80% identical in alignment. Thats how the subsets are constructed.

* Any sequence in the subset is ≥ L% similar to at least one other sequence in the subset

+ score : Given aā pair is more likely to occur in an alignment than by chance
− score : less likely to occur than by chance

Each subset has the weight of a single sequence = 1.
So each sequence within a subset has lower weightage

* Thus the rare substitutions now get more weightage than earlier and don't get masked by dominant alignment pairs. *

$$S(a,b) = \log\left(\frac{P_{ab}}{q_a q_b}\right)$$

P_{ab} : prob of a⟷b alignment obs
$q_i q_j$: prob that a⟷b substitution occurs by chance i.e expected

Using BLOSUM, we comput S matrix for the previous example

The substitutions within a subset are not considered — they're similar above the sequence identity cut off. So only subst. across subsets are considered —

GAHFQAV
GPHFSLV

GPHFDAI

$$S(A,P) = \log\left(\frac{\frac{N_{AP}}{T}}{\frac{N_{Ax}}{T}\,\frac{N_{Py}}{T}}\right)$$

$T = 0.5(7+7) = 7$

GAHFQAV GPHFSLV
GPHFDAI GPHFDAI

$N_{AP} = 0.5(1) = 0.5$

$N_{Ax} = 1.5$ $N_{Py} = 1$

0.5 0.5

$$\Rightarrow S(A,P) = \log\left(\frac{0.5/7}{\frac{1.5}{7}\cdot\frac{1}{7}}\right) = \log\left(\frac{7}{3}\right)$$

Because we're using clusters, we assume that this gives us a better value.

If L% is high then the difference b/w subsets is still pretty high. So we can reduce L%, have more subsets and hence rare substitutions are better represented

So the scores of identical substitutions can also vary
W-W : 15 A-A : 5
Tryptophan is a rare amino aid, so its alignment has a higher score

Another feature : All subst. scores are integers.
What's the advantage?

FASTA: 1. Locate hotspots
2. Extend hotspot to find max scoring ungapped extensions
3. Identify possible gapped alignments
4. Align highest scoring match using dyn. prog. 31/8

Lecture 6

UNIPROT: Database of ~ 200×10^6 sequences of proteins

Can we take a sequence and define / arrive at its relation with any other sequence in the repository

Substitution b/w 2 amino acids at any spot in the protein sequence has the same score. But that doesn't work — aa in different parts of 3D protein structure should have different subst. scores.

Q Given any 2 sequences, after dynamic programming has been done, can we arrive at any conclusions about their relation?

Q Given a subst. matrix, what should be the optimal gap penalty?

Tradeoff b/w speed & sensitivity

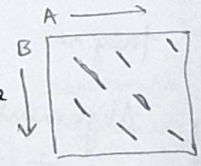
If the Query sequence is aligned (DP) with millions of sequences it'll take a long time [order of (N.M)]

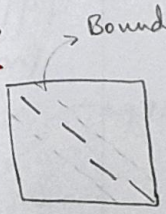
To overcome this, Heuristic algorithm (FASTA, BLAST) is used

FAST All

Dynamic programming calculates the score of every cell in the matrix. But we don't need all

Given 2 sequences, (i, i+1) in A are identical to (j, j+1) in B. All such pairs are identified Heuristically. The optimal alignment should encompass one or more of these 'hotspots'. These hotspots are filtered by applying a threshold (~~ie closer to the dig~~ diagonal score. The top 10 hotspots are retained which are later filtered by applying a threshold. (~~ie close to the diagonal (?)~~). Then Bounded DP is done — solutions only in that range are explored.





Q. How should recursive relation be set up for this?

Running in parallel is expensive

Threshold - the score of those d alignments is above a certain threshold

Score

In BLAST, the hotspots are sequences with similarities instead of identities. They were called HITs and were identified by setting a threshold of certain substitution score.

It also added another feature — the Hit was extended diagonally as long as the score of alignment

31/8

Recorded Lecture – BLAST

Basic Local Alignment Search Tool (BLAST).

- Hits (unlike hotspots in FASTA) allow high scoring similarity substitution. Here, its called High Scoring Pair (HSP)

DLSHGS

DLS	= 14	
DLA	= 11	
DLN	= 11	Threshold
DLD	= 10	= 11

Given a sequence, the all possible k-tuples (whose alignment would score higher than threshold) are well known. So alignment becomes easy

- Then the HSP is diagonally extended until the score doesn't fall below a certain score

It has been observed that usually —
- HSP is longer than word size (for DLS, its = 3) (word size)
- HSPs occur in pairs. Multiple HSPs can fall on the same diagonal.

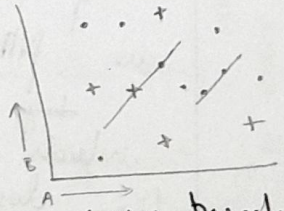
TWO HIT METHOD

BLAST 2.0 ⟹ If the distance b/w 2 HSP on same diagonal is < A, then they simply get linked. If > A, then they're not linked. This is called Two Hit Extension.

- If threshold is lowered – more 1 hit, most are dismissed
- Speed : 2x faster than one-hit
- Sensitivity : almost the same

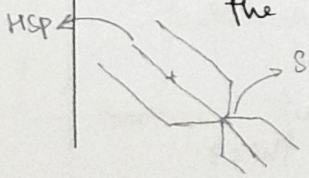
Eg: Alignment of Broadbean leghemoglobin & horse β globin.
+ : identical HSP
• : similar HSP.

Instead of extending all HSPs, we do 2 hit extension



Gapped BLAST – allows for gaps

From the HSP, a seed residue is chosen and the area around it is explored



These paths are explored as long as the score remains above some threshold

Dp algorithms are used to extend seed residue in both directions.

$$S_{ab} = \frac{1}{\lambda} \log\left(\frac{P_{ab}}{q_a q_b}\right) \Rightarrow P_{ab} = q_a q_b e^{\lambda s_{ab}}$$

$$\sum P_{ab} = 1 = \sum_{a,b} q_a q_b e^{\lambda s_{ab}}$$

Then optimal traceback is chosen and this method allows for gaps in the alignment. 1/9

Lecture 7

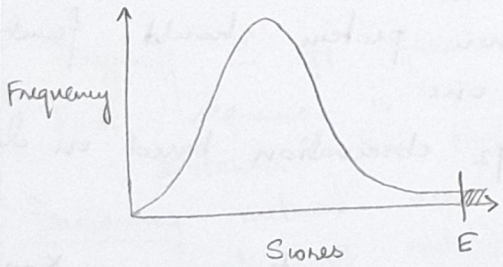
FASTA & BLAST are based on heuristic understanding of alignment

Say you get a score after aligning Query sequence and DB sequences. How do we know the significance of a score?

We get the distribution of scores of that query seq with multiple DB sequences. Then ~~we~~ from the distribution, we get a threshold based on confidence level, say 95% or 99%

But which sequences should be compared with the Query? It should be a random sample of sequences out of 200 M. Can't be a sample from a particular family of proteins.

Turns out, this follows an Extreme value distribution



In this distribution, the threshold for confidence level is given by E value —

$$E = KmNe^{-\lambda S}$$

Karlin-Artschul Equation

The sequences with score higher than E are related/similar to query. Because, if the sequences are unrelated, then coming across it by chance is less than 1% or 0.1% or whatever.

For 1%, E = 0.01. You can decrease E till 0 - identical

E = 0.001 means that ... the chances that you wrongly identified a sequence to be homologous is 1 in 1000

~40 min

M: no. of letters in query N: no. of letters in database

(111)

Normalisation score : $S' = \frac{\lambda S - \ln K}{\ln 2}$ λ, K : scaling constants
S : score of alignment.

So, $E = KmNe^{-\lambda S} = mN2^{-S'} = E$

S' : also called bit score. If 2 sequences have different lengths bit score is enough to compare relatedness of individual sets

BLOSUM 62 matrix — $\frac{2}{\log 2} s(a,b)$ values rounded to nearest integer

Integer operations are much faster than floating pt operations

E values will change with database size. So to get a significant hit, bit score should be modified

PSI-BLAST Lecture
Position Specific Iterated Blast.

Prob that there is a match of score greater than S is —

$P(x > S) = 1 - e^{-E(S)}$

Sequence identity zones
Midnight : 0 - 20% Twilight : 15% - 30% Safe zone : 30 - 100%
35%

5/9

Reading : PAM

Accepted point mutation — replacement of one aa by another that's accepted by nat. sel
To be accepted, the new protein should function similarly to the old one

400 possible changes ; 1572 observation based on closely related proteins.

If $X \to Y$ or $Y \to X$, the score is the same. $X \to X$ is not considered a change. So we get a ~~diagonal~~ lower triangular matrix with 190 cells

71 evolutionary trees were observed. 35 possible mutations never occured — rare aa or requires 1+ nucleotide has to change for codon to differ.

Highest occurrence : Asp ↔ Glu
20% of interchanges (↑ than exp) required more than 1 nucleotide change

If aligned sequences have diverged recently, P_{ab} would be v. small
$\Rightarrow$ $S(a,b)$ would be negative
If a long time has passed, then $P_{ab} \approx q_a q_b \Rightarrow S(a,b) \sim 0$

PAM Substitution matrices So P_{ab} should be normalised by divergence time

Sequences S_1 and S_2 are defined as being one PAM unit diverged if a series of pt. mutations (no insertion/deletion) has converted S_1 to S_2 with an avg of 1 mutation event per ~~1000~~ 100 amino acids.

100 PAM $\Rightarrow$ 100 pt. mut. for 100 aā – doesn't mean the 2 sequences 100 PAM units apart are totally different; certain positions can ~~multiply~~ mutate many times & ~~but~~ back mutations are possible

Derive $S(a,b)$ for PAM 1 evol. time (time period for which we expect 1% of aā to mutate). Then we extrapolate

Construction of PAM 1

1. Align prot. seq. that are at least 85% identical
2. Construct evolutionary tree & infer ancestral sequences
3. Count the no. of aā substitutions that occured in tree
4. Use these counts to estimate probabilities of replacements
5. Scale substitution prob to PAM 1 evolutionary time

- Dayhoff – 71 families, 1572 sequences.
- Each sequence is 85% similar to keep evolutionary distances low. Find reliable data
- Parsimony method was used to build evolutionary tree
- Then, count aā substitutions by counting residue pairing b/w sequences & immediate ancestors.
  Use these numbers to create a count matrix
- Estimate substitution prob.

$$P(A_j|A_k) = \frac{A_{j,k}}{\sum_{m=1}^{20} A_{j,m}}$$

- Scale probabilities to PAM1 evolutionary time

c – scales the off-diagonal terms & adjusts so row sum = 1

$$P_{j,k} = c \cdot p_{j,k} \quad \text{where } j \neq k \qquad P_{j,j} = 1 - c \cdot p_{j,k}$$

$$c = \frac{0.01}{\sum_{j=1}^{20} \sum_{i \neq k} q_j p_{j,k}}$$ scaling factor

c (scaling factor) accounts for evolutionary distance i.e. it makes subst. prob. over 1 PAM evol. time

PAM n substitution matrix

Bayes' theorem : $P^1(b|a) = \frac{P^1(ab)}{q_a}$ — Joint P of $a\bar{a}$ a & b; Prob of occurrence of $a\bar{a}$

$s^1(a,b) = \log \frac{P^1(b|a)}{q_b} \longrightarrow s^n(a,b) = \log \frac{P^n(b|a)}{q_b}$

PAM 250 : widely used matrix

PAM1 corresponds to 1 million years of evolution

PAM 120 – largest info content

Limitations

- Small dataset has been used to derive the scores
- Subst. data is calculated for small evolutionary distance and the extrapolated
- Raising $P^2(a,b) \rightarrow P^{250}(a,b)$ doesn't capture the true difference b/w short & long time substitutions.
  - short time subst. arise from single base change in triplet
  - long time subst. shows all types of codon changes

In BLOSUM, L% determines evolutionary distance

large L ⇒ short evol. distance

PSI BLAST

* Use BLAST search → constructed MSA based on local alignments where E < h
* From this, calculate position-specific scoring matrix
* Search database with PSSM as query – iterate until convergence

PSSM – l x 20 matrix based on observed residue freq

where l : length of seq

In next stage, BLAST search is carried out using PSSM instead of query.

Process is iterated several times with PSSM generated from round i used for round i+1, until alignments with sufficiently low E-values are obtained

→ Relative mutability
Probability that each aā will change in a given small evolutionary time
Its proportional to the ratio of changes to occurrences
Asn, Ser, Asp & Glu (most) & Trp & Cys (least)

→ Mutation Prob Matrix for evolutionary dist. of 1 PAM
M_{ij} : represents the prob that aā in column j will be replaced by aā in row i after evolutionary interval of 1 PAM

Non-diagonal ↓

$$M_{ij} = \frac{\lambda m_j A_{ij}}{\sum_i A_{ij}}$$

λ : proportionality const.
m_j : mutability (for Ala $m = 100$)
A_{ij} : element of accepted pt. mut. matrix

Diagonal : $M_{jj} = 1 - \lambda m_j$

→ Simulation of mutational process
for 1 mut. to take place in 1 PAM interval —
For every aā in the sequence, obtain a number from $x \sim \text{unif}(0,1)$. For Ala, if $x \leq 0.9867$, then its left unchanged For $x = 0.9867 - 0.9868$, Ala → Arg & so on. This is done for whole sequence to simulate mutation
Method can be modified to include long intervals, fixed no. of mutations & so on
Over large evolutionary intervals, most aā in the sequence would have changed

→ Estimation of evolutionary distances
There's a correspondance b/w observed differences and evolutionary distance. Its calculated based on —

$$100\left(1 - \sum_i f_i M_{ii}\right)$$

M_{ii} : element in MPM $j \to i$
f_i : normalised freq i.e. prob. that i will occur in 2nd seq by chance

→ Relatedness odds matrix → symmetrical

$$R_{ij} = \frac{M_{ij}}{f_i}$$

Here, each ij term gives the prob. of replacement per occurrence of i per occurrence of j

(16) → Chemical meaning
aā that have similar chemical properties are more likely to interchange. These patterns are imposed mainly by nat. sel. & secondarily by constraints of genetic code

Imp. properties of aā that determines interactions —
- size, shape, local conc of charge
- conformation of van der Waals surface
- ability to form salt bonds, hydrophobic & hydrogen bond

→ Computing relationship b/w sequences
Log odds matrices are used as scoring matrix to detect very distant relationship b/w proteins.

