Decomposition trees of permutations, and how to use them for a (realistic ?) study of perfect sorting by reversals

Mathilde Bouvel
(Institut für Mathematik, Universität Zürich)

talk based on joint works and ongoing projects with
C. Chauve, M. Mishna, C. Nicaud, C. Pivoteau, D. Rossin

Pattern Avoidance and Genome Sorting at Dagstuhl, Feb. 2016

Perfect sorting by reversals: the problem

The model

- Genome or chromosome = sequence of genes (genes are oriented).
- Restricting to the set of common homologous genes of two species: Genome = a signed permutation (signs indicate orientation).

  W.l.o.g., the genome of one of the species is $12\dots n$.
- One type of evolutionary events only: reversals.
  The reversal of a fragment of a permutation reverses the order of the elements in that fragment while changing their signs.

Example: 1 -7 6 -10 9 -8 2 -11 -3 5 4

⇓ Reversal of the red fragment ⇓

1 -7 6 -10 9 -8 2 -4 -5 3 11

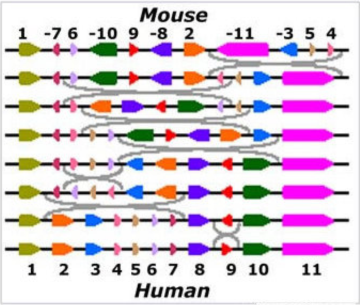
Sorting by reversals

The problem:

- INPUT: A signed permutation σ of size n.
- OUTPUT: A parsimonious scenario from σ to $1\,2 \dots$ or $-n \dots -2\ -1$.

Scenario = sequence of reversals.

Parsimonious = shortest, *i.e.* minimal number of reversals.



© Genome Research

The solution:

- Hannenhalli-Pevzner theory
- Polynomial algorithms: from $\mathcal{O}(n^4)$ to $\mathcal{O}(n\sqrt{n \log n})$

Remark: the problem is *NP*-hard when permutations are unsigned.

Perfect sorting by reversals (1/2)

Perfect sorting by reversals:
further requirement not to break any (common) interval.

Interval of σ =
fragment of σ whose (unsigned) elements form of range (in $\mathbb{N}$).
Example: $\sigma = 4$ -7 -5 6 3 -1 2.

Common interval between σ_1 and σ_2: set of integers such that both in σ_1 and σ_2, there is a fragment containing exactly these (unsigned) integers.
Example: $\sigma_1 = 5$ -1 -3 7 6 -2 4 and $\sigma_2 = 6$ -4 7 1 -3 2 -5

Why this restriction?
Groups of homologous genes appearing together in two species are likely to be

- together in the common ancestor;
- never separated during evolution.

Perfect sorting by reversals (2/2)

The problem:

- INPUT: A signed permutation σ of size n.
- OUTPUT: A parsimonious perfect scenario from σ to $1 2 \ldots$ or $-n \ldots -2\ -1$.

Parsimonious perfect scenario = scenario where reversals never break intervals, and which is shortest among all such scenarios.

Be careful!: Parsimonious perfect $\not\Rightarrow$ parsimonious.

Complexity: *NP*-hard problem [Figeac-Varré, '04].

Algorithm:
FPT algorithm of [Bérard-Bergeron-Chauve-Paul, '07] $\left(\text{in } 2^p \cdot n^{\mathcal{O}(1)}\right)$, representing permutations as trees.

Decomposition trees or strong interval trees

1. Strong interval trees
2. (Substitution) decomposition trees
3. Some applications in algorithms and combinatorics

Decomposition trees
or strong interval trees

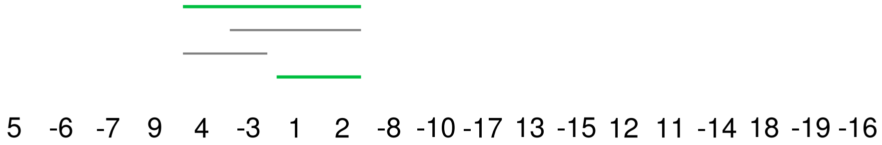
1. Strong interval trees
2. (Substitution) decomposition trees
3. Some applications in algorithms and combinatorics

Strong intervals

Strong interval of σ: one that does not overlap any other interval of σ.

Interval I is strong iff $\forall J$, $I \subseteq J$ *or* $J \subseteq I$ *or* $I \cap J = \emptyset$.

