Bioinformatics

WS 19/20

What is Bioinformatics?

- in Bioinfo: intend to develop, optimize, parallelize models, algos, production-level SW for analyzing, storing, extracting knowledge from biological raw data

DNA

- 4 basic nucleotides:
  - Adenine
  - Cytosine
  - Guanine
  - Thymine / Uracil (RNA)

Ambiguity Code

- char denotes whether it could be A or C, ...

Code	Represents	Complement
A	Adenine	T
C	Cytosine	G
G	Guanine	C
T	Thymine	A
Y	Pyrimidine (C or T)	R
R	Purine (A or G)	Y
W	weak (A or T)	S
S	strong (C or G)	W
K	keto (T or G)	M
M	amino (C or A)	K
D	A, G, T (not C)	H
V	A, C, G (not T)	B
H	A, C, T (not G)	D
B	C, G, T (not A)	V
X/N	any Base	X/N
-	Gap	-

DNA Sequencing

- reading the nucleotide bases in a DNA molecule

> Sanger sequencing:
- accuracy: 99,9%
- sequence length: 300 - 900 nucleotides
- costs: $2400 per 1 000 000 nucleotides ↯
- few sequences: up to 100 ↯

> next-generation sequencing
- accuracy: 98 - 99,9%
- sequence length: 100 - 400 nucleotides
- costs: $1 - $10 per 1 000 000 nucleotides ☺
- many sequences: 500 - 3 000 000 000 per sequencer run

Model Organism

- extensively studied / sequenced species → understand particular biological phenomena
- expectation: discoveries provide insight into workings of other organisms

Criteria

- easy experimental manipulation
- ease of genetic manipulation
- easy to grow → short life-cycle / generation times
- easy to extract DNA data
- economical importance

Examples

- Escheria coli: gut bacterium, fast growing, inexpensive to cultivate
- Drosophila melanogaster: fruit fly, breeds quickly
- Arabidopsis Thaliana: flower, small genome

Terminology

base pair = pairing of A + T / C + G in double-stranded DNA

ACGGT → 5 nucleotides
↳ 5 base pairs

Kb = kilo-bases
Mb = mega-bases
Gb = Giga-bases

Genome

Genome = full genetic information of an organism

- contains all chromosomes
- comprises coding + non-coding sequence data
  - coding → encodes proteins
  - non-coding → does not encode proteins; function: partially known, regulation of protein processes

Genome Size

not nec. correlated with organism complexity

- homo sapiens: 3,2 Gb
- marbled lungfish: 130 Gb

plants: often large genomes (redundant info → hybridization)

Shotgun Sequencing

Reading a genome:

1) Break up genome randomly into fragments
2) Read fragments
3) Assemble fragments into a genome

Characteristics

Coverage: how many fragments/reads cover one nucleotide on the genome
Fragment length
paired-end vs single-end reads
de novo vs by reference assembly

by reference assembly = read mapping
- closely related species Y → genome available
- map reads of X to genome of Y to assemble

de novo assembly
- no closely related species' genome available
- assemble genome out of read soup

paired-end reads
paired-ends = two ends of the same DNA molecule
↳ sequence one → turn around → sequence the other end
↳ 2 sequences = paired-end reads

Gene = coding part of DNA typically ≈ 1000bp long
- encodes either for RNA or protein

Protein, RNA Sequences
T ~> U
proteins: 20 letter alphabet
3 DNA/RNA chars ≙ 1 protein char
= CODON

- 61 triplets encode 20 proteins
↳ redundancy (usually in third codon pos.)

3 triplets for start, stop (start ≙ ATG ≙ Met / M acid)

Synonymous substitution / mutation
GCC → GCT = Alanine → Alanine
Non-synonymous subst. / mut.
GGT → GTT = Glycine → Valine

DNA ↔ Protein Translation

DNA → protein: non ambigous, redundant

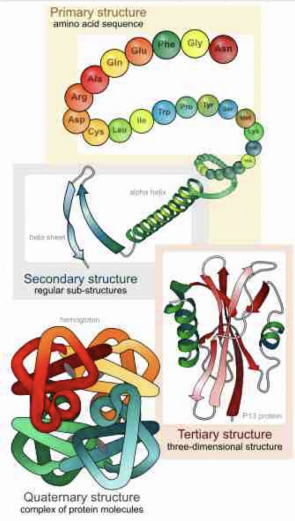
Protein → DNA: ambigous → several DNA triplets = same Amino Acid

Proteins

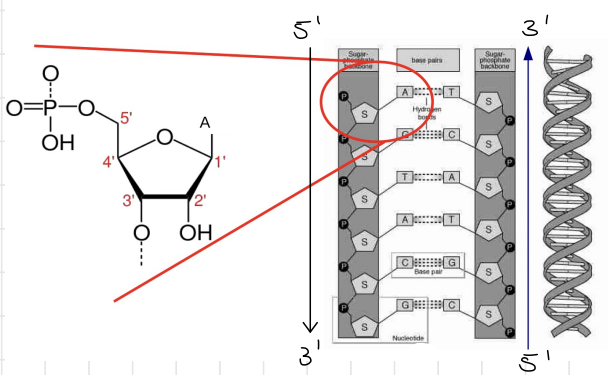
- structural proteins → tissue building blocks
- enzymatic proteins → catalysts of specific biochemical reactions
- homo sapiens ≈ 20 000 proteins
- protein = sequence of amino acid chars
  = protein letters = RESIDUES

Structure

- residues = primary structure
- structure determines function/ effect of a protein



3', 5' end



Convention: DNA sequences in 5' → 3' direction

→ Genes have a direction: depending on which DNA strand encodes

Prokaryotes / Eukaryotes

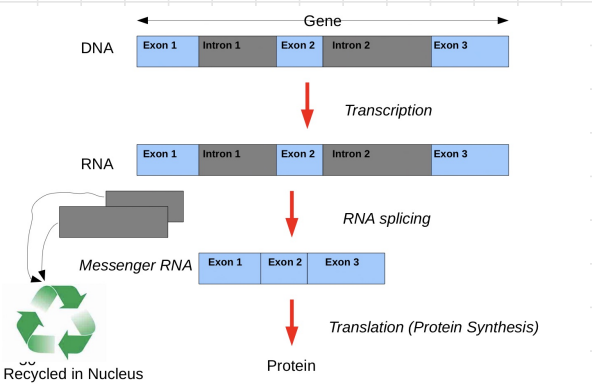
Prokaryota: · no cell nucleus, mostly unicellular organisms
· a gene encodes protein / RNA

Eukaryota: · organism with cell nucleus
· not the entire gene may encode for a protein, just parts of it
Introns: gene not used in protein synthesis
Ekons: parts of the gene used for protein synthesis

RNA

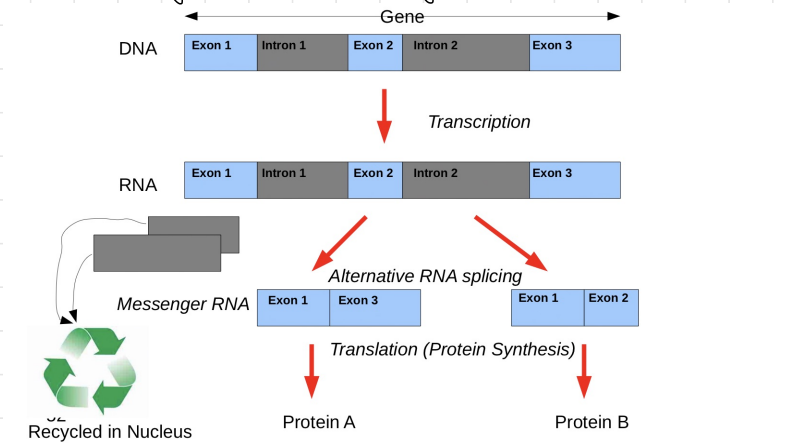
· RNA = copy of coding DNA strand (Gene)
· involved in DNA Transcription to construct
- a protein: DNA → RNA → Protein
  ↳ = translation = coding-RNA
- non-coding RNA: DNA → RNA with other direct cell function

RNA Splicing Eukaryota



Alternative Splicing

Increases coding power of a gene



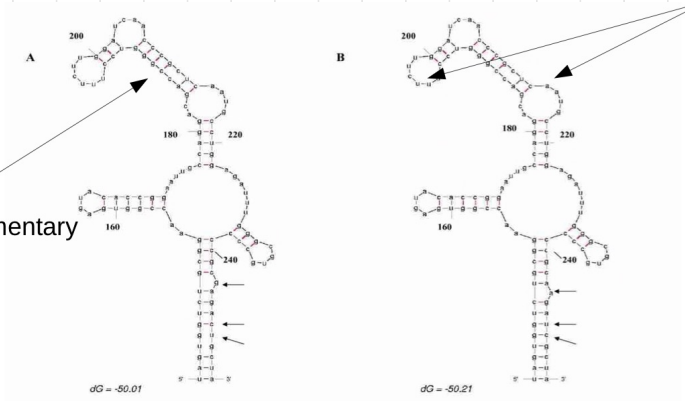
mRNA = messenger RNA
→ transports RNA data to the ribosome for protein synth.

rRNA = ribosomal RNA
→ carries out translation in ribosome via catalysis

tRNA = transfer RNA
→ brings in the amino acids

- only few genes common to all species
  - ↳ e.g. rRNA
  - ↳ can be used to infer evolutionary relationships among all species

Secondary Structure
influences function of the molecule

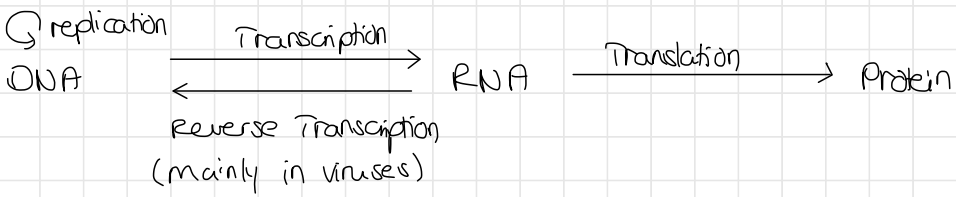


Loops: no Matching bases

Stem:complimentary Bases bind
A ↔ U
C ↔ G

- matching bases in stem cannot mutate independently from each other

Central Dogma of Molecular Biology



Transcriptome
- set of all RNA (mRNA, rRNA, tRNA) in a cell
- reflects activity in a cell
- temporal + spatial component: transcriptome varies
  ↳ different active genes

Meta-Genome
- all genetic material of a community

Chromosomes

- in cell: DNA organized in long molecules = chromosomes
- all chromosomes = genome
- human: 46 (2 x 23 chr.) prok: 1 chr.
  mouse: 40
  donkey: 62

Eukaryotic Chromosomes

- paired chr. = homologous ~> parental / maternal
- some genes in hom. chr. exactly identical
  some not -> different genotypes
- genes that appear in different forms = Alleles
- cells containing pairs of chr. = diploid
  one of each = haploid (sexual reprod.)