6/9

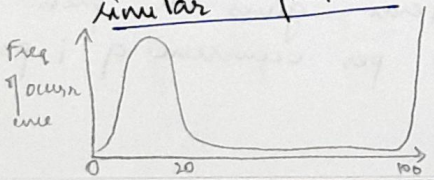
Lecture 08

→ Why have log?
- Taking log gives us an even distribution for likely and unlikely events. For obs/exp < 1, the log will be negative & > 1, the log value will be positive So the sign gives us an idea of favorable and unfavorable substitutions.
- Also if working with subsequences, then with a log value, the probability would be additive and not multiplicative.

→ How do we arrive at optimal gap penalties for the substitution matrix?
For a set of known alignments, we derive a subst. matrix. Then we experiment with different values of gap penalties and try to align the sequences. The gap penalty score for which the alignment closely matches that of gold standard alignment is the best gap penalty.
It changes with different substitution scores.

→ How do we get the gold standard alignments?
We can get it by superimposing structurally similar proteins and aligning the residues of proteins.

Why the spike in frequency?

→ 

N-D dynamic prog: At each pt - $(2^n - 1)$ computations
$O(2^n L^n)$ - time

Multiple sequence alignments (MSA)

Say we want to align 3 sequences, then we use 3D dynamic programming to align them.
What recursive formula can we use?

But for longer sequences and > 3 sequences, it would become too computationally heavy and it would take a long time.

Also, how do we come up with a substitution score matrix for MSA? It would be an n-dimensional (nD) sub. score matrix. This is super hard.

We get around this using heuristic methods. Called — Progressive alignments.

→ This method is fast, uses heuristics but NOT necessarily optimal

- Align the first 2 sequences to one another
- Then, align the third sequence to the alignment of first two
- Iterate with subsequent sequences.

Once 1, 2 are aligned, how do we align sequence 3 with it?

1 A
2 K
3 V

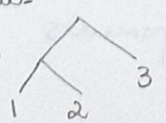
Its alright for identity & alignment residue
But when there's a mismatch b/w 1 & 2,
we can get a substitution score for 3rd sequence by —

$$* \frac{S(AV) + S(KV)}{2} *$$

→ Progressive alignments with trees

Here, we construct a relationship of sequences.
Based on this, we progressively go on aligning the sequences.

We construct the tree by doing pairwise alignment of all sequences and based on similarities, the tree is constructed.



based on distances inferred.

For this, sequences 1 & 2 are aligned first and the 3 is aligned with them.

What can be used as a distance measure to construct trees?
More the similarity — lesser the distance
So we should use some inverse of alignment scores to figure out the distances.

7/9

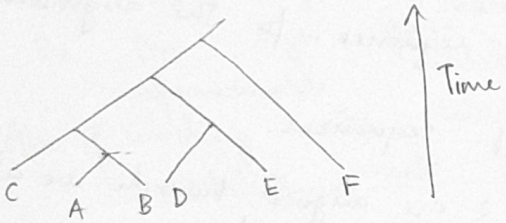
Lecture 09

Explain the distribution of frequency of sequence identity
- Isoforms : functionally similar proteins that have similar but not identical sequence
They are formed by splice variants or point mutations
Strains of bacteria / virus are a major source of this.
So we get a spike in frequency at 98% or 99%.

Discussion of Problem set.

Lecture 10

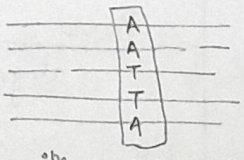
Finishing MSAs



Based on this tree,
- Align A & B cuz they're closer
- Aling AB with C (next closest)
- Then align D & E
- Then CAB with DE
- Finally, CABDE with F.

To do this we need —
* Distance metric : closely related sequence have a smaller score than distantly related ones.

How do we score and align multiple sequences i.e. CAB with DE ?
Say we have alignments, we can think of ~~position~~ residue at ith position as a composition of multiple residues. Then, many such composite residues can be aligned



i^{th} = 0.6A + 0.4T

CLUSTAL : a method that does multiple sequence alignments

First, distance matrix is created from pairwise comparison. Then from the distance map, a phylogentic tree is deduced

Variants of scoring scheme —

$S(R,-) = -x$ $\quad S(-,-) = 0$ $\quad S(R,R')$ from BLOSUM

t — 1
l — 2
k — 3
k — 4
i — 5
v — 6

Score = $\dfrac{\left[S(t,i) + S(l,i) + S(k,i) + \dots + S(k,v) + S(k,v)\right]}{8}$

without sequence weights

With sequence weights (because of nested/related sequences) —

Score = $\left[S(t,i) \times w_1 \times w_5 + S(l,i) \times w_2 \times w_5 + \dots\dots S(k,v) \times w_4 \times w_6 \right]$

Clustal W alignments are not necessarily optimal

Iterative improvement of MSA

- Align the pair with greatest similarity
- Align to this pair, the seq with greatest similarity (tree)
- Obtain MSA based on this method
- Remove 1 sequence from the MSA and align it to the profile of other n-1 sequences from previous step.
- Repeat the iteration x times or until alignment scores converge

Phylogeny

It is a central idea to all of biology

Different methods aspects

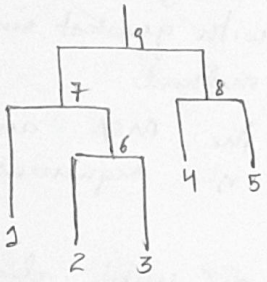
1. UPGMA
2. Neighbour joining — homework (Eddy textbook)
3. Parsimony.

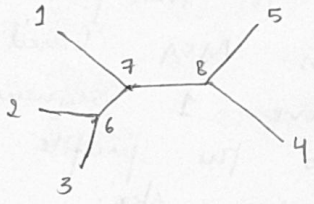
Recorded lecture – Phylogenetic trees. 13/9

The tree of life is central to Darwin's idea of evolution.
Sequence trees are used to construct PAM matrices.
Phylogenetic tree created by aligning haemoglobin sequences shows birds in one branch, mammals in another.

Each terminal pt is called a leaf and its end point is leaf node. Other nodes have bifurcations only.
Two leaves are at a certain distance away. This distance is an indicator of the time elapsed when they diverged from the last common ancestor.
Distance b/w nodes is called edge.
Trifurcation & other higher order divergences can be approximated by bifurcations

Different types of trees.
Rooted tree – all leaves converge to a single ancestor.
In an unrooted tree, there's no common ancestor.





5 leaves – 9 nodes
n leaves – $n+n-1 = 2n-1$ nodes

Assuming only bifurcations

5 leaves – 8 nodes
n leaves – $2n-2$ nodes

Q. What's the leaf-node-edge relationship in 2 kinds of trees?

→ Unweighted Pair Group Method using Arithmetic Averages [UPGMA]

Define distance b/w two clusters –

$$d_{ij} = \frac{1}{C_i C_j} \sum_{p \in C_i} \sum_{q \in C_j} d_{pq}$$

where C_i, C_j : sizes of clusters i and j

Steps

1. Each sequence is its own cluster with initialized branch size = 0
2. Measure the distance b/w all two points and determine the pair with minimal $d(i,j)$ value
3. New cluster $k = i+j$ → If there a 5 leaves s. 1,2 are closest, then their parent node is node no. 6
4. Introduce node at height $d(i,j)/2$ — not unweighted
5. Remove i and j from cluster list. Define $d(k,l)$ where l = all nodes other than i, j, k
6. Repeat the process until the last 2 nodes are joined

Homework –

Read Ch. 7 of Biological Seq. Analysis – Durbin

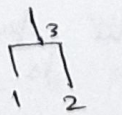
Show that $d_{kl} = \dfrac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}$

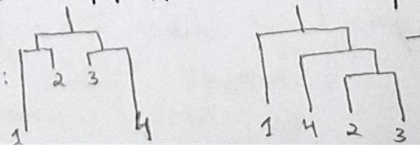
$$d_{il} = \frac{1}{C_i C_l} \sum_{p,q} d_{pq} \Rightarrow C_i \cdot d_{il} = \frac{1}{C_l} \sum_{p,q} d_{pq} \qquad C_j \cdot d_{jl} = \frac{1}{C_l} \sum_{\substack{p \in C_j \\ q \in C_l}} d_{pq}$$

$$\Rightarrow \text{RHS} = \frac{1}{C_k C_l} \left[\sum_{\substack{p \in C_i \\ q \in C_l}} d_{pq} + \sum_{\substack{p \in C_j \\ q \in C_l}} d_{pq} \right] = \frac{1}{C_k C_l} \left[\sum_{q \in C_l} \left\{ \underbrace{\sum_{p \in C_i} d_{pq} + \sum_{p \in C_j} d_{pq}}_{\sum_{p \in C_k} d_{pq} \;\because\; C_i + C_j = C_k} \right\} \right]$$

$$\therefore \text{RHS} = \frac{1}{C_k C_l} \sum_{\substack{p \in C_k \\ q \in C_l}} d_{pq} = d_{kl} = \text{LHS}$$

Limitations of UPGMA –

* The $d(3,1) = \dfrac{d(2,1)}{2}$ for  tree. The branches can have unequal lengths & the distance to another leaf might be shorter than the one to which you're closely related. So UPGMA cannot capture relations like these



↳ ∵ it gives equal weightage

Neighbour joining uses the following formula —

$$D_{(ij),k} = \frac{1}{2}\left(D_{i,k} + D_{j,k} - D_{i,j}\right)$$

→ Tree construction by Parsimony

Here, the tree is computed by such that a minimum number of mutations link up the given → assuming they're related set of sequences

Eg: Ancestor – 00000 & we have sequence of 1s & 0s we can construct a tree based on this principle

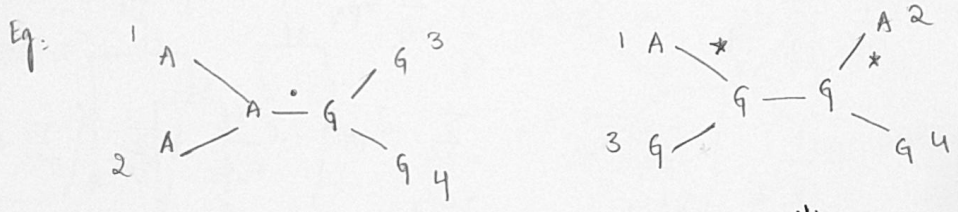
General formulation of parsimony

Based on number sequences, we construct possible trees. To figure out the most parsimonious one i.e. one that requires least no. of mutations, we consider some specific ~~align~~ sequence positions.

If two leaves have same sequence, then their common ancestor is also considered to have the same nucleotide.

If they have different residues, then one of the leaves must have mutated.

Eg:



Choosing the parsimonious - indicating position – the sequences should have some common nucleotides / residues in a subset of sequences, not all.

The trees are constructed for several seq. positions and the tree that's parsimonious in most no. of cases is considered the most parsimonious one

Ancestral sequences for PAM matrices were constructed using this method

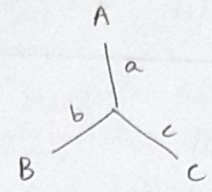
Figure out the ancestral sequence – when equally likely, then keep either letter (nucleotide), otherwise go for the most common one

Different algorithms

1. Fitch-Margoliash distance matrix method – based on the additive tree model where distances are expected to to equal the sums of branch lengths b/w species.

Construct an unrooted matrix for the following ~~method~~ matrix

	A	B	C
A		22	39
B			41
C			



$a + b = 22$
$a + c = 39$
$b + c = 41$
$c - b = 17$

$2c = 58$

$c = 29$
$a = 10$
$b = 12$

Uses of trees – tracing epidemiology i.e phylogeny of different strains of viruses.

Elephant-Mammoth story

1995 → Alignment of segment of cytochrome b gene of woolly mammoth, Asian elephant, African elephant, human and bovine samples.

From tree construction, they found that African elephant is more closely related to Asian elephant.

2015 → Multiple gene alignment : Asian elephant is actually closer to woolly mammoth. Makes sense geographically.

Assignment 1: Virus migration

6 virus strains – originated in Africa.

Construct a rooted phylogenetic tree using UPGMA & indicate on the map where it spread.

Assignment 2: Different characters – body shape, body colour, head colour, with/without antennae, visible jaws, spots on the back, colour of spots.

Construct a distance matrix (6×6) and based on it, create a phylogenetic tree using UPGMA.

Neighbour joining

Property of UPGMA : additive lengths that represent the same rate i.e. molecular clock.

In neighbour joining, additivity holds

Say, i, j are neighboring leaves i.e. their parent node is k

Remove i, j from leaf nodes & add k to the list.

Then the distance b/w k and m is —

$$d_{km} = \frac{1}{2}\left(d_{im} + d_{jm} - d_{ij}\right)$$

This way, we can strip away leaves till we arrive at the last remaining pair

Choosing neighbouring leaves —

* DO NOT pick two nodes whose d_{ij} is minimal
* Instead we —

$$D_{ij} = d_{ij} - \left(r_i + r_j\right)$$

$$r_i = \frac{1}{|L| - 2} \sum_{k \in L} d_{ik}$$

$|L|$: size of set of leaves

If D_{ij} is minimal — then, i, j are neighboring

* Define a new node k & set $d_{km} = \frac{1}{2}\left(d_{im} + d_{jm} - d_{ij}\right) \; \forall \; m \in L$
* Add k to set of leaves with edge length —

$$d_{ik} = \frac{1}{2}\left(d_{ij} + r_i - r_j\right) \; ; \; d_{jk} = d_{ij} - d_{ik}$$

20/9

Lecture 11

Hidden Markov Models

Markov process : Your current state depends ONLY on the immediately preceding state — none of the older ones

Eg: Dynamic programming

Genomic sequences have CpG islands — overall, they're quite rare — because a methylation changes C → T

They can be common in regions where they're protected from methylation.

CpG : common in promoter regions.

