Algorithms for Genome Rearrangements

Pedro Feijão

Summer 2015

pfeijao@cebitec.uni-bielefeld.de

Preliminaries

- Wiki Page:
  `http://wiki.techfak.uni-bielefeld.de/gi/Teaching`

- Organization

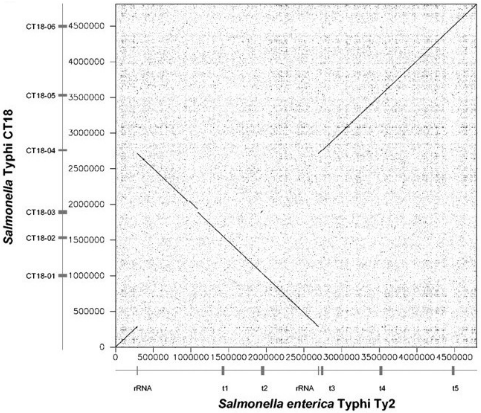
Turnip (Speiserübe) vs. Cabbage (Weißkohl)

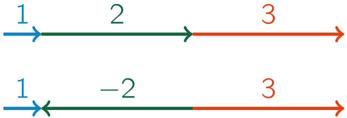
Although cabbages and turnips share a recent common ancestor, they look and taste different.

Genome Rearrangements - Background

- In the 1980s Jeffrey Palmer studied evolution of plant organelles by comparing mitochondrial genomes of cabbage and turnip.
- He found 99% similarity between genes.
- These surprisingly similar gene sequences differed in gene order.
- This study helped pave the way to analyzing genome rearrangements in molecular evolution.

Genome Rearrangements - Background

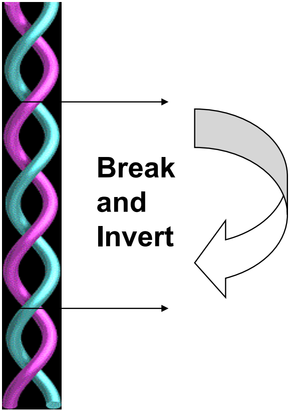




1 2 3

1 -2 3

Reversal Example



Break and Invert

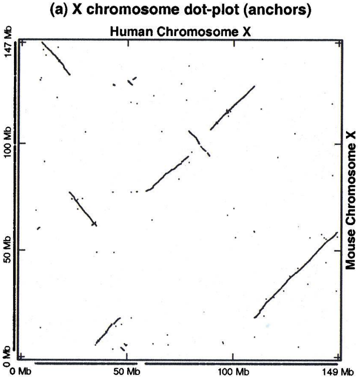
5’ ATGCCTGTACTA 3’

3’ TACGGACATGAT 5’

5’ ATGTACAGGCTA 3’

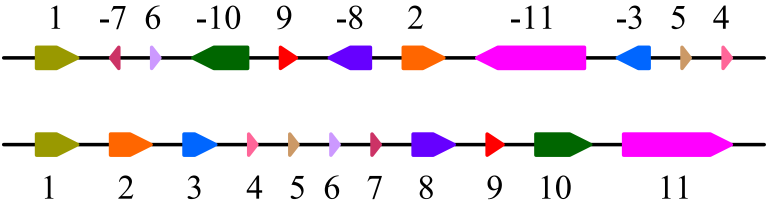
3’ TACATGTCCGAT 5’

Human vs. Mouse



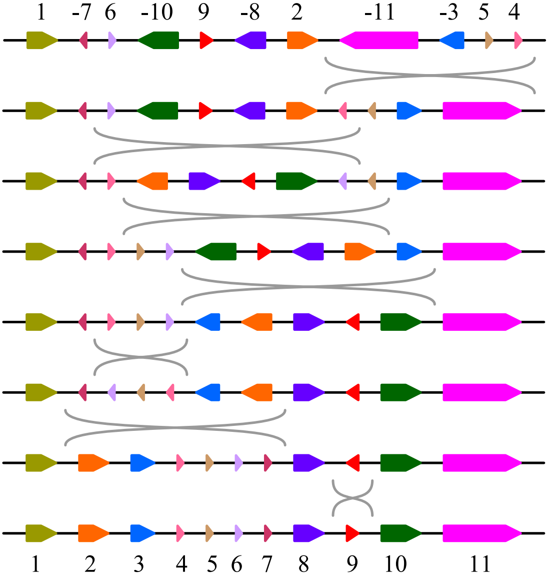
Pevzner, P.A. and Tesler, G. 2003. **Genome rearrangements in mammalian evolution: Lessons from human and mouse genomic sequences**. *Genome Res.* **13**: 13-26.