Species

by reproduction:
- 2 species that can reproduce
- bacteria / viruses?

evolutionary species concept:
- via ancestral decent in evolutionary tree

general lineage concept:
- an independently evolving lineage

phylogenetic species concept
by sequence similarity + statistical methods

Taxonomy

- group biological organisms (species) into groups with similar charact.
- define char. of groups at different hierarchy levels
- taxonomic ranks
  - domain -> 3 domains of life
  - kingdom
  - phylum
  - class
  - order · family · genus · species

Phylogeny

- unrooted binary tree
- leafs = currently living organisms represented by their DNA/protein sequences
- inner nodes = hypothetical common ancestors
- outgroup = one/more closely related but different species

Taxon

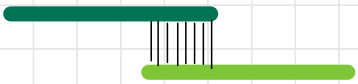
- used to denote clades/subtrees in taxon. / phylog.
- a group of species that form a biological unit
- in phylogenetics: refer to single leaf as taxon

Pair-wise sequence alignment

DNA, protein sequences = strings

global alignment = align full strings

local alignment = align similar substrings



Approach:

- online algorithms: sequences cannot be preprocessed
  → no index build
- dynamic programming: break down problem into smaller subprobl.
  → store results to avoid redundant comp.

Substring

x = substring of y if u, v exist with $y = uxv$

String Distance

- positivity $d(x,y) \geq 0$
- separation $d(x,y) = 0 \iff x = y$
- symmetry $d(x,y) = d(y,x)$
- triangle inequality $d(x,y) \leq d(x,z) + d(z,y) \quad \forall z$

String Operations

- Substitution of a letter
- Deletion of a letter
- Insertion of a letter

Hamming Distance

- for strings with $|x| = |y|$
- counts # differing chars

$\delta_h(\text{ping}, \text{pong}) = 1$
$\delta_h(\text{lala}, \text{alall}) = 4$ ÷

⊕ easy to compute

Edit Distance

$\delta_e(x, y)$ = minimum # of operations to transform x into y

⊕ more sensful in sequence comparison

String Alignment

- insert gaps into both strings to achieve same length

$$\begin{pmatrix} ACG--A \\ ATGCTA \end{pmatrix}$$

Cost: hamming dist. btw new sequences
min. cost: edit dist. btw origin sequences

Compute Edit Distance

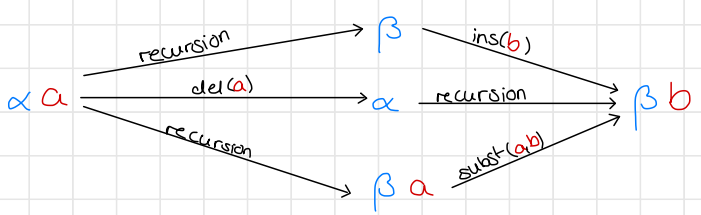
Let $X = \alpha a$ $Y = \beta b$

Compute $\delta(\alpha a, \beta b)$ assuming we know

- $\delta(\alpha a, \beta)$
- $\delta(\alpha, \beta b)$
- $\delta(\alpha, \beta)$

→ Recursion until $X = \varepsilon$, $Y = \varepsilon$

- 3 possible sequences of operations:



Costs:

$\delta_e(\alpha a, \beta) + 1$
$\delta_e(\alpha, \beta b) + 1$
$\delta_e(\alpha, \beta)$ if $a == b$
$\delta_e(\alpha, \beta) + 1$ else

4) Take shortest path:

$$\delta(x,y) = \min \begin{cases} \delta e(\alpha a, \beta) + 1, \\ \delta e(\alpha, \beta b) + 1, \\ \delta e(\alpha, \beta) + 1 * (a == b) \end{cases}$$

Needleman-Wunsch algorithm

- computes best global alignment
- complexity $O(|X| \cdot |Y|)$ → filling is most expensive

Algo: 1) Setup table with x, y zB x = monkey
y = money

2) Init first row, first column

3) Fill each element with min of three prev values
store best path

4) backtrace to get chain of operations

		M	O	N	K	E	Y
	0	1	2	3	4	5	6
M	1	0	1	2	3	4	5
O	2	1	0	1	2	3	4
N	3	2	1	0	1	2	3
E	4	3	2	1	1	1	2
Y	5	4	3	2	2	2	1

Adding weights

penalize less likely operations with weights

e.g. subst. happen 5x more often than ins/del

↳ $\delta e(x,y) = \min(\ \delta e(\alpha a, \beta) + 5,\ \delta e(\alpha, \beta b) + 5,\ \delta e(\alpha, \beta) + 1 * (a == b)\ )$

- similarity: pos./neg. weights to favor similarities
  - ins/del/non-identity subst. neg. weight
  - identity subst. pos. weight

! min distance ↔ max similarity

Local Alignment

- more useful for dissimilar sequences with possible similiar regions

Approaches:

a) Find substrings with min. distance
   ↳ ⊖ short strings selected

b) Find substrings with max. similarity

Smith-Waterman algorithm

≈ Needleman-Wunsch

1) Init first row, column with 0
2) Fill table with same recursion
3) Find largest value in table → best similarity
4) Traceback until a 0 reached

Substitution Matrices

- DNA / proteins alter over time (DNA mutations)
- subst. matrices describe rate at which a char in a sequence changes to other char states over time

- Transitions $A \leftrightarrow G$ / $C \leftrightarrow T$ more often than transversions

↳ take into account that e.g. $subst(A,G) < subst(A,C)$

⇒ $\delta(x,y) = \min($
$\delta e(\alpha a, \beta) + ins(b),$
$\delta e(\alpha, \beta b) + del(a),$
$\delta e(\alpha, \beta) + subst(a,b))$

BLOSUM matrices

Blocks of Amino Acid Substitution Matrix

- used for sequence alignment of protein sequences

- high numbers (eg BLOSUM 80) for closely related sequences

Log-odd scores

$$S_{i,j} = \frac{1}{\lambda} \log\left(\frac{p_{i,j}}{\pi_i \times \pi_j}\right)$$

$p_{i,j}$: observed frequency of subst.

$\pi_i \times \pi_j$: expected frequency of subst.

π_i = stationary frequency of state i

$S_{i,j} > 0$ → very likely
< 0 → unlikely

Hamming Distance with mismatches

X = long string, y = short string
↳ find all substr. x of X with $|x| = |y|$ and $Sh(x,y) \leq k$

Recursion / DP formula

1) init first row with 0
2) init first col with $k+1$
3) Fill table: $T(i,j) = T(i-1, j-1) + subs(i,j)$
   (no del / ins with hamming dist.!)
4) Last line: keep all alignments with score $\leq k$

BLAST Basic Local Alignment Search Tool

· fast heuristic to find similar sequences

Assumption: sequence similarity ⇔ evolutionary / functionally relatedness

Naïve Approach: Smith-Waterman algo
→ build pair-wise alignment of query sequence Q with every sequence in DB $S_1, \ldots, S_d$
↳ report best matches
⊖ DP matrix for every sequence
$O(m \times n)$ compl. for one SW-algo

Algorithm

1) Seeding: find seeds = common subwords btw query and db sequences

2) Extension: starting from seed: extend alignment in both directions
→ high-scoring segment pairs (HSP)

3) Evaluation: assess statistical significance of each HSP

Seeding

a) Build L_1: all subwords (factors) of length ω of query sequence

Query ACTT GCTA , $\omega = 5$

ACTT G

CTTGC

TTGCT

TGCTA

b) $\forall w_i \in L_1$: build neighborhood N_i with similar subwords
↳ add only subwords with similarity score $> T$

Similarity: via substitution matrix
proteins: BLOSUM
DNA: +2 for match, -3 for mismatch (+5/-4)

w_i = TTGCT
↳ TTGAT
+2+2+2-3+2 = 5 > 4 ✓

TTGCT
TGCCT
+2-3-3+2+2 = 0 < 4 X

c) · build $L_2 = \cup N_i$ combination of all neighborhoods
· build final list $L = L_1 \cup L_2$
· Scan DB for exact matches of subwords in L

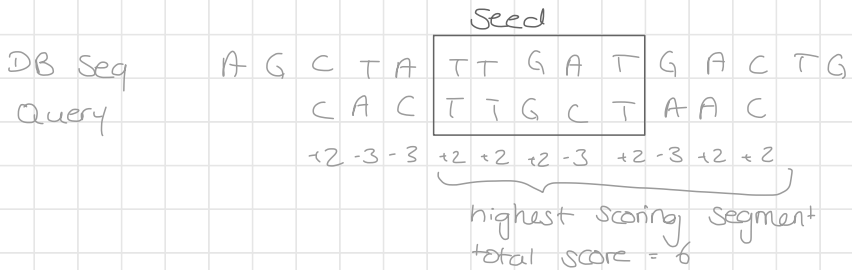
DB sequence

AGCTA TTGAT GACTG
seed

Extension

- try to extend alignment to the left + right from the seed
- stop if current total score drops by more than X compared to current max
- trim alignment back to max score

X = 3



Evaluation

- probability p of observing score $S \geq x$ by chance:

$$p(S \geq x) = 1 - \exp\left(e^{-\lambda(x-\mu)}\right)$$

- alignment scores follow Gumbel extrem value distribution

↳ E-value: expectation value (takes DB size d into account)

$$E \approx 1 - e^{-p(S>x)d}$$

E-value = expected # of times unrelated DB sequence would obtain score $S \geq x$ by chance

⇒ small E-value for biologically related sequences

Genome Assembly

- reconstructing original DNA from its factors

De novo

1) assemble reads into contigs using overlaps between them

2) use mate pairs to combine contigs into scaffolds
(contig order + orientation out of paired-end reads)

3) manual joining of scaffolds

metrics of assembly quality: length of contigs, scaffolds

1) Overlappings

a) Overlap Graphs

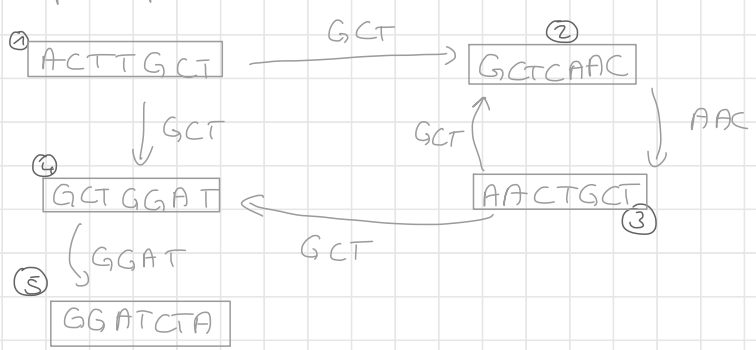
- each read = node
- compute pairwise alignments btw reads, overlap → edge

-> walk along Hamiltonian path (every node exactly once)
↳ reconstruct original sequence

start: · circular genome: any node
· linear genome: node with no inbounding edges

Genome ACTTGCTCAACTGCTGGATCTA

Overlap Graph:



⊖
- no efficient algo to find Hamiltonian path
- pair-wise alignments $O(n^2)$ (n = # reads)
- only good for small # of reads with significant overlap