I guess these regions are called CpG islands (see P)

Example from — Sean. Eddy

Given a set of sequences, we can analyse how frequently 2 base pairs occur —

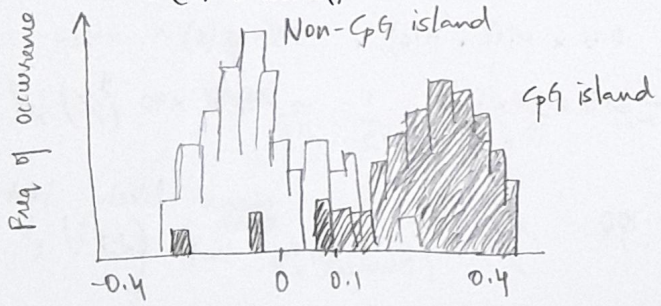
CpG island : $P(CG) = 0.274$ Non-CpG island: $P(CG) = 0.078$

If we compare these values —

$$\log\left(\frac{P(CpG)\ \text{isl}}{P(CpG)\ \text{non-isl}}\right) = \text{Bit score}$$

~ similar to $\frac{[obs]}{[exp]}$

Bit score > 0.1 — CpG island



$\Sigma = \{ A, T, G, C \ldots \}$ — letters

$\Pi = \{ \Pi_1, \Pi_2, \Pi_3 \ldots \}$ — states for proteins — α, β, coil (c)

Emission probability : Prob of some letter to be in a particular state

$e(\Pi_i, \Sigma_i)$ = If the Π_i state, Σ_i letter is emitted with that prob

Transition prob: The prob of transitioning from one state to another

$a(\pi_i \to \pi_j) = ?$

Consider a protein sequence –
- the sequence of letters is visible
- sequence of states ($\pi_1, \pi_2 \ldots$ which can take the form α, β or c) is hidden.

Example: Rolling a die – biased

FAIR $P(6) = \frac{1}{6}$

LOADED $P(6) = \frac{1}{2}$ $P(1 \ldots 5) = \frac{1}{10}$

1 2 1 5 6 1 1 5 2 4 $n = 10$

What is the prob. that this sequence arose from a loaded die?

Initially, the prob of choosing either die = 0.5

~~Fair~~ die: $0.5 \times p(1) \times p(2) \cdots p(4)$

$0.5 \times \left(\frac{1}{6}\right)^{10} = 0.5 \times 10^{-9}$

Another seq: 1 6 6 5 6 2 6 6 3 6

What is prob that seq of states for this is FFF....F ?

It will be the same – $0.5 \times \left(\frac{1}{6}\right)^{10} \approx 10^{-9}$

$\pi = LLL \cdots L$: $0.5 \times p(1) \times p(6) \times \cdots p(6)$

$= 0.5 \times \frac{1}{10} \times \frac{1}{2} \times \cdots \frac{1}{2} \simeq 0.5 \times 10^{-7}$

$\frac{P(LLLLLL\ldots L)}{P(FF\ldots F)} = 100 \Rightarrow$ 100 times more likely that its loaded & not fair.

Properties of HMM –

All emission prob for all states $\{e\}_\pi$, $a(\pi_i \to \pi_j)$, $a(0 \to \pi_i)$

We can ask 3 questions –

Find $P(x|\theta)$

Evaluation – Given 2 HMMs, can I arrive at which one is more likely?

Find seq of states π that maximizes $P(x, \pi|\theta)$ (?)

Decoding – based on seq, can I decipher the hidden states?

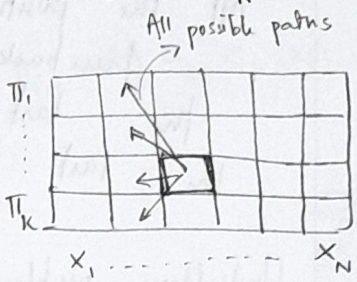
Learning – how do I learn ~~from~~ ~~given~~ values & use them?

Lecture (Recorded)

Given a sequence, there are hidden states. So we construct HMMs —

$A_{\pi_i \to \pi_j}$, $E_{\pi_l}(x_i)$, $A_{0 \to \pi_i}$

Given $X = x_1, x_2 \dots x_N$ find $\pi = \pi_1, \pi_2 \dots \pi_N$ that maximises $P(x, \pi)$



So we create a grid which has the probabilities of each x_i being in π_i

We compute the optimal path through this grid by —

$e_{\pi_i}(x_1) \cdot a_{\pi_i \to \pi_j} \cdot e_{\pi_j}(x_2) \dots$

VITERBI MATRIX

We use dynamic programming for this matrix.

It's called the Viterbi algorithm.

V matrix : $V_{\pi_j}(x_i)$ — probability to be in π_j given the character is x_i

states k, positions i

$$V_k(x_i) = \max_{\{\pi_1, \dots \pi_{i-1}\}} P(x_1, \dots x_{i-1}, \pi_1 \dots \pi_{i-1}, x_i, \pi_i = k)$$

Inductive method

$$V_l(i+1) = \max_{\{\pi_1 \dots \pi_i\}} P(x_1 \dots x_i, \pi_1 \dots \pi_i, x_{i+1}, \pi_{i+1} = l)$$

$$= \max_{\{\pi_1 \dots \pi_i\}} P(x_{i+1}, \pi_{i+1} = l \mid x_1 \dots x_i, \pi_1 \dots \pi_i) \cdot P(x_1 \dots x_i, \pi_1 \dots \pi_i)$$

Conditional prob — prob ~~given~~ of these condition

Say we're only interested in P of getting to l when prev. state was k.

$$= \max_{\{\pi_1 \dots \pi_i\}} P(x_{i+1}, \pi_{i+1} = l \mid \pi_i) \, P(x_1 \dots x_{i-1}, \pi_1 \dots \pi_{i-1}, x_i, \pi_i)$$

Prob of going to π_{i+1} through π_i × prob of π_i

$$= \max_{\{\pi_1 \dots \pi_i\}} P(x_{i+1}, \pi_{i+1} = l \mid \pi_i = k) \max_{\{\pi_1 \dots \pi_{i-1}\}} P(x_1 \dots x_{i-1}, \pi_i \dots \pi_{i-1}, x_i, \pi_i = k)$$

$$V_\ell(i+1) = \max_{\{\pi_1 \cdots \pi_i\}} P(x_{i+1} | \pi_{i+1} = \ell)\, P(\pi_{i+1} = \ell | \pi_i = k)\, \underbrace{V_k(i)}_{\text{From } ①}$$

$$\Rightarrow V_\ell(i+1) = \max_k e_\ell(x_{i+1}) \cdot a_{\pi_k \to \pi_\ell} \cdot V_k(i)$$

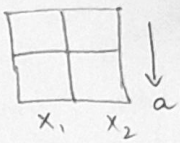
$$* \left\{ V_\ell(i+1) = e_\ell(x_{i+1}) \max_k a_{k\ell} V_k(i) \right\} * \quad : \text{Recursive formula}$$

Initialization : $\boxed{V_0(0) = 1 \qquad V_k(0) = 0}$ ⇒ Always, position 1 is in state 1

Time $\sim O(K^2 N)$ Space $\sim O(KN)$

All the positions in the matrix are computed & a traceback is done from the highest value in the last column i.e. most likely state for the last position x_n

Underflow problem : V can become too small to compute. → $0.5 \times 0.2 \times 0.5 \times \cdots$ very small no. $\sim 10^{-300}$ So we take log —



$$V_\ell(i) = \log e_\ell(i) + \max \left[V_k(i-1) + \log a_{k\ell} \right]$$

Example of Fair vs loaded die — to show how good Viterbi is.

* Viterbi can miss small changes
* It gets the transitions right but the boundary predictions are not accurate.

Evaluation

What's the prob. of getting a seq. x, given HMM ?

$$P(x) = \sum_\pi P(x, \pi) = \sum_\pi P(x|\pi) \cdot P(\pi)$$

Instead of computing max, we compute sum of all possible paths This is called the Forward Algorithm —

$$F_k(i) = P(x_1 \cdots x_i, \pi_i = k)$$

$$e_k(x_i) \sum_\ell F_\ell(i-1)\, a_{\ell \to k}$$

Final $P(x)$ is calculated by adding the values in the last column

Computational complexity : $O(nm^2)$ n: no. of positions m: no. of states

Viterbi Computational complexity

$O(K^2 N)$

Refer to Khan Academy

22/9

Recorded lecture

Learning - if e_i / a_i are not specified
find parameters $\theta = (e_i, a_{ij})$ that maximizes $P[x, \theta]$

Derivation of forward matrix -

$$F_k(i) = p(x_1 \dots x_i, \pi_i = k)$$

$$= \sum_{\pi_1 \dots \pi_{i-1}} P(x_1 \dots x_{i-1}, \pi_1 \dots \pi_{i-1}, \pi_i = k)\, e_k(x_i)$$

$$= \sum_{l} \sum_{\pi_1 \dots \pi_{i-2}} P(x_1 \dots x_{i-1}, \pi_1 \dots \pi_{i-2}, \pi_{i-1} = l)\, a_{lk}\, e_k(x_i)$$

$$= \sum_{l} F_l(i-1)\, a_{lk}\, e_k(x_i)$$

$$\boxed{F_k(i) = e_k(x_i) \sum_{l} F_l(i-1) \cdot a_{lk}}$$

Initialization : $\boxed{F_0(0) = 1}$ $\boxed{F_k(0) = 0}$ $\forall\ k > 0$

Termination : $\boxed{P(x) = \sum_k F_k(N)}$ — prob of whole seq

Backward Algorithm

We want to compute $P(\pi_i = k \mid x)$, prob dist of i^{th} position given seq. x.

$$P(\pi_i = k, x) = P(\underbrace{x_1 \dots x_i, \pi_i = k}_{F_k(i)}, \underbrace{x_{i+1} \dots x_N}_{\text{backward } B_k(i)})$$

$$= P(x_1 \dots x_i, \pi_i = k)\, P(x_{i+1} \dots x_N \mid x_1 \dots x_i, \pi_i = k)$$

$$= P(x_1 \dots x_i, \pi_i = k)\, \underbrace{P(x_{i+1} \dots x_N \mid \pi_i = k)}$$

"starting from i^{th} state $= k$, generate the rest of x"

Then, $P(\pi_i = k \mid x) = P(\pi_i = k, x) / P(x)$

Both F & B algorithms ultimately give the likelihood of occurrence of $X\ (= x_1 \dots x_n)$ based on the HMM model (i.e θ parameters)

Computational complexity - $O(K^2 N)$ time, $O(KN)$ - space } F & B

Backward algorithm derivation

$B_k(i) = P(x_{i+1} \cdots x_N \mid \pi_i = k)$ — starting from i^{th} state $= k$, generate the rest of x

$$= \sum_{\pi_{i+1} \cdots \pi_N} P(x_{i+1} \cdots x_N, \pi_{i+1} \cdots \pi_N \mid \pi_i = k)$$

$$= \sum_{l} \sum_{\pi_{i+1} \cdots \pi_N} P(x_{i+1} \cdots x_N, \pi_{i+1} = l, \pi_{i+2} \cdots \pi_N \mid \pi_i = k)$$

$$= \sum_{l} e_l(x_{i+1}) \, a_{kl} \sum_{\pi_{i+1} \cdots \pi_N} P(x_{i+2} \cdots x_N, \pi_{i+2} \cdots \pi_N \mid \pi_{i+1} = l)$$

$$\boxed{B_k(i) = \sum_{l} e_l(x_{i+1}) \, a_{kl} \, B_l(i+1)}$$

Initialization : $B_k(N) = 1 \quad \forall \; k$ (?)

Termination : $P(x) = \sum_{l} a_{0l} \, e_l(x_1) \, B_l(1)$ — sum over the last column

↓ Inside the summation

Problem — 31:32 min

Use min in Viterbi

\# $\sum_{x} P(x) = 1$ — Summing over all possible sequences where $P(x) = \sum_{k} F_k(N)$

Learning

Re-estimate the parameters of model based on training data

1. Estimation when "right answer" is known
2. Estimation when "right answer" is unknown.

Q : update the parameters θ of model to maximize $P(x \mid \theta)$

1. When right path is known

Given $x = x_1 \ldots x_N$ for which $\pi_1 \ldots \pi_N$ is known

We can show that max. likelihood parameters θ are —

$$\boxed{a_{kl} = \frac{A_{kl}}{\sum_{i} A_{ki}}} \qquad \boxed{e_k(b) = \frac{E_k(b)}{\sum_{c} E_k(c)}}$$

A : no. of times $k \to l$ transmission occurs in π

E : no. of times state k emits b in x

Drawback –

If there's little data, there may be overfitting
$P(x|\theta)$ is maximised but θ is unreasonable.

0 probability – BAD!

Overfitting ⇒ same HMM doesn't explain a bigger data set. cannot.

Pseudocounts

Solution to prevent overfitting

A_{kl} = no. of $k \to l$ + r_{kl}

$E_k(b)$ = no. of time k emits b + $r_k(b)$

r_{kl} & $r_k(b)$ are pseudocounts representing our prior belief.

Larger pseudocounts ⇒ stronger prior belief.

Example – dishonest casino ~45 min

2. When the right answer is unknown.

Idea :
- We start with our best guess for A_{kl}, $E_k(b)$
- We update parameters of the model, based on our guess
- We Repeat

Given $x_1 \dots x_N$ for which $\pi_1 \dots \pi_N$ is unknown,

Principle : Expectation maximisation i.e. θ that increases $P(x|\theta)$

1. Estimate A_{kl}, $E_k(b)$ is training data
2. Update θ according to A_{kl}, $E_k(b)$
3. Repeat 1, 2 until convergence

Iteration –

1. Perform Viterbi to find π^*
2. Calculate A_{kl}, $E_k(b)$ according to π^* + pseudocounts
3. Calculate new parameters a_{kl}, $e_k(b)$

Repeat until convergence

(?) Initial evaluation of Viterbi – by taking arbitrary values of e and a?

Notes: Not guaranteed to increase $P(x|\theta)$

* Guaranteed to increase $P(x|\theta, \pi^*)$ *

In general, worse performance than Baum - Welch

22/9

Lecture 12 - Discussion

Youtube : Mathematical-Monk + Viterbi

Overview + Discussion

25/9

Recorded lec : HMM learning

Posterior decoding - Predicting optimal state at a specific position

To find the prob. that state of the i^{th} position is k -

$$P(\pi_i = k \mid x) = \frac{F_k(i)\, B_k(i)}{P(x)}$$

Because of F & B we have the P of getting a particular state in i^{th} position for a given seq X

$$P(\pi_i = k \mid x) = \frac{P(\pi_i = k, x)}{P(x)} = \frac{P(x_1 \dots x_i, \pi_i = k, x_{i+1} \dots x_n)}{P(x)}$$

$$= \frac{P(x_1 \dots x_i, \pi_i = k)\, P(x_{i+1} \dots x_n \mid \pi_i = k)}{P(x)}$$

seq before i and after it, is independent so F_k & B_k can be multiplied

$$\boxed{P(\pi_i = k \mid x) = \frac{F_k(i)\, B_k(i)}{P(x)}}$$

Then we can ask, what is the most likely state at position i of sequence X

Define $\hat{\pi}$ by posterior decoding -

$$\boxed{\hat{\pi}_i = \text{argmax}_k\, P(\pi_i = k \mid x)}$$

Estimating new parameters.