Human vs. Mouse

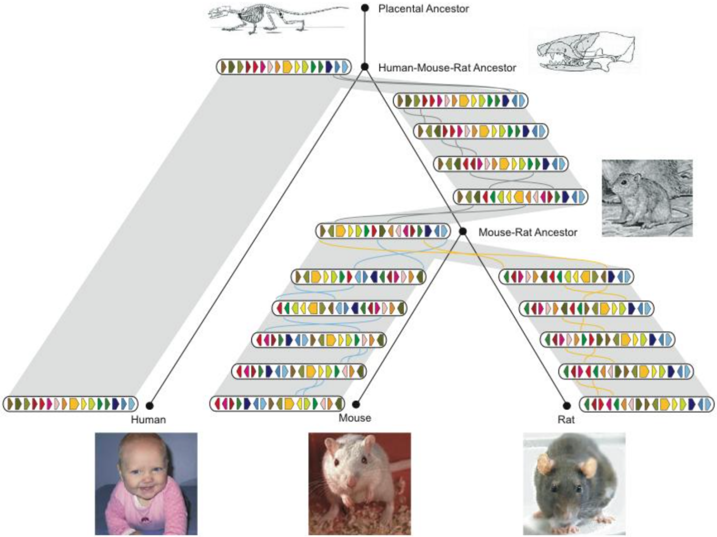


- How many rearrangements do we need to *transform* one genome into the other?

Human vs. Mouse



X Chromosome history

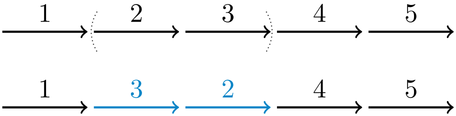


Rat Consortium, *Nature*, 2004

Genome Rearrangements

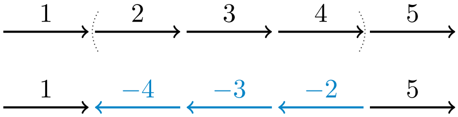
- **Genome rearrangements** are evolutionary events that *shuffle* the genome.
- Important questions:
    - What is the **minimum number** of rearrangement operations needed to transform one genome into another? (Distance)
    - Can we find a **rearrangement scenario** with this minimum number of operations? (Sorting)
- Several types of **rearrangement operations** were studied:

Common Rearrangement Operations



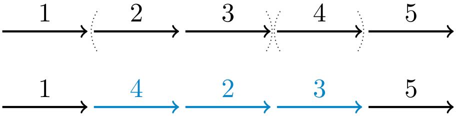
Unsigned Reversal/Inversion

Common Rearrangement Operations



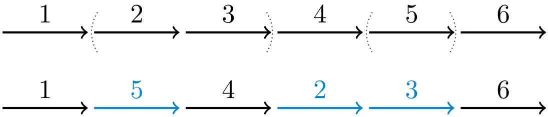
Signed Reversal/Inversion

Common Rearrangement Operations



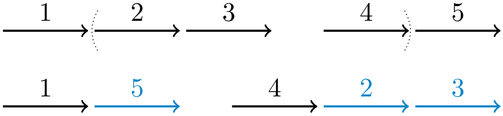
Transposition

Common Rearrangement Operations



Block Interchange

Common Rearrangement Operations



Translocation (*multichromosomal* operation)

Genome Rearrangement Models

- Several models were proposed, allowing only one operation or combining two or more.

- Each different models results in a *combinatorial problem* that must be solved.

- Usually polinomially solvable, notable exceptions: Unsigned reversal and Transposition (NP-hard)

Reversal Models

- Since 1990, beginning with Sankoff in 1992, many papers have been devoted to the subject of **reversal distance**.
- The *unsigned reversal* distance is NP-hard (Caprara 1997)
- The *signed reversal* was solved polynomially by Hannenhalli and Pevzner in 1995.

Definitions

- A genome will be represented by a **permutation**, which is a bijection on the set $\{1, \ldots, n\}$.

$$\pi = (\ \pi_1 \quad \pi_2 \quad \cdots \quad \pi_n\ )$$

For example:

$$\pi = (\ 2 \quad 1 \quad 4 \quad 3 \quad 5 \quad 8 \quad 6 \quad 7\ )$$

Definitions

- A **reversal** $\rho(i,j)$ reverts the *order* of elements in interval (i,j).

$$\pi = (\ 2 \quad 1 \quad 4 \quad 3 \quad 5 \quad 8 \quad 6 \quad 7\ )$$

Applying $\rho(3,6)$:

$$\pi = (\ 2 \quad 1 \quad 8 \quad 5 \quad 3 \quad 4 \quad 6 \quad 7\ )$$

Reversal Distance Problem

- **Problem**: Given two permutations π and σ, find the shortest series of reversals that transforms π into σ.
- **Input**: Permutations π and σ.
- **Output**: A series of reversals $\rho_1, \dots, \rho_d$ transforming π into σ, such that d is minimum.