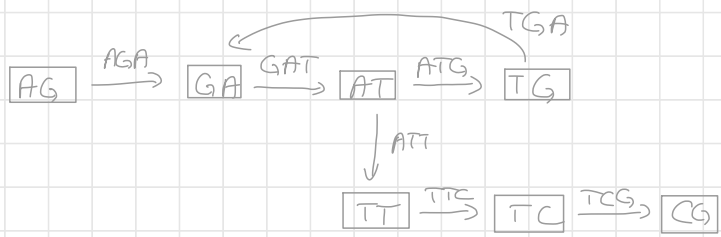
b) De Bruijn graphs

- decompose reads into k-mers (us. $k = 20-50$)
  $k <$ length of shortest read

- unique k-mer = edge
- nodes = $(k-1)$-mers → pre/suffix of connecting k-mer edge

↳ find original sequence by finding Eulerian path (each edge ex. once)

AGATGATTCG
k=3 AGA
GAT
⋮



⊕
- compact repr. of repeats
- building graph: time $O(N)$ (N = total length of all reads)
  space $O(\min(G, N))$ (G = genome size)
- efficient algos to find Eul. path exist

⊖
- information loss → k-mer extraction
- if reads error-free: Eul. path always exists
- repeats: multiple alternative paths
- disjoint graphs if lack of coverage in some regions

Error correction

- reads contain sequencing errors
- bubbles = errors in middle of the read
- tips = errors at ends of the read
- detection/fixing: use coverage info

By reference

Sliding window approach

- slide each read along reference genome, mark matching pos.
- if gaps allowed: DP algo (eg Smith-Waterman)

⊖ huge complexity

Hashing

- build genome index → hash table stores positions of all k-mers
  $k \ll$ read length

- for each read:
  - select proxy k-mer (leftmost / middle part)
  - hash table lookup: find all positions of this k-mer in genome → seeds
  - for each seed: try to extend alignment (z. SW. / Wat. M. algo)

Variations

- inexact matches with spaced seeds
  ↳ binary mask defines pos where mismatches allowed 111 00 1

- multiple k-mers per read
  ↳ require ≥ n seed matches for mapping location to be considered

- inverted approach: build hash table from reads, search for k-mers present in the reference

Burrows-Wheeler Transformation BWT

1) Write down all cyclic rotations of source string S
2) sort rows lexicographically
3) store last column → BWT(S)

Rows = sorted list of suffixes
-> efficient substring search

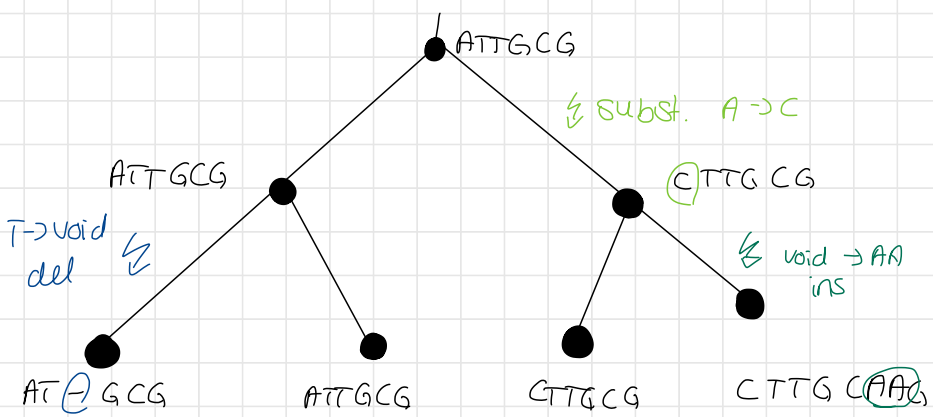
Pattern search:

1st column obtained by sorting last one
2nd analogous
~> proceed and check rows for suffix pattern

Multiple Sequence Alignment

- asses origin sequence for multiple homologous sequences

Indel = insertion / Deletion
↳ indel lengths 1-3 frequent (3 = codon length!)



Aligned Data

AT-GC--G
ATTGC--G
CTTGC--G
CTTGCAAG

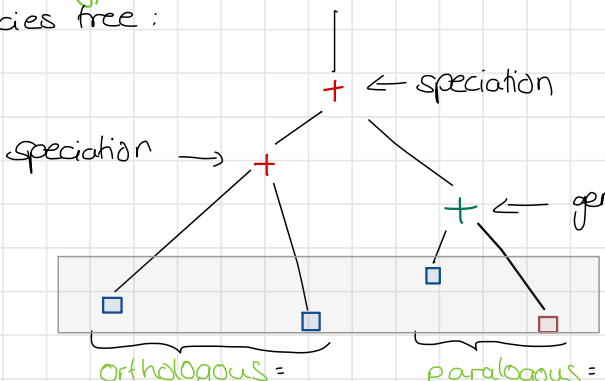
goal: infer homology
-> which chars share common evolutionary history

Importance of MSA

- input for phylogenetic reconstruction
- discover important parts of protein family (group of evolutionary related genes/proteins in different species with similar function/structure)

Terminology

species tree:



descended from same ancestral sequence separated by speciation event

mutation of duplicated genes during separate speciation events

- orthologous sequences = sequences in different species that evolved from same ancestral gene
- homologous characters = chars that share common evolutionary history
  - ! sequence similarity ≠ homology
    - ↳ convergent evolution possible (independently evolved twice)
    - short sequences: by chance

Algorithm Outline

given n orthologous sequences $S_1, \dots, S_n$ of different lengths

Gap Insertion

insert '-' such that:
- all sequences have same length
- some criterion is optimized
- homologous chars s_i, s_j aligned to each other (same alignment col)
- not allowed: columns of entirely gaps

Alignment Criteria

how to define alignment quality

a) Sum of pair measure

- score each site (column), add up scores overall sites
  -> penalize mismatches, gaps
  favor matches
- per-site score: sum of all pairwise scores btw chars of a site
- example:

S1	A	A	G	A	A	–	A
S2	A	T	–	A	A	T	G
S3	C	T	G	–	G	–	G
	1	1	1	1	1	1	1
p() = edit-dist.	1	1	1	1	1	1	1

$p(-,-) = 0$

$\Sigma = 14$

(–)
- optimal MSA-computing NP-complete
- complexity: time + space $O(m^n)$ m = length of seq.
- not guaranteed that SP is biologically most plausible criterion
- dependant on arbitrary p()

b) Star Alignment Approximation

- pick S_c = center sequence
  - ↳ compute all $\frac{n^2}{2} - n$ optimal pair-wise alignments
    - → select seq. with largest similarity to all others
- align remaining seq to S_c (pairwise-seq. alignment algo)
- "once a gap, always a gap"
  - ↳ gaps inserted to S_c cannot be removed

- produces MSA with SP score $< 2 \cdot$ optimum

Example:

S1: A T T G C C A T T ← S_c
S2: A T G G C C A T T
S3: A T C C A A T T T T
S4: A T C T T C T T
S5: A C T G A C C

↳
S1: A T T G C C A T T - -
S2: A T G G C C A T T - -

S1: A T T G C C A T T - -
S3: A T C - C A A T T T T

S1: A T T G C C A T T - -
S4: A T C T T C - T T - -

S1: A T T G C C A T T - -
S5: A C T G A C C - - - -

← Gap inserted

'once a gap, always a gap'

↳ Alignment:
S1 A T T G C C A T T - -
S2 A T G G C C A T T - -
S3 A T C - C A A T T T T
S4 A T C T T C - T T - -
S5 A C T G A C C - - - -

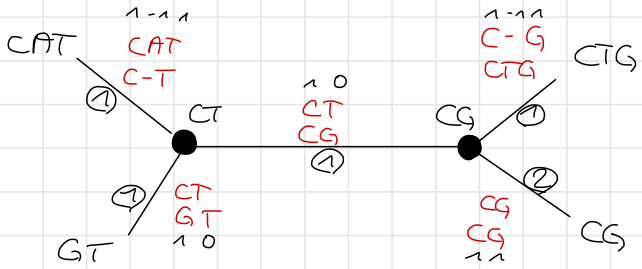
c) Tree Alignment if evolutionary tree for the seq. available

· find assignment of seq. to the inner nodes to maximize sum over similarity scores on all branches

Example: $p(a,b) := 1$ if $a == b$

$= 0$ else

$p(a,-) = -1$



⇒ overall score = 6

! NP-hard → ancestral states not given

⊖ · hen&egg problem: need MSA to build tree
need tree to compute MSA

↳ if alignment wrong, tree might be wrong
if tree wrong, MSA might be wrong

Idea: simultaneous inference of tree & MSA
↳ solving 2 NP-hard / NP-complete problems at once???

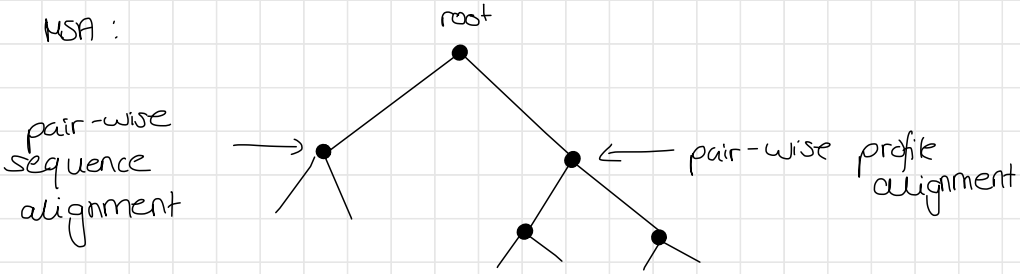
d) Progressive MSA

= Guide tree approach

1) Build guide tree
· compute all $\frac{n^2}{2} - n$ pair-wise distances btw. the n sequences
→ hierarchical clustering (eg NJ algo)

2) calculate pair-wise sequence-sequence
seq -profile
profile -profile alignment bottom-up

MSA:



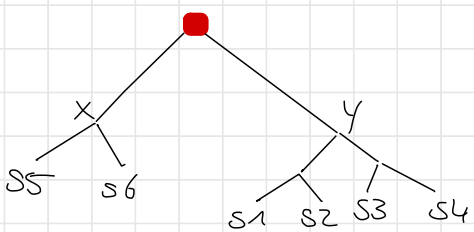
- start with the 2 closest seq. in guide tree
- calc. pair-w. alignment (Needl. Wunsch)
- calc. profile ("mischsequenz")
- set parent = profile

Loop: take next 2 seq. (profiles that are closest
calc. profile

End: one profile left → MSA

Profile Alignment:
- average over all possibilities

example:



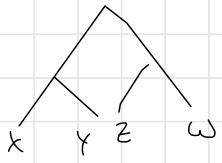
	0	1	2	3	4	5	6	7	8	9
S1	P	E	E	K	S	A	V	T	A	L
S2	G	E	E	K	A	A	V	L	A	L
S3	P	A	D	K	T	N	V	K	A	A
S4	A	A	D	K	T	N	V	K	A	A

	0	1	2	3	4	5	6	7	8	9
S5	E	G	E	W	G	L	V	L	H	V
S6	A	A	E	K	T	K	I	R	S	A

→ compute score btw. pos 6 of x and pos 7 of y
→ weighted average over all 8 possibilities (2*4)

$$\frac{1}{8} * [p(T,V) + p(T,I) + p(L,V) + p(L,I) + p(K,V) + p(K,I) + p(K,V) + p(K,I)]$$