Now that max likelihood of states is known, we can calculate the transition probabilities.

At each position i, probability transition $k \to l$ is used:

$$P(\pi_i = k, \pi_{i+1} = l \mid x) = \frac{P(\pi_i = k, \pi_{i+1} = l, x_1 \dots x_n)}{P(x)} = \frac{Q}{P(x)}$$

$$* P(X|\theta) = \sum_{k=1}^{N} \sum_{l=1}^{N} F_k(i)\, a_{kl}\, B_l(i+1)\, e_l(i+1)$$

$$Q = P(x_1 \dots x_i, \pi_i = k, \pi_{i+1} = l, x_{i+1} \dots x_N)$$

$$= P(\pi_{i+1} = l, x_{i+1} \dots x_n \mid \pi_i = k)\, P(x_1 \dots x_i, \pi_i = k)$$

$$= P(x_{i+2} \dots x_n \mid \pi_{i+1} = l) * P(x_{i+1} \mid \pi_{i+1} = l) * P(\pi_{i+1} = l \mid \pi_i = k) * F_k(i)$$

$$= B_l(i+1)\, e_l(x_{i+1})\, a_{kl}\, F_k(i)$$

So, $$P(\pi_i = k, \pi_{i+1} = l \mid x, \theta) = \frac{F_k(i) * a_{kl} * e_l(x_{i+1}) * B_l(i+1)}{P(x|\theta)}$$

So A_{kl} (# of times $k \to l$, given current θ) is —

$$A_{kl} = \sum_i P(\pi_i = k, \pi_{i+1} = l \mid x, \theta) = \sum_i \frac{F_k(i)\, a_{kl}\, e_l(x_{i+1})\, B_l(i+1)}{P(x|\theta) *}$$

Similarly, $$E_k(b) = \frac{1}{P(x|\theta)} \sum_{i | x_i = b} F_k(i)\, B_k(i)$$

Does this give us the optimal values of a & e?
This is not guaranteed — its a maxima but it need not be the absolute maxima

This algorithm (Baum-Welch) is better than viterbi, but still doesn't give the optimal (it converges to local optimum)

Time complexity : # iteration $\times$ $O(k^2 N)$

But this does guarantee to increase the log likelihood

When there are too many parameters / too large models — there is the risk of overtraining

Algorithm : * Initialization : best-guess for θ

* Iteration : F, B
  Calculate A_{kl} $E_k(b)$ given θ
  Calculate new θ : a, e
  Calculate new log-likelihood $P(x|\theta_{new})$
  until it converges.

Viterbi learning finds/maximises the ~~p of x~~ state(s) π^* that gives max probability of X

Whereas Baum-Welch finds the state of max probability in each of the positions

Limitation of HMMs —

* If transition prob are high, then predictions / decoding doesn't work well $\Rightarrow a_{\pi_1 \to \pi_2} = 0.95$: bad.

27/9

Live lecture

Discussion of HMM learning

Question: Code the algorithms.

20/10/24

BI3134 - Part II

Sequence motifs

Conserved sequences of identical or similar patterns
They can be biologically important - eg. nuclear localisation signal in proteins. (Helps infer functional similarity)
Motifs can found in -
DNA, RNA, proteins
within different molecules
across species

Finding motifs using Gibbs sampling

Example - in genomic sequences, we're looking for a motif upstream of a particular gene
The motifs are similar, not identical
They can be present at slightly different positions in a particular region.

Motif discovery by alignment

1. local alignment of multiple sequences
2. Highly conserved regions are isolated
3. Construct a profile matrix of the conserved region

PSI BLAST : PSSM :: Motif finding : ~~PS~~ PWM
profile weight matrix

Constructing a profile matrix

From the local alignment, in each position we calculate how many A T G C s there are

A	0	0	3
T	1	2	1
G	0	2	0
C	3	0	0

This can be written in probability i.e. occurrence frequency
But prob can never be 0, so we use pseudocounts for where its 0.
Other values are adjusted so that each column adds up to 1

Assumptions –

1. Every sequence has exactly one instance of the motif
2. The motif has a fixed length L.

Probability of l-mers

$Prob(a/p)$ = P of l-mer 'a' detected with profile P

$$P(a/p) = \prod_{i=1}^{l} P_{a_i,k}$$

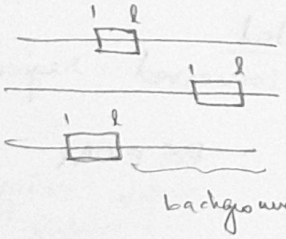
$P_{a_i,k}$ is the prob of letter a_i at position k.

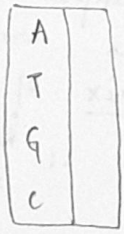
Finding the best scoring l-mer

* Sliding windows of l-mers, shifting one nucleotide at a time
* Compute prob of all possible l-mers
* Brute force method – not computationally friendly.

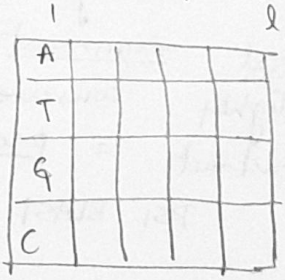
Note: Even if we find a high scoring l-mer, it need not mean something. The difference b/w highest scoring l-mer and other l-mers can be very less.

So we use the obs/exp to be sure of significance





Background model B
(exp)



Motif model M
(obs)

Using this, we score a word of length l –

$$P(w) = \prod_{i=1}^{l} B(w[i])$$

$$Q(w) = \prod_{i=1}^{l} M(i, w[i])$$

$$R(w) = \log\left(\frac{Q(w)}{P(w)}\right)$$

Motif finding dilemma

This is circular — if we know $(a_1 \dots a_n)$, we can construct M and B, and if we know M & B, we can determine the most likely motif locations in sequences $S_1 \dots S_n$

The **Gibbs sampling algorithm**

We construct M' and B' using n-1 sequences and use them to determine a_i for sequence S_i.

Algorithm:

Input : sequences $S_1 \dots S_n$

Output : $(a_1 \dots a_n)$, M', B'

Init : choose $(a_1 \dots a_n)$ arbitrarily

For $h = 1, n$ {

Compute M' & B' from $a_1 \dots a_{h-1}, a_{h+1}, \dots a_n$

Use M' & B' to find a_n in S_h

Compute new M & B

Compute score $L(a_1 \dots a_n)$*

}

* iterate until convergence

Refer to: Rouchka & Rouchka 2008

Youtube - Xiaole Shirley Liu - Stat115 ch. 10.3

1. Randomly initialize a prob. matrix — construct θ_1
2. Take out one sequence with its sites from current motif
3. Score each possible segment using θ_1 without S_1 — θ_1'
   i.e seg. 1-6, 2-7, 3-8 and so on in S_1
   * Scoring is done using $\left(\frac{obs}{exp}\right)$
   Also called giving weights to these segments.
4. Sample a site from S_1 based on the prob. distribution created using θ_1'
5. Repeat the process by removing sequences until motif converges

Lecture - 25/10/21 30/10

Structural biology

Modelling the 3D structures of proteins / biomolecules

Protein structure representation - All atom representation
Stick representation
Ribbon representation

Utility of 3D structures

The folding of polymeric chain brings together residues that are distant in sequence but close spatially. The function of molecule is heavily dependent on its structure

Evolution conserves structure, more than sequence

Patterns in space are frequently more recognizable than patterns in seq.

History and Context

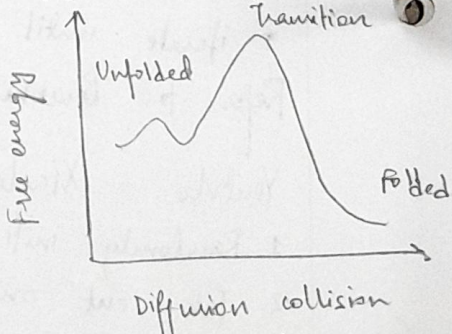
Kendrew and Perutz } - models of 3D structure
Dorothy Hodgkin } - X-ray crystallography - efficient but slow
Hence the need for comp. modelling

Elphinson's dogma?

Given a sequence of aa, it forms a unique 3D structure in a particular environment.

Protein folding

A folded protein is considered to be most stable i.e least free energy.



Homology

Similar proteins (homologous ones) have similar structure

MSA of globin proteins from different sources show that some regions / residues are highly conserved to preserve the function : binding of heme with O_2.

Phenylalanine & Histidine are conserved in all sequences.

If we find another protein with the same residues in same places, we can predict that it binds O_2.

Visualisation tool - Chimera

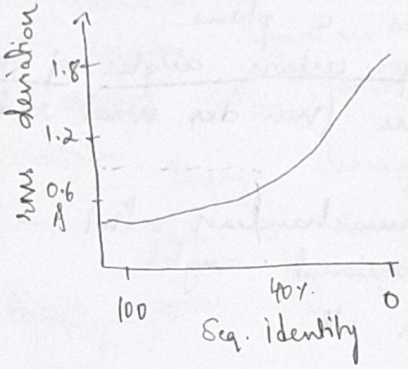
Recognising homology

Mammalian & plant globins look alike but share only 10–20% seq. identity. That's why we use PSI BLAST to recognise similarity.

Root means square deviation – the measure of difference in atomic positions when overlapped/superimposed

The rms deviation is ~ 1.4 Å b/w myoglobin & α haemoglobin chain, despite the low seq identity.

Chothia-Lesk plots



This realisation/plot gave rise to comparative modelling/homology modelling.

If a sequence has recognisable homologs with another seq whose structure had be resolved, then that knowledge can be used as a template to build/predict/model a new structure

Ab initio modelling – using the knowledge of physics to predict the 3D model, without a template

30/10

Lecture – 26/10

Unique protein folds – of the order of ~ 1000. (fold domain families)

There are called templates, based on which new proteins can be modeled – through comparative modelling. (also called threading)

Ab initio modelling,

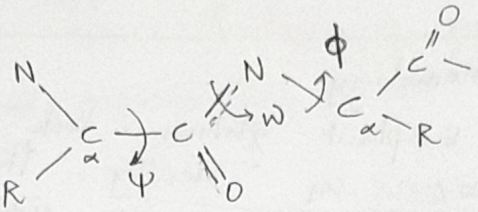
This is based on physical laws. For this, we need to know the structure of polymeric chain.

Consider: a tri-alanine

The atoms carry a charge & there are dipoles based on the orientation.

The polymeric chain has torsional degrees of freedom

The molecule can rotate about a single bond. But this rotation is constrained by potential collision / overlap of atoms



Two important contributions –

Linus Pauling : the amide bond has partial double bond and it cannot be rotated along this

$\omega = 0^\circ$ cis or 180° trans

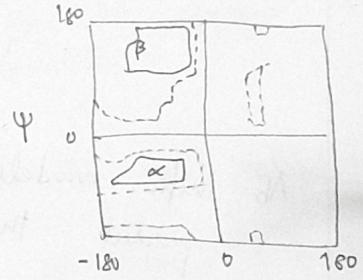
G N Ramachandran & team :-

Because of partial double bond, the atoms are contrained to move as a plane which allows for only certain angles of ψ and ϕ at which the atoms (van der waal radii) don't collide.

This gave rise to Ramachandran Plot – 2D map of allowed torsional angles.

This remains the same for all side chain residues.

Except : Glycine (permissive)
Proline (restrictive)



Observation : The hydrophobic residues coalesce to the inside of the globular protein, forming the core. This can be a test for a potential model

Levinthal's paradox.

Simplify the problem –

- Each residue can adopt one of 3 discrete groups from R plot (α, β, loop)
- Change in conformation : $\sim 10^{-12}$ s. Possible conformations $= 3^{150} \sim 10^{71}$ A protein with 150 residues would need 10^{50} years

The free energy landscape of a protein should be viewed like a funnel, not a golf course

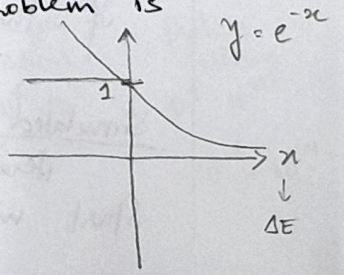
Every successive fold should take us to closer to the global minima, without trying all possible conformations.

Monte Carlo simulations
It is a minimum energy search choosing states by their probability of occurence
If is not minimisation Doesn't guarantee to find the minima
The energy minima is found by considering the energy of transition from one conformation to another
Energy change : ΔE
Prob. of change : $e^{-\Delta E/kT}$
Accept moves by probability of their occurence

30/10

Lecture - 27/10
Monte Carlo Simulations
We need to find the 3D conformation with minimum energy. Usually E is calculated using Coulombic / Van der waal interactions.
Through MC simulations, we search for minimum energy state Moves (conformation changes) are accepted acc. to prob of the move
$P(0 \rightarrow 1) = e^{-\Delta E/kT}$

Energy landscape of this protein folding problem is shaped like a funnel



If ΔE is negative i.e $E_1 < E_0$ (i.e. more stable), then, $P(0 \rightarrow 1) = 1$
Based on this, moves are accepted.

Metropolis MC algorithm

If $P(0 \to 1) = 0.3$, then acc. to this algorithm, this move is accepted 3 times out of 10.

This is done using a version of Uniform Random number generator.

$E_i \to E_j$ is not rejected just because $E_j > E_i$.

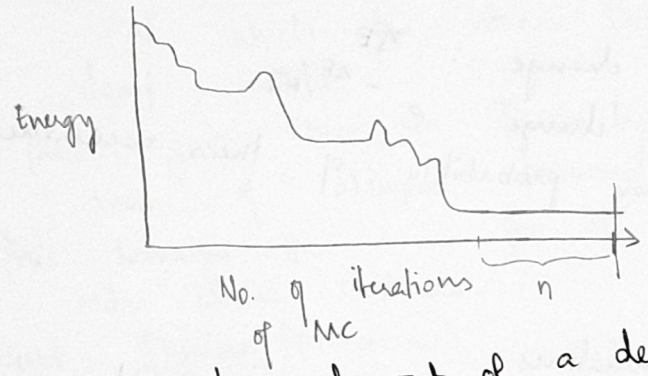
The move is accepted based on its probability

If $P = 0.3$, and the generated random number is $> P$ ⇒ Move rejected

rand $\leq P$ ⇒ Move accepted

MC simulation needs to have —

1. Move set — generate random no.
2. Storing scheme — ΔE calculation.
3. Accept - Reject criteria — prob $\geq$ rand → accept
   Metropolis criteria

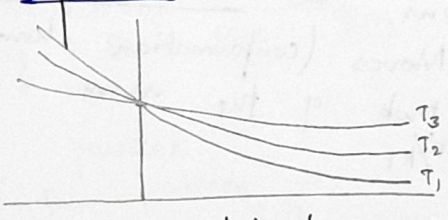


Since the minima existed for longer than n steps, we accept this as global minima & stop the simulations

But how to get out of a deep local minima?

