Assessing the robustness of predictions for ancestral adjacencies

A. Rajaraman[1,2], C. Chauve[1] and Y. Ponty[1,3,4,5] and J. Zanetti[1,3,6,7]

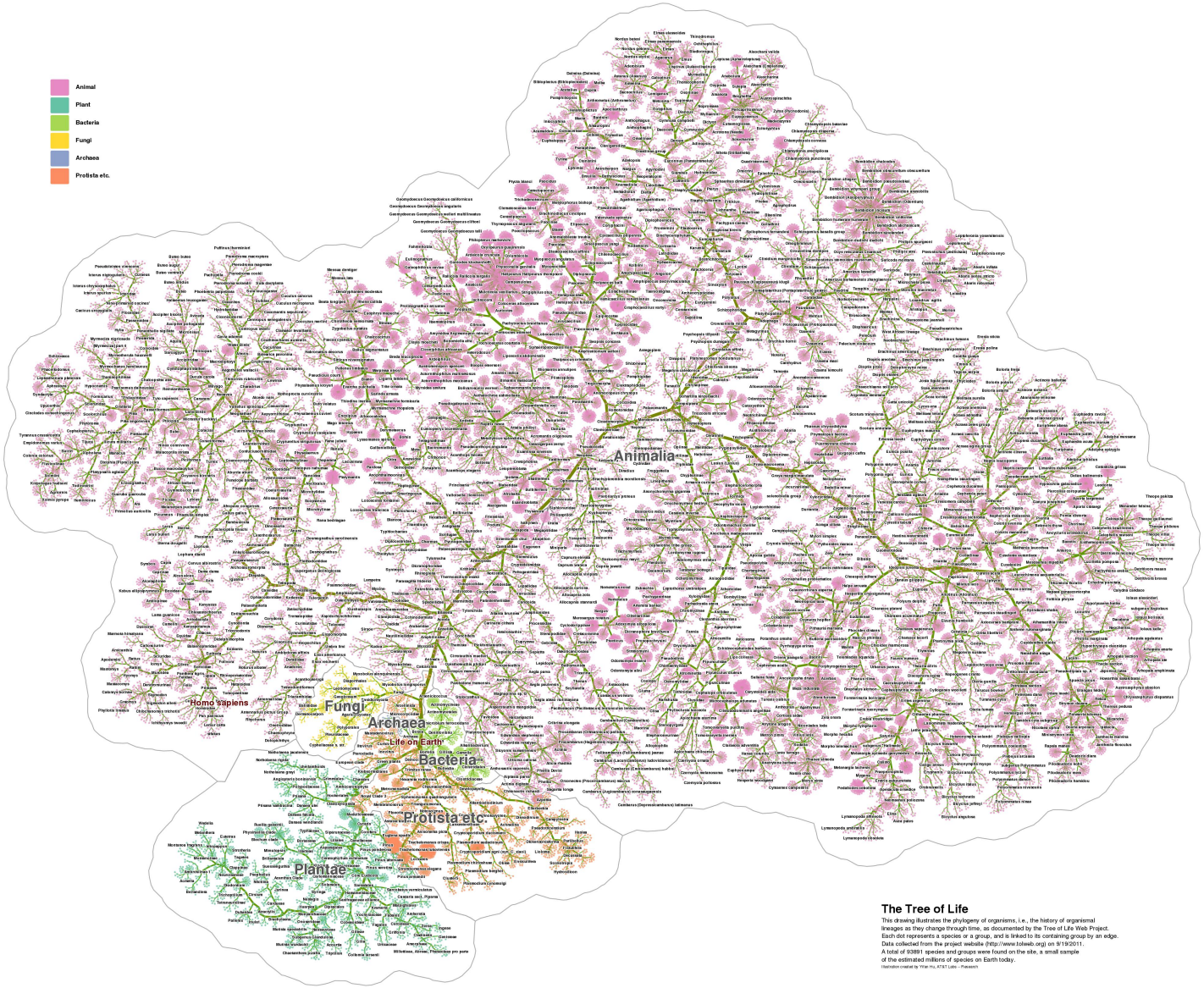
1. Department of Mathematics, Simon Fraser University,
2. Department of Bioengineering, UIUC
3. Pacific Institute for the Mathematical Sciences,
4. CNRS/LIX, École Polytechnique, Palaiseau
5. AMIB, Inria Saclay
6. Institute of Computing, UNICAMP, Campinas, Brazil
7. São Paulo Research Foundation, FAPESP, São Paulo, Brazil

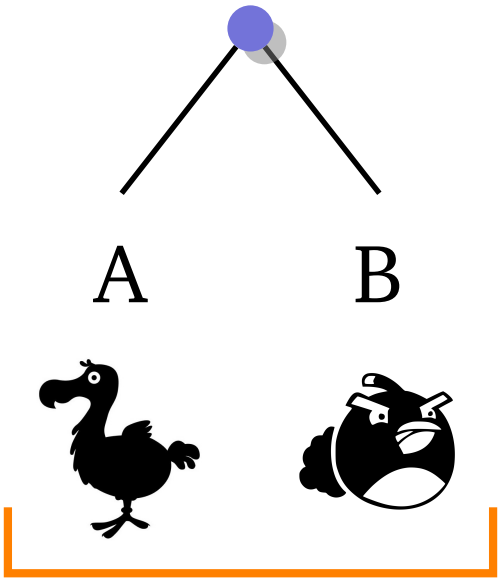
Some things I learned during my

~~holidays~~ sabbatical in Vancouver

Anything misleading is probably false, as distractions were many and time was scarce.

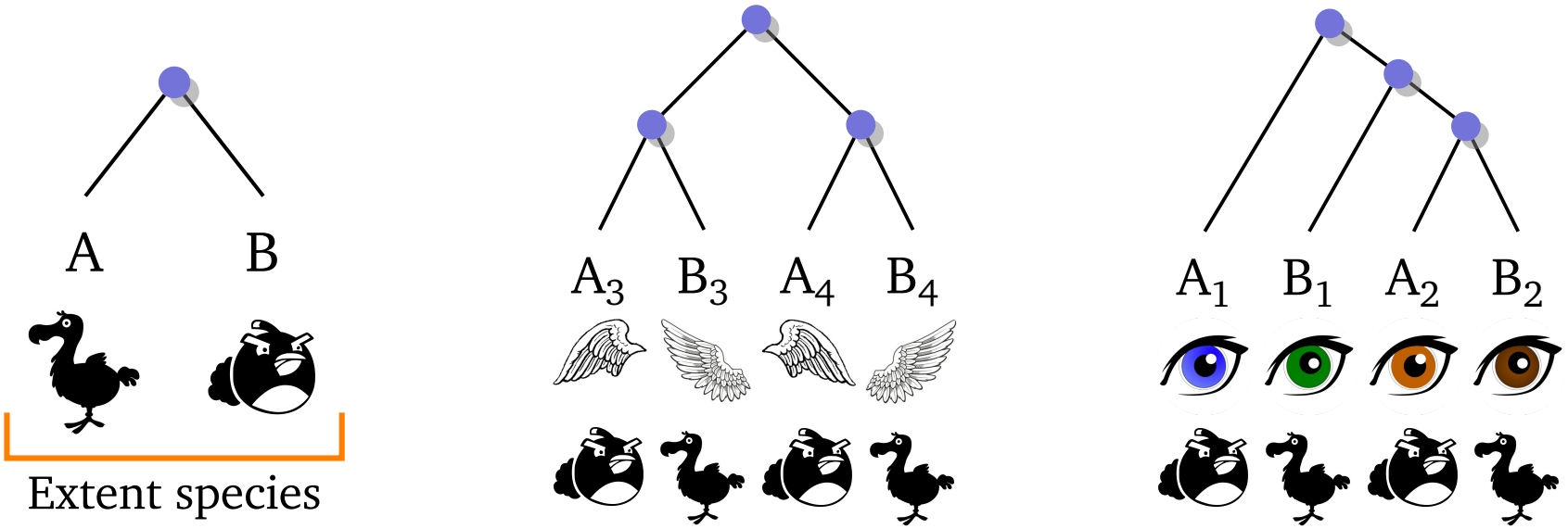
The tree of life





Extent species

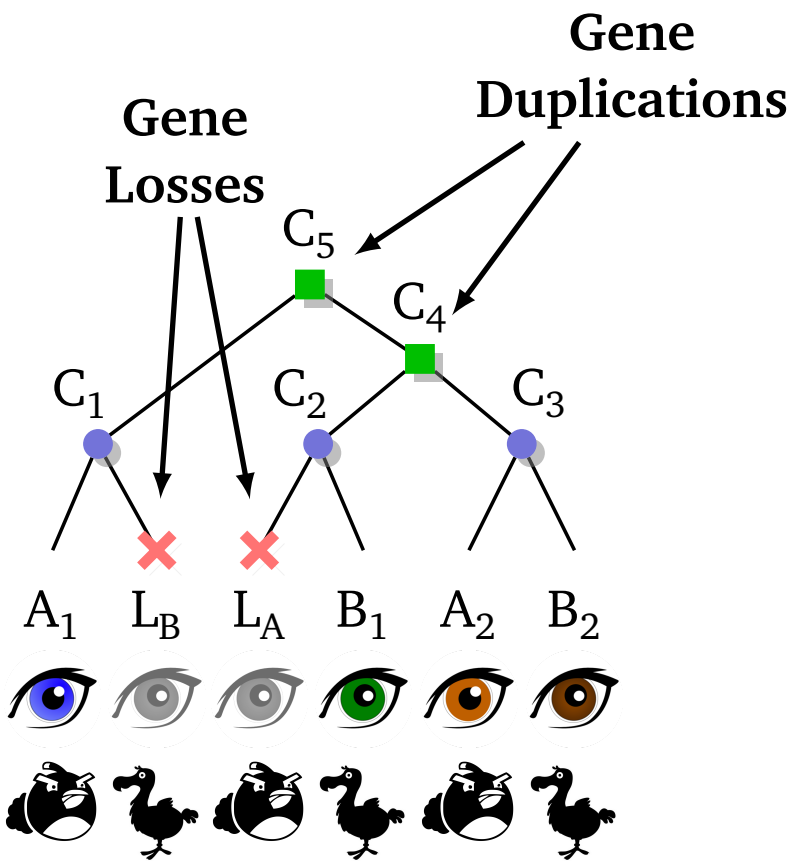
Main event: Speciation



How best to explain (reconcile) those?