Example:



5 -6 -7 9 4 -3 1 2 -8 -10 -17 13 -15 12 11 -14 18 -19 -16

🟩 strong, ▬ overlapping

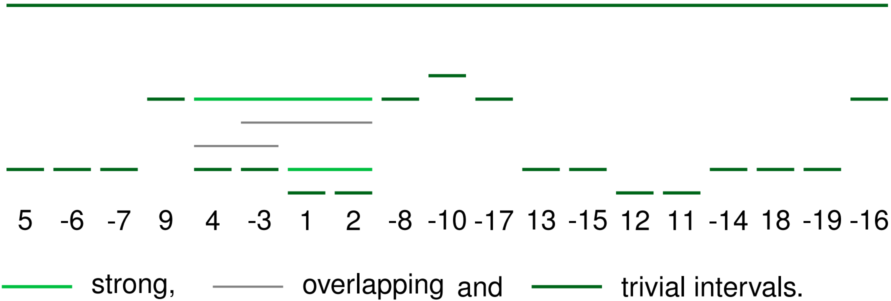
Strong intervals

Strong interval of σ: one that does not overlap any other interval of σ.

> *Interval I is strong iff $\forall J$, $I \subseteq J$ or $J \subseteq I$ or $I \cap J = \emptyset$.*

Remark: Trivial intervals (=singletons and whole set) are strong.

Example:



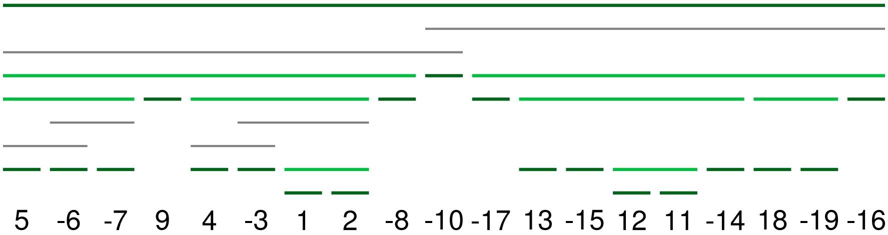
Strong intervals

Strong interval of σ: one that does not overlap any other interval of σ.

Interval I is strong iff $\forall J$, $I \subseteq J$ *or* $J \subseteq I$ *or* $I \cap J = \emptyset$.

Remark: Trivial intervals (=singletons and whole set) are strong.

Example:



— strong, — overlapping and — trivial intervals.

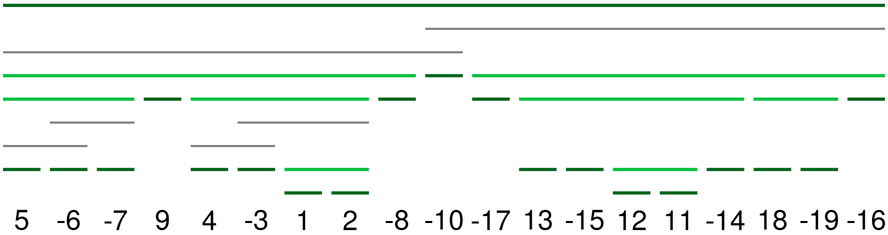
Strong intervals

Strong interval of σ: one that does not overlap any other interval of σ.

Interval I is strong iff $\forall J$, $I \subseteq J$ *or* $J \subseteq I$ *or* $I \cap J = \emptyset$.

Remark: Trivial intervals (=singletons and whole set) are strong.

Example:

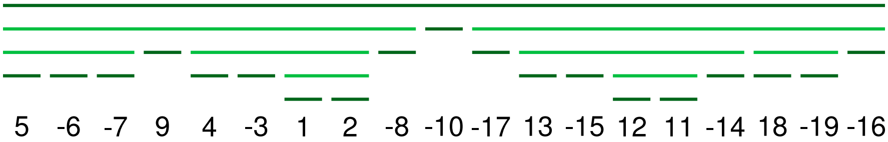


— strong, — overlapping and — trivial intervals.

Remark: Identical definition on signed and unsigned permutations.