Reversal Distance Problem

- **Problem**: Given a permutation π, find the shortest series of reversals that transforms π into the identity $(\, 1 \quad 2 \quad \cdots \quad n \,)$. (Sorting π)
- **Input**: Permutation π.
- **Output**: A series of reversals $\rho_1, \ldots, \rho_d$ transforming π into the identity, such that d is minimum.

d is called the **reversal distance**, and it is denoted here $d(\pi)$.

Example

How can we sort this permutation?

$$\pi = (\ 2 \quad 1 \quad 5 \quad 3 \quad 4\ )$$

First Algorithm Idea

- In the *i*-th step, put the *i*-th element in position

$$(2 \quad 1 \quad 5 \quad 3 \quad 4)$$

$$(1 \quad 2 \quad 5 \quad 3 \quad 4)$$

$$(1 \quad 2 \quad 3 \quad 5 \quad 4)$$

$$(1 \quad 2 \quad 3 \quad 4 \quad 5)$$

At most $n - 1$ reversals are needed.

Worst case scenario

$$\pi = (\ 5 \quad 1 \quad 2 \quad 3 \quad 4\ )$$

is sorted with 4 reversals, but it is possible to sort it with only 2.
In general:

$$\begin{array}{l} (\ n \quad 1 \quad \ldots \quad n-1\ ) \\ (\ n \quad n-1 \quad \ldots \quad 1\ ) \\ (\ 1 \quad 2 \quad \ldots \quad n\ ) \end{array}$$

In this case, the ratio between this and the optimal solution is $\frac{n-1}{2}$, which is not good.

Approximation Algorithms

- Find **approximate** rather than optimal solutions.
- The **approximation ratio** of an algorithm A on input π is:

$$r = \frac{A(\pi)}{\mathrm{OPT}(\pi)}$$

where:

- $A(\pi)$ is the solution by algorithm A.
- $\mathrm{OPT}(\pi)$ is the optimal solution.

Fixing the algorithm

- Why is the algorithm bad?
- Because it breaks some “good” parts.

$$\pi = (\ 5 \quad 1 \quad 2 \quad 3 \quad 4\ )$$

Applying the reversal $\rho(1,2)$ to put 1 in its position breaks the “good” connection $1, 2$

Adjacencies and Breakpoints

- For a permutation $\pi = (\ \pi_1 \quad \pi_2 \quad \cdots \quad \pi_n\ )$
  - Two elements π_i and π_{i+1} are **adjacent** if $|\pi_i - \pi_{i+1}| = 1$.
  - Otherwise, π_i and π_{i+1} form a **breakpoint**.

Example:

$$\pi = (\ 2 \quad 1 \quad 4 \quad 3 \quad 5 \quad 8 \quad 6 \quad 7\ )$$

- $(1, 4)$, $(3, 5)$, $(5, 8)$, and $(8, 6)$ are breakpoints.

The number of breakpoints in π, denoted by $b(\pi)$, is already a simple measure of rearrangement distance.

Reversals and Breakpoints

- To solve the reversal problem, it is common to add two new elements: 0 and $n+1$.

For instance,

$$\pi = (\ 1 \quad 3 \quad 2 \quad 4 \quad 6 \quad 5\ )$$

becomes

$$\pi = (\ \mathbf{0} \quad 1 \quad 3 \quad 2 \quad 4 \quad 6 \quad 5 \quad \mathbf{7}\ )$$

- This is to verify if the elements 1 and n form adjacencies (in the correct position) or breakpoints.

Reversals and Breakpoints

- The only permutation *without breakpoints* is the identity.

$$i = (\ 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7\ )$$

We can think of a rearrangement problem as *reducing the number of breakpoints*, until reaching 0.

Idea: Apply a reversal that reduce the maximum number of breakpoints.

Reversals and Breakpoints

Let's sort the permutation $\pi = (\, 2 \quad 3 \quad 1 \quad 4 \quad 6 \quad 5 \,)$.

$$\pi = (\, 0 \mid 2 \;\; 3 \mid 1 \mid 4 \mid \underline{\; 6 \quad 5 \;} \mid 7 \,) \qquad \textbf{(5 BPs)}$$

$$\pi = (\, 0 \mid \underline{\; 2 \quad 3 \mid 1 \;} \mid 4 \quad 5 \;\; 6 \;\; 7 \,) \qquad \textbf{(3 BPs)}$$

$$\pi = (\, 0 \;\; 1 \mid \underline{\; 3 \quad 2 \;} \mid 4 \quad 5 \;\; 6 \;\; 7 \,) \qquad \textbf{(2 BPs)}$$

$$\pi = (\, 0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6 \;\; 7 \,) \qquad \textbf{(0 BPs)}$$

Reviewing the Worst Case Example

$$\pi = (\ 5\ \ 1\ \ 2\ \ 3\ \ 4\ )$$

$$\pi = (\ 0 \mid 5 \mid 1\ \ 2\ \ 3\ \ 4 \mid 6)$$ **(3 BPs)**

There are 3 BPs, and each BP defines a reversal that fixes that BP.
Which of the 3 reversals eliminates more breakpoints?