We can repeat the MC simulations after increasing T which will change the transition probabilities.

The 'unfavorable' transition prob. will increase with T



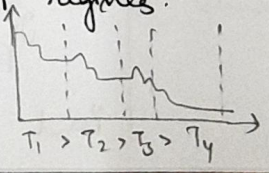
$T_3 > T_2 > T_1$

Greater the T, higher the 'unfavorable' probability.

If the same state/conformation is achieved across all T, then it's the global minima

Simulated annealing — a trick in MC simulation where iterations are carried out in different T_i regimes.

Start with v. high T, then keep decreasing
↳ won't get stuck in local min.



Defining the move set – changing conformation of ϕ & ψ, 2/3 positions at a time, using coarse moves initially, then fine moves.

Coarse move – 3x (ϕ, ψ) from 0° to 90°

Fine moves – change ψ 0 to 10°

Limitations of MC

Its NOT a minimisation algorithm

Won't work well if the energy landscape is not funnel shaped

* Genetic Algorithm – simulating evolution *

Reproduction – Mutation – Shifts – Cross overs – Selection

This uses evolutionary mechanisms to create new combinations of 3D structure and evaluate the new models.

1/11/21

Lecture Rec – Molecular dynamics simulations

Recall : MC sampling – move set

scoring – energy value

optimisation – based on prob. of move

Given a 3D structure, what sequence would best fit this structure?

What is the MC move set and score for this?

If the change in a torsion angle causes a collision b/w different parts of protein, the E_1 would be very very high. So the prob. of this will be v. v. low.

Rates of motion	Bond stretching	Angle bending	Bond rotatn	Protein tumbling	Enzyme reaction	Protein folding
Time (ps)	0.01	0.1	1	20×10^3	$10^6 - 10^9$	$10^9 - 10^{12}$
Frequency (cm^{-1})	3000	300	30	0.0015	$10^{-5} - 10^{-8}$	$10^{-8} - 10^{-11}$

Molecular potential energy

It's calculated by summing over different terms which account for torsional energy, bond flexing and so on. It's based on harmonic potential (as it's considered as a spring).

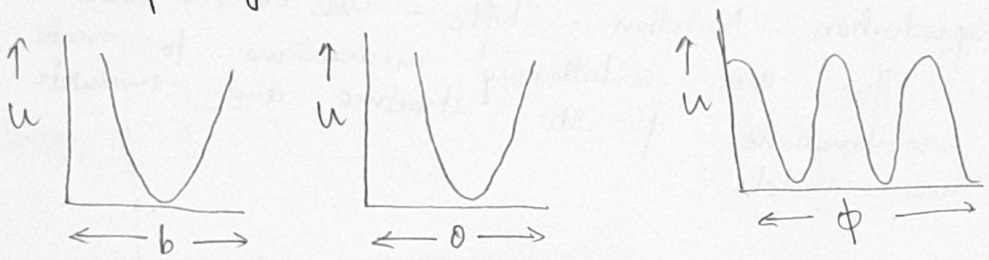
$$U = \underbrace{\sum_{\text{All bonds}} \frac{1}{2} K_b (b - b_0)^2}_{\text{vibration}} + \underbrace{\sum_{\text{bond angles}} \frac{1}{2} K_\theta (\theta - \theta_0)^2}_{\text{bond bending}} + \underbrace{\sum \frac{1}{2} K_\phi [1 - \cos(n\phi + \delta)]}_{\text{torsional}}$$

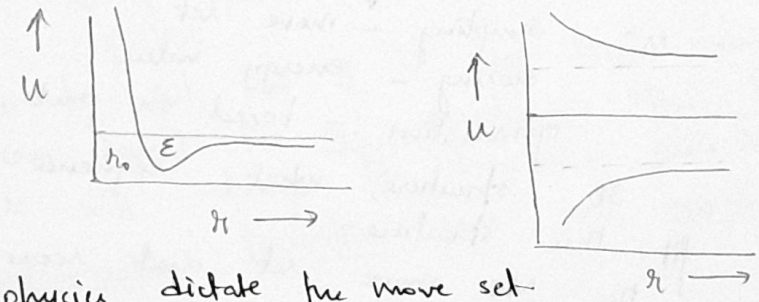
$$+ \underbrace{\sum_{\substack{\text{all pairs of non-bonded atoms}}} \left(\varepsilon \left[\frac{r_0}{r} \right]^{12} - 2 \left[\frac{r_0}{r} \right]^{6} \right)}_{\text{van der waal}} + \underbrace{\sum_{i,j} \frac{332 q_i q_j}{r}}_{\text{electrostatic}}$$

Vibrational & bending have a single minima

But torsional angle has multiple minima – trans + gauche or – gauche. So, $\cos\theta$ takes care of that

When atoms collide, the van der waal term will be very high (∵ r will be very low)





We let physics dictate the move set

Molecular dynamics simulation

This uses the energy definition to simulate a fold based on the atoms around it.

BUT our equation is for a pair of atoms. This is where time period comes in.

Assume: How do all other atoms influence the position of an atom in a timestep (10^{-15} s) smaller than the time it takes to flex a bond (10^{-14} s)?

Then with the new position of atom 1, calculate the shift in atom 2, and ~~so~~ so on.

$\Delta t = 10^{-15}$ s

We get a trajectory of all the atoms

Position of atom 1 $x(t+\Delta t) = x(t) + v(t)\cdot\Delta t + \frac{[4a(t) - a(t+\Delta t)]\Delta t^2}{6}$

$v(t+\Delta t) = v(t) + \frac{(2a(t+\Delta t) + 5a(t) - a(t-\Delta t))\Delta t}{6}$

$v(t)$ is dependent on temperature - Boltzmann distribution

$\sum \frac{1}{2}mv^2 = \frac{1}{2}k_B T$

If there are attractive forces, then the trajectory based on their velocities (& ultimately PE) will evolve to give a folded state, after 10^{12} steps. (upto 1ms to fold).

Steps: Starting coordinates of atoms
Initialize atomic velocities
Iterate: position → forces → new position

Lecture - 2/11/21

Potentially energy in MDS doesn't account for covalent bonds like disulphide or hydrogen bonds. This has to be added externally

Alfafold 2, Rosetta fold - programs that use Artificial Neural Networks (ANN) to predict 3D structure

Homology / Comparative modelling: Satisfation of spatial restraints
Considers that the target sequence is part of a protein family whose template protein structure is known.
We assume that their structures will be similar.

Steps:
1. Template search (Protein Data Bank - 50,000 unique protein structure) * PSI BLAST
2. Target - template alignment (accuracy of this alignment - v. imp.)
3. Deduce restraints from the aligment + build model
4. Evaluate the model - $\frac{obs}{exp}$

Step 2 : Tg - Tm alignment - global DP

If residues in Tg & Tm align, then we say that they occupy similar spatial position — distance, angle, torsion angle & volume.

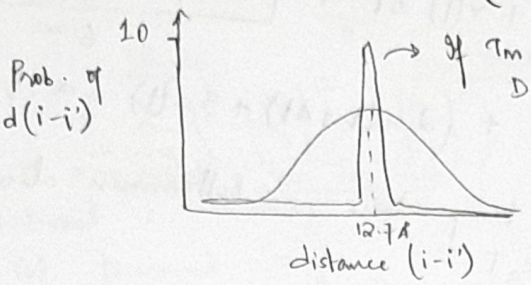
Tg — i — i' — i"

Tm — j — j' — j"

We know, $d(j-j') = 12.7$ Å

If our alignment is accurate,

$d(i-i') \cong (12.0 - 13.0)$ Å



Live lec - 3/11/21

Comparative modelling

$d(i-i') = f(d(j-j'))$

The peak of the distribution of this function would be centred at $d = j-j'$

The closer the alignment of Tm & Tg, the narrower the distribution of $d(i-i')$.

Say there's template 2 (Tm2) where $k-k'$ corresponds to $i-i'$.

Something about how the distribution will look.

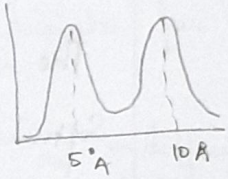
Just like distance, other spatial features (bond angle, torsion angle, volume etc) of Tm and Tg are also similar and follow a distribution.

For any feature i : $P_i = P(f_{(i,j,k)})$

The probability of the molecule : $P = \prod_i P_i$
↳ prob. distribution fn.

By this definition, we assume that these probabilities are independent. Because the system of molecules is so interconnected, optimising the pdf will give us the optimal structure

Question
If there's a bimodal pdf distribution, then what's the most likely optimal value?



Constructing pdf
we know that, if Tm & Tg were identical, then it would be a delta function & the less identical the sequences, flatter the pdf.
So, based on seq alignment, we can instruct the program to construct a pdf.

Modeller - program that uses satisfaction of spatial constraints.

The width of pdf is decided based on
- local sequence identities
- the distance from gaps (closer to gap, larger the width)

There is a linear combination of all these variables.
How do we get the weights in the linear comb?
By using it on the models / structures we know already. Those structures are used to train this.

After we have P_i from all features, we maximise $\prod_i P_i$ with respect to all the features —

$$\frac{\partial \prod_i P_i(x)}{\partial x} = 0$$

Expectation of bimodal pdf distribution?

Live lec - 8/11/2021

Recap :

Based on a template model, we have a distribution of probability of various spatial restraints.

The pdf's were deduced based on sequence alignment

Software : MODELLER

Scoring function (for points & restraints)

$$P(R/I) = \prod_i P_i(r_i/I_i)$$ - we have to maximise this

R - all deg of freedom

I - all information

r_i - i^{th} restrained feature

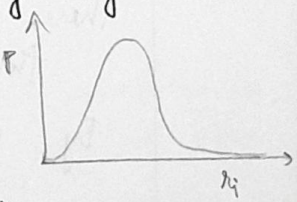
I_i - information about i^{th} feature

P is the molecular probability distribution.

Optimisation of Pseudo-energy score

$$E = -\ln P(R/I)$$: consider it as a energy function lower the E, better

The pdf is asymptotic, so sometimes sampling might lead to a feature outside the restraint, but since it has a v low p, it won't contribute to E.



If a pdf is bimodal, after many iterations, we will get two highly scored solutions. This can be resolved with external information, or maybe the molecule exists in two states.

Threading

If we don't know a suitable template for a target sequence, we 'thread' it through all possible templates and find the energy of each, so we can pick the right template

"Biological" significance of modelling errors

Errors margins are bigger when alignments don't work

Common reasons —

1. Incorrect template
2. Misalignment - because Tm & Tg are distant homologues
3. Regions without a template - we won't get restraints for this region (MC for a small stretch)
4. Distortions / shifts in aligned region
5. Sidechain packing - the sidechains of Tm & Tg are v. different based on seq identity. There'd be holes because of inefficient packing of sidechain

Methods to get restraints / verify structure

NOE - NMR

FRET

Cross-linking. proteolyse. then mass spectrography

Model Evaluation - 4th step

* One test of the model is to make sure that the predicted structure has a hydrophobic core.

Rarely, the template chosen is wrong.

Commonly, the target-template alignment has to be redone

But this is just a qualitative / categorical evaluation.

* But we need an evaluation that gives feedback on what to tweak to make it better

Why use protein structure modelling? - Baker & Sali 2001

Example : human BRCA1 and its 2 BRCT domains

The models of the BRCT domains were useful in figuring out why some mutations lead to cancer while others don't.

A decision tree was constructed to predict the functional impact of genetic variants.

Based on a training set, the tree classified the unknown mutations into harmful or not harmful

They found that mutations in amino acids clustered in a certain spatial scale region led to cancer.

Williams et al. 2004

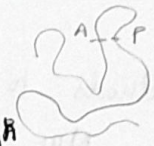
Live lec - 9/11/2021

Model evaluation

Accessible surface area - area available to solvent should have more hydrophylic residues

In a protein, say there are 2 residues A & P

In the PDB, there are 50k structures.



To compute the observed A-P interaction, we calculate the no. of times $d(AP) \leq 4\text{Å}$

$$\frac{obs}{exp} \cong \frac{[AP]}{[A][P]}$$

Total no. of aa interactions

$$\Delta G = -RT \ln \frac{obs}{exp}$$

For the reaction $[A] + [P] \longrightarrow [AP]$

$$\Delta G = -RT \ln(k) = -RT \ln \frac{[AP]}{[A][P]}$$

Similar to $-\ln\left(\frac{obs}{exp}\right)$

Using the known structures, we'll set up a scoring scheme so we get minimal ΔG

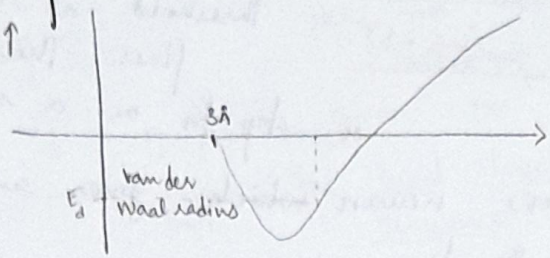
How to calculate the expected value of A-P interaction?

Exp value is : if residues could pick this interaction. we shuffle the protein seq. 1000 times, and each time, we calculate how often A-P occur together. This gives us how often they'd interact at random - the expected value $[A][P]$

in the same protein structure

for instance $\frac{obs}{exp}$ of cys-cys is > 1

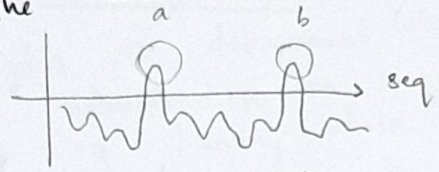
we define $d(A-P) \leq d_{thr}$



For each of possible 210 pairs of aa interactions, we can get a graph like this

So, for a d(X,Y), we can get a E_d which is the score for that pair.

With this score, we can find the energy of the molecule, with seq specific energy



Because energy of a, b regions is positive, it means that region is not stable.

Such a graph shows us which areas are problematic & need to be re-aligned / re-evaluated

Fold It - game

Live lecture - 10/11/2021

Something about threading, exp. value of A-P interactions and Fold It demonstration.

Redo lec!

