Topics of today:

1. Small Parsimony / Big Parsimony
2. Fitch's algorithm
3. Small Parsimony with SCJ
4. Quiz-review

Small (and big) parsimony problems

$\mathrm{d}_{\mathrm{M}}(\mathbb{G}_1, \mathbb{G}_2)$: distance between two genomes under some model M

Binary tree $\mathtt{T}$ $\begin{cases} \text{topology of } \mathtt{T} \text{ is known} \\ \text{each leaf } \mathtt{u} \text{ of } \mathtt{T} \text{ corresponds to a given genome } \mathbb{G}_{\mathtt{u}} \end{cases}$

Function $\mathcal{A}$: assigns a genome $\mathcal{A}(\mathtt{u})$ to each node $\mathtt{u}$ of $\mathtt{T}$
(if $\mathtt{u}$ is a leaf of $\mathtt{T}$, then $\mathcal{A}(\mathtt{u}) = \mathbb{G}_{\mathtt{u}}$)

Weight $w(\mathtt{uv})$ of a branch $\mathtt{uv}$ of $\mathtt{T}$ under $\begin{cases} \text{assignment } \mathcal{A} \\ \text{model } \mathrm{M} \end{cases}$: $w(\mathtt{uv}) = \mathrm{d}_{\mathrm{M}}(\mathcal{A}(\mathtt{u}), \mathcal{A}(\mathtt{v}))$

Small parsimony problem under the model M:

Find an assignment $\mathcal{A}$ that minimizes the total branch weight of $\mathtt{T}$:

$$\mathcal{W}(\mathtt{T}) = \min_{\mathcal{A} \in \mathfrak{A}} \sum_{\mathtt{uv} \in E(\mathtt{T})} \mathrm{d}_{\mathrm{M}}(\mathcal{A}(\mathtt{u}), \mathcal{A}(\mathtt{v}))$$

Big parsimony problem under the model M:

Given a set of k genomes $\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_k$,

find $\begin{cases} \text{tree } \mathtt{T} \text{ whose } k \text{ leaves are in one-to-one correspondence with the genomes } \mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_k \\ \text{assignment } \mathcal{A} \text{ of genomes to the nodes of } \mathtt{T} \end{cases}$

minimizing the total branch weight $\mathcal{W}(\mathtt{T})$

Small parsimony with Fitch's algorithm

General model:

$$\begin{cases} \text{binary rooted tree } \mathtt{T}\text{, with } n \text{ nodes} \\ \text{genomes are represented by sequences of a fixed length } \ell\text{, over a finite alphabet } \Sigma \text{ (with } |\Sigma| = m\text{)} \\ \text{hamming distance (hd) gives the weight of the branches} \end{cases}$$

Assuming positions being mutually independent, the problem can be solved for each position separately:

The algorithm determines an optimal value for each position p of each node $\mathtt{v}$, denoted by $s_{\mathtt{v}}[p]$

Bottom-up phase: defines set $B(\mathtt{v}, p)$ of possible values for each $s_{\mathtt{v}}[p]$, based on $\mathtt{v}$'s children $\mathtt{x}_1$ and $\mathtt{x}_2$

$$\begin{cases} \text{if } \mathtt{v} \text{ is a leaf : } B(\mathtt{v}, p) = \{s_{\mathtt{v}}[p]\} \\ \text{else} \begin{cases} \text{compute } B(\mathtt{v}, p) = B(\mathtt{x}_1, p) \cap B(\mathtt{x}_2, p) \\ \text{if } B(\mathtt{v}, p) = \emptyset : B(\mathtt{v}, p) = B(\mathtt{x}_1, p) \cup B(\mathtt{x}_2, p) \end{cases} \end{cases}$$

Complexity: $O(mn)$ (bottom-up traversal takes $O(n)$, computation of each node takes $O(m)$)

Top-down phase: defines final value $s_{\mathtt{v}}[p]$, based on set $B(\mathtt{v}, p)$ and $\mathtt{v}$'s parent $\mathtt{u}$