⊖ Initial pair-wise alignment can't be changed / corrected



x : GAAGTT
y : GAC-TT
z : GAACTG
w : GTACTG

flipping C- to -C would fit better but cannot be done

} y: GA-CTT

e) Iterative progressive MSA

- execute progr. MSA multiple times → refine alignment
- eg. MUSCLE Refinement

f) Motif-based approaches

- find small motif (substring) common to all sequences
  = anchor / block / region / ...
- if motif found: align seq. to motif
- align regions around motifs using e.g. progressive alignment

Benchmarking MSAs

- using MSA benchmarks
- using simulation

Simulation

- infer true MSA through simulation from tree
  → disaligns sequences
  → align using MSA algo
  → compare result with true MSA:
    count correct sites
    compare SP scores
  → infer tree
  → compare trees

Phylogenetics

Taxonomy = group of species with similar characteristics
↳ define charact. at different hierarchy levels
-> taxonomic ranks
domain (3 domains of life Bacteria, Archaea, Eukaryota)
kingdom
phylum
class
order
family
genus
species

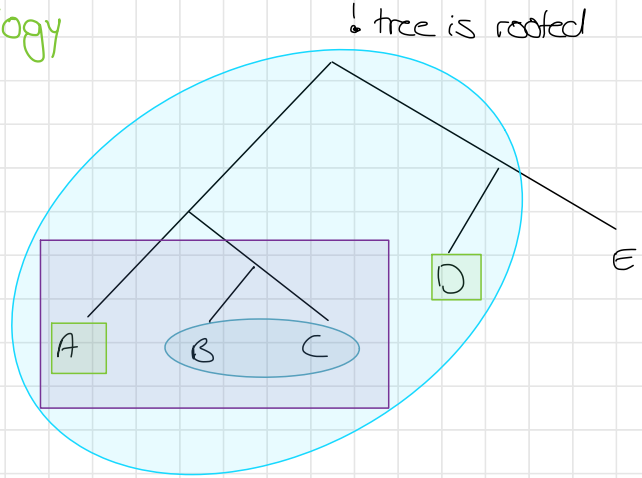
Phylogeny

- unrooted, binary tree
- leafs = extant (currently living) organisms represented by their DNA/protein sequences
- inner nodes = hypothetical common ancestors
- outgroup = closely related, but different species (?1)
  ↳ allows rooting of tree

Taxon

- denote clades (subtrees in phylog. /taxonom.
- = group of ≥1 species forming a biological unit
- defined by taxonomists -> controversial debates!
- phylogen.: single leaf = taxon (pl. taxa)

Terminology

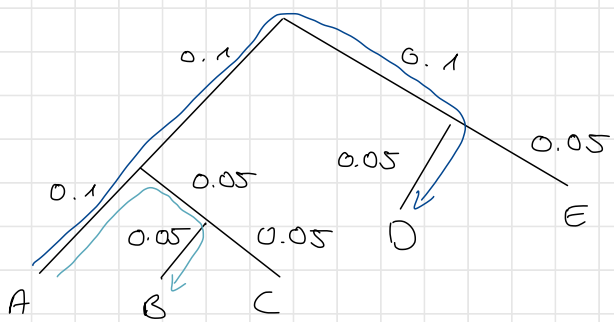


B, C = monophyletic group = sister species -> sisters to each other

(A, B, C) = monophyletic group = sister species -> sister to (D, E)

(A, B, C, D) = paraphyletic g: includes ancestor but not all decendants

(A, D) = polyphyletic group = most recent common ancestor excluded



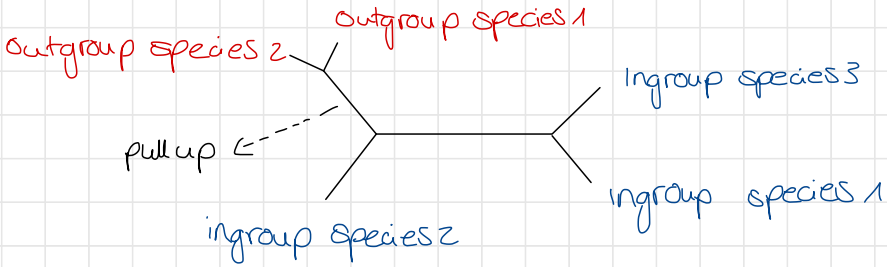
patristic distance between 2 taxa: sum over branch length along path in tree

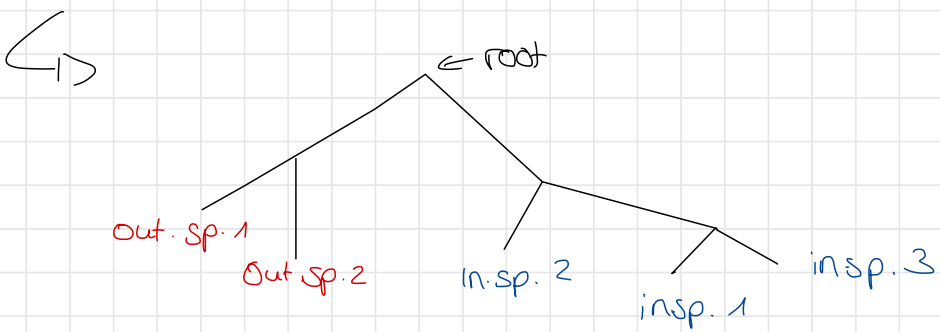
e.g. A <-> B = 0.1 + 0.05 + 0.05 = 0.2

A <-> D = 0.1 + 0.1 + 0.1 + 0.05 = 0.35

Tree Rooting

- root tree using outgroups
  - ↳ root tree on connecting edge between outgroup and ingroup





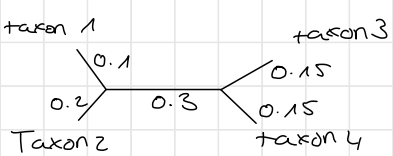
→ choice of outgroup influences tree topology!

Tree Inference

1) Obtain homologous sequences from the same gene of different species
2) MSA program calculates MSA → alignment-free: no MSA calc.
   - ↳ less accurate
3) Tree inference program calculates tree

Newick Tree Format

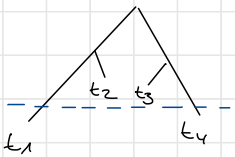
(Taxon 1: 0.1, Taxon 2: 0.2,
(Taxon 3: 0.15, Taxon 4: 0.15): 0.3);



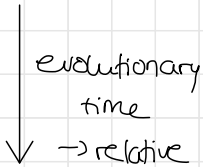
Problem: meta-data ↔ branch length association → different interpretations possible

Tree Shapes

ultrametric tree



non-ultrametric tree



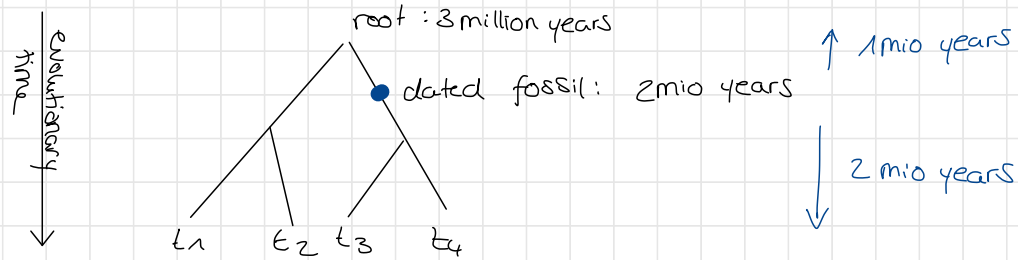
Dating Trees

relative evolutionary time $\xrightarrow{?}$ real time

Requirement: rooted + ultrametric tree

→ rooting: outgroups

ultrametricity: programs for divergence time estimation



→ how to place fossil? no DNA data available

↳ ad hoc empirical knowledge / computationally using morphological data (Struktur + Form)

Tree counts (binary trees)

Rooted trees: tree with n taxa = # of unrooted trees for $n+1$ taxa

→ $n-1$ inner nodes

$2n-2$ branches

$$(2n-3)!! = \frac{(2n-3)!}{2^{n-2}(n-2)!} \text{ trees} \quad (n \geq 2)$$

e.g. $n = 4$

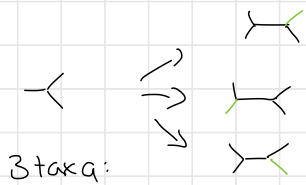
$$\rightarrow (2n-3)!! = \frac{(8-3)!}{2^2(2)!} = \frac{5!}{4 \cdot 2!} = \frac{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}{4 \cdot 2 \cdot 1} = 5 \cdot 3 = 15$$

Unrooted trees: tree with n taxa:

→ $n-2$ inner nodes

$2n-3$ branches

$$(2n-5)!! = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$



3 taxa:
1 tree

4 taxa:
3 trees

u = # trees of size $4-1=1$

v = # branches in tree of size $4-1=3$

↳ # trees with 4 taxa $= u \cdot v = 3$

5 taxa:

$u = 3$

$v = 5$

$\Rightarrow u \cdot v = 15$

6 taxa

$u = 15$

$v = 7$

$\Rightarrow u \cdot v = 105$

Use of phylogenetic trees

- identifying unknow species
- divergence time estimates
- diversification rates
- viral outbrakes
- forensics
- ...

Building Phylogenetic Trees

distance-based methods

- compute pair-wise distance matrix using MSA
- build tree using these distances
- heuristics (≈ hierarch. clustering methods)
  - e.g. Neighbor Joining / UPGMA
- least-squares method: explicit optimality criterion

distance matrix: memory problem
↳ out-of-core / external memory algos for data that is too large for main memory
→ explicit data transfer RAM ↔ disk in application to circumvent OS paging (use of knowledge → faster)

⊕ faster ⊖ less accurate

character-based methods

- optimality criteria f() operate directly on MSA
  e.g. parsimony
  max likelihood } ? memory intensive!
  bayesian inference
- calculate score based on current tree topology + MSA
  ↳ score = how well does MSA data fit tree?