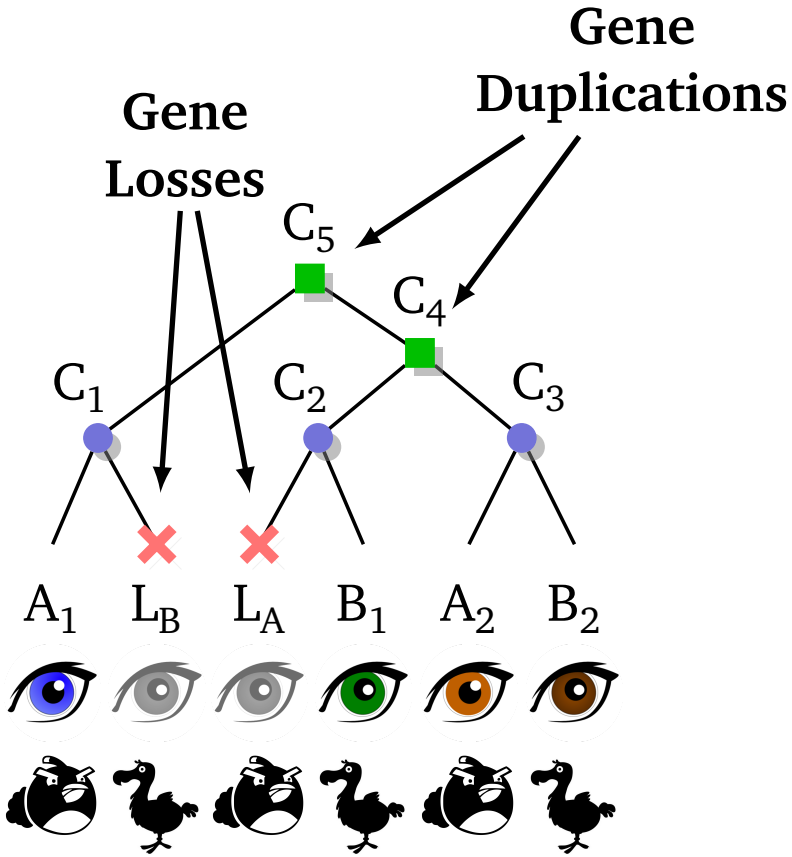
(and hopefully learn something in the process)

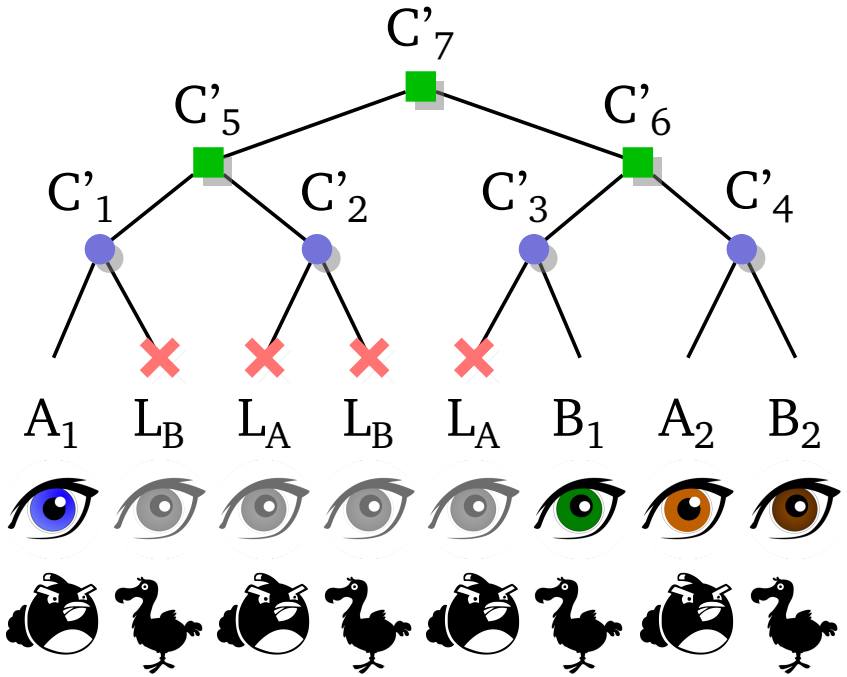
Gene/Species Trees Reconciliation



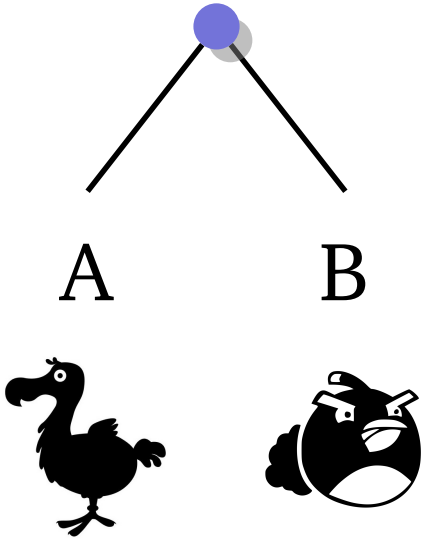
Find (parsimonious) evolutionary scenario
using (min# of) Duplications/Losses(/Transfers)

Gene/Species Trees Reconciliation



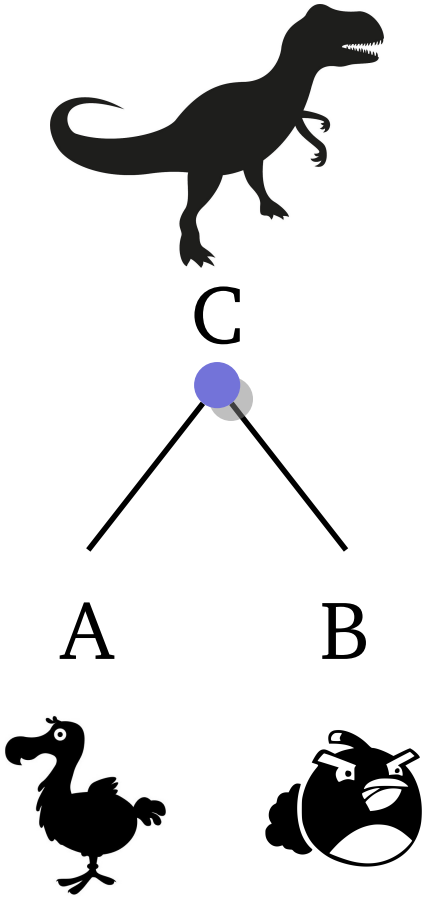


Find (parsimonious) evolutionary scenario using (min# of) Duplications/Losses(/Transfers)



Goal: To infer features of ancestral genomes

Inferring ancestral characters



Goal: To infer features of ancestral genomes

Inferring ancestral syntenic characters

- Efficient algorithms exist for parsimonious scenarios for genomic characters (genes under DTL, for example).

Inferring ancestral syntenic characters

- Efficient algorithms exist for parsimonious scenarios for genomic characters (genes under DTL, for example).
- Bérard *et al* (2012) extended methods to gene adjacencies under gene Speciation, Duplication, Loss.
- `DeCo` considers cost of gains/ breaks due to rearrangements.

Inferring ancestral syntenic characters

- Efficient algorithms exist for parsimonious scenarios for genomic characters (genes under DTL, for example).
- Bérard *et al* (2012) extended methods to gene adjacencies under gene Speciation, Duplication, Loss.
- `DeCo` considers cost of gains/ breaks due to rearrangements.