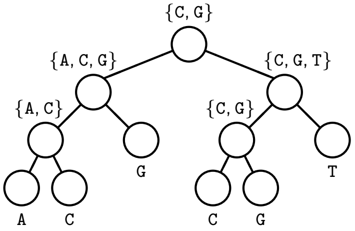
$$\begin{cases} \text{if } \mathtt{v} \text{ has a parent } \mathtt{u} \text{ and } s_{\mathtt{u}}[p] \in B(\mathtt{v}, p) : s_{\mathtt{v}}[p] = s_{\mathtt{u}}[p] \\ \text{otherwise (including the root), arbitrarily assign any value from } B(\mathtt{v}, p) \text{ to } s_{\mathtt{v}}[p] \end{cases}$$

Complexity: $O(mn)$ (top-down traversal takes $O(n)$, computation of each node takes $O(m)$)

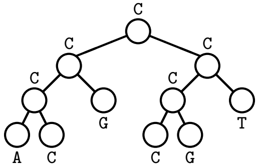
Total complexity: $O(mn)$ per position; with ℓ positions: $O(\ell mn)$

Small parsimony with Fitch's algorithm

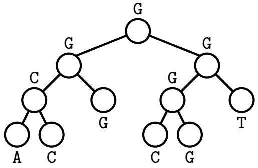
Bottom-up phase:



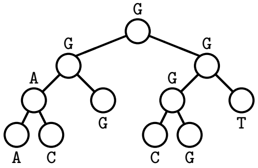
Top-down phase:



$\mathcal{W} = 4$



$\mathcal{W} = 4$



$\mathcal{W} = 4$

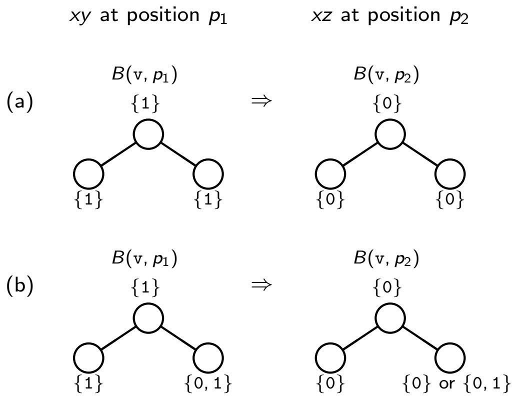
Small parsimony under SCJ with Fitch's algorithm

$$\begin{cases} \text{set of gene families } \mathcal{F} \\ \\ \text{set of extremities } \xi(\mathcal{F}) = \{f^t : f \in \mathcal{F}\} \cup \{f^h : f \in \mathcal{F}\} \quad (|\xi(\mathcal{F})| = 2|\mathcal{F}|) \\ \\ \text{set of all possible adjacencies } \widehat{\alpha}(\mathcal{F}) = \{\text{subsets of } \xi(\mathcal{F}) \text{ with size } 2\} \\ \qquad \Rightarrow \quad |\widehat{\alpha}(\mathcal{F})| = \binom{|\xi(\mathcal{F})|}{2} = \binom{2|\mathcal{F}|}{2} = \frac{2|\mathcal{F}|(2|\mathcal{F}|-1)}{2} = O(|\mathcal{F}|^2) \\ \qquad \text{Note: } \widehat{\alpha}(\mathcal{F}) \text{ has many pairs of } \textbf{conflicting adjacencies } xy, xz \text{ with } y \neq z \end{cases}$$

$$\text{genome assigned to vertex } \mathtt{u} \begin{cases} \text{represented by a sequence } s_{\mathtt{u}} \text{ of length } \ell = |\widehat{\alpha}(\mathcal{F})| \text{ over the binary alphabet } \{0,1\} \\ \\ \text{cannot contain conflicting adjacencies:} \\ \quad \text{for each pair of conflicting adj. in } \widehat{\alpha}(\mathcal{F}) \text{, at most one of the two can be set to 1 in } s_{\mathtt{u}} \end{cases}$$

Small parsimony under SCJ with Fitch's algorithm

Possible bottom-up scenarios involving two conflicting adjacencies



Conflicts can be avoided by assigning the value 0 to each ambiguous position of the root genome:

$$\mathtt{r} \text{ is the root of } \mathtt{T}\text{: for each position } p \text{ of } s_{\mathtt{r}} \begin{cases} s_{\mathtt{r}}[p] = 1 & \text{if } B(\mathtt{r}, p) = \{1\} \\ s_{\mathtt{r}}[p] = 0 & \text{otherwise} \end{cases}$$

It is easy to verify that: $d_{\text{SCJ}}(\mathtt{u}, \mathtt{v}) = \text{hd}(s_{\mathtt{u}}, s_{\mathtt{v}})$

Close-related problems are NP-hard...