Strong interval tree [Heber-Stoye, ’01](+ PQ-trees of [Booth-Lueker, ’76])

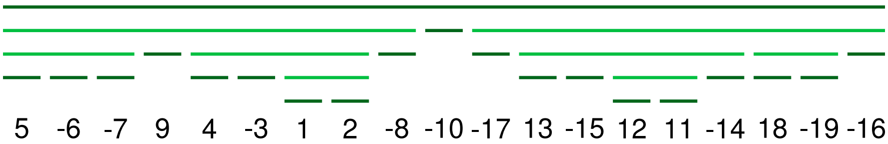
Example (continued):



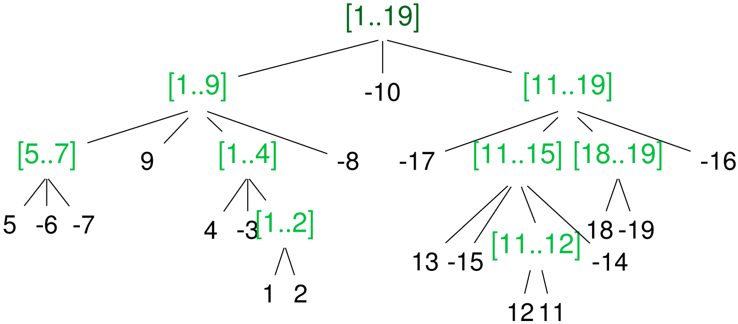
The inclusion order among strong intervals is a tree-like ordering.

Strong interval tree [Heber-Stoye, ’01](+ PQ-trees of [Booth-Lueker, ’76])

Example (continued):



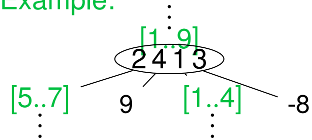
The inclusion order among strong intervals is a tree-like ordering.



Enriching strong interval trees

To every node, associate a quotient permutation = the order of the children. (**Remark:** children are intervals.)

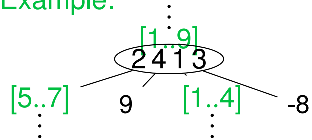
Example:



Enriching strong interval trees

To every node, associate a quotient permutation = the order of the children. (**Remark:** children are intervals.)

Example:



Two types of nodes:

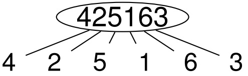
- Linear nodes (□):
  - increasing, *i.e.* quotient permutation = $1\ 2 \ldots k$;

  ⇒ label ⊞
  - decreasing, *i.e.* quotient permutation = $k\ (k-1) \ldots 2\ 1$;

  ⇒ label ⊟
- Prime nodes (◯): the quotient permutation is simple.

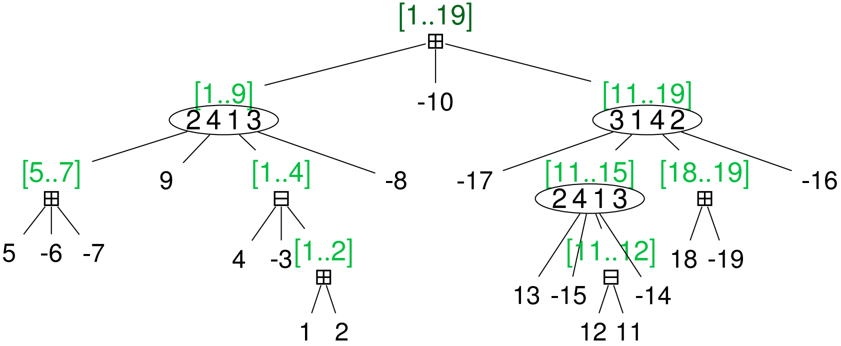
Simple permutations = the only intervals are the trivial ones: $\{1\}$, $\{2\}$,. . ., $\{n\}$ and $[1, \ldots, n]$.

Example: 425163, *i.e.*



Simplifying strong interval trees

In the full tree obtained, some information is redundant.

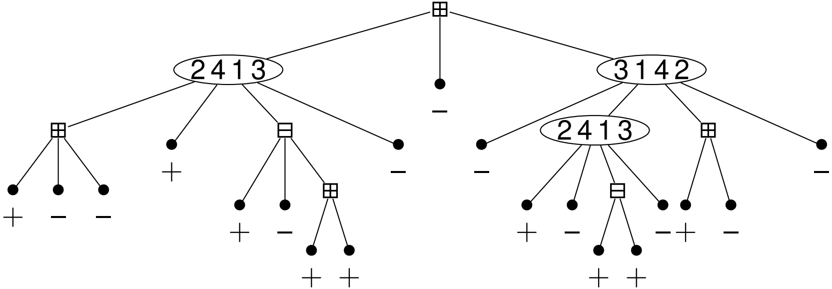


The full tree and the permutation can be recovered keeping only:

- the quotient permutations labeling the internal nodes;
- in the signed permutation case: the signs of the leaves.