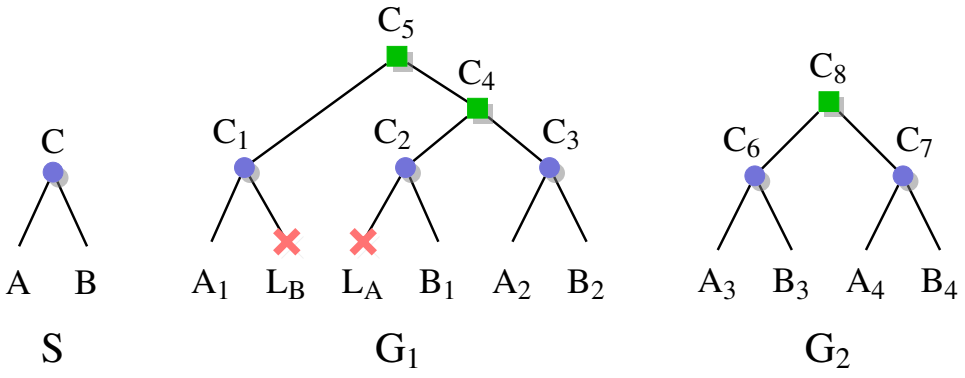
Robustness of `DeCo` predictions

- Zanetti *et al* (2014): Boltzmann-Gibbs probability as support;
- Rajaraman *et al* (2015): Parameter dependence of predictions.

DeCo : Parsimoniously inferring ancestral adjacencies

Problem

Given two reconciled gene trees + extant adjacencies, construct a parsimonious evolutionary scenario

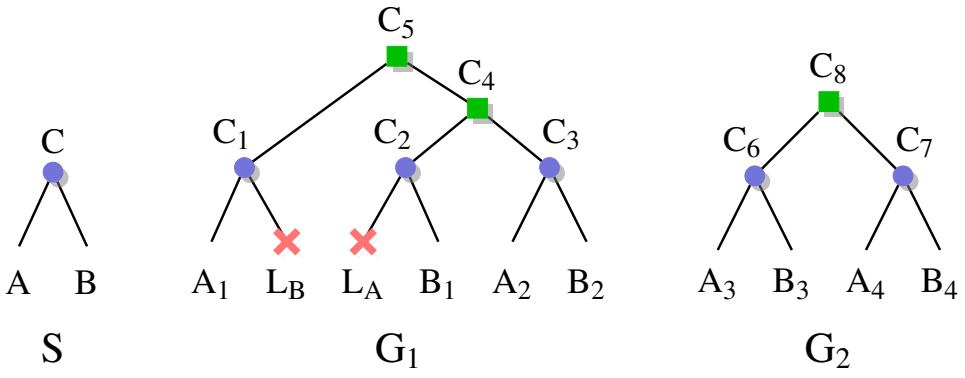


+ Adj. $\{A_1A_2, B_1B_3, B_2B_4\}$

DeCo : Parsimoniously inferring ancestral adjacencies

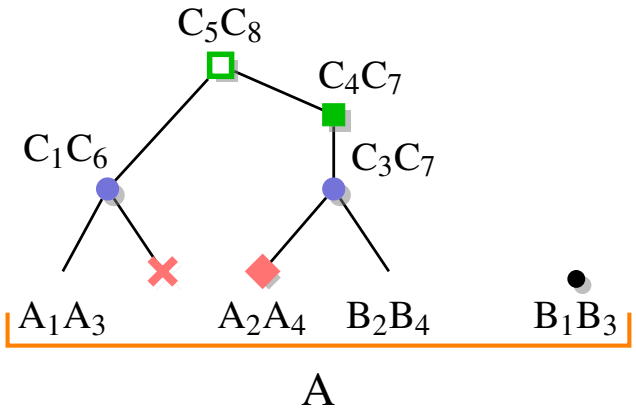
Problem

Given two reconciled gene trees + extant adjacencies, construct a parsimonious evolutionary scenario



+ Adj. $\{A_1A_2, B_1B_3, B_2B_4\}$

Output: Adjacency forest



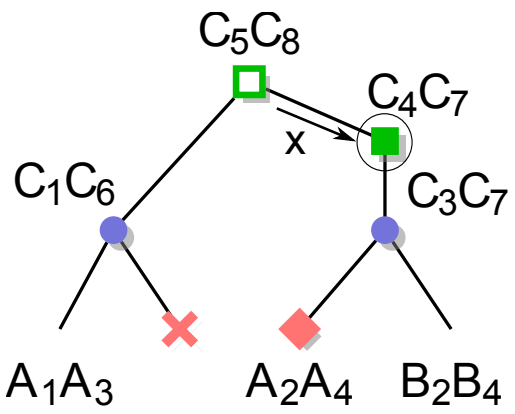
+ Ancestral genome architecture

Ancestral adjacency events

Event costs/parameters

- Adjacency gain: x
- Adjacency break: y

W.l.o.g., $x^2 + y^2 = 1$.

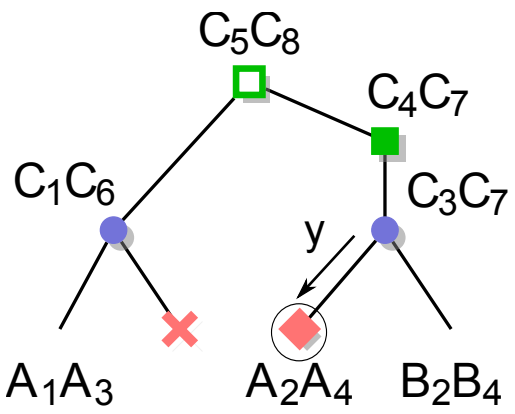


Ancestral adjacency events

Event costs/parameters

- Adjacency gain: x
- Adjacency break: y

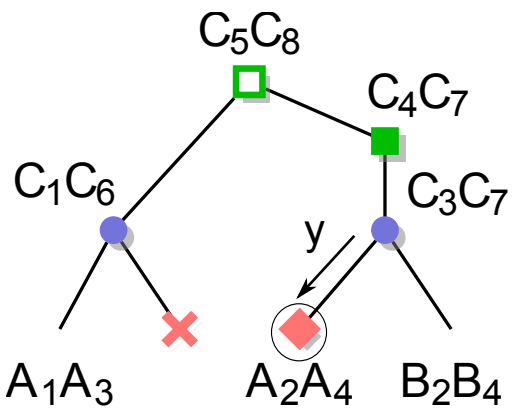
W.l.o.g., $x^2 + y^2 = 1$.



Event costs/parameters

- Adjacency gain: x
- Adjacency break: y

W.l.o.g., $x^2 + y^2 = 1$.



Signature of a scenario

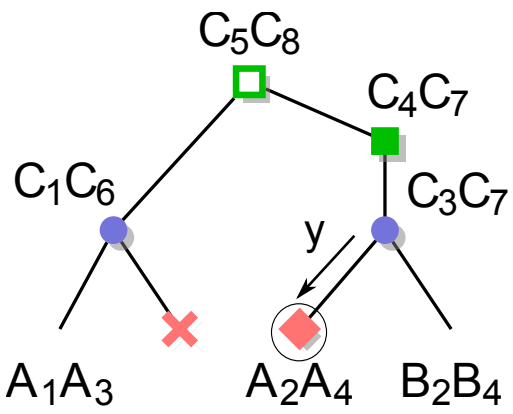
(g, b): Number of gains(g) and breaks(b).

$$Cost\left((g, b)\right) = x \cdot g + y \cdot b.$$

Event costs/parameters

- Adjacency gain: x
- Adjacency break: y

W.l.o.g., $x^2 + y^2 = 1$.



Signature of a scenario

(g, b): Number of gains(g) and breaks(b).

$$Cost\left((g, b)\right) = x \cdot g + y \cdot b.$$

Extended signatures for a given adjacency

(g, b, d): $d = 1$ if adjacency present, 0 otherwise.

$$Cost\left((g, b, d)\right) = x \cdot g + y \cdot b + 0 \cdot d.$$

A peek (Bérard et al., 2012)

1. $e(g_1) = \mathsf{Extant}$ and $e(g_2) = \mathsf{Extant}$:

$$c_1(g_1, g_2) = \begin{cases} 0 & \text{if } g_1 g_2 \text{ is an adjacency} \\ 0 & \text{otherwise} \end{cases}$$

$$c_0(g_1, g_2) = \begin{cases} 0 & \text{if } g_1 g_2 \text{ is not an adjacency} \\ 0 & \text{otherwise} \end{cases}$$

2. $e(g_1) = \mathsf{GLoss}$ and $e(g_2) \in \{\mathsf{Extant}, \mathsf{Spec}, \mathsf{GDup}\}$:

$$c_1(g_1, g_2) = c_0(g_1, g_2) = 0^{\#\mathsf{NonGDup}(g_2)}$$

3. $e(g_1) \in \{\mathsf{Extant}, \mathsf{Spec}, \mathsf{GDup}\}$ and $e(g_2) = \mathsf{GLoss}$:

$$c_1(g_1, g_2) = c_0(g_1, g_2) = 0^{\#\mathsf{NonGDup}(g_1)}$$

4. $e(g_1) = \mathsf{GLoss}$ and $e(g_2) = \mathsf{GLoss}$:

$$c_1(g_1, g_2) = c_0(g_1, g_2) = 0$$

5. $e(g_1) \in \{\mathsf{Extant}, \mathsf{Spec}\}$ and $e(g_2) = \mathsf{GDup}$:

$$c_1(g_1, g_2) = \min \begin{cases} c_1(g_1, b_{g_2}) + c_0(g_1, a_{g_2}), & c_0(g_1, b_{g_2}) + c_1(g_1, a_{g_2}), \\ c_1(g_1, b_{g_2}) + c_1(g_1, a_{g_2}) + x, & c_0(g_1, b_{g_2}) + c_0(g_1, a_{g_2}) + y \end{cases}$$

$$c_0(g_1, g_2) = \min \begin{cases} c_0(g_1, b_{g_2}) + c_0(g_1, a_{g_2}), & c_0(g_1, b_{g_2}) + c_1(g_1, a_{g_2}) + x, \\ c_1(g_1, b_{g_2}) + c_0(g_1, a_{g_2}) + x, & c_1(g_1, b_{g_2}) + c_1(g_1, a_{g_2}) + 2x \end{cases}$$

