Sorting by Reversals

David Letscher

Saint Louis University

Intro to CS: Bioinformatics
Genome Rearrangement

Turnips and Cabbages

- 99% identical
- Nearly identical gene sequences
- Genes in different order!

How does this happen:
- DNA forms loops and reversed a segment or two strands cross and swap segments.

Letcher (SLU)
Genome Rearrangement

Turnips and Cabbages

- 99% identical
- Nearly identical gene sequences
- Genes in different order!

How does this happen

DNA forms loops and reversed a segment or two strands cross and swap segments.
Genome Rearrangement

Human and Mouse Genome

[Image of a genome rearrangement diagram comparing human and mouse chromosomes.]
Genome Rearrangement

Human and Mouse X Chromosomes
Reversals

Sorting by Reversals
The more reversals, the further back their common ancestor lived.

**Parsimony assumption**

Want to count the minimum number of reversals needed to convert one sequence to another.
Sort a sequence of numbers by reversing segments

Examples:

5 2 1

3 4

1 2 5 3 4

1 2 5 4 3

1 2 3 4 5

What's the minimum number of reversals needed?
Related Problem: Sorting by Reversals

Sort a sequence of numbers by reversing segments

An example

\[
\begin{array}{cccccc}
5 & 2 & 1 & 3 & 4 \\
\overline{1} & 2 & 5 & 3 & 4 \\
1 & 2 & 5 & 4 & 3 \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
\]
Related Problem: Sorting by Reversals

Sort a sequence of numbers by reversing segments

An example

5 2 1 3 4
1 2 5 3 4
1 2 5 4 3
1 2 3 4 5

What’s the minimum number of reversals needed?
Challenge

Find the minimum number of reversals to sort

- 1 4 6 5 7 8 3 2
- 1 9 3 4 7 8 2 5 6
Notation

Permutation

\[ \pi = \pi_1 \pi_2 \pi_3 \ldots \pi_{n-1} \pi_n \]

\[ \begin{align*}
\pi_1 & \quad \pi_2 & \quad \pi_3 & \quad \pi_4 & \quad \pi_5 & \quad \pi_6 & \quad \pi_7 & \quad \pi_8 \\
1 & \quad 4 & \quad 6 & \quad 5 & \quad 7 & \quad 8 & \quad 3 & \quad 2
\end{align*} \]
Extending permutations

For a permutation of 1..n, add a 0 at the beginning and n+1 at the end.

1 9 3 4 7 8 2 5 6 becomes 0 1 9 3 4 7 8 2 5 6 10

This changes the permutation to $\pi_0 \, \pi_1 \, \pi_2 \, \pi_3 \, \ldots \, \pi_{n-1} \, \pi_n \, \pi_{n+1}$

Breakpoints

Positions in the extended permutation where numbers are not consecutive:

0 1 | 9 | 3 4 | 7 8 | 2 | 5 6 | 10

This sequence has 6 breaks points.

How many breakpoints can a reversal remove?
Lower Bounds

\[ b(\pi) \] Number of breakpoints in the (extended) permutation \( \pi \)

\[ d(\pi) \] The minimum number of reversals need to sort \( \pi \)

Each reversal removes at most 2 breakpoints so

\[ d(\pi) \geq b(\pi) / 2 \]
Lower Bounds

\[ b(\pi) \] Number of breakpoints in the (extended) permutation \( \pi \)

\[ d(\pi) \] The minimum number of reversals need to sort \( \pi \)

Each reversal removes at most 2 breakpoints
**Lower Bounds**

\[ b(\pi) \] Number of breakpoints in the (extended) permutation \( \pi \)

\[ d(\pi) \] The minimum number of reversals need to sort \( \pi \)

Each reversal removes at most 2 breakpoints so

\[ d(\pi) \geq b(\pi)/2 \]
An Example

\[ \pi = 231465 \]

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Breakpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 2 3 1 4 6 5 7</td>
<td>5</td>
</tr>
<tr>
<td>0 1 3 2 4 6 5 7</td>
<td>4</td>
</tr>
<tr>
<td>0 1 2 3 4 6 5 7</td>
<td>2</td>
</tr>
<tr>
<td>0 1 2 3 4 5 6 7</td>
<td>0</td>
</tr>
</tbody>
</table>

Can we always remove a break point?
**Strip** A sequence between consecutive breakpoints

**Increasing strip** a strip where the numbers are in increasing order

**Decreasing strip** a strip where the numbers are in decreasing order

**An example:** \( \pi = 1 \ 9 \ 8 \ 7 \ 3 \ 4 \ 2 \ 5 \ 6 \)

0 1 9 8 7 3 4 2 5 6 10 has 6 strips

Increasing strips: 0 1, 3 4, 2, 5 6, 10

Decreasing strips: 9 8 7, 10
Fact 1

If a sequence has a decreasing strip then there is a reversal that removes a breakpoint.
Fact 1