Small parsimony under breakpoint distance

Big parsimony under SCJ

References

Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology

(Walter M. Fitch)

Systematic Zoology, vol. 20, pp. 406–416 (1971)

SCJ: A Breakpoint-Like Distance that Simplifies Several Rearrangement Problems

(Pedro Feijão and João Meidanis)

TCBB volume 8 Number: 5 (2011)

Quiz - Review

Inversion model 1

1 In the DCJ model any operation reconstructing a target adjacency is optimal, but the same is not true for the inversion model because...

- A a target adjacency can be bad
- B a target adjacency can be already present in the genome
- C reconstructing a target adjacency can be unsafe

2 A cycle is bad when...

- A it cannot be sorted by inversions
- B it interleaves another bad cycle
- C it contains only bad target adjacencies

3 Which data structure helps finding safe inversions?

- A relational diagram
- B overlap graph
- C component tree

4 A bad component can be fixed...

- A with a neutral inversion
- B with a split inversion
- C with a safe inversion

Inversion model 2

1 Each leaf of the component tree represents...

- A a bad component
- B a hurdle
- C a fortress

2 The cost of covering a component tree can be expressed in terms of...

- A the number of bad nodes
- B the length of the longest traversal of the tree
- C the number of leaves

3 Fixing a super hurdle with a neutral inversion

- A is a good strategy
- B creates a new hurdle
- C destroys a good component

4 Merging two good (or trivial) components..

- A can merge bad components into a good one
- B creates a new bad component
- C is never recommended

DCJ-indel model 1

1 The indel-potential is defined as...

- A the number of runs in a component
- B the smallest number of runs that can be obtained after sorting with internal gaining DCJs
- C the number of indel-edges in a componenta bad component

2 The indel-potential of a component depends on..

- A its number of runs
- B its number of indel-edges
- C its length

3 The number of runs in a cycle can be...

- A 0,1,2,4,6,8,...
- B any non-negative integer
- C any positive integer

4 The number of runs in a path can be...

- A 0,1,2,4,6,8,...
- B any non-negative integer
- C any positive integer

DCJ-indel model 2

1 A recombination can reduce the overall number of runs by at most...

A 1   B 2   C 3

2 A recombination can reduce the overall overall indel-potential by at most...

A 1   B 2   C 3

3 A recombination involving a cycle is...

A gaining

B neutral

C losing

4 A recombination involving a cycle can be...

A deducting

B part of an optimal sorting sequence

C none of those

DCJ-indel model - Path recombinations

With respect to the endpoints:

$$\mathbb{A}\!-\!\mathbb{A} \;+\; \mathbb{B}\!-\!\mathbb{B} \begin{cases} \mathbb{A}\!-\!\mathbb{B} \;+\; \mathbb{A}\!-\!\mathbb{B} & \text{(gaining)} \\ \mathbb{A}\!-\!\mathbb{B} \;+\; \mathbb{A}\!-\!\mathbb{B} & \text{(gaining)} \end{cases}$$

$$\mathbb{A}\!-\!\mathbb{B} \;+\; \mathbb{A}\!-\!\mathbb{B} \begin{cases} \mathbb{A}\!-\!\mathbb{A} \;+\; \mathbb{B}\!-\!\mathbb{B} & \text{(losing)} \\ \mathbb{A}\!-\!\mathbb{B} \;+\; \mathbb{A}\!-\!\mathbb{B} & \text{(neutral)} \end{cases}$$

$$\mathbb{A}\!-\!\mathbb{A} \;+\; \mathbb{A}\!-\!\mathbb{A} \begin{cases} \mathbb{A}\!-\!\mathbb{A} \;+\; \mathbb{A}\!-\!\mathbb{A} & \text{(neutral)} \\ \mathbb{A}\!-\!\mathbb{A} \;+\; \mathbb{A}\!-\!\mathbb{A} & \text{(neutral)} \end{cases}$$