6. $e(g_1) = \mathsf{GDup}$ and $e(g_2) \in \{\mathsf{Extant}, \mathsf{Spec}\}$:

$$c_1(g_1, g_2) = \min \begin{cases} c_1(a_{g_1}, g_2) + c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) + c_1(b_{g_1}, g_2), \\ c_1(a_{g_1}, g_2) + c_1(b_{g_1}, g_2) + x, & c_0(a_{g_1}, g_2) + c_0(b_{g_1}, g_2) + y \end{cases}$$

$$c_0(g_1, g_2) = \min \begin{cases} c_0(a_{g_1}, g_2) + c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) + c_1(b_{g_1}, g_2) + x, \\ c_1(a_{g_1}, g_2) + c_0(b_{g_1}, g_2) + x, & c_1(a_{g_1}, g_2) + c_1(b_{g_1}, g_2) + 2x, \end{cases}$$

7. $e(g_1) = \mathsf{Spec}$ and $e(g_2) = \mathsf{Spec}$:

$$c_1(g_1, g_2) = \min \begin{cases} c_1(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + 0, & c_1(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + y + 0, \\ c_0(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + y + 0, & c_0(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + 2y + 0, \\ c_1(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + 0, & c_1(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + y + 0, \\ c_0(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + y + 0, & c_0(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + 2y + 0 \end{cases}$$

$$c_0(g_1, g_2) = \min \begin{cases} c_0(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + 0, & c_1(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + x + 0, \\ c_0(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + x + 0, & c_1(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + 2x + 0, \\ c_0(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + 0, & c_1(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + x + 0, \\ c_0(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + x + 0, & c_1(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + 2x + 0, \end{cases}$$

8. $e(g_1) = \mathsf{GDup}$ and $e(g_2) = \mathsf{GDup}$:

$$c_1(g_1, g_2) = \min \begin{cases} c_1(a_{g_1}, g_2) + c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) + c_1(b_{g_1}, g_2), \\ c_1(a_{g_1}, g_2) + c_1(b_{g_1}, g_2) + x, & c_0(a_{g_1}, g_2) + c_0(b_{g_1}, g_2) + y, \\ c_1(g_1, a_{g_2}) + c_0(g_1, b_{g_2}), & c_0(g_1, a_{g_2}) + c_1(g_1, b_{g_2}), \\ c_1(g_1, a_{g_2}) + c_1(g_1, b_{g_2}) + x, & c_0(g_1, a_{g_2}) + c_0(g_1, b_{g_2}) + y, \\ c_1(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}), \\ c_1(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + x, \\ c_1(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + x, \\ c_1(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + 2x, \\ c_1(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + y, \\ c_1(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + x + y, \\ c_1(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + x + y, \\ c_0(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + y, \\ c_0(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + x + y, \\ c_0(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + x + y, \\ c_0(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}), \\ c_0(a_{g_1}, a_{g_2}) + c_1(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + x, \\ c_1(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + x, \\ c_0(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_1(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + y, \\ c_0(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_1(b_{g_1}, a_{g_2}) + y, \\ c_0(a_{g_1}, a_{g_2}) + c_0(b_{g_1}, b_{g_2}) + c_0(a_{g_1}, b_{g_2}) + c_0(b_{g_1}, a_{g_2}) + 2y, \end{cases}$$

$$c_0(g_1, g_2) = \min \begin{cases} c_0(a_{g_1}, g_2) + c_0(b_{g_1}, g_2), & c_0(a_{g_1}, g_2) + c_1(b_{g_1}, g_2) + x, \\ c_1(a_{g_1}, g_2) + c_0(b_{g_1}, g_2) + x, & c_1(a_{g_1}, g_2) + c_1(b_{g_1}, g_2) + 2x, \\ c_0(g_1, a_{g_2}) + c_0(g_1, b_{g_2}), & c_0(g_1, a_{g_2}) + c_1(g_1, b_{g_2}) + x, \\ c_1(g_1, a_{g_2}) + c_0(g_1, b_{g_2}) + x, & c_1(g_1, a_{g_2}) + c_1(g_1, b_{g_2}) + 2x. \end{cases}$$

`Deco` (Berard *et al* (2012)): unambiguous dynamic programming scheme for the most parsimonious adjacency forest in $\mathcal{O}(|G_1||G_2|)$.

A peek (Bérard et al., 2012)

.
.
.

7. If $e(g_1) = \text{Spec}$ and $e(g_2) = \text{Spec}$:

$$c_1(g_1, g_2) = \min \begin{cases} c_1(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)),\ c_1(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)) + \text{ABreak}, \\ c_0(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)) + \text{ABreak}, \\ c_0(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)) + 2\text{ABreak}, \\ c_1(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)),\ c_1(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)) + \text{ABreak}, \\ c_0(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)) + \text{ABreak},\ c_0(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)) + 2\text{ABreak} \end{cases}$$

$$c_0(g_1, g_2) = \min \begin{cases} c_0(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)),\ c_1(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)) + \text{AGain}, \\ c_0(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)) + \text{AGain},\ c_1(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)) + 2\text{AGain}, \\ c_0(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)),\ c_1(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)) + \text{AGain}, \\ c_0(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)) + \text{AGain},\ c_1(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)) + 2\text{AGain} \end{cases}$$

8. If $e(g_1) = \text{GDup}$ and $e(g_2) = \text{GDup}$:

.
.
.

`Deco` (Berard *et al* (2012)): unambiguous dynamic programming scheme for the most parsimonious adjacency forest in $\mathcal{O}(|G_1||G_2|)$.

Why do we do exact optimization, again?

- For fun: Because, who doesn't just *love* algorithms?? :)

Why do we do exact optimization, again?

- For fun: Because, who doesn't just *love* algorithms?? :)
- For money: Operations research, network design...

Why do we do exact optimization, again?

- For fun: Because, who doesn't just *love* algorithms?? :)
- For money: Operations research, network design...
- For love (of exhaustivity): Real negative results (vs heuristics)

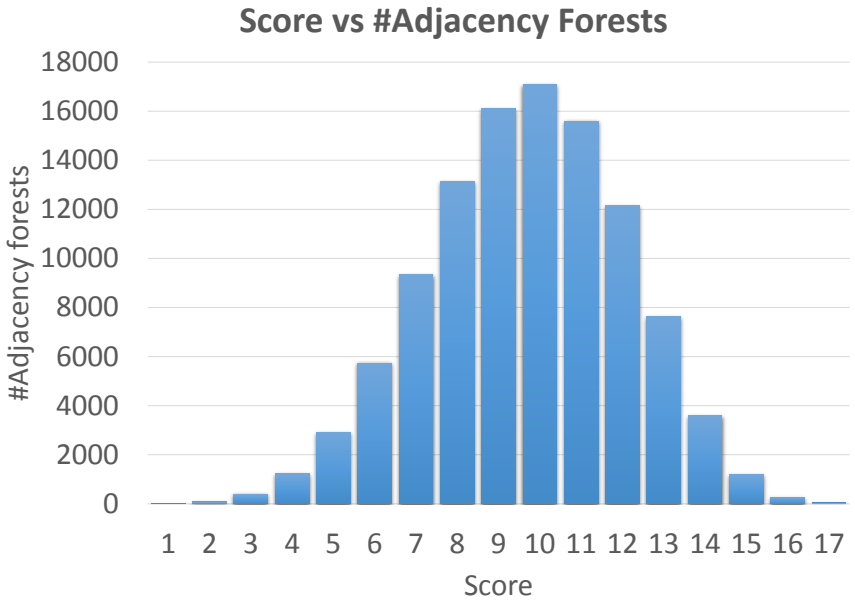
Why do we do exact optimization, again?

- For fun: Because, who doesn't just *love* algorithms?? :)
- For money: Operations research, network design...
- For love (of exhaustivity): Real negative results (vs heuristics)
- To predict the unobservable: RNA folding, phylogeny...

Why do we do exact optimization, again?

- For fun: Because, who doesn't just *love* algorithms?? :)
- For money: Operations research, network design...
- For love (of exhaustivity): Real negative results (vs heuristics)
- To predict the unobservable: RNA folding, phylogeny...