⊕ more accurate ⊖ slow

NP-Problem

- least squares, parsimony, likelihood, bayesian all NP-hard
  ↳ super-exponential increase in possible trees

Neighbor Joining Algorithm

distance-based

- Start with the distance matrix $D_{i,j}$ $(i,j = 1, \dots, n)$
- For each tip: compute $u_i = \frac{\sum_j D_{ij}}{n-2}$
- Find pair of tips (i,j) with min. $D_{ij} - u_i - u_j$
  ↳ connect (i,j) to new ancestral node X
- compute new branch lengths from X to i and j:
  $b_i = 0.5 D_{ij} + 0.5 (u_i - u_j)$
  $b_j = 0.5 D_{ij} + 0.5 (u_j - u_i)$
- Update distance matrix: compute dist. btw X and remaining tips

$$D_{(i,j),k} = \frac{(D_{ik} + D_{jk} - D_{ij})}{2}$$

$(i,j \hat{=} X \text{ children})$

- Replace tips i,j with X → treat X as tip now
- Loop until 2 nodes remain
  ↳ connect them

Complexity:
Space $O(n^2)$
Time $O(n^3)$

UPGMA

distance-based

Unweighted Pair Group Method with Arithmetic mean
→ produces ultrametric, rooted trees
can be used if known that we have ultrametric tree → usually not the case

- start with distance matrix ①
- find min value $D_{i,j}$
- merge (i,j) → new group has $n(i,j) = n_i + n_j$ members
- assign branch lengths $D_{i,j}/2$ to $i \to (i,j)$, $j \to (i,j)$
- update distances btw (i,j) and all $k \neq i,j$:

$$D_{j,k} = \frac{n_i}{n_i + n_j} \cdot D_{i,k} + \frac{n_j}{n_i + n_j} D_{j,k}$$

Complexity:

- naive impl. : $O(n^3)$
  - ↳ maintain list of per-col / per-row minima
    - ↳ update-list $O(n)$
    - look for min $O(n)$
    - } $O(n^2)$
- time $O(n^2)$

non-ultrametric trees:
can yield misleading results
↳ most trees non-ultrametric! (unequal evolut. rates among all lineages)

Least squares optimization

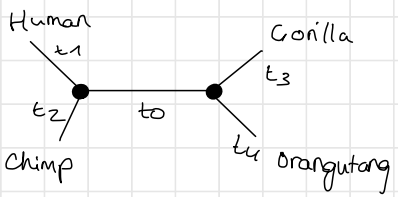
distance-based

- given fixed, fully binary tree T with n taxa and pair-wise distance matrix D

- Find branch lengths $t_1, \dots, t_{2n-3}$ such that:
  sum of squared differences between pair-wise patristic dist. d_{ij} and plain pair-wise dist. D_{ij} is minimized
  ⇝ minimize $Q = \sum_{i<j} (D_{ij} - d_{ij})^2$

  complexity: $O(n^3)$ if derivative + system of linear equations
  $O(n^2) \dots O(n)$ if tree-like structure taken into account

- Find tree topology for that minim. Q
  -> NP-hard!
  (! single tree in $O(n) - O(n^3)$!)



patristic distances:
$d[H][C] = t_1 + t_2$
$d[H][G] = t_1 + t_0 + t_3$
$d[H][O] = t_1 + t_0 + t_4$
⋮

Minimum Evolution Method

≈ Least squares
- explicit criterion
- min $Q = t_0 + t_1 + t_2 + \dots$

Parsimony

character-based

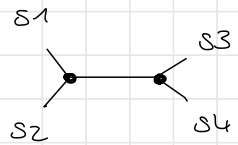
- given a MSA → find tree that explains data with least amount of mutations

Count Parsimony

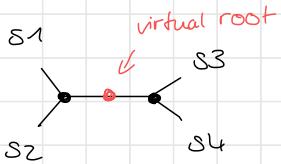
MSA

S1	A	A	G	G
S2	A	A	A	-
S3	A	G	A	G
S4	T	T	A	T

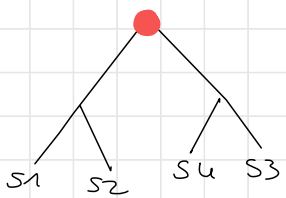
→ indel → can be A/C/T/G



→ Find sequences of inner nodes to min. # of mutations



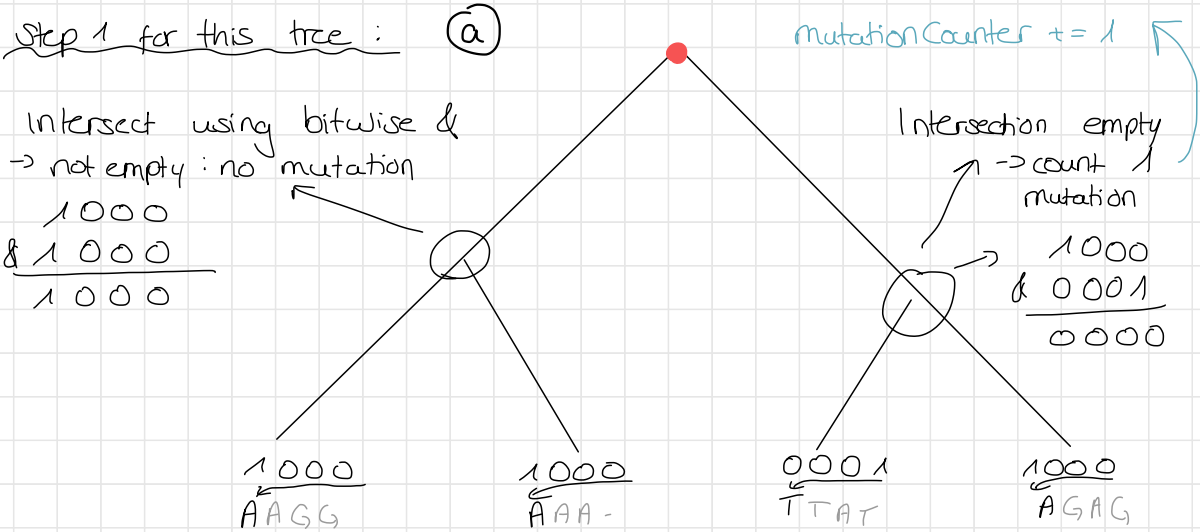
---->



! score independent of root position

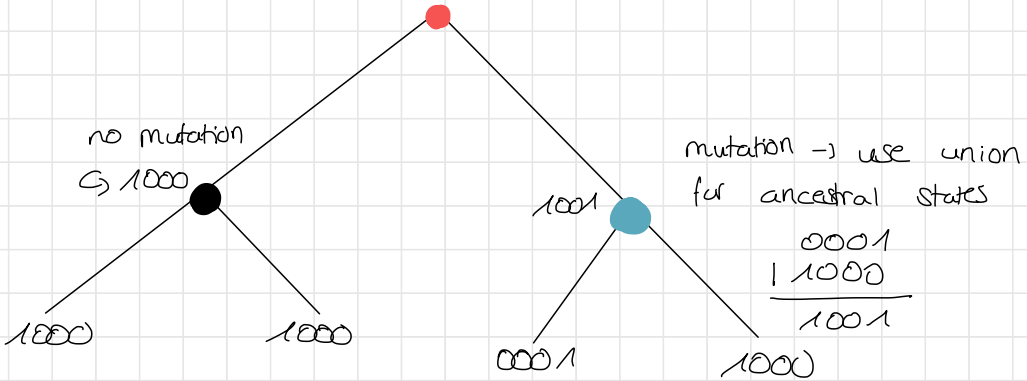
→ Compute scores by comparing sites (columns) individually (assumption: independent evolution of sites)

Step 1 for this tree: (a)

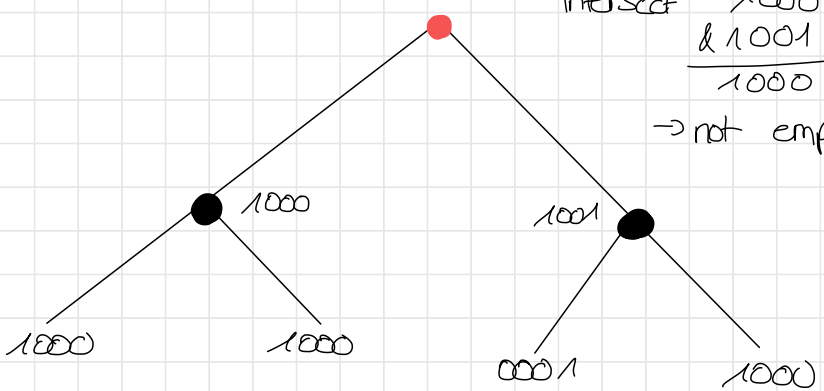


(Encode A = 1000 C = 0100 G = 0010 T = 0001)

(b)



(c)



=> overall score for 1st site = 1 = mutation Counter

Repeat for all sites, add mutationCounter over all sites

Complexity

Time: to score one tree given MSA with n taxa, m sites
- -> (n-2) * m calculations (n-2 = #inner nodes)
- -> O(nm), small constant in O(1)

Space : · alignment : $n * m * 4$ bits
· ancestral nodes $(n-2) * m * 4$ bits
· score counter : $(n-2) * 32$ bits
-> $O(nm)$, small constant in $O()$

Tree Search

how to obtain initial starting tree?
↳ comprehensive tree -> change tree to improve score
· NJ / UPGMA
· random tree
· stepwise addition algo

Random Tree

· insert sequences at random positions

Randomized Stepwise addition algo

· build tree using first 3 randomly picked sequences
· insert following sequences according to best parsimony insertion score

Alter Topology to improve score

alteration mechanisms:
- Hill Climbing
- Simulated annealing
- ...

Basic moves:
· NNI nearest neighbor interchange
· SPR subtree pruning and re-grafting
· TBR tree bisection and reconnection

NNI
- exchange connectivity of 4 subtrees within main tree

SPR
- select + remove subtree from main tree
- reinsert it elsewhere on main tree as new node

TBR
- detach subtree from main tree at inner node
- attempt all possible connections btw. the 2 trees

Long branch attraction
- error where distantly related lineages are incorrectly infered to be closely related

→ happens if both lineages underwent large amount of change

Felsenstein zone = settings where parsimony recovers wrong tree

↳ Solution: max likelihood

Maximum Likelihood

character-based

- calculate likelihood of tree given MSA $L(T|D) = P(D|T)$ — T = tree, D = MSA data

assuming sites evolved independently

$L(T|D) = \prod_{i=1} P(S_i|T)$ = probability that site $1, \dots, i$ evolved to tree

↳ use log (small values cause underflow!)