With respect to the runs:

$$\mathcal{AB} \;+\; \mathcal{AB} \begin{cases} \mathcal{AA} \;+\; \mathcal{BB} & (\Delta_\lambda = -2) \\ \mathcal{ABBA} \;+\; \varepsilon & (\Delta_\lambda = -2) \end{cases}$$

$$\mathcal{A}(\mathcal{B}) \;+\; \mathcal{A} \begin{cases} \mathcal{AA} \;+\; (\mathcal{B}) & (\Delta_\lambda = -1) \\ \mathcal{AA}(\mathcal{B}) \;+\; \varepsilon & (\Delta_\lambda = -1) \end{cases}$$

$$(\mathcal{A})\mathcal{B} \;+\; \mathcal{B} \begin{cases} (\mathcal{A}) \;+\; \mathcal{BB} & (\Delta_\lambda = -1) \\ (\mathcal{A})\mathcal{BB} \;+\; \varepsilon & (\Delta_\lambda = -1) \end{cases}$$

Deducting path recombinations: $\begin{cases} \text{gaining with } \Delta_\lambda = -2 \\ \text{gaining with } \Delta_\lambda = -1 \\ \text{neutral with } \Delta_\lambda = -2 \end{cases}$

DCJ-indel model - Path recombinations

Putting together (examples):

$$\overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} = \overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\varepsilon}}{\mathbb{A}-\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -1\text{)}$$

$$\overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} = \overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\varepsilon}}{\mathbb{A}-\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -1\text{)}$$

$$\overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{BA}}}{\mathbb{A}-\mathcal{BA}-\mathbb{B}} = \overset{\mathbb{AB}_{\mathcal{A}}}{\mathbb{A}-\mathcal{AB}\mathcal{BA}-\mathbb{B}} + \overset{\mathbb{AB}_{\varepsilon}}{\mathbb{A}-\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -2\text{)}$$

$$\overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{BA}}}{\mathbb{A}-\mathcal{BA}-\mathbb{B}} = \overset{\mathbb{AB}_{\mathcal{A}}}{\mathbb{A}-\mathcal{A}\mathcal{A}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{B}}}{\mathbb{A}-\mathcal{B}\mathcal{B}-\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -2\text{)}$$

$$\overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{BA}}}{\mathbb{A}-\mathcal{BA}-\mathbb{B}} = \overset{\mathbb{AB}_{\mathcal{B}}}{\mathbb{A}-\mathcal{BA}\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\varepsilon}}{\mathbb{A}-\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -2\text{)}$$

$$\overset{\mathbb{AB}_{\mathcal{AB}}}{\mathbb{A}-\mathcal{AB}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{BA}}}{\mathbb{A}-\mathcal{BA}-\mathbb{B}} = \overset{\mathbb{AB}_{\mathcal{B}}}{\mathbb{A}-\mathcal{B}\mathcal{B}-\mathbb{B}} + \overset{\mathbb{AB}_{\mathcal{A}}}{\mathbb{A}-\mathcal{A}\mathcal{A}-\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -2\text{)}$$

Λ	λ
0	0
1	1
2	2
3	2
4	3
5	3
6	4
7	4

DCJ-indel model - Path recombinations

Putting together (examples):

$$\overset{\mathbb{A}\mathbb{B}_{\mathcal{A}\mathcal{B}}}{\mathbb{A}\text{–}\mathcal{A}\mathcal{B}\text{–}\mathbb{B}} + \overset{\mathbb{A}\mathbb{B}_{\mathcal{B}\mathcal{A}}}{\mathbb{A}\text{–}\mathcal{B}\mathcal{A}\text{–}\mathbb{B}} = \overset{\mathbb{A}\mathbb{B}_{\mathcal{A}}}{\mathbb{A}\text{–}\mathcal{A}\mathcal{B}\mathcal{B}\mathcal{A}\text{–}\mathbb{B}} + \overset{\mathbb{A}\mathbb{B}_{\varepsilon}}{\mathbb{A}\text{–}\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -2\text{)}$$

$$\overset{\mathbb{A}\mathbb{B}_{\mathcal{A}\mathcal{B}}}{\mathbb{A}\text{–}\mathcal{A}\mathcal{B}\mathcal{A}\mathcal{B}\text{–}\mathbb{B}} + \overset{\mathbb{A}\mathbb{B}_{\mathcal{B}\mathcal{A}}}{\mathbb{A}\text{–}\mathcal{B}\mathcal{A}\text{–}\mathbb{B}} = \overset{\mathbb{A}\mathbb{B}_{\mathcal{A}}}{\mathbb{A}\text{–}\mathcal{A}\mathcal{B}\mathcal{A}\mathcal{B}\mathcal{B}\mathcal{A}\text{–}\mathbb{B}} + \overset{\mathbb{A}\mathbb{B}_{\varepsilon}}{\mathbb{A}\text{–}\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -2\text{)}$$