But optimal solution may be isolated/poorly representative

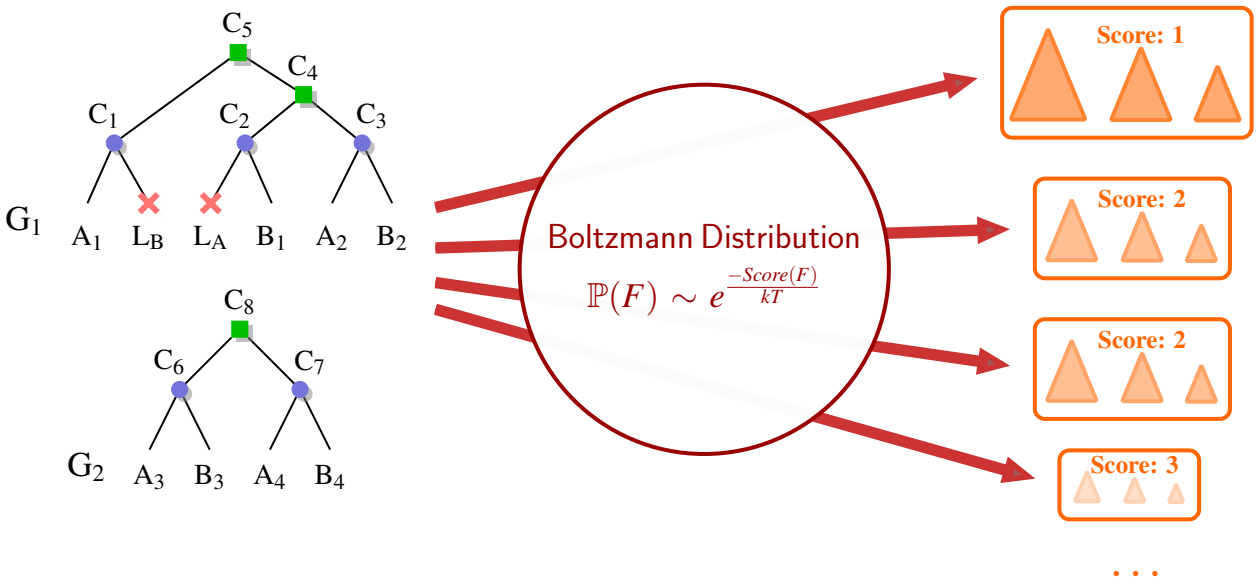


$\Rightarrow$ How to capture such a diversity?

Part I. A Boltzmann-Gibbs framework to correct syntenic conflicts

[Chauve-P-Zanetti, BMC Bioinformatics 2015]

Boltzmann-Gibbs distribution



Analysis

Average properties

Ancestral adjacencies

Synthenic inconsistencies

Recurrent patterns ...

Max. expected accuracy predictions

Robustness assessment

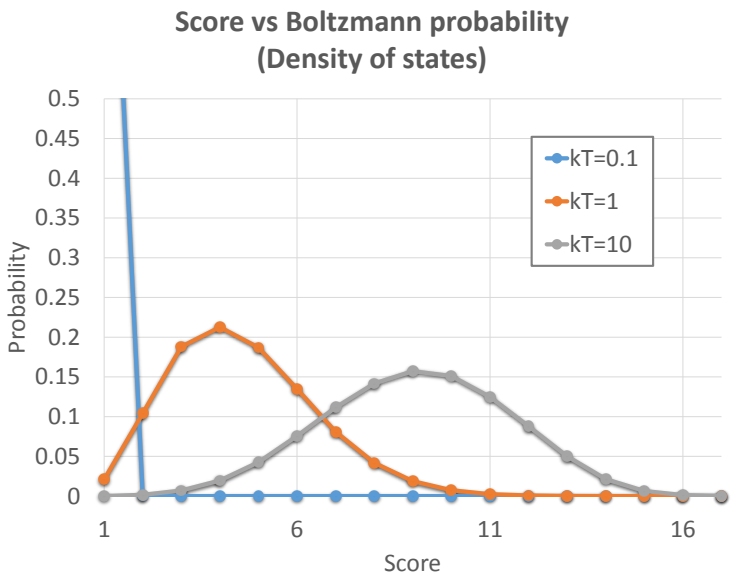
...

Bolzmann-Gibbs prob. of forest F

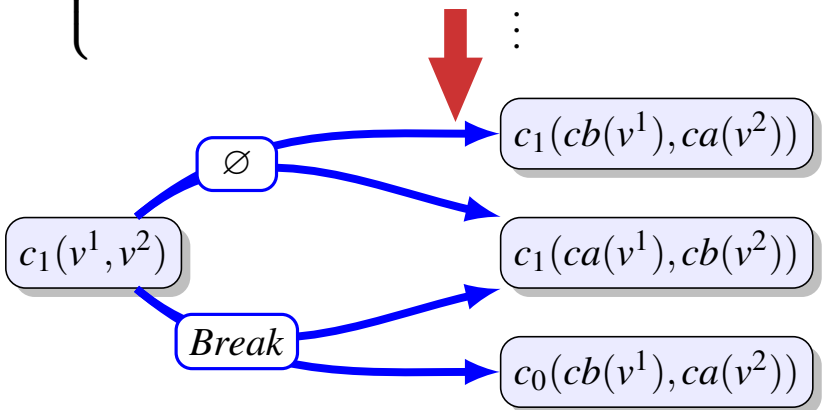
$$\mathbb{P}(F) = \frac{e^{-\frac{Score(F)}{kT}}}{\mathcal{Z}} \quad \mathcal{Z} := \sum_{F'} e^{-\frac{Score(F)}{kT}}$$

kT: Pseudo-temperature

(how *deep* within subopts should you look?)



$$c_1(v^1,v^2) = \min \begin{cases} c_1(ca(v^1),cb(v^2)) + c_1(cb(v^1),ca(v^2)) \\ c_1(ca(v^1),cb(v^2)) + c_0(cb(v^1),ca(v^2)) + Break \\ \vdots \end{cases}$$

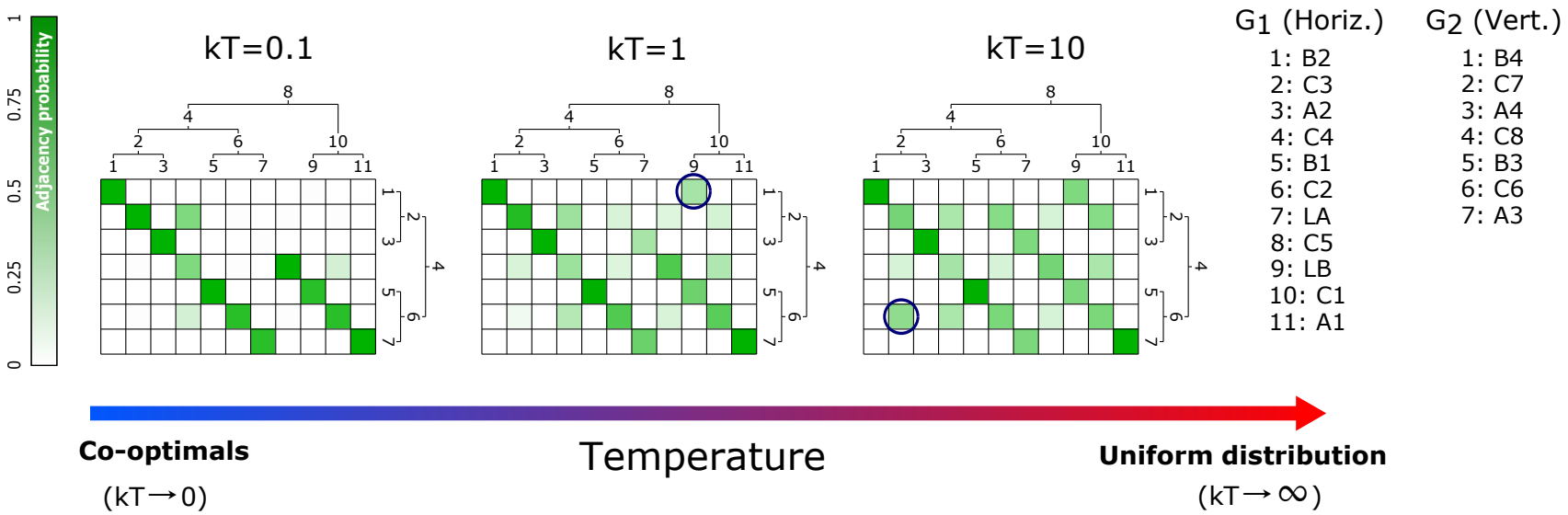


Remark: The DeCo DP scheme is unambiguous and complete

Automatic algorithmic corollaries

- Partition function $\mathcal{Z}$ in $\mathcal{O}(|G_1||G_2|)$: $(\min, +, C) \to (\sum, \times, e^{-\frac{C}{kT}})$
- Sampling: Chose transition w.p. prop. to its contribution to $\mathcal{Z}$.
- Adjacency dot-plot: $\mathcal{O}(|G_1||G_2|)$ inside/outside algorithm.

Ancestral adjacency probabilities



Remarks

- Boltzmann-Gibbs: Tradeoff between scoring and combinatorics;
- Pseudo-temperature kT: Restricted to cooptimals ($kT \to 0$) up to the fully uniform distribution ($kT \to +\infty$);
- Adjacencies probabilities not necessarily monotonous on kT;
- Syntenic conflicts: >2 neighbors predicted in genomes.