Artificial Neural networks

It was used in Alphafold 2 by Deepmind

Several neurons are connected to others through 'synapses'

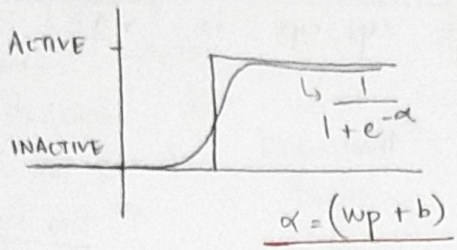
If two neurons are linked, when one fires, the subsequent one also fires

The inputs from each neuron is weighted and there is also a bias : $(wp + b)$

↓

basal level

Based on the total/net input received, the neuron decides whether or not to go from Inactive → Active



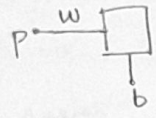
Each neuron has an activation function, $f(wp+b)$ which determines the threshold at which neuron fires. This can be a step fn or a sigmoidal fn.

This is for one neuron which gives an output of 0 or 1.

Live lec - 15/11/2021

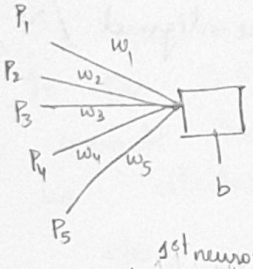
Neural networks

f: activation function



$f(wp+b)$

net input: α

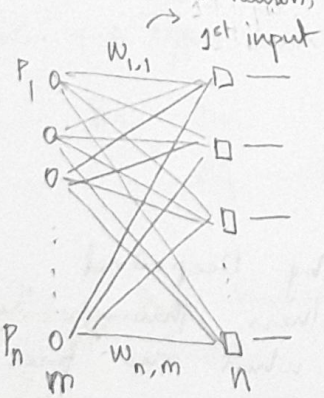


Each neuron can have multiple inputs

s.t. $\alpha = (w_1 p_1 + w_2 p_2 \cdots w_n p_n + b)$

Then this input goes into the activating fn.

There can be layers of neurons —



Such a structure will give us a

Weight matrix —

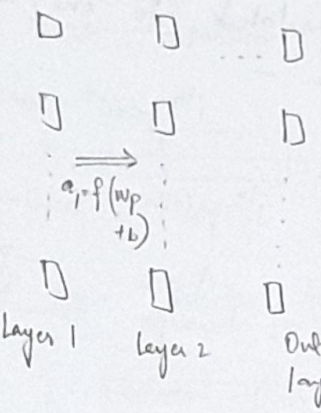
$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & & & \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

n rows × m columns

$$a = f(Wp + b)$$

p: m rows × 1 column vector

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}$$



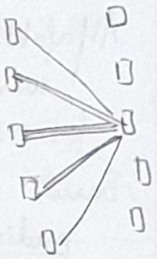
Network with a multiple layers has several weights and biases for each layer

They can be used to predict the secondary structure of a protein, given the sequence.

The input layer is as long as the sequence, there are hidden layers and output layer has 3 neurons - α, β, coil

The inputs at i^{th} position receives decreasing weights from previous layer as neurons are away from i.

'Closer' the input neuron, stronger the weight



The weights are determined by training the network on a huge amount of training data

The output is maximised when tweaking the weights doesn't improve the efficiency anymore.

Because residues closer affect 2° structure rather than far away residues

Live lecture - 16/11

Alphafold 2

The input signal of i^{th} residue would be weighted propensity of neighbouring residues to take on α-helix or β-sheet state

S_α: α-propensity

$$\alpha\left(w_1 S_\alpha(i-3) + w_2 S_\alpha(i-2) \dots + w_5 S_\alpha(i+3)\right)$$

If the input is strong enough, then output of i^{th} residue will be α

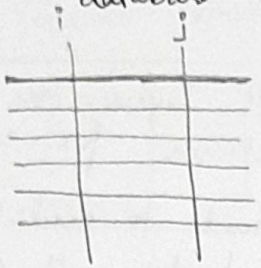
A well trained network has optimal weights.

Foldit - the game uses a software called Rosetta.

Recall: homology modelling doesn't capture the model well when the sequence segments don't align

Alphafold has been successful in capturing the differences

Alphafold does a MSA of Tg seq. with all other database sequences (say, using PSI BLAST).

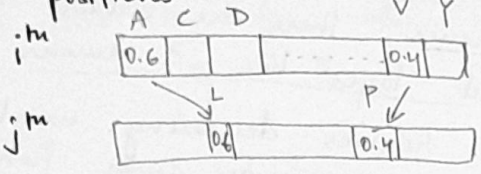


Then it tries to see if i^{th} & j^{th} positions are correlated to one another.

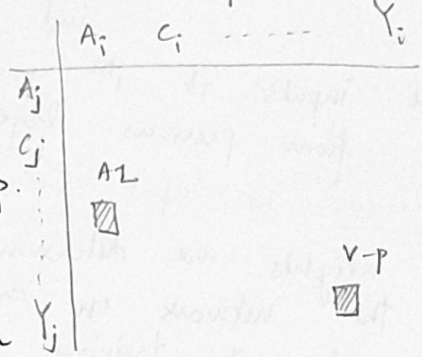
Eg: of correlation –

A	L	i^{th} positn : alanine or valine
A	L	
A	L	j^{th} positn : leucine or proline
V	P	
V	P	A – L V – P
A	L	always correlated.

Correlation coefficient can be calculated for any 2 positions –



Correlation coeff. matrix



Alfafold creates such a correlation coefficient heatmap for all positions.

Based on this, it tries to predict the distance between i and j.

If there is high correlation, the implication is that i & j are very close in 3D space

By rule of thumb, $d(i,j) \leq 8\text{Å}$

If i changed, j also needed to change to preserve the 3D structure.

If all residues are the same at a position, then its useless – we won't be able to find a correlation. So, completely conserved positions and no conservation / random positions have to be ignored.

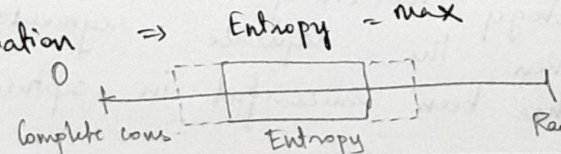
How to selectively choose semi-conservative positions?

Introduce **Entropy** : $\sum_i p_i \ln p_i$

p_i : prob. of finding i amino acid at the position.

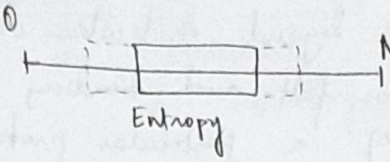
Complete conservation $\Rightarrow p_i = 1 \Rightarrow$ Entropy = 0

No conservation $\Rightarrow$ Entropy = max



Where and how does the sequence distance b/w i & j come into the pictures?

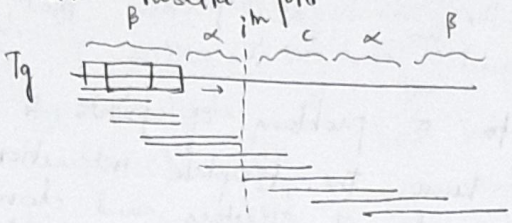
~~Live lecture - 22/11~~



Positions whose entropy is in the mid range should be considered for correlations.

If we expand the limit, then we'll get a weaker correlation

How Rosetta fold works



Cut Tg into segments & compare with all T_m sequences & align it. Slide the window by 1 position & align again — so for each position i, we get a 2° structure probability, which is compiled

In Alphafold, based on all sequences in PDB, we can construct prob. distribution of distance b/w any two ~~correlated~~ amino acids (?).

Then for Tg sequence, we find the configuration of pairwise distances that maximises the total score of all probabilities.

In Alphafold, ANNs, infer the probability of the dist (i,j)

Instead of predicting 2° structure, they predict the dist (i,j)

This prediction can be based on torsion angles

Ultimately, we want the combination of probabilities that give minimum energy.

The moves are done in torsion angle space, and by gradient descent, neural networks predict the model.

Assignment — ChimeraX UCSF — build models.

Live lecture - 22/11/2024

Paxlovid : SARS CoV2 protease — that inhibits the growth of SARS CoV2

Molecular Docking

Strategies for docking small ligand molecules onto proteins

→ Ligands bound to human heart fatty acid binding protein

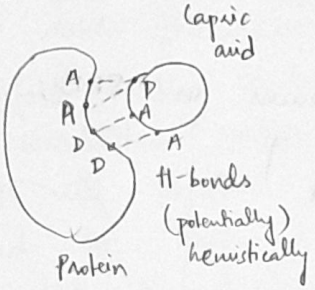
We knew the structure of a particular protein bound to capric acid.

We need to find the analogue protein that binds to lauric acid, and the way in which they bind

This can be reduced to a problem of points & restraints

We know the template interaction — with points of acceptors and donors.

We can proceed as we did with homology modelling.



Rational drug/ligand design

Properties of drugs

1. Should bind tightly to the target
   - complement the chemistry (A-D) and geometry (stereospecific) of the target
   - inhibit the function of target protein
2. Be specific
3. Not toxic ie no negative interactions with other molecules in the body
4. Lipinski's rule of 5
   - Molecular mass of 5000 Daltons or lower
   - No more that 5 H-bond donors
   - No more than 10 H-bond acceptors
   - Octanol-water partition coefficient : $\log P < 5$

Docking strategies

1. **Docking by analogy**
   The structure of the site of protein and interactions of homologous protein is known.

2. **Directed docking**
   If we only know the structure of protein and its active site, we can do trial and error — See which of the thousands of library of compounds (ZINC) fits best into the active site. Pick the best one and generate variants of the drug and see which one works best.
   Eg: Nutlin bound to MDM2 → suppresses p53 - a cancer suppressing protein
   ↳ inhibits MDM2 - better for cancer!

3. **Blind docking**
   We don't know which site to inhibit. So, we take the whole compound library & for each ligand, try to dock it in every crevice and see which docking gives the best bind. This can be done considering the ligands & protein to be rigid or flexible
   How flexible? — we Monte Carlo or Mol. dynamics

Docking questions
Where on the target will the ligand bind?
What ligand binds best?
How would it bind? (Energetics)

Molecular docking

1. Find binding site on protein
2. Choose tight binders from a library
3. Identify correct docking pose
4. Fine tune by modifying the lead compound

Programming requirements —

Efficient screening program

Accurate screening scheme

Consider ligand & receptor flexibility

Binding free energy

Binding eqns:

$$R + L \underset{k_d}{\overset{k_a}{\rightleftharpoons}} R'L' \quad \Rightarrow \quad K_a = \frac{1}{k_d} = \frac{[R'L']}{[R][L]}$$

$\frac{[R'L']}{[R][L]}$ → $\frac{obs}{exp}$

Free energy equations:

$$\Delta G^\circ = -RT \ln(K_a)$$

$$\Delta G^\circ = \Delta H^\circ - T\Delta S^\circ$$

(ΔH° ↓ enthalpy, $T\Delta S^\circ$ ↓ entropy)

Entropy reduces, but its v. hard to determine by how much. So, ΔG can't be calculated very accurately.

Docking scoring functions

1. Molecular mechanics based — (Q) Representation, Scoring, Optimisation? what ??
2. Empirical energy calculations : Autodock.

   Instead of enthalpy, we find free energy of different interactions —

$$\Delta G_{binding} = \Delta G_{vdw} + \Delta G_{elec} + \Delta G_{Hbond} + \Delta G_{desolv} + \Delta G_{tors}$$

   Desolvation : the process of displacing all the water molecules between the ligand and the protein.

3. Knowledge based — $\frac{obs}{exp}$ method

→ Empirical energy calculations

ΔG_{vdW} - 12-6 Lennard-Jones potential

ΔG_{elec} - coloumbic with Solmajer-dielectric

ΔG_{hbond} - 12-10 potential with Goodford directionality

ΔG_{desolv} - Stouten pairwise atomic solvation parameters

ΔG_{tors} - no. of rotatable bonds

→ Knowledge-based

$$\text{quantity} = -\ln \frac{g_{obs}}{g_{exp}}$$

Calculating exp value - using a large library of docked structures

$$\Delta W_{i,j}(r) = W_{i,j}(r) - W(r) = -\ln \frac{g_{i,j}(r)}{g(r)}$$

$$g(r) = \frac{\sum_i \sum_j g_{i,j}(r)}{i \times j}$$

Example of autodock - Camphor binding to cytochrome PP450

Autodock predicts docking based on empirical energy calculation

There are many many docking software

→ Docking prediction is difficult -

$\Delta G_{binding}$ given by computational predictions is very high - its not realistic, but it helps in ranking the interactions of various ligands.

This happens because energy of binding is not equal to energy of interaction.

$$\Delta G_{binding} = \Delta G_{interaction} - \Delta G_{solvation}$$

30/11

Next Generation Sequencing (NGS) - 23/11/2021

Initial method - shotgun sequencing

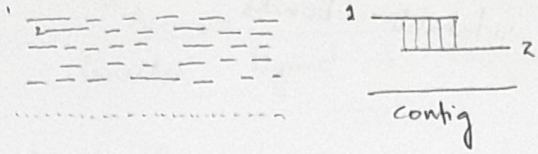
Sequence 200-300 bp length of segments and align them against a reference sequence.

If reference is not there! we need to do de novo assembly

Problem: Given a collection of reads, we reconstruct the genome from which it was derived.

Challenges: Repeats in the genome
Sequencing errors (substitution, deletion, insertion)
Huge size of the data

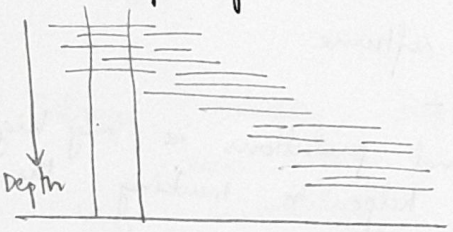
→ Overlap layout consensus (Celera, Newbler)
Suitable for long reads, highly parallelizable



Contig is the stretch including 1 & 2 i.e union of the two segments.

Rule for overlap
If the suffix of one segments & prefix of another has k-matches i.e k-mer region, then overlap can be accepted.
k can be 50 bp or so, and we set (k-mer) sequence identity to be >98%.
After all the overlaps, at each position, we take the consensus from those overlaps.

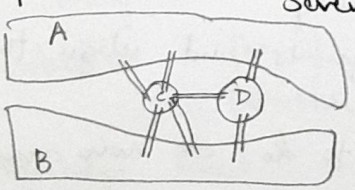
Because of difference in flanking region of the overlaps, repeating sequences will be identified separately & not last

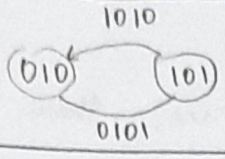


Depth of sequencing: no. of overlaps for a certain position
Greater the depth, more accurate the sequencing.
Based on repeating segments, they will occur n times (approx) more than other segments.