The strong interval trees we want

We use the simplified version of the strong interval tree.



Remark: Strong interval trees (simplified or not) can be computed in linear time [Uno-Yagiura, '00] [Bergeron-Chauve-de Montgolfier-Raffinot, '08].

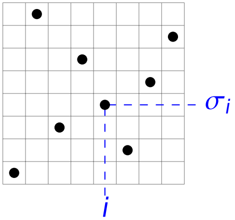
Decomposition trees
or strong interval trees

1. Strong interval trees
2. (Substitution) decomposition trees
3. Some applications in algorithms and combinatorics

Substitution in permutations

Easily explained on permutation diagrams.

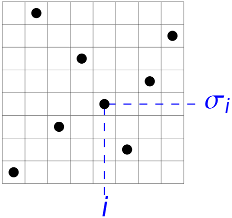
Example: $\sigma = 1\ 8\ 3\ 6\ 4\ 2\ 5\ 7 =$



Substitution in permutations

Easily explained on permutation diagrams.

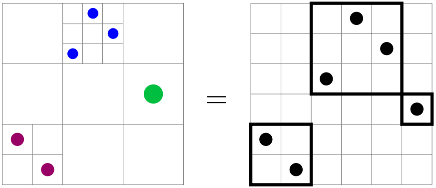
Example: $\sigma = 1\,8\,3\,6\,4\,2\,5\,7 =$



The substitution of $\pi_1, \dots, \pi_k$ in σ of size k is $\sigma[\pi_1, \dots, \pi_k]$ obtained as:

Example:

$1\,3\,2[2\,1, 1\,3\,2, 1] =$

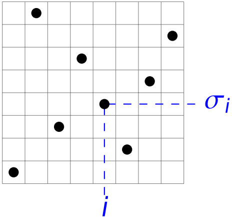


$= 2\,1\,4\,6\,5\,3$

Substitution in permutations

Easily explained on permutation diagrams.

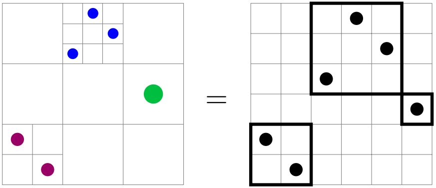
Example: $\sigma = 1\ 8\ 3\ 6\ 4\ 2\ 5\ 7 =$



The substitution of $\pi_1, \ldots, \pi_k$ in σ of size k is $\sigma[\pi_1, \ldots, \pi_k]$ obtained as:

Example:

$1\,3\,2[2\,1, 1\,3\,2, 1] =$



$= 2\,1\,4\,6\,5\,3$

Remark: Every π_i corresponds to an interval in $\sigma[\pi_1, \ldots, \pi_k]$.

Substitution decomposition theorem [Albert-Atkinson, '05]

Theorem: Every permutation of size $\neq 1$ is uniquely decomposed as

- $12\dots k[\pi_1, \dots, \pi_k]$, where the π_i are $\oplus$-indecomposable; or
- $k\dots 21[\pi_1, \dots, \pi_k]$, where the π_i are $\ominus$-indecomposable; or
- $\sigma[\pi_1, \dots, \pi_k]$, where σ is simple of size $k \geq 4$.

Remark: Simple permutations (*i.e.* those with only trivial intervals, like before) are 12, 21 or of size ≥ 4.

Notation: $\oplus$-indecomposable = that cannot be written as $12[\pi_1, \pi_2]$;
$\ominus$-indecomposable = that cannot be written as $21[\pi_1, \pi_2]$.

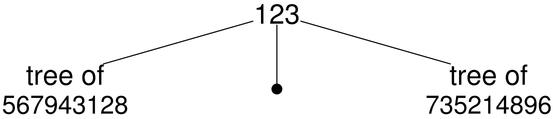
Remark: The π_i are the maximal strong intervals of the decomposed permutation.

(Substitution) decomposition trees

The theorem gives the first level of the decomposition tree.