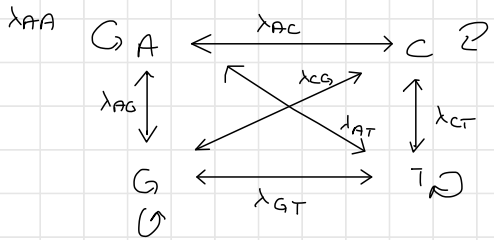
$$\log(L(T|D)) = \sum_{i=1}^{n} \log((P(S_i|T))$$

Equilibrium Frequencies

$\vec{\pi} = (\pi_A, \pi_C, \pi_G, \pi_T)$ -> must sum to 1

Assumption: time reversibility => $\pi_i P_{i,j}(t) = \pi_j P_{j,i}(t)$

Find substitution model



$$\lambda_{AA} = -\sum_{j \neq A} \lambda_{jA}$$

	A	C	G	T
A	0			
C	λ_{AC}	0	symm	
G	λ_{AG}	λ_{CG}	0	
T	λ_{AT}	λ_{CT}	λ_{GT}	0

→ rate matrix

I Jukes Cantor Model

everything equally likely

$$\begin{array}{c} \\ A \\ C \\ G \\ T \end{array}\begin{pmatrix} & & & \\ \alpha & & & \\ \alpha & \alpha & & \\ \alpha & \alpha & \alpha & \end{pmatrix}$$

$\vec{\pi} = (1/4, 1/4, 1/4, 1/4)$

II Felsenstein

$$\begin{array}{c} \\ A \\ C \\ G \\ T \end{array}\begin{pmatrix} & & & \\ \alpha & & & \\ \alpha & \alpha & & \\ \alpha & \alpha & \alpha & \end{pmatrix}$$

$\pi_i \neq \pi_j$

III Kimura 2-parameter Model

$$\begin{array}{c} \\ A \\ C \\ G \\ T \end{array}\begin{pmatrix} \cdot & & & \\ \beta & \cdot & & \\ \alpha & \beta & \cdot & \\ \beta & \alpha & \beta & \cdot \end{pmatrix}$$

$\vec{\pi} = (1/4, 1/4, 1/4, 1/4)$

IV HKY

$$\begin{array}{c} \\ A \\ C \\ G \\ T \end{array}\begin{pmatrix} \cdot & & & \\ \beta & \cdot & & \\ \alpha & \beta & \cdot & \\ \beta & \alpha & \beta & \cdot \end{pmatrix}$$

$\pi_i \neq \pi_j$

V GTR General time reversible model

STR/GTR

$$\begin{pmatrix} \cdot & & & \\ \alpha & \cdot & & \\ \beta & \gamma & \cdot & \\ \delta & \varepsilon & \zeta & \cdot \end{pmatrix}$$

$\pi_i \neq \pi_j$

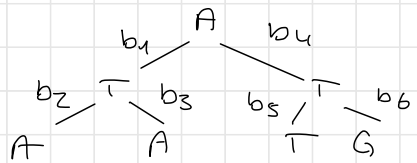
-> all rates relative to each other

Computing Likelihood

$P_{i,j}(t) =$ probability of starting in state $i \in \{A, C, T, G\}$ and after time t being in state j

1) Assume inner states of tree known:

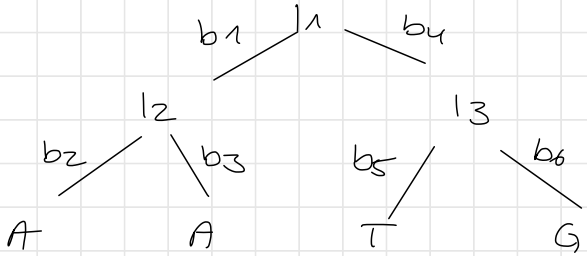
$b_1 \Rightarrow$ from A to T in time b_1



$\rightsquigarrow \pi_A \cdot P_{A,T}(b_1)\, P_{T,A}(b_2) \cdot P_{T,A}(b_3) \cdot P_{A,T}(b_4) \cdot P_{T,T}(b_5) \cdot P_{T,G}(b_6)$

π_A: likelihood of A as root

2) Now: inner states unknown



$\rightsquigarrow$ Likelihood of all possible combinations:

$L\left(\begin{smallmatrix} A \\ A \quad A \end{smallmatrix}\right) + L\left(\begin{smallmatrix} A \\ A \quad C \end{smallmatrix}\right) + \dots$

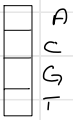
$$\Rightarrow \quad L = \sum_{l_1=A}^{T} \sum_{l_2=A}^{T} \sum_{l_3=A}^{T} \pi_{l_1} \cdot P_{l_1,l_2}(b_1)\, P_{l_1,l_3}(b_4) \cdot \dots$$

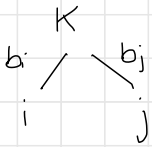
OR (over l_1, l_2, l_3) — AND (over the factors)

Combinatorial problem: n unknown nodes $\Rightarrow 4^n$ possibilities

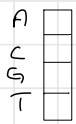
$\Rightarrow$ dynamic programming: Conditional likelihood vector

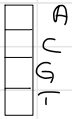
Calculate CLV of k with respect to i, j



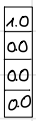


$$CLV_k(A) = \left(\sum_{S_i=A}^{T} P_{A\to S_i}(b_i) \cdot CLV_i(S_i)\right) \cdot \left(\sum_{S_j=A}^{T} P_{A\to S_j}(b_j) \cdot CLV_j(S_j)\right)$$





CLV of leaves is e.g. leaf = A →



⇒ Likelihood of the root given CLV_{root}

$$L = \sum_{S=A}^{T} \pi_S \cdot CLV_{root}(S)$$

Computing transition matrix P

- Introduce Q matrix

$Q = diag(\vec{\pi})\, R$ → time reversability for R : $\pi_i q_{ij} = \pi_j q_{ji}$

$$q = \begin{pmatrix} -3\lambda & \lambda & \lambda & \lambda \\ \lambda & -3\lambda & \lambda & \lambda \\ \lambda & \lambda & -3\lambda & \lambda \\ \lambda & \lambda & \lambda & -3\lambda \end{pmatrix}$$

(det = 0 !)
mit $\pi = \lambda$, $\alpha = 1$ for JC-M.

⇒ $P(t) = e^{Qt}$

- $Q = U V U^{-1}$ → EV-decomposition, Q not symmetric → non real values

⇝ define symmetric $Q' = diag(\sqrt{\vec{\pi}})^{-1} \cdot Q \cdot diag(\sqrt{\vec{\pi}})$

$= U' V (U')^T$

$$\Leftrightarrow \quad Q = \underbrace{\left(\text{diag}\left(\sqrt{\pi}\cdot u^{1}\right)\right)}_{U} \cdot V \cdot \underbrace{\left(u^{1\top}\,\text{diag}\left(\sqrt{\pi}\right)^{-1}\right)}_{U^{-1}}$$

$$\Rightarrow \quad P(t) = e^{Qt} = \sum_{k=0}^{\infty} \frac{1}{k!}(Qt)^k$$

$$= \sum_{k=0}^{\infty} \frac{1}{k!}(UVU^{-1})^k \cdot t^k$$

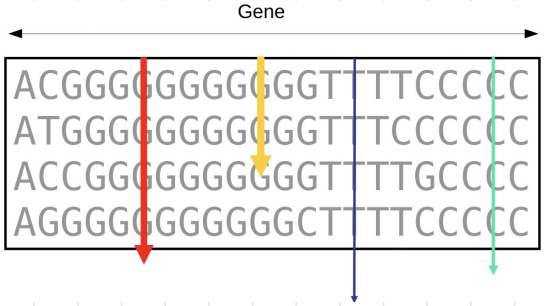
$$= U\left(\sum_{k=0}^{\infty} \frac{1}{k!} V^k t^k\right) U^{-1}$$

$$= U\, e^{\text{diag}(\lambda_i)t}\, U^{-1}$$

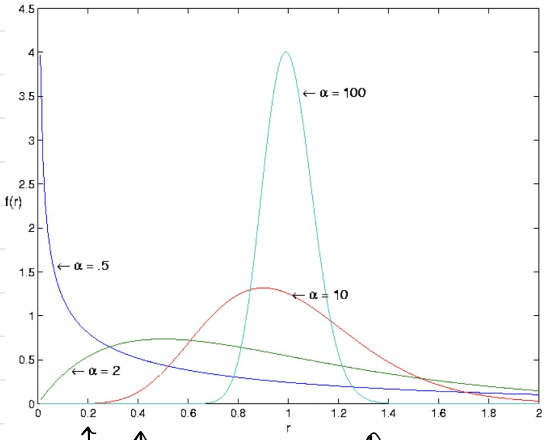
$\text{diag}(\lambda_i) = \text{diag}(\text{Eigenvalues})$

Among Site Rate Heterogeneity

- different sites evolve at different speeds



-> model using Γ-distribution



small α : rates differ significantly
-> high rate heterogeneity

e.g. r_0 r_1 . . . . r_{w-1}

Discrete Γ-model

- discret. Γ distribution at ω rates e.g. $\omega = 4$

→ log likelihood at site i is then

$$ln(L(i)) = \log\left(\frac{1}{\omega} * (L_0 + L_1 + \ldots + L_{\omega-1})\right)$$

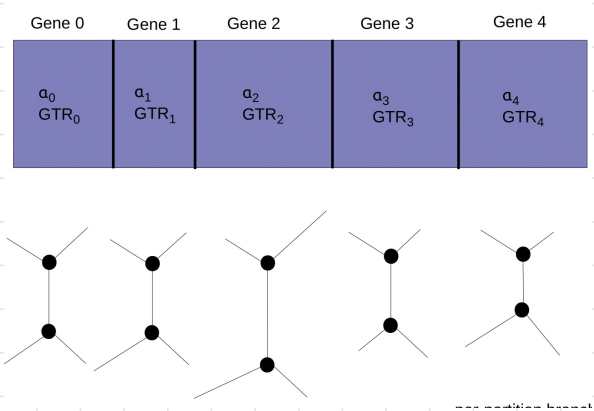
rate 0: $P(t) = e^{Q r_0 t}$ ... rate $\omega-1$: $P(t) = e^{Q r_{\omega-1} t}$

Mixture Models

- Γ model: 4 discrete evolutionary rates
  - ↳ consider all at once without assigning specific rate to a specific site

partitioned Dataset

use GTR for each Gene
↳ results in same tree topology per gene



→ increases # of params in likelihood model by $4 * (2n-3)$

Model Testing

more params → improved likelihood
→ but overparametrized?

Likelihood ratio test if models nested
Models A nested in model B if A's params subset of B's params

→ $LR = P(D|A) / P(D|B) = L(A)/L(B)$
$\Delta = ln(LR^2) = 2(ln(L(A)) - ln(L(B)))$

→ compare Δ to χ^2 distribution with degr. of fr. $k_A - k_B$ to determine if Δ is significant

→ D: can only compare likelihoods for same data

Bootstrapping

Goal: find best tree → to what degree does Data support resulting tree?