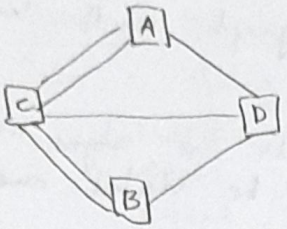
→ De Bruijn graph construction
Suitable for high coverage, short reads
Fast but memory intensive
Sensitive to sequencing errors

Graph theory: Started with Euler solving Konnigsberg Seven bridges problem
Start at one district, traverse each bridge exactly one time and come back to where you started





3-mers
edges - 4



☐ Nodes
— Edges

Euler abstracted the problem to nodes and edges

for you to cross the bridge only once, the degree of every node should be even

Degree - Number of edges leaving a node should be even

So if you don't have an even degree, then the Konnigsberg bridge problem cannot be solved

De Bruijn - Mathematician

What is the smallest circular string of 0s and 1s that will contain all the k mers

If k = 3 : 010 001 100 000 .... 8 segments

01011010 ?

De Bruijn graph — he considered each possible segment as a node and connecting edges were overlaps.

All nodes' degrees are even — he started at one place and went without travelling back

So, we create a de-Bruijn graph of shotgun sequences.

Constructing the Eulerian path

Small change: Here, the fragments will be edges and the prefixes will be nodes.

Because there's a 5' to 3' directionality, we can only go in one direction

The starting and ending points will have odd degree (=1) if its a linear segment. So the graph can have exactly 2 nodes with degree = 1.

Then, we trace a path s.t. each edge is traversed exactly once in the right direction.

NGS 2 - 24/11/2024

30/11

If all the fragments occur only once, then the edges are traversed once

If there are branches in the graph, we'll have 2 or more possible solutions.

Hamiltonian cycle: here, the fragments are ~~edges~~ nodes, so each node should be visited once - harder

Eulerian cycle: prefix/suffix - nodes, fragments are edges. So each edge can be traversed once

This is easier

Overlap = 2

Eg problem: ATG TAA TTA TGA - 8 bp. Find sequence

Here, Hamiltonian is more intuitive. Eulerian is used in algorithms.

TAA
TTA

ATG
TGA
ATGA

TTA
TAA
TTAA

→ Bulges in de Bruijn graphs

The erroneous fragments are present in very few numbers - 1x compared to 8x or 16x of other fragments

Branching vertex is caused by repeat in original sequence or a sequencing error. Errors are detected by a coverage cutoff threshold.

Each of these bulges need to be systematically resolved

→ Using NGS data

Sequencing CSF fluid of an ill patient - 0.016% corresponded to leptospira bacteria and corresponding treatment was done

Burrows-Wheeler Transform

Take a string, end with $ and circularly permute it.

MAT$ ↻
$MAT
T$MA
AT$M

Arrange alphabetically, $ before A →

$MAT
AT$M
MAT$
T$MA

Last column of this arrangement is the BWT of MAT

MAT$ —BWT→ TM$A

GOOGOL$ → LO$OOGG
LO$2(O)2(G)

Similar letters are bunched together. So this transform was initially used to compress data

30/11

Lecture BWT - 29/11

BANANA$ —BWT→ ANNB$AA

1 2 3 4 5 6 7 8 9 10 0
GATACACATA$ —BWT→ A_4 A_1 A_2 A_3 A_0
ATTCCGAA$AA
10 9 3 5 7 1 4 6 0 8 2 → Suffix array

The letters in original string are numbered and kept track of.
Also coded as: $G_0 A_0 T_0 A_1 C_0 A_2 C_1 A_3 T_1 A_4 \$$

Burrows-Wheeler matrix – The arrangement of sequences in alphabetical order

In this matrix, both in first and last column, the letter A appears in the same order: $A_4 A_1 A_2 A_3 A_0$
The same is true for the order of other letters.
Order of occurrence of each character is the same in first and last column. Using this, data can be retrieved

Reversing BWT – method 1 : Last first method.

1st BWTgf BWT is : LO$OOGG

$ ← L
G_i ← O_i
G_j ← $
L ← O_j
O_i ← O_k
O_j ← G_i
O_k ← G_j

From 1st row, we know that L precedes $ because all are circularly permuted.
Based on tracing the path of 1st & Last column, we can recreate the string: GOOGOL$

Constructing a Trie

Trie of words is such that all words begin at the root and end at leaf. If words have common starts, then branching happens when the words diverge

If we have a sequence, GATACACATA$, then suffixes from this can be found in BWO matrix.

Then Suffix Trie is made from BWM

Because constructing this trie is based on Tools (ATGC), retrieving segments from this - identifying where it lies / how many times it occurs - becomes very easy.

Suffix tree - putting / collapsing unbranched letters to form a word

30/11

Lecture - 30/11/2021 - Problem solving

Take home : Building models from data

Representation

Scoring $-\ln\left(\frac{obs}{exp}\right)$ ratio!

Optimisation

Don't blackbox the software - understand the shortcomings

(A)

BIOINFORMATICS - Summary (1st half)

Problem solving : 1. Representation 2. Scoring 3. Sampling

1. Sequence alignments

→ Dynamic programming - Global (Needleman-Wunsch) ; Local (Smith-Waterman)

↳ We find optimal solution by optimising the parts.

	Global	local
Recurrence relation	$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_i) \\ F(i,j-1) - d \\ F(j-1,j) - d \end{cases}$	$\max \begin{cases} \text{—same—} \\ 0 \\ \text{other 3 expressions} \end{cases}$
Traceback	Starts with LAST value	Start with HIGHEST value not penalised
Overhangs	Penalised	
Sensitivity	Detecting sequence with high similarity	Better at detecting remote similarities
Comp. complexity	~ O(mxn) - time & memory	~ O(mxn) - time & mem

→ Substitution scores

Identical > Conservative > Non-conservative

$\left(\frac{obs}{exp}\right)$: > 1 favourable, = 1 random, < 1 unfavourable

Usually $\log\left(\frac{obs}{exp}\right)$

$$s(A,P) = \log\left(\frac{p_{A,P}^{obs}}{p_{A,P}^{exp}}\right)$$

$$p_{A,P}^{obs} = \frac{N_{A,P}}{\sum_i \sum_j N_{i,j}}$$

$$p_{A,P}^{exp} = \frac{\sum_x N_{A,x}}{\sum_i \sum_j N_{i,j}} \cdot \frac{\sum_y N_{P,y}}{\sum_i \sum_j N_{i,j}}$$

Derived from unimpeachable alignments

→ BLOSUM

- Rare but favourable subs are given a v. low score — to fix that
- Multiple seq are aligned in a block & any subset that has ≥ L% seq. identity is grouped
- These subsets are given the weightage of a single sequence So rare subs are better represented. ↓ L% ⇒ better rep

→ Gap penalty : linear - same penalty for all gaps

Affine : Opening penalty > Extension penalty

→ Heuristic alignments (FASTA, BLAST)
- Seq in DB are searched for short, high scoring stretches (hotspots)
- These stretches are sequentially extended until highest scoring segment pair is found (until max scoring ungapped extension)

* FASTA : Hotspots (ktup = 2) → Threshold → Bounded DP (Recursion ?) Fig 2
* BLAST : 3. Identify high scoring pairs → extend diagonally to find max scoring pair
1. Compilation of query words (word score threshold)
2. Database Scan
* BLAST 2.0 : If dist. between 2 HSPs on same diagonal < A, then they get linked : 2-HIT EXTENSION
* Gapped BLAST : gapped extension is started for any HSP score > S_g
Only those cells are considered for which optimal local alignment score doesn't fall $< X_g$

→ Detecting homology — significance of a score
Extreme value distribution of alignment score vs freq.
Karlin-Artschul eqn : $E = Kmn e^{-\lambda s}$
Normalisation score : $s' = \frac{\lambda s - \ln K}{\ln 2}$ $\quad E = mN 2^{-s'}$

→ Multiple Sequence Alignment (MSA)
- Progressive alignment (with phylogenetic trees)
- Clustal W alignments
↳ Distance matrix is created from weighted pairwise comparison. Then phyl. tree is deduced → then progressive alignment using guide tree
- Iterative improvement of MSA.

2. Phylogenetic trees
Leaves - Nodes - Edges : Rooted, Unrooted trees

→ UPGMA : $d_{ij} = \frac{1}{C_i C_j} \sum_{p \in C_i} \sum_{q \in C_j} d_{pq}$
- Start with leaves whose distance $d(i,j)$ is minimal
- Introduce a new node height at $\frac{d(i,j)}{2}$

→ Neighbours joining method — additive but not necessarily same rate
Choose $D_{ij} = d_{ij} - (r_i + r_j)$ minimal $r_i = \frac{1}{|L|-2} \sum_{k \in L} d_{ik}$
$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$ where m : parent node of i, j

→ Parsimony : least no. of mutations ; several seq. positions are considered

(B)

Hidden Markov Model

Σ - alphabet of symbols
Q - set of states capable of emitting Σ
A - set of transition prob. - a_{ij}
E - set of emission prob - $e(\pi_i, z_i)$
$\pi = (\pi_1 \dots \pi_L)$: Sequence of states
$x = (x_1 \dots x_L)$: Sequence of emitted symbols

Properties of HMM : $\{e\}_\pi$, $a(\pi_i \to \pi_j)$, $a(0 \to \pi_i)$

1. Evaluation : How likely is the sequence x, given our HMM?
   Find $P(x|M)$ - Forward algorithm
   $P(x_i \dots x_j)$ - Backward algorithm (prob of a substring of x given M)
   $P(\pi_i = k|x)$ - "posterior" prob that ith state is k, given M
2. Decoding : Finding most likely parse of a sequence
   Maximise $P(x, \pi)$ - Viterbi algorithm
3. Learning : Re-estimate parameters of model based on training data - Update parameters θ to maximise $P(x|\theta)$
   - Correct path is known
   - Correct path is unknown - Viterbi, Baum Welch

	Viterbi	Forward	Backward
Initialisation	$V_0(0) = 1$ $V_k(0) = 0 \quad \forall\ k > 0$	$f_0(0) = 1$ $f_k(0) = 0 \ \forall\ k > 0$	$B_k(N) = 1 \quad \forall\ k$
Iteration	$V_l(i) = e_l(x_i) \max_k V_k(i-1) \cdot a_{kl}$	$f_l(i) = e_l(i) \sum_k f_k(i-1)\, a_{kl}$	$b_l(i) = \sum_k e_k(x_{i+1}) \cdot a_{kl}\, B_k(i+1)$ l , k
Termination	$P(x, \pi^*) = \max_k V_k(N)$	$P(x) = \sum_k f_k(N)$	$P(x) = \sum_k a_{0k}\, e_k(x_1)\, B_k(1)$

Posterior decoding : $P(\pi_i = k|x) = \dfrac{F_k(i)\, B_k(i)}{P(x)}$

$\hat{\pi_i} = \text{argmax}_k\ P(\pi_i = k|x)$ - most likely state at position i of seq x

Learning

→ Given x, $\pi = \pi_1 \dots \pi_N$ is known

$$\Rightarrow a_{kl} = \frac{A_{kl}}{\sum_i A_{ki}} \qquad e_k(b) = \frac{E_k(b)}{\sum_c E_k(c)}$$

Use pseudocounts (r_{kl} & $r_k(b)$) to prevent overfitting

→ When π is unknown

Expectation maximisation — θ that increases $P(x|\theta)$

* Viterbi Training

1. Perform viterbi to find π^* — by taking random values of e/a?
2. Calculate E, A acc to $\pi^* + r$
3. Calculate new parameters a_{kl}, $e_k(b)$ # This finds the path π^* that maximises $P(x$

Repeat until convergence

* Baum-Welch Algorithm

1. Initialisation : best guess for θ
2. F, B algorithm
3. Calculate A_{kl} & $E_k(b)$ given θ —

$$A_{kl} = \frac{\sum_i F_k(i)\, a_{kl}\, e_l(x_{i+1})\, B_l(i+1)}{P(x|\theta)} \qquad E_k(b) = \frac{\sum_{i|x_i=b} F_k(i)\, B_k(i)}{P(x|\theta)}$$

4. Calculate θ_{new} : a_{kl} and $e_k(b)$
5. Calculate new log likelihood $P(x|\theta_{new})$

Reiterate until convergence

$\log_2(x) = y$

x	0.1	0.2	0.25	0.3	0.4	0.5	0.6	0.7
y	-3.322	-2.322	-2	-1.737	-1.322	-1	-0.737	-0.5146

x	0.75	0.8	0.9	1.0	0.33	0.05	0.45
y	-0.415	-0.3219	-0.152	0	-1.6	-4.32	-1.15

B13134 - Quiz 01

VASUDHA KULKARNI
Reg No. 2019105[illegible]

Q.1 Seq A : ATTA
Seq B : TATA

Bounded dynamic programming
Identity matrix : Match - +1
Mismatch - 0
Gaps - -1

(i)

		A	T	T	A
	0	-1	-2	-3	-4
T	-1	0	0	-1	-2
A	-2	0	-1	-1	0
T	-3	-1	1	0	-1
A	-4	-2	0	0	+1

Since gaps & overhangs are penalised, the n+1 row & n+1 column have -1, -2, -3 ... gap penalties.

3 alignments i.e. 3 tracebacks are possible

(a) 1 → 0 → 1 → 0 → -1

Seq A : _ A T T A
Seq B : T A T _ A

(diagonal line)

(b) 1 → 0 → -1 → 0 → 0

Seq A : A T T A
Seq B : T A T A

(c) 1 → 0 → -1 → 0 → -1

Seq A : A T _ T A
Seq B : _ T A T A

B13134 - Quiz 1

VASUDHA KULKARNI
Reg No. 20191057

Q.2. BLAST search

	E value
Seq 1	0
Seq 2	2e-62
Seq 3	0.01

1) Sequence 1 is closest to our query. Its E value = 0, which means that the chances of wrongly identifying seq 1 as a homologue, by fluke or random chance, is 0.

2) Karlin-Altschul equation : $E = KmNe^{-\lambda s}$ where N: size of database

$\Rightarrow E \propto N$

So if the same search was done in a 10x larger database, then, E value of seq 3 would be 0.1

$E' = \frac{N'}{N} \cdot E \qquad E = 0.01 \qquad \frac{N'}{N} = 10$

$\Rightarrow E' = 10 \times 0.01$

$E' = 0.1$

BIOINFORMATICS (B13134) - QUIZ 2

Name : Vasudha Kulkarni
Roll No: 20191057

Question 2 –

Red channel (R) Blue channel (B)

R, B are the possible states

High & low signals are the sequence

From the figure –

Emission probabilities : –

		H	L
$\mathbb{E}_1$	B	0.7	0.3
	R	0.3	0.7

Transition probabilities : a channel conducts for 10 units of time before another channel takes over.

