Counting Sorting Scenarios, Intermediate Genomes and Dealing With Missing Genes With the Rank Distance

J. P. P. Zanetti, L. Chindelevitch, J. Meidanis

University of Campinas, Brazil; Simon Fraser University, Canada

December 2019
Summary

1. Introduction

2. Counting Scenarios and Intermediates

3. Genomes With Indels

4. Final Remarks
Introduction
Comparing genomes

- Multichromosomal genomes
- Each chromosome: linear or circular
- Genomes have the same genes (just positioned differently)
- Correspondence between genes is given
- Compute “distance” between two genomes
- Rank distance: related to Double Cut and Join (DCJ)
- Both easy to compute, biologically sound
Comparing genomes

- Multichromosomal genomes
- Each chromosome: linear or circular
- Genomes have the same genes (just positioned differently)
- Correspondence between genes is given
- Compute “distance” between two genomes
- Rank distance: related to Double Cut and Join (DCJ)
- Both easy to compute, biologically sound
Rank distance: DCJ with average number of adjacencies

### Diagram

- **Snapshot 1:**
  - Three linear sequences: $a$, $b$, and $c$.
  - Adjacencies represented by arrows.

- **Snapshot 2:**
  - A circular arrangement involving $a$, $b$, and $c$.
  - The circular structure indicates a rearrangement event.

---

**ZCM2019 (Unicamp, SFU) Scenarios, Intermediates, Indels for Rank December 2019**
Rank distance: DCJ with average # of adjacencies

• Build breakpoint graph with caps

\[ \begin{align*}
\text{a} & \quad \text{b} \\
\text{c} & \\
\text{a} & \quad \text{b} \\
\text{c} & \\
\end{align*} \]

\[ \begin{align*}
\text{a} & \quad \text{b} \\
\text{c} & \\
\end{align*} \]
Rank distance: DCJ with average \# of adjacencies

- Build breakpoint graph with caps

- Formula: \# adjacencies - \# of components

- Genomes **may differ** in \# of adjacencies; which one to use?
Rank distance: DCJ with average # of adjacencies

- Build breakpoint graph with caps

- Formula: # adjacencies - # of components
- Genomes **may differ** in # of adjacencies; which one to use?
- DCJ approach: close paths into cycles add "null" chromosomes to equalize # of adjacencies

\[ d_{DCJ} = 5 - 3 = 2 \]
Rank distance: DCJ with average # of adjacencies

- Build breakpoint graph with caps

- Formula: # adjacencies - # of components

- Genomes may differ in # of adjacencies; which one to use?

- DCJ approach: close paths into cycles
  
  add “null” chromosomes to equalize # of adjacencies

- Rank distance approach: use average # of adjacencies
  
  double formula to avoid fractions

\[ d_{\text{DCJ}} = 5 - 3 = 2 \]
\[ d_{\text{rank}} = 2 \left( \frac{4+5}{2} - 3 \right) = 3 \]
Allowed rearrangements:
cuts, joins, double swaps

Weights:
cut = join = 1, double swap = 2

Rank distance:
weight of optimal (smallest weight) series of genome rearrangements
taking $A$ to $B$
Rank distance by rearrangements

- Allowed rearrangements:
  - cuts, joins, double swaps
- Weights:
  - cut = join = 1, double swap = 2
- Rank distance:
  - weight of optimal (smallest weight) series of genome rearrangements taking A to B

\[ a \rightarrow b \rightarrow c \quad \text{cut} = \quad a \rightarrow b \rightarrow \text{join} \]

Diagram:

- cut: \[ a \rightarrow b \rightarrow c \]
- join: \[ a \rightarrow b \rightarrow c \]
- double swap: \[ a \rightarrow b \rightarrow c \]

ZCM2019 (Unicamp, SFU)  Scenarios, Intermediates, Indels for Rank  December 2019  12 / 32
Rank distance by breakpoint graph components

- Forget about caps: not needed
- Breakpoint graph without caps has paths and cycles only
- Path contributes its \# of edges to rank distance
- Cycle contributes its \# of edges minus 2 to rank distance

\[ d_{\text{rank}} = 1 + 2 + 0 = 3 \]
Interesting property

- Optimal rank series never mix breakpoint graph components
- Some DCJ optimal series do mix breakpoint graph components
We can represent genomes as square, symmetric, 0-1 matrices.