Example: 5 6 7 9 4 3 1 2 8 10 17 13 15 12 11 14 18 19 16

$= 1\,2\,3\,[\,5\,6\,7\,9\,4\,3\,1\,2\,8\,,1\,,7\,3\,5\,2\,1\,4\,8\,9\,6\,]$



(Substitution) decomposition trees

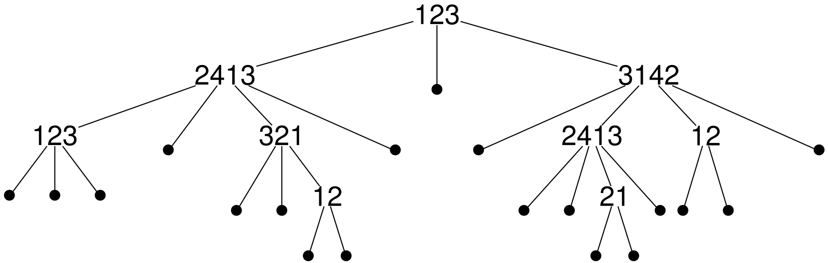
The theorem gives the first level of the decomposition tree.

Decomposing recursively the π_i's gives the full decomposition tree.

Example: $5\ 6\ 7\ 9\ 4\ 3\ 1\ 2\ 8\ 10\ 17\ 13\ 15\ 12\ 11\ 14\ 18\ 19\ 16$

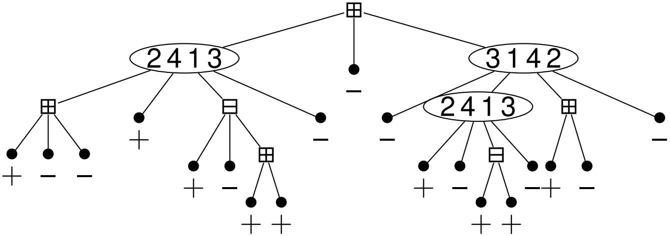
$= 1\ 2\ 3\ [\ 5\ 6\ 7\ 9\ 4\ 3\ 1\ 2\ 8\ ,1\ ,7\ 3\ 5\ 2\ 1\ 4\ 8\ 9\ 6\ ]$

$= 1\ 2\ 3\ [\ 2\ 4\ 1\ 3\ [\ 1\ 2\ 3\ ,1\ ,4\ 3\ 1\ 2, 1], 1, \ldots] = \ldots$

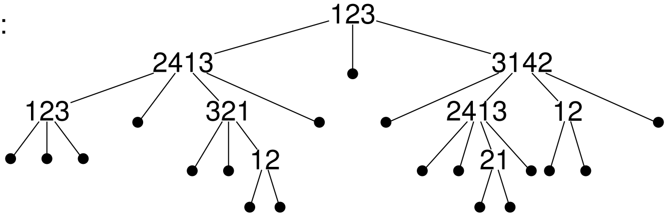


Decomposition tree or strong interval tree?

Strong interval tree:



Decomposition tree:



They are the same (in the unsigned case) up to the change of notation $12\dots k \leftrightarrow \boxplus$, $k\dots 21 \leftrightarrow \boxminus$ and $\sigma \leftrightarrow \textcircled{\sigma}$ for simples.

Decomposition trees or strong interval trees

1. Strong interval trees
2. (Substitution) decomposition trees
3. Some applications in algorithms and combinatorics

Strong interval trees in algorithms

- Computating modular decomposition trees of graphs through factorizing permutations.
  [Habib-Paul-Viennot, '98] [Habib-de Montgolfier-Paul, '04] [Tedder-Corneil-Habib-Paul, '08] [Capelle-Habib-de Montgolfier, '02] [Bui Xuan-Habib-Paul, '05] [Bergeron-Chauve-de Montgolfier-Raffinot, '08]
- Pattern matching of permutations, in restricted cases.
  [Bose-Buss-Lubiw, '98] [Ibarra, '97] [B-Rossin, '06] [B-Rossin-Vialette, '07]
- Computing scenarios of perfect sorting by reversals.
  [Bérard-Bergeron-Chauve-Paul, '07] [Bérard-Chateau-Chauve-Paul-Tannier, '08] [B-Chauve-Mishna-Rossin, '09]
- …

Decomposition trees in combinatorics

- Enumeration of simple permutations.
  [Albert-Atkinson-Klazar, '03]
- Number of intervals in random permutations.
  [Corteel-Louchard-Pemantle, '06]
- Properties of classes closed by substitution.
  [Atkinson-Stitt, '02] [Brignall, '07] [Atkinson-Ruškuc-Smith, '09]
- Exhibit the structure of classes.
  [Albert-Atkinson, '05] [Brignall-Huczynska-Vatter, '08]
  [Brignall-Ruškuc-Vatter, '08] [Bassino-B-Rossin, '08]
  [Bassino-B-Pierrot-Rossin, '15] [Bassino-B-Pierrot-Pivoteau-Rossin, '16]
- . . .