R R B B B B B B B B B R R

Transition probabilities :

$P_{B \to R} = 1/10 = P_{R \to B}$

$P_{B \to B} = 9/10 = P_{R \to R}$

Given seq X : H L H

We need to find the least likely seq of channel i.e. path that gives rise to X.

Initial condition : Equal prob that either channel is firing in the beginning

	H	L	H
Blue	0.35 ←	0.0945 ←	0.0595
Red	0.15 ←	0.0945 ←	0.0255

Least likely final

$(1,2) = 0.3 \times \max \begin{cases} 0.9 \times 0.35 \\ 0.1 \times 0.15 \end{cases}$ should've been min

$(2,2) = 0.7 \times \max \begin{cases} 0.9 \times 0.15 \\ 0.1 \times 0.35 \end{cases}$

$(1,3) = 0.7 \times \max \begin{cases} 0.9 \times 0.0945 \\ 0.1 \times 0.0945 \end{cases}$

$(2,3) = 0.3 \times \max \begin{cases} 0.9 \times 0.0945 \\ 0.1 \times 0.0945 \end{cases}$

Tracing back from least likely final state –

∴ The least likely sequence of channels : RRR

Question 1

Name: Varsdna Kulkarni
Roll no: 20191057

5 features : Flower size (big / small)
Petal colour (red / black)
Centre colour (blue / green / violet)
Pattern (presence / absence of yellow lines)
Odour (present / absent ; if present - green / red)

Flower no.	Size	Petal color	Centre colour	Pattern	Odour
1	B	R	Purple	P	G
2	B	Bla	Blue	A	R
3	S	R	Blue	A	Abs
4	B	Bla	Green	P	G
5	B	R	Purple	A	Abs
6	S	Blue (K)	Green	P	R

Based on these sequences, construct a distance matrix and phylogenetic tree using PHYLIP.
(I couldn't do it in time).

BI3134 - Quiz 03 17/11

Name: Vaindha Kulkarni

Reg No: 20191057

Question 2

Based on the potential energy score of the three templates, we can see that neither of the 3 ~~fit~~ result in ~~the model~~ a suitable model independently.

A plausible 3D model has negative PE at all residues. To obtain such a model for our target protein using the 3 given templates, we can use the energy graph.

The target sequence can be cut up into segments and aligned with the corresponding part of the template which gives a negative PE score.

For ~~our~~ the given graph -

Residue 1 - 4 : Template 1

Residue 5 - 8 : Template 2

Residue 9 - 13 : Template 3

If we align different segments with different templates, we would get the most optimal model.

B13134 - Quiz 3

Name: Vaindhe Kulkarni Reg. No: 20191057

Question 1: Gibbs sampling to find 2 motifs

→ Gibbs sampling is an algorithm in which the steps iterate until they converge to a single solution

→ So, given n sequences $(S_1\ S_2 \cdots S_n)$, we could split each sequence in half to get $(S_{11}\ S_{12}\ S_{21} \cdots S_{n1}\ S_{n2})$ 2n no. of sequences.

→ Then we perform Gibbs sampling on set of first-half sequences & second half sequences.

Input : $S_{11}, S_{21} \cdots S_{n1}$

Output : $(a_1 \cdots a_n)$, M', B'

Init : Choose $(a_1 \cdots a_n)$ arbitrarily in S_{11}

For k = 1, n {

Compute M, B from $a_1 \cdots a_{k-1}, a_{k+1} \cdots a_n$

Use M, B to find a_k in S_{k1}

Compute new M' & B'

Compute score $L(a_1 \cdots a_n)$*

}

* iterate until convergence

Use a similar algorithm for sequences $(S_{12}, S_{22} \cdots S_{n2})$

→ Sometimes motifs might get cut off because of splitting the sequence, so we can use Gibbs sampling on sequences that have arbitrary splitting ratios, say 60:40 instead of 50-50

→ The two motifs may not occur in the same order or same half of the sequence. So we could iterate the algorithm on different combination of sequences, as in, $(S_{11}, S_{21} \cdots S_{k2} \cdots S_{n2})$ and $(S_{12}, S_{22} \cdots S_{k1} \cdots S_{n1})$.

This way we can find 2 motifs in a set of sequences.

B13134 - Bioinformatics Quiz 4 1/12/2021

Name: Varudha Kulkarni
Reg No: 20191057

01. The number of forward and reverse reads should be equal. This means that fragments of low counts are probably erroneous sequences.

Forward reads

TGC	GCB	ZZZ	ATT	TTG	CBB	CAC
746	700	34	691	716	699	41

Erroneous sequence

ATT
TTG
TGC
GCB
CBB

ATTGCBB
1 2 3 4 5 6 7

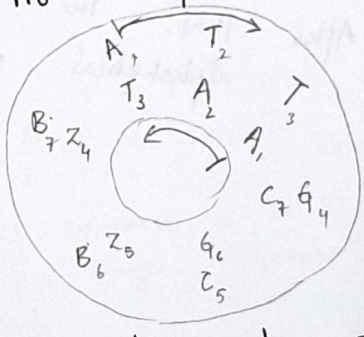
Reverse reads

CCG	TZZ	ATZ	AAT	ZGC	ZZG
17	705	721	688	716	734

Erroneous sequence

AAT
ATZ
TZZ
ZZG
ZGC

AATZZGC
1 2 3 4 5 6 7



The genome was assembled based on the overlaps of the fragments. After sequencing the forward and reverse reads, the only way to make them align properly is by considering them as part of a circular genome.

(2)

B13134 Quiz 4

Name : Vaudha Kulkarni

Reg No: 20191057

2) Would BWT of a string in alphabetical order also be in alphabetical order? Neglecting $.

Checking with a random example : ACGT$

ACGT$		$ACGT	
$ACGT	Arranging	ACGT$	T$ACG
T$ACG	→	CGT$A	
GT$AC	alphabetically	GT$AC	
CGT$A		T$ACG	

ACGT$ —BWT→ T$ACG

Ignoring $ in the BWT, we would get, TAGG, which is not in alphabetical order

This is because, the first letters of the BWT will come from the segment starting with '$ij...k', which means the first letter will always be the last letter of the string.

After that, the following letters will be in alphabetical order.

Questions on Alignments

01) local dynamic programming

(a) ATTA , (b) TATA

Identity matrix
Gap penalty : -1

TA
TA
Score = 2

ATTA
AT-A
Score = 2

Doing it by hand gives 4 solutions
Another would be : ATA TA
ATT TA

02) Human protein : N-AFTER-C

(a) VFITE (b) ETIFV

h ↔ a = 4 h ↔ b = -10

Gap penalty : -8

N-VFITE-C must be the correct sequence because it gives a higher score

PAM250 matrix

AFTER
VFITE
0 9 0 0 -1 = 8

AFTER
ETIFV
0 -3 0 -5 -2 = -10

To make sure that the sequence is a homolog, we compute E value

03) Semi-global dynamic programming → overhangs in shorter sequence are penalised i.e horizontal lines

+1 for identity -1 for mismatch

(a) AGT (b) AAGC

		A	G	T
	0	-2	-4	-6
A	0	1	-1	-3
A	0	+1	0	-2
G	0	-1	2	-1
C	0	-1	-1	+1

No penalisation

_ A G T
A A G C
0 1 1 -1 ⇒ Score = 1

The n+1 column is a 0 column as the longer sequence is not penalised

04)

	A	C	M	S
A	4	0	-1	1
C		9	-1	-1
M			5	-1
S				4

Global dynamic programming
Gap pen = -1
M C A
S C A
Score = 12

Gap pen = 0
M - C A
- S C A
score = 13

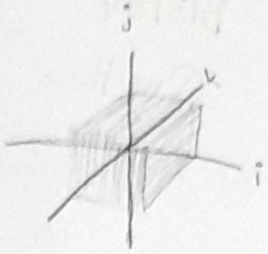
local
C A
C A
Score = 13

(Yes, local is same as global with gap = 0) X

Duh, because d = 0 in global dyn means extra row & column are 0.

5) local dynamic programming in 3D

Substitution matrix would also be 3 dimensional



Base conditions : $F(i,0,0) = F(0,j,0) = F(0,0,k) = 0$

Constant gap penalty = $-D$

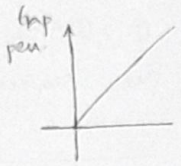
Recursive relationship = Max of

$$\begin{cases} 0 \\ F(i-1,j-1,k-1) + S(i,j,k) \\ F(i-1,j,k) - D + \\ F(i-1,j,k-1) - D + S(i,0,k) \\ F(i,j,k-1) - D \\ F(i,j-1,k) - D \\ F(i,j-1,k-1) - D + S(0,j,k) \\ F(i-1,j-1,k) - D + S(i,j,0) \end{cases}$$

This method of MSA would be very computationally heavy

$X_1 X_2 X_3$ vs $Y_1 Y_2 Y_3$ vs $Z_1 Z_2 Z_3$ — $X_1 X_2 X_3$ vs $Y_1 Y_2 \_$ vs $Z_1 Z_2 \_$?

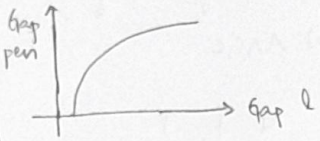
6) E - linear gap penalty



The penalty increases at a const rate

could be C also ✓

F - affine gap penalty



The rate of increase of gap penalty decreases with length after the initial jump (Not sure about initial leeway)

Can't be (B) because the score doesn't go down

07) Distance substitution matrix : identity = 0 mismatch score > 0

Gap penalties would be a positive number because in this scheme, the most optimal alignment score would give you something close to 0.

Recursive relationship — minima of different things

How to choose optimal gap penalties? Try different values of penalties and minimise the score? But for distantly related sequence, seq. identity is not v. useful.

Use : Sometimes, extending a gap is better than some substitutions

Question 4 : Gap pen w/ d=0 ≠ local dyn. prog

Two main differences – In global, we start with last cell whereas in local we start with highest score

Also, in local we stop at 0 but that's not true for glob

Question 5 : If a residue in one sequence is aligned to 2 gaps, then what would be the gap penalty, as compared to single gap?

Work this out!

Question 6 : Graph A – an affine with extension cost = 0

Technically affine should be linear cost for extension.

So, E & C are affine with opening cost = 0.

BIOINFORMATICS

HMM Problem Set

1. Seq : ATG – 2 states : high GC (H) low GC (L)

Figure out if seq x is from species A or Z

HMM values in the problem set.

So, we need to maximise $P(x)$ given θ

$a_{0\to 1} = 0.5 \quad \log_2(0.5) = -1$

$\log_2(P_{H\to H}) = -1.0$

$\log_2(P_{L\to L}) = -0.74$

$$F_k = \log_2(e_k(i)) + \sum_l \left(\log_2 F_l(i-1) + \log_2 a_{lk}\right)$$

a) Animal A.

	A	T	G
$\pi_1 = H$	-3.32	-8.70	-22.3
$\pi_2 = L$	-2.74	-9.54	-22.3

$\log_2 P(x) = -44.6$

b) Animal Z

	A	T	G
H	-4.32	-13.96	-28.49
L	-3.0	-11.06	-28.84

$\log_2 P(x) = -57.33$

$P(x|A) > P(x|Z)$ i.e its more likely that HMM of A gave rise to seq x than that of Z.

WRONG!

Convert to normal prob. and redo.

---

2. Seq A T G G → most likely seq of states

	A	T	G	G
High	-3.34	-6.66	-9.4	-12.14
Med	-3.92	-6.66	-9.98	-13.3
Low	-3.92	-7.56	-10.62	-13.46

Seq of states : H H H H

$\because \sum_i a_{ki} \neq 1 \Rightarrow$ needs to be corrected

4) Two states. π_1 = Home π_2 = Abroad

Sequence : 78 12 14 54

$e_H(50+) = \frac{23}{56} = 0.41$ $\quad e_A(50+) = \frac{10}{49} = 0.20$ $\quad ; \pi(1) = H$

Transition probabilities : $a_{AH} = a_{HA} = \frac{7}{52} = 0.134$ $\quad a_{HH} = a_{AA} = \frac{45}{52} = 0.866$

$\hookrightarrow \frac{1}{8} = 0.125$ $\quad \hookrightarrow \frac{7}{8} = 0.875$

	78	12	14	54
H	0.41 ←	0.21 ←	0.106 ←	0.037
A	0	0.04 ←	0.027 ←	0.006

=> Best q sequence HHH

Prob. that all these matches were away from home -

$P(50+|A) = \frac{10}{33}$ $\quad P(50-|A) = \frac{39}{72}$

$P(\text{all } A) = \frac{10}{33} \times \frac{39}{72} \cdot \frac{39}{72} \times \frac{10}{33} = 0.026 \simeq 2.6\%$

---

5) Emitted characters : A T G C - X

Prob q nucleotides > e q gaps. If A. $e_- = 0$.

Seems like e_- in l_1 < e_- in l_2.

e_G. e_C would be greater in CpG areas

Then to compute e_s and a_s ?

$e = \frac{E}{\Sigma E}$ transmission prob ?

---

25 Emitted characters : $\binom{A}{A}$ $\binom{A}{-}$ $\binom{A}{C}$ $\binom{-}{A}$ ...

in 3 states

NGS Question

3) BWT of AGTGA

- AGTGA$
- $AGTGA
- A$AGTG
- GA$AGT
- TGA$AG
- GTGA$A

BWT → AG$TAG

$AGTGA
A$AGTG
AGTGA$
GA$AGT
GTGA$A
TGA$AG

01) Overlaps = 2

TTA	TGA	ATG	TAC	CCC	CCA	CCG	CCT
12	12	12	12	4	3	1	4

a) Assemble the genome. b) How many chromosomes are there?

TTA + TAC	ATG + TGA	CCC + CCA	CCC + CCT	
TTAC	ATGA	CCCA	CCCT	→ 4 chromosomes
12	12	2	2	→ No. of copies per chromosome

c) CCG which occurs only once is probably an error. If coulda been a CCA.

02) NGS run gave 10 sequences.

What is the ploidy — how many variants?

What is the consensus of the variants?

Are there any sequencing errors?

Which algorithm to use for 1000 copies instead of 10?

Look at recording ~ 49 mins for solution.

Likely triploid organism