With this representation, rank distance is just $r(A - B)$.
Counting Scenarios and Intermediates
Scenarios and intermediates

- Scenario from $A$ to $B$: optimal series of operations taking $A$ to $B$
- Intermediate genome: genome appearing in some scenario
- In graph language:
  - Scenario = minimum weight path
  - Intermediate = vertex in scenario
- Both give an idea of the **diversity** of ways to get from $A$ to $B$
- In general, $\#$ intermediates $<< \#$ scenarios

3 paths (sorting scenarios) 6 intermediate genomes
Scenarios(\(A, B\)) = \[\begin{cases} (k + 1)^{k-1} & \text{for a } 2k\text{-cycle} \\ \text{complicated} & \text{for a path} \end{cases}\]

- Formula for cycles already known (same as DCJ)
- For paths, no simple formula, but fast code
- Depends on type and size of components only

```bash
count_rank_scenarios2.py -c 2 4 6 -p 2 3 5
```

result: 283983840
Formulas / Code for Intermediates

\[ \text{Intermediates}(A, B) = \begin{cases} \frac{1}{k+1} \binom{2k}{k} & \text{for a 2k-cycle} \\ k + 1 \left\lceil \frac{(k + 1)}{2} \right\rceil & \text{for a } k\text{-edge path} \end{cases} \]

- Depends on type and size of components only

```python
count_intermediates2.py -c 2 4 6 -p 2 3 5
```

result: 3600
### Experiments

#### Counts for human genome vs. other eutherians

<table>
<thead>
<tr>
<th>Species</th>
<th>Distance</th>
<th>Scenarios</th>
<th>Intermediates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chimpanzee</td>
<td>27</td>
<td>$6.54 \times 10^{11}$</td>
<td>$2.46 \times 10^{4}$</td>
</tr>
<tr>
<td>Orangutan</td>
<td>53</td>
<td>$6.03 \times 10^{38}$</td>
<td>$1.29 \times 10^{10}$</td>
</tr>
<tr>
<td>Rhesus</td>
<td>150</td>
<td>$1.21 \times 10^{138}$</td>
<td>$1.45 \times 10^{28}$</td>
</tr>
<tr>
<td>Elephant</td>
<td>336</td>
<td>$[10^{583}, 10^{609}]$</td>
<td>$8.56 \times 10^{86}$</td>
</tr>
<tr>
<td>Goat</td>
<td>393</td>
<td>$[10^{548}, 10^{588}]$</td>
<td>$5.30 \times 10^{85}$</td>
</tr>
<tr>
<td>Mouse</td>
<td>509</td>
<td>$[10^{830}, 10^{980}]$</td>
<td>$2.44 \times 10^{131}$</td>
</tr>
<tr>
<td>Rat</td>
<td>788</td>
<td>$[10^{1251}, 10^{1476}]$</td>
<td>$2.30 \times 10^{189}$</td>
</tr>
</tbody>
</table>
Genomes With Indels
Representing genomes with indels as matrices

A missing extremity is represented by a 0 row and 0 column

\[
A - B = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\
\end{bmatrix}
\]

We get \( d(A, B) = r(A - B) = 8 \). How can we compute it faster?
The Augmented Breakpoint Graph

Components: standard, A-null, B-null, AA-null, BB-null, AB-null path

A path is proper if both ends are free or it is an AB-null singleton

Result 1 The rank distance equals

\[ n - 2c(A, B) - p_0(A, B) - p_{AB}(A, B) \]

where \( n \) counts extremities, \( c \), cycles, \( p_0 \), proper paths, and \( p_{AB} \), non-singleton \( AB \)-paths
A problem with the rank distance

These genomes are at distance 8 and cannot be optimally sorted using only insertions, deletions, and marker substitutions.

Some intermediate genome must have a **semi-chromosome**: a tail extremity with no corresponding head or vice versa.
Result 2: The rank distance (with indels) equals the smallest cost of a sequence of operations going from $A$ to $B$ using:

- Cuts or joins (cost 1);
- Double swaps (cost 2);
- Substitutions of one extremity (cost 2);
- Insertions/deletions of whole chromosomes or semi-chromosomes (cost = number of extremities used).
**An alternative version: the rank-indel distance**

**Definition**: The rank-indel distance equals the smallest cost of a sequence of operations going from $A$ to $B$ using:

- Cuts or joins (cost 1);
- Double swaps (cost 2);
- Substitutions of one extremity (cost 2);
- Insertions/deletions of whole chromosomes or semi-chromosomes (cost = number of extremities used).

**Result 3** The rank-indel distance equals

\[ n - 2c(A, B) - p_0(A, B) + p_{AB}(A, B) \]

The only difference with the rank distance is the sign of the last term.
An Application: Phylogenetic Tree of Fungal Pathogens

ML on nearly 3,000 CDS

rank distance + neighbor-joining
Final Remarks
Conclusion and open problems

- We have a basic $O(n)$ algorithm for rank distance with indels
- We have a similar $O(n)$ algorithm for rank-indel distance
- They differ by a single term, which is often small in practice
- We observe qualitative agreement with an existing phylogeny

- Do the “semi-chromosomes” make biological sense?
- Can these distances help ancestral genome reconstruction?
- Can we generalize these ideas to handle gene duplications?
Acknowledgments

João Paulo Pereira Zanetti

Leonid Chindelevitch

FAPESP
NSERC / CRSNG
CIHR / IRSC
GenomeCanada
PRONON
Related References


Thanks!!

Get this presentation:

http://www.ic.unicamp.br/~meidanis/research/rear/