Creating Replicas

given alignment A with length n
→ resample n columns/sites (Ziehen mit Zurücklegen)
→ create tree (via Max. Likl. / Max. Pars.)
→ do this n times
⇒ n trees = bootstrap trees

Probabilities

- prob. of site being sampled per draw: $\frac{1}{n}$
- prob. of site not -"- : $1 - \frac{1}{n}$
- prob. of site not being sampled in any of the n draws: $(1 - \frac{1}{n})^n$
- prob. of site being sampled at least once in n draws
  → $Pr(X \geq 1) = 1 - (1 - \frac{1}{n})^n \approx 1 - \frac{1}{e} \approx 63{,}2\,\%$

↳ expectation: searches on alignment and replicates are somehow different

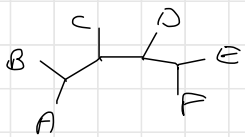
Bipartitions: Inner Branches

- inner branch = smallest unit of evolutionary relationship

ABICDEF → A, B more closely related to each other than taxa C, D, E, F

- trivial bipartition: outer taxa A|BCDEF

possible bipartitions for



A|BCDEF
AB|CDEF
ABC|DEF
ABCD|EF

Algorithm

1) Infer ML tree on full alignment
2) Create bootstrap tree set
3) Extract bipartitions from ML tree and bootstrap tree set
4) Annotate ML tree with relative frequency of its bipartitions in the bootstrap tree set

Consensus of Trees

with consensus threshold t

1) extract bipartitions from tree set
2) Determine consensus bipartitions → occur in $\geq t$% of trees
3) Transform consensus biparts into consensus tree

Consensi Flavors

- majority rule consensus : freq. > 50%
- strictly consensus : freq. > 95%
- extended majority rule consensus

Extended majority rule consensus

Bipartitions b_1, b_2 cannot be in same tree if incompatible

$b_1 = B|\bar{B}$, $b_2 = C|\bar{C}$ **compatible** if

$(B \cap C = \emptyset) \vee (B \cap \bar{C} = \emptyset) \vee (\bar{B} \cap C = \emptyset)$

→ Algo : given consensus Bips B_C, non-consensus Bips B_n

1) remove most frequent bip $b \in B_n$ from B_n
2) if b compatible to all $c_i \in B_C$: $B_C = B_C \cup \{b\}$
3) Loop until $|B_C| = (n-3)$ or $B_n = \emptyset$

Hashing

- store biparts. in hash-table
  → compute biparts. from adjacent biparts

$a = 001100 = CD|ABEF$

$b = 110000 = AB|CDEF$

$? = a|b = 111100 = ABCD|EF$

- r[] one random number per taxon
  ↳ hash: xor of all corresp. random numbers

hash(a) = r[C] xor r[D]

hash(b) = r[A] xor r[B]

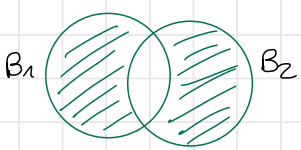
hash(?) = hash(a) xor hash(b)

Distances between trees

Robinson-Foulds Distance (metric)

Given unrooted trees T_1, T_2 with biparts B_1, B_2

$RF(T_1, T_2) = |B_1 \cup B_2| - |B_1 \cap B_2|$



T_1: AB|CDEF, ABD|CEF, ABDE|FC → B_1

T_2: AB|CDEF, ABCD|EF, CD|ABEF → B_2

$|B_1 \cup B_2| = 6$

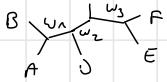
$|B_1 \cap B_2| = 2$

$= RF(T_1, T_2) = 4$

Flavors

Δ = symmetric set difference $X \Delta Y = (X \setminus Y) \cup (Y \setminus X)$

w_i = support value



	unweighted	weighted
absolute	$\lvert B_1 \cup B_2 \rvert - \lvert B_1 \cap B_2 \rvert$	$\sum_{b \in B_1 \Delta B_2} w(b)$
relative	$\frac{\lvert B_1 \cup B_2 \rvert - \lvert B_1 \cap B_2 \rvert}{2(n-3)}$	$\frac{\sum_{b \in B_1 \Delta B_2} w(b)}{2(n-3)}$

How many replicates?

1) Randomly split tree in subsets t', t''
2) Compute consensus trees $c(t')$, $c(t'')$
3) Computed weighted RF distance $wRF(c(t'), c(t''))$
4) Repeat 1) – 3) with cutoff values l, m
   if less than l permutations have $wRF < m$:
   Stop -> stable support values

Triplets / Quartets

alternative to biparts.

→ triplets for rooted
quartets for unrooted trees

⇝ for 4 taxa: only 3 possible trees
⇝ $O(n^4)$ to extract all quartets

Bayesian Methods

- sample posterior probability distribution

→ ML: searches best peak value

Probabilities

		A: a_1	A: a_2	A: Σ
B	b_1	·	·	·
	b_2	·	·	
	Σ			

- Joint probability $Pr(a_1, b_1)$: prob. of observing a_1 and b_1
- Marginal probability: $Pr(b_1) = Pr(b_1, a_1) + Pr(b_1, a_2)$
- Conditional probability: Prob. of observing a_1 given b_1 has occured

$$Pr(a_1 | b_1) = \frac{Pr(a_1, b_1)}{Pr(b_1)}$$

- Statistical independence: two events A, B independend if

$$Pr(A, B) = Pr(A) \cdot Pr(B)$$

Bayes Theorem

$$Pr(A, B) = \frac{Pr(B|A)}{Pr(A)} \Rightarrow Pr(B|A) = \frac{Pr(A, B)}{Pr(A)}$$

$Pr(B|A)$: unobserved outcome

$Pr(A, B)$: observed outcome

$$Pr(B|A) = \frac{Pr(A|B) \cdot Pr(B)}{Pr(A)}$$

$Pr(B|A)$: posterior prob.

$Pr(A|B)$: likelihood

$Pr(B)$: prior probability

$Pr(A)$: marginal probability

→ Phylogenetics

$$\underbrace{Pr(Tree, Params \mid Alignment)}_{\text{distr. over all possible trees and all model parameter values}} = \frac{\overbrace{Pr(Alignment \mid Tree, Params)}^{\text{does alignment fit tree + model?}} \cdot \underbrace{Pr(Tree, Params)}_{*}}{Pr(Alignment)}$$

* introduces prior knowledge/assumptions about prob. distn. of trees + params (GTR rates/ ...)

$n+1$ = # of possible trees

$$Pr(Alignment) = Pr(Alignment, t_0) + \dots + Pr(Alignment, t_n)$$
$$= Pr(Alignment \mid t_0) \cdot Pr(t_0) + \dots + Pr(Alignment \mid t_n) Pr(t_n)$$

→ General Bayes Theorem

$$f(\theta \mid A) = \frac{f(A \mid \theta) f(\theta)}{\int f(\theta) f(A \mid \theta) d\theta}$$

θ = tree topology, model params, branch lengths, ...

f = density function

$f(x) > 0 \quad \forall x$

$\int f(x) = 1$

$$Pr(0.2 \leq x \leq 0.3) = \int_{0.2}^{0.3} f(x)$$

Monte - Carlo Methods

how to compute $\int f(\theta) f(A|\theta) d\theta$?

Idea: approximate with samples

$$\int f(\theta) d\theta = \frac{1}{N} \sum_i^N f(\theta_i) \quad \text{with } \theta \in \mathbb{R}^m \text{ (domain)}$$

↳ domain volume $V \cdot \frac{1}{N} \cdot \sum_i^N f(\theta_i)$

→ problem: pure MC samples completely random

→ we are interested in the peaks

Markov Chain Monte Carlo Methods

higher sample density in interesting regions

→ biased random walks: probability to find sample in area with high posterior prob. ($f(\theta|A)$) is proportional to posterior distribution

Robot metaphor

- robot has to explore unknown landscape
- Altitude = probability
- generate step to take → evaluate whether to take it

accept/reject step from P1 → P2 based on ratio R of their posterior densities

$$R = \frac{Pr(P2|data)}{Pr(P1|data)} = \frac{Pr(P2)\,Pr(data|P2) \,/\, Pr(data)}{Pr(P1)\,Pr(data|P1) \,/\, Pr(data)}$$

$$= \frac{Pr(P2)\,Pr(data|P2)}{Pr(P1)\,Pr(data|P1)}$$

$\frac{Pr(P2)}{Pr(P1)}$ = prior ratio -> for uniform priors = 1

$\frac{Pr(data|P2)}{Pr(data|P1)}$ = likelihood ratio

Target distribution = posterior distr. -> the one to sample

proposal distribution: decides which point to randomly go to next
- small variance pro: seldom refuses step
  con: smaller steps, more steps required

↳ difficult to find!
dataset dependent, trial & error

Metropolis Algorithm (for Phylogenetics)

Init: choose random tree with random branch lengths as first sample

Loop:
- propose either - new tree topology
  - new branch length
- calculate acceptance ratio of proposal
- accept/reject change
- print current tree + branch lengths for every k iters = thinning

Finish: Summarize Sample using means, histograms, ...

Hastings Correction

asymmetric proposal distribution => bias

→ hastings ratio $HR = \frac{Q(P1|P2)}{Q(P2|P1)}$

if Q symmetric: $Q(P1|P2) = Q(P2|P1) \Rightarrow HR = 1$

$\Rightarrow R = \frac{Pr(P2)}{Pr(P1)} \cdot \frac{Pr(data|P2)}{Pr(data|P1)} \cdot HR$

Bipartitions

posterior prob. of bipart. AB|CDE

→ count relative occurence in generated samples

↳ approx. true proportion if MC converged!

Convergence

- can't say that MCMC-chain has converged
  ↳ can only say it hasn't

=> Run multiple chains

Metropolis-coupled MCMC Methods

Cold Chain: sees landscape as is

Hot chain: sees flatter landscape version → moves more quickly between peaks

↳ Run several chains simultaneously
1 cold chain
several heated chains

"flattened" acceptance Ratio: $R^{1/1+H}$ with H = temperature
cold chain: $H = 0$

Prior Probabilities

- convey scientist's beliefs before having seen the data
  → can bias analysis

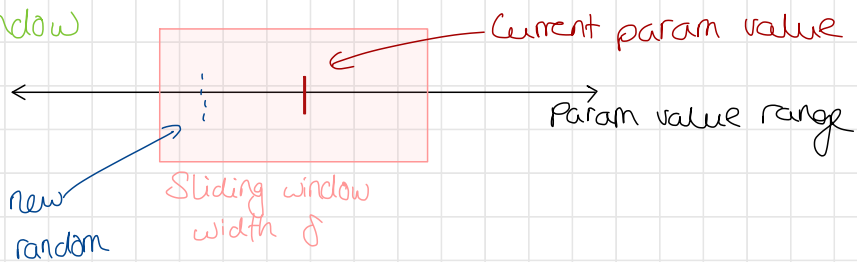
Proposal Mechanisms

- Univariate params + branch lengths:
  - sliding window
- branch lengths:
  - node slider
- topologies:
  - local proposal

Proposal Requirements:

- either no Hastings Ratio needed
- or it can be calculated
- acceptance rate ≈ 25%

Sliding Window



- Hastings Ratio = 1
- δ can be auto-tuned ⇝ acceptance rate 1/4 can be obtained
- used for α-shape param in Γ function for rate heterogeneity

Node Slider

1) Pick 2 contiguous branches randomly b_1, b_2
2) Multiply both by same random number
3) Propose new branch ratio b_1/b_2 at random

! HR ≠ 1

Tree proposal

1) Pick 3 contiguous branches at random that define 2 subtrees X, Y
2) Shrink / grow selected 3 branch segments by random amount
3) Prune X or Y (selected rand.)
4) Reinsert pruned tree at random position in the 3 branch segment

Different Models MCMC

goal: integrate over different models

- encode rate configurations for models as strings:
  - 111111 = JC Model
  - 123456 = GTR Model

→ Bell number: given n models there are B_n combinations possible

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

1) Model proposal

Split move: choose set of substitution rates with >1 member
- 111 222 (2-param model)
- → split it randomly into two rates
- 111 223 (3-param model)

merge move: choose 2 subst. rate sets

111 223

→ merge into one subst. rate set

111 222

2) Sampling Different Models

→ cannot compare likelihoods for different models

↳ use reversible jump MCMC to jump btw. models with different # of params

⇒ proposal ratio calculation changed

→ acceptance ratio:

r = likelihood ratio * prior ratio * proposal ratio * Jacobian

Jacobian defines linear map $\mathbb{R}^n \to \mathbb{R}^m$ at point x if $f(x)$ at x differentiable

→ posterior probability for one spec. model:

fraction of time (samples) the MCMC chain visited within that model

Evolution = change over time through inheritance

Model param: population size constant or time variant?
→ lecture: constant

Units of Evolution

Population

- evolves due to changing frequency of features of individuals

4 main forces:

- Genetic Drift (chance / random events)
- Migration 
- Mutation 
- Natural selection as response to some pressure

Gene

- inherited from generation to generation
- inheritence via Alleles
  Allele = specific form a or A of gene **A**
- diploid: 2 sets of corresponding chromosomes (homologous)
- if allele sequences in the 2 chrom. identical: homozygous
  -"- different: heterozygous
  ↳ cause phenotyp. differences
- humans inherit one allel from father, one from mother

Genotype: set of corresponding alleles in diploid organism
Phenotype: observation for the trait / property the gene controls

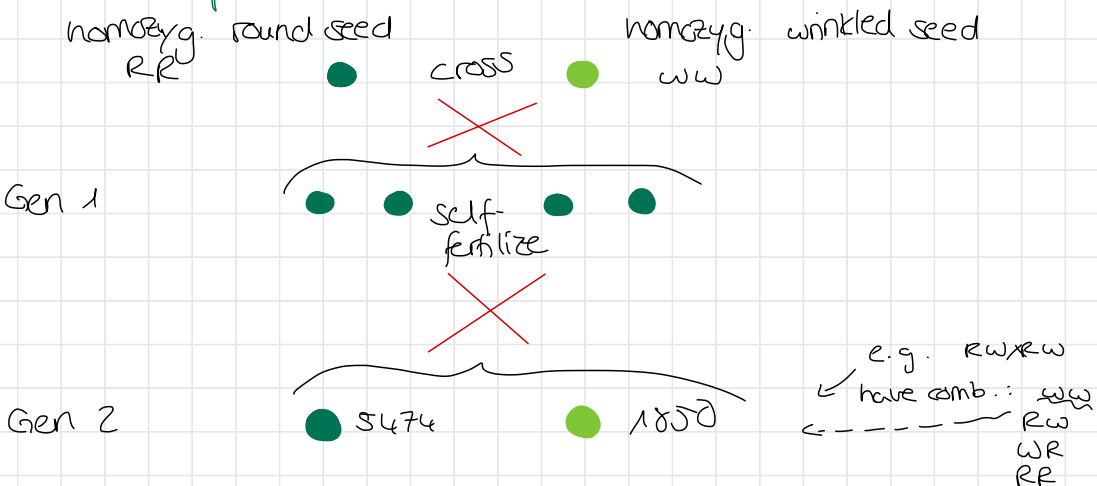
Mendelian Inheritance

Dominance

- genotype Rw → R
  - ⇒ R is dominant
  - w is recessive

- if no dominance-recession relationship: intermediate phenotype

Mendel Experiment



⇒ traits of 2 parent plants don't blend
→ both traits passed to Gen2 without showing in Gen 1

Allele Inheritance

Gene = **A**
corresp. alleles = a/A if only 2 else $A_1, A_2, \ldots$

Polymorphic gene

Gene **A** is polymorphic when there exist multiple alleles

Polymorphic site = SNP ('Snip')

after HGP: observe sites in certain genes with more than one state
- called snip SNP = Single Nucleotide Polymorphism

In population genetics: few mutations => MSA has to be correct!

Does dominance affect allele frequency?

assumptions:

- infinit population size
- random mating (= independent of phenotypes)
- gene **A** has 2 alleles A, a

- given current frequencies at generation 0 of alleles defining genotype

$f_0(A) = p$

$f_0(a) = q$

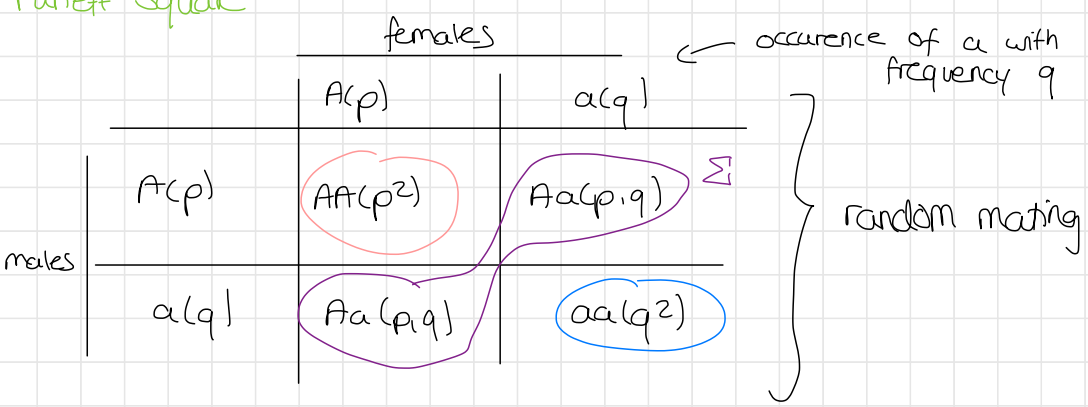
with $p + q = 1$

Generation t: $f_t(A) = f_t(AA) + \frac{1}{2} f_t(Aa)$

$f_t(a) = f_t(aa) + \frac{1}{2} f_t(Aa)$

→ how to compute $f_0 \rightarrow f_1$?

Punett Square



→ $f_1(AA) = p^2$ $f_1(aa) = q^2$

$f_1(Aa) = 2pq$

$\Rightarrow \quad f_0(AA) \neq f_1(AA) \qquad ! \; f_0(A) \neq f_0(AA)$

for $t > 0$: $\quad f_t(AA) = f_1(AA)$
$f_t(aa) = f_1(aa)$
$f_t(Aa) = f_1(Aa)$

$\Rightarrow \quad f_1(A) = f_1(AA) + \frac{1}{2} f_1(Aa) = p^2 + pq = p(p+q) \overset{q=1-p}{=} p = f_0(A)$

$\rightarrow$ frequency of allele occurence does not change
proportion of genotypes AA : Aa : aa constant after gen 1

Hardy-Weinberg Equilibrium

= null hypothesis in evol. biol.
$\rightarrow$ what happens to allel freq. without influence of the 4 evolutionary forces

Finite Population Sizes

- N individuals in diploid population $\Rightarrow$ 2N chrom.
- Frequency of allel A = p

Random Genetic Drift

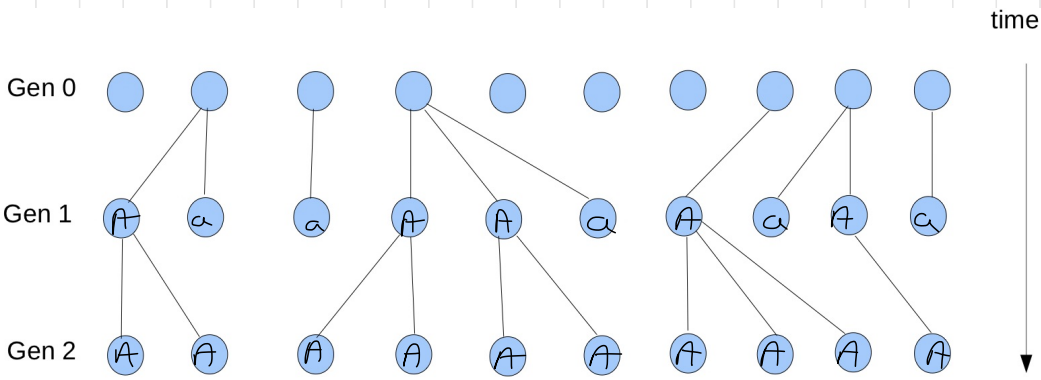
- Some alleles passed to next generation disproportionally without being advantageous / harmful
- happens esp. often in small populations (sampling errors)

Wright-Fisher Model

- population size N (2N chromosomes) constant per generation
- random mating
- non-overlapping generations
- no natural selection
- equal distribution of sexes

Simulation

- each individual from offspring picks random parent
  → all parents equally likely to be picked
  parent can be picked > 1 times
- each offspring inherits genetic info of the parent



→ a vanished due to random parent picking

- probability to pick allel A as parent is

$$p = \frac{\#A}{2N}$$

if population remains constant: sample $2N$ times from the current generation to construct next gen. with $2N$ offsprings

→ p remains constant

⇒ probability to pick A as ancestor k times

$$P(X=k) = \binom{n}{k} p^k (1-p)^{n-k} \qquad p = \frac{\#A}{2N};\ n = 2N$$

$$E(\#A) = 2N * p$$
$$Var(\#A) = 2N * p * (1-p)$$

! next state depends only on current state ⇒ model as Markov Chain

Transition Probabilities

Probability of changing from i alleles in gen t to j alleles in gen $t+1$:

$$\text{Prob}\{X(t+1) = j \mid X(t) = i\}$$

$$= p_{ij} = \binom{2N}{j} \left(\frac{i}{2N}\right)^j \left(1 - \frac{i}{2N}\right)^{2N-j}$$

$$i, j = 0, 1, \ldots, 2N$$

($2N \rightarrow$ either N diploid organisms or $2N$ haploid)

Absorbing state in Markov chain
- probability to exit this state is 0

- Wright-Fisher model: only one allele left
  -> given frequency of allele A $\#A/2N$
  probability that A will become fixed: $\#A/2N$

Heterozygosity and Genetic Drift

- Reduction of polymorphism quantified by degree of homozygosity
  G = probability that 2 alleles are identical

Het_t = heterozyg. at gen. t
$= Het_{(t-1)}(1 - 1/2N)$
$= Het_0 (1 - 1/2N)^t$

$1 - \frac{1}{2N}$ = prob. that 2 randomly chosen alleles differ

Mutation - Drift Balance

- genetic drift removes polymorphism from population
- mutations introduce polymorphism

Balance

1) · Loss of heterozygocity per generation due to genetic drift:
$$-\frac{1}{2N} \cdot Het$$

· μ = mutation rate per gene (! 2 alleles per gene) per generation

· gain of heteroz. due to mutation:
$$2\mu(1-Het)$$

2) · Pick two alleles
· Consider transition from generation $t \rightarrow t+1$
· Probability that they are identical: $1-Het$
· If identical: probability that one of them will mutate:
$$2\mu$$

$$\Rightarrow \quad Het_{t+1} = Het_t - \frac{1}{2N} \cdot Het_t + 2\mu(1-Het_t)$$

$$\Delta Het = -\frac{1}{2N} \cdot Het_t + 2\mu(1-Het_t)$$

$$\Delta Het = 0 \Rightarrow Het = \frac{4\mu N}{1+4\mu N}$$

Mutation Rate

· μ = mutation prob. per gen. per individual
· $2N$ indiv. $\rightarrow$ $2N\mu$ mutations per gen.
· probability that mutation will be fixed = $\frac{1}{2N}$

$\Rightarrow$ rate of mutation arising and being fixed in population
$$\frac{1}{2N} \cdot 2N\mu = \mu$$

Natural Selection

- fitness = ability of an individual to survive + reproduce
  → depends on genotype

- selection = one genotype reproduces more freq. than others

- if genotype has better fitness : (e.g. AA)
  - will fix in the population after several gen.
  ⇒ allel A will also fix
  ⇝ natural selection has favored allele A
  ⇝ natural selection on A = Positive Selection

Models of Selection