If a sequence has a decreasing strip then there is a reversal that removes a breakpoint

1. Find the smallest number $k$ in any decreasing strip
2. Find $k - 1$ in the sequence
3. Perform the reversal that make $k - 1$ and $k$ adjacent

0 1 4 6 5 7 8 3 2 9
0 1 2 3 8 7 5 6 4 9

0 1 6 5 4 2 3 7 8 9
0 1 6 5 4 3 2 7 8 9
Fact 2

If there is no decreasing strips then reversing an increasing strip will not change the number of breakpoints.
Fact 2

If there is no decreasing strips then reversing an increasing strip will not change the number of breakpoints

0 1 2 5 6 7 3 4 8 9
0 1 2 7 6 4 3 4 8 9

Notice that to remove a breakpoint either the strip before or after must have been increasing.
A Greedy Algorithm

1. While \( b(\pi) > 0 \)
2.   If \( \pi \) has a decreasing string
3.     Among all possible reversals choose one that removes the most breakpoints
4.   Else
5.     Perform a reversal of an increasing strip
A Greedy Algorithm

1. While $b(\pi) > 0$
2. If $\pi$ has a decreasing string
3. Among all possible reversals choose one that removes the most breakpoints
4. Else
5. Perform a reversal of an increasing strip

Note, if you’re trying to remove a breakpoint, use the idea a couple of slides ago to quickly find one. Find the smallest element $k$ of a decreasing strip and perform the reversal that makes $k - 1$ and $k$ adjacent.

Note that at most two reversals are needed in the algorithm to remove a breakpoint.
A Greedy Algorithm

1. While $b(\pi) > 0$
2. If $\pi$ has a decreasing string
3. Among all possible reversals choose one that removes the most breakpoints
4. Else
5. Perform a reversal of an increasing strip

Note, if you’re trying to remove a breakpoint, use the idea a couple of slides ago to quickly find one. Find the smallest element $k$ of a decreasing strip and perform the reversal that makes $k - 1$ and $k$ adjacent.

Note that at most two reversals are needed in the algorithm to remove a breakpoint. So $d(\pi) \leq 2b(\pi)$
Challenge

Try this algorithm on

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

1. How many breakpoints are in each sequence?
2. How many reversals does the greedy algorithm take to sort the sequence?
3. What is the minimum number of reversals needed?
Approximation Algorithms

How many reversals are necessary?

\[ \frac{b(\pi)}{2} \leq d(\pi) \leq 2b(\pi) \]

The greedy algorithm might not find the minimum number of reversals, but at worst is it off by a factor of 4! This algorithm has an approximation factor of 4.
How many reversals are necessary?

\[ b(\pi)/2 \leq d(\pi) \leq 2b(\pi) \]

The greedy algorithm might not find the minimum number of reversals, but at worst is it off by a factor of 4!
How many reversals are necessary?

\[ \frac{b(\pi)}{2} \leq d(\pi) \leq 2b(\pi) \]

The greedy algorithm might not find the minimum number of reversals, but at worst is it off by a factor of 4!

This algorithm has an **approximation factor** of 4.
In the previous algorithm we either have a decreasing strip and can remove a breakpoint. If this is all we every saw, then we would have $d(\pi) \leq b(\pi)$. 
In the previous algorithm we either have a decreasing strip and can remove a breakpoint. If this is all we every saw, then we would have $d(\pi) \leq b(\pi)$. But, we don’t always have a decreasing strip. So it might take two reversals to remove a breakpoint.
In the previous algorithm we either have a decreasing strip and can remove a breakpoint. If this is all we ever saw, then we would have $d(\pi) \leq b(\pi)$.

But, we don’t always have a decreasing strip. So it might take two reversals to remove a breakpoint.

Think about what happens to create a sequence with no decreasing strips.
Claim

If every reversal that reduce the number of breakpoints results in a permutation without decreasing strips then there is a reversal that removes two breakpoints!

If true, this would show $b(\pi)/2 \leq d(\pi) \leq b(\pi)$ and give an algorithm with approximation factor 2.
If you’re interested in how to show this is true, I’m happy to show you.
Multiple Sequences

Goal
Simulate multiple DNA strands and how many changes are required to put each strip on the correct chromosome and in the correct order.

Moves
1. Reversal on a single strand
2. Cut two strands and swap the ends
3. Cut two strands combining the front ends together and and back ends together
How should we define the number of breakpoints
In particular, what do we do at the ends?
Bounds

What the most breakpoints that can be removed by a single reversal or other move?

What’s the most moves need to remove a breakpoint?

Can we bound the distance to the sorted order based on the number of breakpoints?
Challenge

How many breakpoints? Bounds? What’s the shortest move sequence?