Solving perfect sorting by reversals: an algorithm and its analysis

Idea of the algorithm of [Bérard-Bergeron-Chauve-Paul, '07]

Starting point: Compute the strong interval tree of σ.

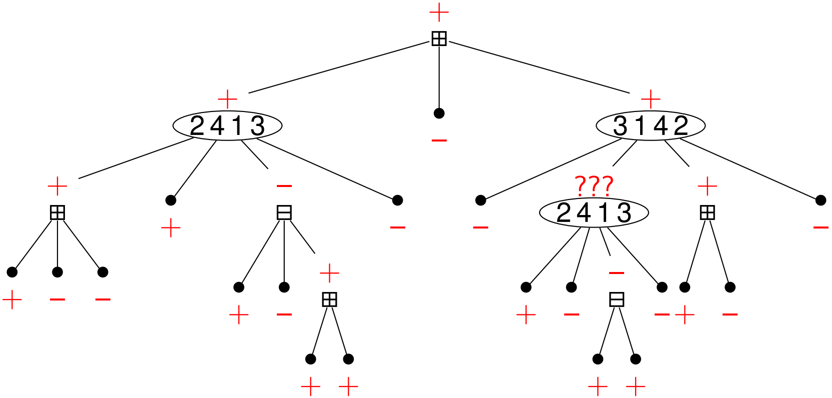
Pre-processing: Put labels $+$ or $-$ on the nodes of the strong interval tree of σ:

- Leaf: sign of the element in σ;
- Linear node: $+$ for $\boxplus$ (increasing) and $-$ for $\boxminus$ (decreasing);
- Prime node whose parent is linear: sign of its parent;
- Other prime node: ???
  - $\hookrightarrow$ Test labels $+$ and $-$ and choose the shortest scenario.

Main part of the algorithm:

- Perform Hannenhalli-Pevzner (or improved version – solving (normal) sorting by reversals) on prime nodes.
- A signed node belongs to the scenario **iff** it has a linear parent and its sign is different from the one of its parent.

Example of labeled decomposition tree



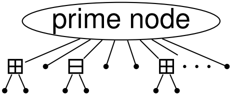
Complexity

- The algorithm runs in $\mathcal{O}(2^p n \sqrt{n \log n})$, with $p = \#$ prime nodes.
- It is polynomial when there are no prime nodes;
  this corresponds to separable permutations or commuting scenarios.

[Bérard-Bergeron-Chauve-Paul, '07]

Under the uniform distribution on signed permutations, it is:

- Polynomial with probability 1 asymptotically.
  Because a tree is of the shape shown opposite with probability tending to 1:



- Polynomial on average.
  Bounding the number of permutations whose strong interval tree contains p prime nodes.

[B-Chauve-Mishna-Rossin, '09]

Separable permutations and commuting scenarios

Commuting scenarios

- A scenario for perfect sorting by reversals is commuting when all its reversals pairwise commute (=do not overlap).

Nice surprise: Examples of commuting scenarios arise in the study of mammalian genome evolution.

Commuting scenarios

- A scenario for perfect sorting by reversals is commuting when all its reversals pairwise commute (=do not overlap).

Nice surprise: Examples of commuting scenarios arise in the study of mammalian genome evolution.

Remark: A commuting scenario can be described as a set (instead of sequence) of reversals.

- A (signed) permutation is commuting if there exists a commuting scenario sorting it.

Remark: If σ is commuting, all permutations obtained changing the signs in σ also are.

Separable permutations

Separable permutations:

- Those avoiding the patterns 2413 and 3142.
- Those whose decomposition tree contains no prime node.
- Those having a separation tree.

Consequence: Separable permutations and commuting permutations (rather, their unsigned version) coincide.

Consequence: The algorithm is polynomial on separable permutations ($p = 0$).