Boltzmann prob. vs syntenic conflicts

Dataset: 6,074 `Deco` instances from 36 genomes (Ensembl 2012)

Parsimony: syntenic inconsistencies for 5,817/112,188 genes

(Selecting subset of adjacencies while removing syntenic issues = NP-Hard)

Proba.	Ancestral genes			Ancestral adjacencies			Syntenic conflicts		
Cutoff	kT=0.5	kT=0.1	kT=0.01	kT=0.5	kT=0.1	kT=0.01	kT=0.5	kT=0.1	kT=0.01
≥ 0.1	122,593	120,495	120,531	137,133	116,671	116,873	31,415	14,655	14,863
≥ 0.2	121,896	119,792	119,864	130,928	113,835	114,109	26,415	12,631	12,871
≥ 0.3	120,869	118,871	118,913	122,073	110,472	110,676	19,729	10,259	10,471
≥ 0.4	118,240	117,883	117,990	111,224	107,492	107,845	11,569	8,439	8,738
≥ 0.5	113,023	116,476	116,810	100,265	103,963	105,015	5,658	6,578	7,373
≥ 0.6	104,970	114,699	114,902	88,943	100,064	100,511	3,044	5,081	5,387
≥ 0.7	92,647	112,834	112,924	75,063	96,546	96,789	1,363	4,104	4,301
≥ 0.8	75,408	110,272	110,503	58,665	92,181	92,621	492	3,290	3,496
≥ 0.9	45,753	107,769	107,863	34,005	88,132	88,373	65	2,717	2,886
$= 1$	13	16,470	106,903	7	10,798	87,048	0	0	2,690

$\Rightarrow$ Boltzmann-Gibbs provides a tractable notion of support that may be used to correct syntenic conflicts

Boltzmann prob. vs syntenic conflicts

Dataset: 6,074 `Deco` instances from 36 genomes (Ensembl 2012)

Parsimony: syntenic inconsistencies for 5,817/112,188 genes

(Selecting subset of adjacencies while removing syntenic issues = NP-Hard)

Proba.	Ancestral genes			Ancestral adjacencies			Syntenic conflicts		
Cutoff	kT=0.5	kT=0.1	kT=0.01	kT=0.5	kT=0.1	kT=0.01	kT=0.5	kT=0.1	kT=0.01
≥ 0.1	122,593	120,495	120,531	137,133	116,671	116,873	31,415	14,655	14,863
≥ 0.2	121,896	119,792	119,864	130,928	113,835	114,109	26,415	12,631	12,871
≥ 0.3	120,869	118,871	118,913	122,073	110,472	110,676	19,729	10,259	10,471
≥ 0.4	118,240	117,883	117,990	111,224	107,492	107,845	11,569	8,439	8,738
≥ 0.5	113,023	116,476	116,810	100,265	103,963	105,015	5,658	6,578	7,373
≥ 0.6	104,970	114,699	114,902	88,943	100,064	100,511	3,044	5,081	5,387
≥ 0.7	92,647	112,834	112,924	75,063	96,546	96,789	1,363	4,104	4,301
≥ 0.8	75,408	110,272	110,503	58,665	92,181	92,621	492	3,290	3,496
≥ 0.9	45,753	107,769	107,863	34,005	88,132	88,373	65	2,717	2,886
$= 1$	13	16,470	106,903	7	10,798	87,048	0	0	2,690

$\Rightarrow$ Boltzmann-Gibbs provides a tractable notion of support that may be used to correct syntenic conflicts

Boltzmann prob. vs syntenic conflicts

Dataset: 6,074 `Deco` instances from 36 genomes (Ensembl 2012)

Parsimony: syntenic inconsistencies for 5,817/112,188 genes

(Selecting subset of adjacencies while removing syntenic issues = NP-Hard)

Proba.	Ancestral genes			Ancestral adjacencies			Syntenic conflicts		
Cutoff	kT=0.5	kT=0.1	kT=0.01	kT=0.5	kT=0.1	kT=0.01	kT=0.5	kT=0.1	kT=0.01
≥ 0.1	122,593	120,495	120,531	137,133	116,671	116,873	31,415	14,655	14,863
≥ 0.2	121,896	119,792	119,864	130,928	113,835	114,109	26,415	12,631	12,871
≥ 0.3	120,869	118,871	118,913	122,073	110,472	110,676	19,729	10,259	10,471
≥ 0.4	118,240	117,883	117,990	111,224	107,492	107,845	11,569	8,439	8,738
≥ 0.5	113,023	116,476	116,810	100,265	103,963	105,015	5,658	6,578	7,373
≥ 0.6	104,970	114,699	114,902	88,943	100,064	100,511	3,044	5,081	5,387
≥ 0.7	92,647	112,834	112,924	75,063	96,546	96,789	1,363	4,104	4,301
≥ 0.8	75,408	110,272	110,503	58,665	92,181	92,621	492	3,290	3,496
≥ 0.9	45,753	107,769	107,863	34,005	88,132	88,373	65	2,717	2,886
$= 1$	13	16,470	106,903	7	10,798	87,048	0	0	2,690

$\Rightarrow$ Boltzmann-Gibbs provides a tractable notion of support that may be used to correct syntenic conflicts

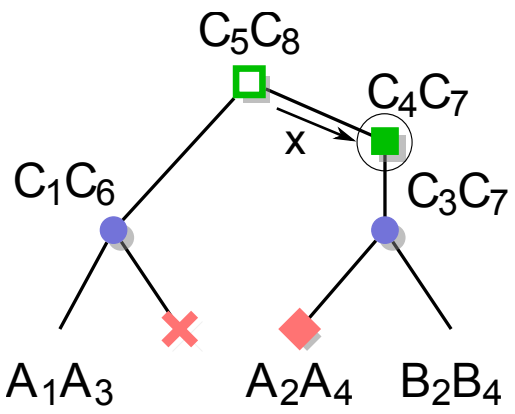
Part II. Testing the robustness of parsimony predictions to uncertain event costs

[Rajaraman-Chauve-P, ISBRA'15]

Event costs/parameters

- Adjacency gain: x
- Adjacency break: y

W.l.o.g., $x^2 + y^2 = 1$.



Signature of a scenario

(g, b): Number of gains(g) and breaks(b).

$$Cost\left((g, b)\right) = x \cdot g + y \cdot b.$$

Extended signatures for a given adjacency

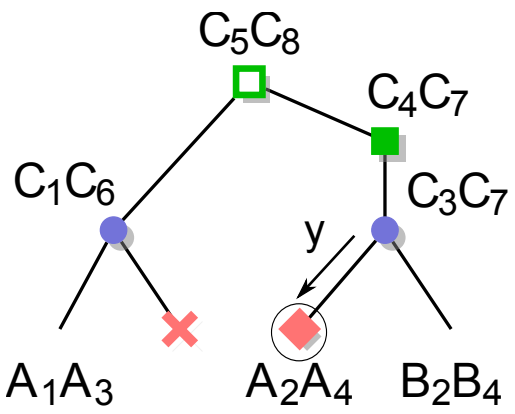
(g, b, d): $d = 1$ if adjacency present, 0 otherwise.

$$Cost\left((g, b, d)\right) = x \cdot g + y \cdot b + 0 \cdot d.$$

Event costs/parameters

- Adjacency gain: x
- Adjacency break: y

W.l.o.g., $x^2 + y^2 = 1$.



Signature of a scenario

(g, b): Number of gains(g) and breaks(b).

$$Cost\left((g, b)\right) = x \cdot g + y \cdot b.$$

Extended signatures for a given adjacency

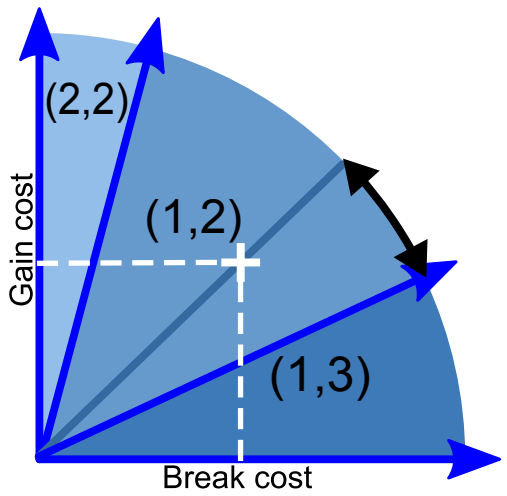
(g, b, d): $d = 1$ if adjacency present, 0 otherwise.

$$Cost\left((g, b, d)\right) = x \cdot g + y \cdot b + 0 \cdot d.$$

Unit-costs for evolutionary events: How *bad* can they be?

Signature robustness

How stable is the parsimonious signature at cost (x, y)?



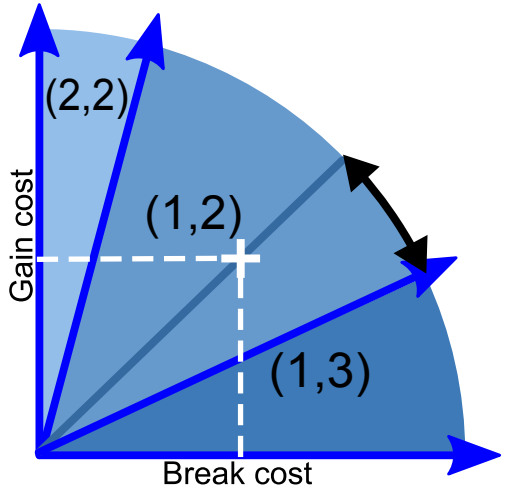
Unit-costs for evolutionary events: How *bad* can they be?