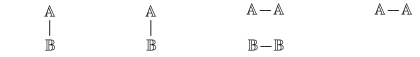
Λ	λ
0	0
1	1
2	2
3	2
4	3
5	3
6	4
7	4

$$\overset{\mathbb{A}\mathbb{B}_{\mathcal{A}\mathcal{B}}}{\mathbb{A}\text{–}\mathcal{A}\mathcal{B}\mathcal{A}\mathcal{B}\text{–}\mathbb{B}} + \overset{\mathbb{A}\mathbb{B}_{\mathcal{B}\mathcal{A}}}{\mathbb{A}\text{–}\mathcal{B}\mathcal{A}\mathcal{B}\mathcal{A}\text{–}\mathbb{B}} = \overset{\mathbb{A}\mathbb{B}_{\mathcal{A}}}{\mathbb{A}\text{–}\mathcal{A}\mathcal{B}\mathcal{A}\mathcal{B}\mathcal{B}\mathcal{A}\mathcal{B}\mathcal{A}\text{–}\mathbb{B}} + \overset{\mathbb{A}\mathbb{B}_{\varepsilon}}{\mathbb{A}\text{–}\mathbb{B}} \quad \text{(neutral, with } \Delta_\lambda = -2\text{)}$$

DCJ and DCJ-indel models - Capping

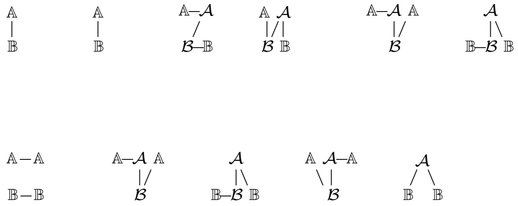
Add caps to close all paths of the graph into cycles, preserving the distance

Canonical capping (no indel edges): maximizes the number of cycles



paths	linking cycle	Δn	Δc	$\Delta(2\mathbb{AB})$	Δ_{DCJ}
$\mathbb{AB}$	$(\mathbb{AB})$	$+0.5$	$+1$	-0.5	0
$\mathbb{AA} + \mathbb{BB}$	$(\mathbb{AA}, \mathbb{BB})$	$+1$	$+1$	0	0
$\mathbb{AA}$	$(\mathbb{AA}, \Gamma_{\mathbb{B}})$	$+1$	$+1$	0	0
$\mathbb{BB}$	$(\mathbb{BB}, \Gamma_{\mathbb{A}})$	$+1$	$+1$	0	0

Singular capping (with indel edges): optimizes the number of cycles and of runs at the same time



paths	linking cycle	Δn	Δc	$\Delta(2\mathbb{AB})$	$\Delta\lambda$	$\Delta^{\lambda}_{\mathrm{DCJ}}$
$\mathbb{AA}_{\mathcal{AB}} + \mathbb{BB}_{\mathcal{AB}}$	$(\mathbb{AA}_{\mathcal{AB}}, \mathbb{BB}_{\mathcal{BA}})$	$+1$	$+1$	0	-2	-2
$2 \times \mathbb{AA}_{\mathcal{AB}} + \mathbb{BB}_{\mathcal{A}} + \mathbb{BB}_{\mathcal{B}}$	$(\mathbb{AA}_{\mathcal{AB}}, \mathbb{BB}_{\mathcal{B}}, \mathbb{AA}_{\mathcal{BA}}, \mathbb{BB}_{\mathcal{A}})$	$+2$	$+1$	0	-4	-3
$\mathbb{AB}_{\mathcal{AB}} + \mathbb{AB}_{\mathcal{BA}}$	$(\mathbb{AB}_{\mathcal{AB}}, \mathbb{AB}_{\mathcal{BA}})$	$+1$	$+1$	-1	-2	-1
$\mathbb{AB}$	$(\mathbb{AB})$	$+0.5$	$+1$	-0.5	0	0
$\mathbb{AA} + \mathbb{BB}$	$(\mathbb{AA}, \mathbb{BB})$	$+1$	$+1$	0	0	0