Reversals in commuting scenarios

In general, in the computed scenario, a reversal is

- either a linear node or leaf with label different from its linear parent,
- or inside a prime node.

Consequence: For separable permutations, a reversal is a node with a label different from its parent.

Prop.: No $\boxplus - \boxplus$ nor $\boxminus - \boxminus$ edge in decomposition trees.

Consequence:

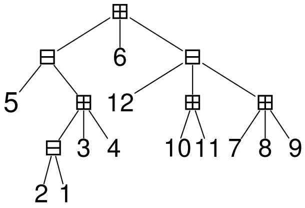
The set of reversals is $\begin{cases} \text{all internal nodes except the root} \\ +\text{leaves with a label different from their parent.} \end{cases}$

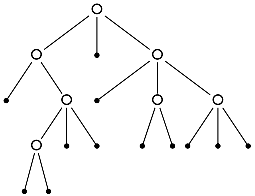
$$\boxed{\text{Reversals} \approx \text{internal nodes – the root} + \text{half of the leaves}}$$

Parameters of commuting scenarios on Schröder trees

The shape of the tree is sufficient to study reversals.

Decomposition trees of (unsigned) separable permutation	Schröder trees $+$ label ⊞ or ⊟ on the root





size of σ	$\longleftrightarrow$	number of leaves
reversal of length ≥ 2	$\longleftrightarrow$	internal node except the root
reversal of length 1	$\longleftrightarrow$	some leaves (half of them)
length of a reversal	$\longleftrightarrow$	size ($=\#$ leaves) of the subtree

Parameters on Schröder trees

Study two parameters on Schröder trees:

- Number of internal nodes, and
- Pathlength = sum of the sizes of the subtrees.

Their average give access to:

- the average number of reversals, and
- the average length of a reversal

in a scenario for a separable permutation.

Analytic combinatorics:
Average of parameters is obtained from bivariate generating functions $S(x,y) = \sum s_{n,k} x^n y^k$ where $s_{n,k}$ = number of Schröder trees with n leaves and k internal nodes (resp. pathlength k).

Results on parameters

Application of the methodology of [Flajolet-Sedgewick, '09].

(Almost direct application; but note that for us the size is the number of *leaves*.)

In Schröder trees with n leaves:

- Average number of internal nodes: $\sim \frac{n}{\sqrt{2}}$
- Average pathlength: $\sim 1.27 n^{\frac{3}{2}}$

In scenarios for separable permutations of size n:

- Average number of reversals: $\sim \frac{1+\sqrt{2}}{2} n$
  (among which on average $n/2$ are of length 1)
- Average length of a reversal: $\sim 1.054 \sqrt{n}$

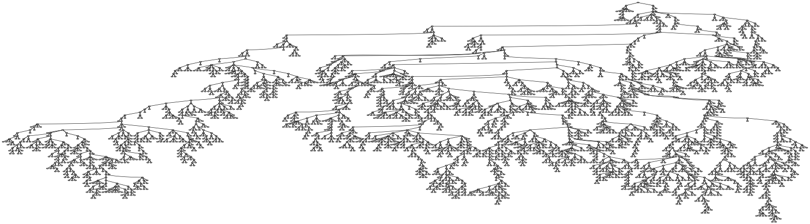
Allowing prime nodes of bounded arity

Comparing models with data

- Data: tree comparing Gorilla and Bos Taurus:

- Random tree under the uniform distribution on permutations:

- Random tree under the uniform distribution on separables:



Neither the uniform distribution nor the restriction to separable permutations represent the data well.
Can we do better by allowing some prime nodes?

Allowing prime nodes of bounded arity

Fix a maximal arity k for the prime nodes.

Remark: This is not a simple variety of trees.

- Number of permutations of size n in this class: $\sim c_1 \cdot \rho_k^{-n} n^{-3/2}$.
- Average number of prime nodes in such trees: $\sim c_2 \cdot n$
- Average number of internal nodes in such trees: $\sim c_3 \cdot n$
- Average pathlength in such trees: $\sim c_4 \cdot n^{3/2}$

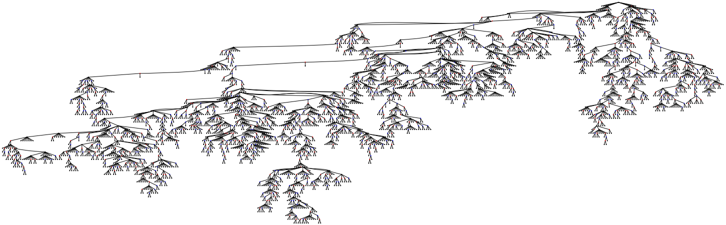
These parameters are related to the perfect sorting by reversals (but less directly than in the separable case).