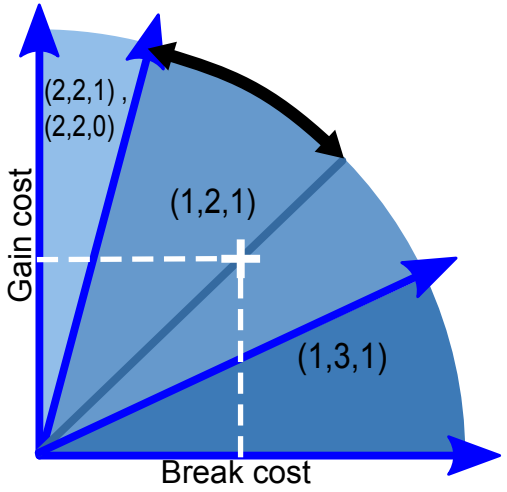
Signature robustness

How stable is the parsimonious signature at cost (x, y)?

Adjacency robustness

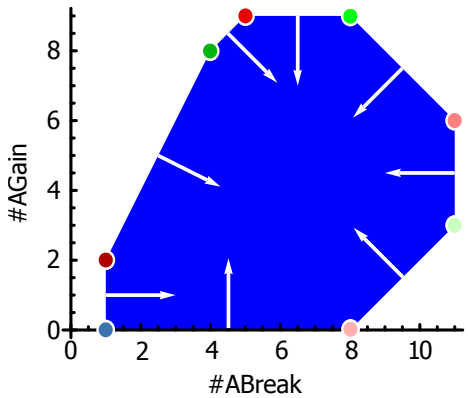
How stable is an adjacency predicted in a parsimonious scenario at cost (x, y)?





Convex polytopes and parameter partitions

Sturmfels,Pachter (2004)

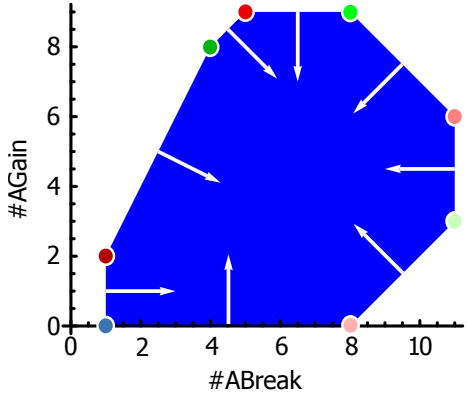


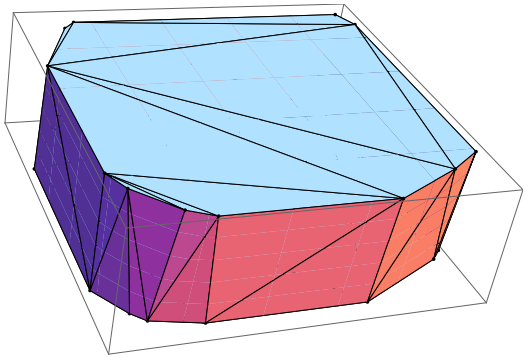
Vertices of convex polytope of all signatures	Unique optima for a cost scheme
Cone of a vertex (signature)	Converging cost schemes

Cones partition the parameter space.

Convex polytopes and parameter partitions

Sturmfels,Pachter (2004)





Vertices of convex polytope of all signatures	Unique optima for a cost scheme
Cone of a vertex (signature)	Converging cost schemes

Cones partition the parameter space.

Tracking non-scoring parameters

Construct the polytope on extended signatures (g, b, d).

Intersection of cones with $z = 0$ gives parameter space partition.

A peek (Bérard et al., 2012)

.
.
.

7. If $e(g_1) = \text{Spec}$ and $e(g_2) = \text{Spec}$:

$$c_1(g_1, g_2) = \min \begin{cases} c_1(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)),\ c_1(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)) + \text{ABreak}, \\ c_0(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)) + \text{ABreak}, \\ c_0(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)) + 2\text{ABreak}, \\ c_1(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)),\ c_1(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)) + \text{ABreak}, \\ c_0(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)) + \text{ABreak},\ c_0(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)) + 2\text{ABreak} \end{cases}$$

$$c_0(g_1, g_2) = \min \begin{cases} c_0(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)),\ c_1(a(g_1), b(g_2)) + c_0(b(g_1), a(g_2)) + \text{AGain}, \\ c_0(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)) + \text{AGain},\ c_1(a(g_1), b(g_2)) + c_1(b(g_1), a(g_2)) + 2\text{AGain}, \\ c_0(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)),\ c_1(a(g_1), a(g_2)) + c_0(b(g_1), b(g_2)) + \text{AGain}, \\ c_0(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)) + \text{AGain},\ c_1(a(g_1), a(g_2)) + c_1(b(g_1), b(g_2)) + 2\text{AGain} \end{cases}$$

8. If $e(g_1) = \text{GDup}$ and $e(g_2) = \text{GDup}$:

.
.
.

$(\min, +)$ algebra.

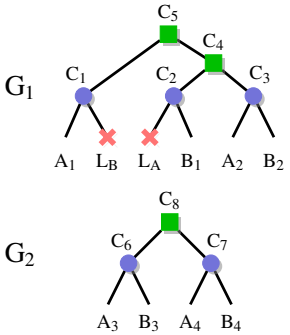
Translating the DP scheme

DeCo	2D	3D
Gain cost	$(1, 0)$	$(1, 0, 0)$
Break cost	$(0, 1)$	$(0, 1, 0)$
Adjacency	-	$(0, 0, 1)$
0	$(0, 0)$	$(0, 0, 0)$
∞	Empty polytope	
$+$	Minkowski sum	
min	Convex hull	

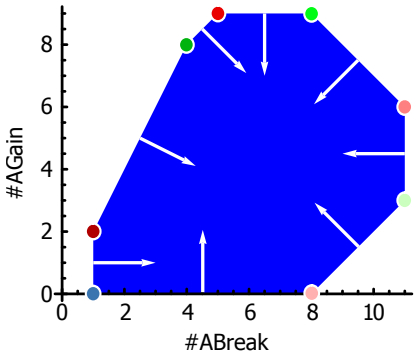
Compute cones using elementary linear algebra.

Complexity: $\tilde{O}\left(N^{d-1}\right)$ time, $O\left(N^{d-1}\right)$ space.
$N = n_1 \times n_2 =$ size of DP table, $d =$ dimension.

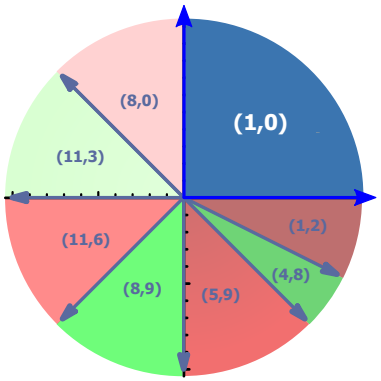
Data and Experiments



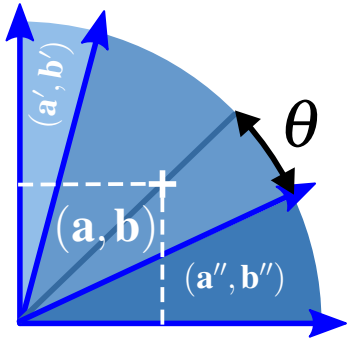
a. Input: Gene trees + Extant adjacencies