The constants c_i are expressed in terms of τ_k, ρ_k and $\Lambda_k''(\tau_k)$, defined by:

- $\Lambda_k(x) = \frac{x^2}{1-x} + \sum_{j=4}^{k} s_j \left(\frac{x}{1-x}\right)^j$ where $s_j = \#$ simples of size j;
- τ_k is the smallest root of $\Lambda_k'(\tau_k) = 1$;
- $\rho_k = \tau_k - \Lambda_k(\tau_k)$.

From genome rearrangements to analytic combinatorics

- Random tree under the uniform distribution on permutations whose decomposition tree has prime nodes of arity at most 7:



Does not seem a good model of data.
But those trees have another interest, for analytic combinatorics.

Families of trees converging to permutations

Combinatorial objects:

- $\mathcal{P}$ = the set of all permutations; $\mathcal{P}_n$ = those of size n.
- $\mathcal{P}^{(k)}$ = the set of all permutations whose decomposition tree contains prime nodes of arity at most k; $\mathcal{P}_n^{(k)}$ = those of size n.
- $\mathcal{P}_n^{(k)} = \mathcal{P}_n$ as soon as $k \geq n$.
- Consequently, $\lim_{k\to\infty} \mathcal{P}^{(k)} = \mathcal{P}$.

Asymptotics:

- Stirling estimates: $|\mathcal{P}_n| \sim_n (n/e)^n \sqrt{2\pi n}$.
- Tree estimates: For any fixed k, $|\mathcal{P}_n^{(k)}| \sim_n \alpha_k \rho_k^{-n} n^{-3/2}$.
- For any fixed k, we have an upper bound on $\alpha_k \rho_k^{-n} n^{-3/2}$ as $n \to \infty$; Illegally applying this bound for $k = n$ gives cst × Stirling estimates.
- **Open:** Can we reconcile both asymptotics properly? Difficulty: the OGF of permutations is not analytic.

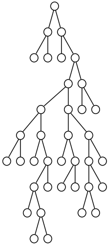
Other non-uniform distributions

Getting closer to the data?

Galton-Watson trees

These are trees with prescribed offspring distribution:
for all i, $p_i =$ probability that a node has i children.

Estimating the offspring distribution on the data (by frequencies of number of children, forgetting about the root), we obtain random trees of the form:



These trees should represent those seen under the prime root in the data.

(Obviously) not a good model.

It is however not so obvious to prove it using the classical method of comparing the data to the model for some estimator.

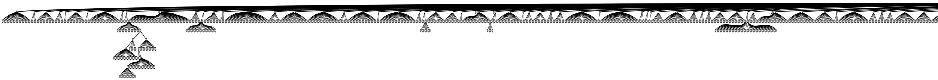
Mixed model

In this model, trees are a forest of 175 subtrees under one prime root, each subtree being obtainded as:

- Draw a random Galton-Watson binary tree, with $Proba(leaf) = 0.8$;
- Replace each leaf by $k + 1$ leaves, k being randomly chosen according to a geometric law of parameter 0.85.

Remark: 175 is the arity of the root in one tree from our data.
Parameters 0.8 and 0.85 are heuristic.

Typical tree obtained:



It seems much more like our data!

Statistical methods to compare trees

The mixed model seems:

- to represent the data well;
- to be simple enough to be studied mathematically.

Questions are:

- Prove properties of the trees in this model.
- Are some of them transferable to the data? Does this give a better understanding of the biological data?

- How to express that our model represents well the data?
- Can we prove it? and how? (Method of the two-sample problem?)

Statistical methods to compare trees

The mixed model seems:

- to represent the data well;
- to be simple enough to be studied mathematically.

Questions are:

- Prove properties of the trees in this model.
- Are some of them transferable to the data? Does this give a better understanding of the biological data?

- How to express that our model represents well the data?
- Can we prove it? and how? (Method of the two-sample problem?)

Questions are very much open, and suggestions very welcome!

Thank you!