b. Polytope computation



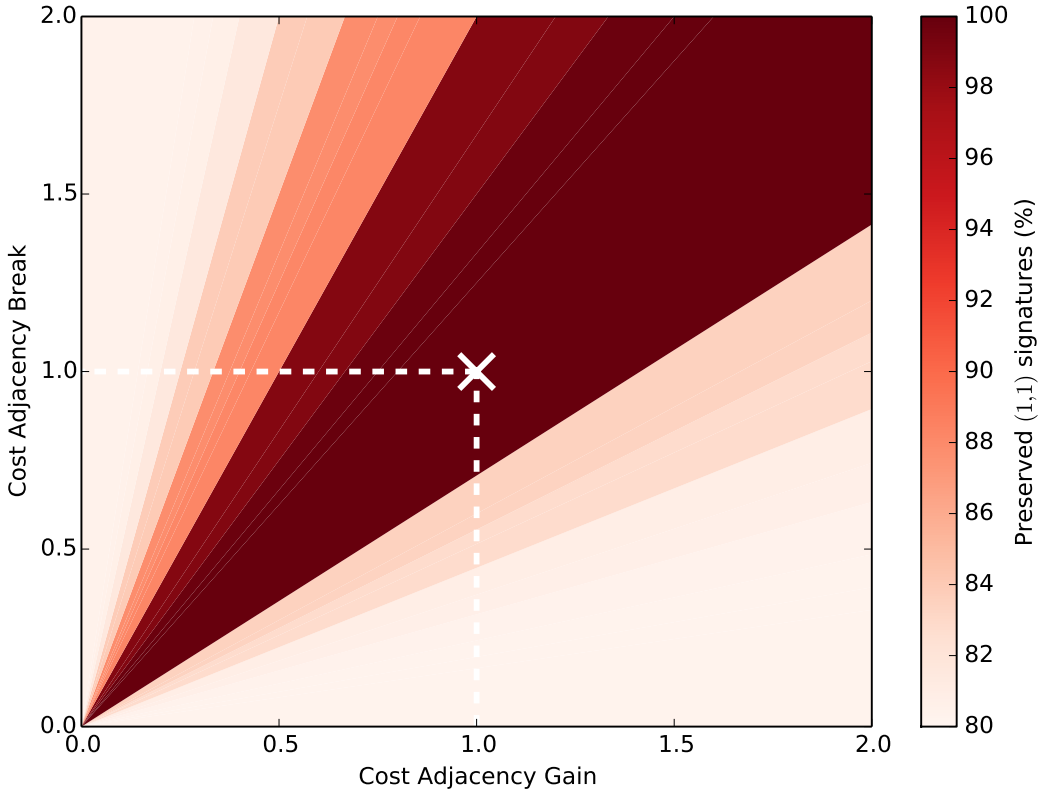
c. Cone segmentation



d. Robustness analysis

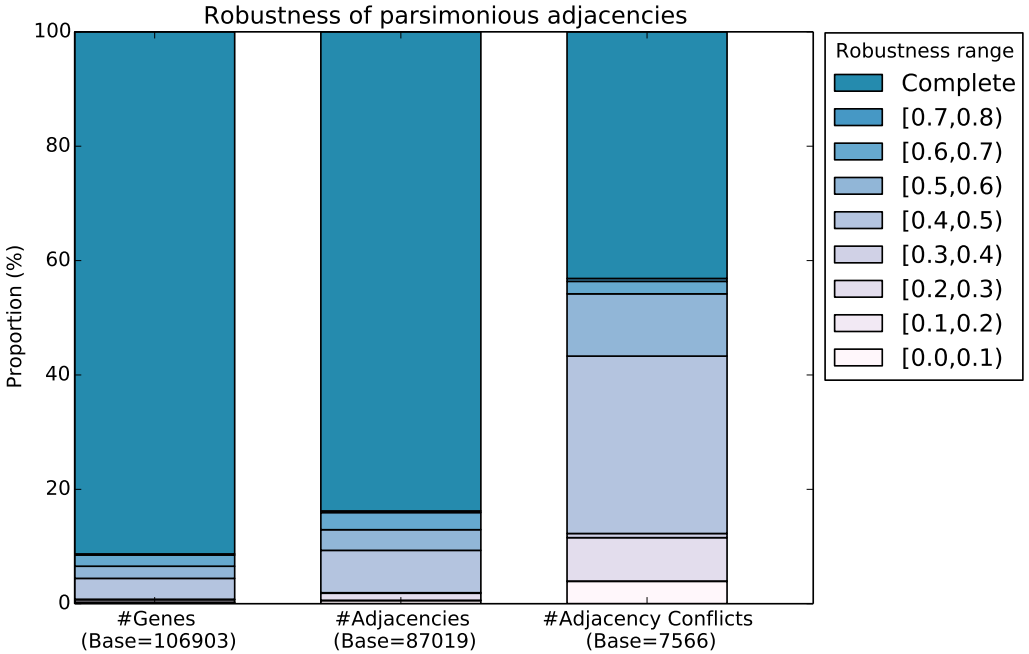
- 50,389 extant mammalian adjacencies over 36 species.
- 6,074 input instances, using 5,039 Ensembl gene trees.
- Default DeCo results: 96,482 ancestral adjacencies predicted.
- 16,039 universally parsimonious adjacencies have a gene involved in > 2 adjacencies (*conflict*).

Results: Signature robustness



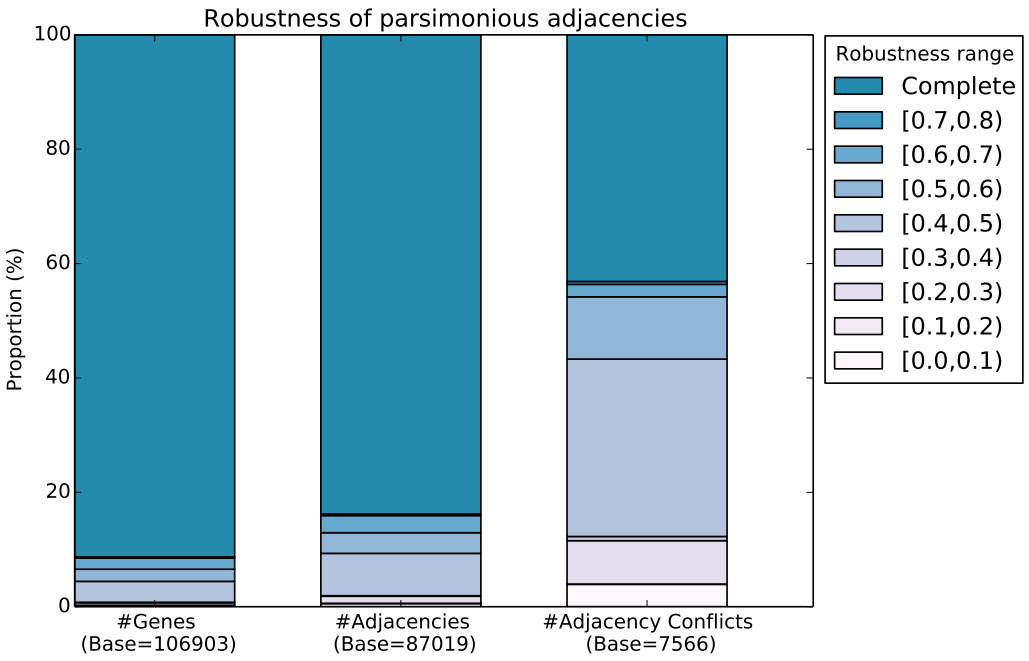
- $> 50\%$ of instances are fully robust.
- 975 instances not robust to any change.
- $\sim 80\%$ of instances parsimonious at some (x, y) cost scheme are also parsimonious at $(1, 1)$.

Results: Adjacency robustness



- $\sim$ 85% of ancestral adjacencies are fully robust, but ...

Results: Adjacency robustness



- $\sim$ 85% of ancestral adjacencies are fully robust, but ...
- $\sim$ 45% of conflicting parsimonious adjacencies are fully robust.

Conclusion

Inferences

- The `DeCo` model is robust, i.e. $\pm$ independent of parameters;
- Predicted numbers of adjacency gain/ break events are very low, consistent with rarity of genome rearrangement events;
- Number of syntenic conflicts remains high; may be an artefact of errors in gene tree reconstruction or reconciliation;
- Boltzmann probability discriminates erroneous predictions;
- Generic dynamic programming principles/applications.

Conclusion

Inferences

- The `DeCo` model is robust, i.e. $\pm$ independent of parameters;
- Predicted numbers of adjacency gain/ break events are very low, consistent with rarity of genome rearrangement events;
- Number of syntenic conflicts remains high; may be an artefact of errors in gene tree reconstruction or reconciliation;
- Boltzmann probability discriminates erroneous predictions;
- Generic dynamic programming principles/applications.

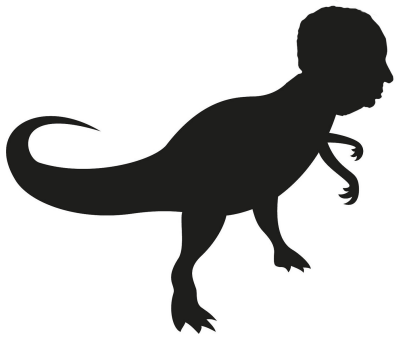
Closing points

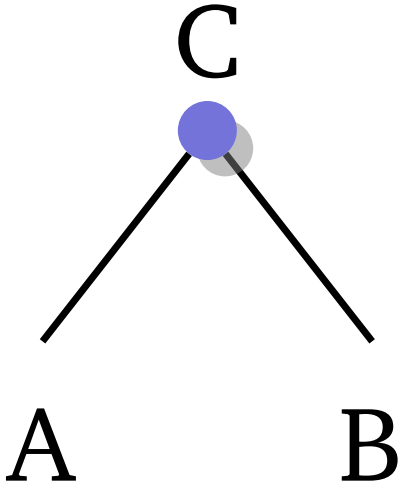
- Simultaneously explore robustness to > 1 ancestral adjacency;
- Explore relation to Pareto-optimality (Bansal et al. 2013);
- Add lateral transfers

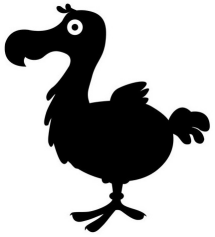
Acknowledgements

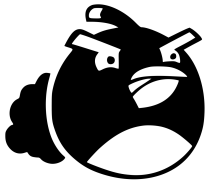
DeClone generalizes `DeCo` to other algebras:
`https://github.com/yannponty/DeClone`

Convex hull computations in DeClone rely on a `Quickhull` implementation by Anatoliy V. Tomilov at the Ural Federal University.









Thank you!