Lower Bound with Breakpoint

- Each reversal can eliminate at most 2 breakpoints (one at the left end and one at the right end).
- Therefore, we have the following **lower bound** for the reversal distance $d(\pi)$:

$$d(\pi) \geq \frac{b(\pi)}{2}$$

Approximation Algorithm

If we can find an algorithm that always eliminates at least 1 BP per reversal, we have a **2-approximation algorithm**. Why?

Let $A(\pi)$ be the number of reversals that this algorithm needs to sort π. Since $A(\pi) \leq \mathrm{BP}(\pi)$, we have

$$d(\pi) \geq \frac{b(\pi)}{2} \geq \frac{A(\pi)}{2}$$

which implies

$$r = \frac{A(\pi)}{d(\pi)} \leq 2$$

Breakpoint Algorithm

Is is always possible to destroy at least one breakpoint? **NO! :-(**

Example:

$$\pi = (\ 0\ \ 4\ \ 5\ \ 6\ \ 1\ \ 2\ \ 3\ \ 7\ )$$

How can we know when these “bad cases” happen?

Increasing and Decreasing Strips

- Any permutation can be partitioned into **increasing strips** (overlined) and **decreasing strips** (underlined).

$$\pi = (\ \overline{0}\ \ \underline{2\ \ 1}\ \ \overline{3\ \ 4\ \ 5}\ \ \overline{7\ \ 8}\ \ \underline{6}\ \ \overline{9}\ )$$

- A strip with one element is increasing for 0 and $n + 1$, decreasing otherwise.

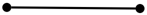
In the example, 0 and 9 are increasing strips, but 6 is a decreasing strip.

Decreasing Strip Reversal

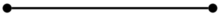
- **Observation**: If there is at least one decreasing strip, there is a reversal that reduces the number of breakpoints.

Proof: Consider the smallest element k in all decreasing strips, say, element 5. The element $k-1$ must be in a increasing strip. (Why?)

- Case 1: $(\ldots\ \underline{7\ 6\ 5}\ \ \ldots\ \ \overline{3\ 4}\ \ \underline{9}\ \ldots)$



- Case 2: $(\ldots\ \overline{3\ 4}\ \underline{9}\ \ \ldots\ \underline{7\ 6\ 5}\ \ \ldots)$



Dealing with Bad cases

All the strips in this permutation are *increasing*:

$$\pi = (\ \overline{0}\ \ \overline{4\ \ 5\ \ 6}\ \ \overline{1\ \ 2\ \ 3}\ \ \overline{7}\ )$$

- What can we do to guarantee that we can decrease the number of breakpoints?

We can apply a reversal in a increasing strip (the number of breakpoints does not change), and then we will have a decreasing strip. For instance,

$$\pi = (\ \overline{0}\ \ \underline{6\ \ 5\ \ 4}\ \ \overline{1\ \ 2\ \ 3}\ \ \overline{7}\ )$$

A Breakpoint reversal sorting algorithm

```
procedure BreakpointReversalSort($\pi$)
    while $b(\pi) > 0$ do
        if $\pi$ has a decreasing strip then
            $\rightarrow$ Apply a reversal that decreases $b(\pi)$
        else
            $\rightarrow$ Apply a reversal in an increasing strip
        end if
    end while
end procedure
```

Approximation Factor

- BreakpointReversalSort is a 4-approximation algorithm

Proof:

- In the worst case, this algorithm will need $2b(\pi)$ reversals to sort a permutation. (Why?)

How to improve?

- If there is a decreasing strip, there is a “good” reversal.
- The bad case is that when there are no decreasing strips.
- Can we always apply a reversal that decreases $b(\pi)$ **and** always mantains a decreasing strip?
- No, but there is the following result:

Theorem: If all reversals on π that reduce $b(\pi)$ create a permutation without decreasing strips, then there exists a reversal that reduces $b(\pi)$ by two [Kececioglu and Sankoff, 1993].

2-Approximation by Kececioglu & Sankoff

```
procedure KS_ReversalSort($\pi$)
    while $b(\pi) > 0$ do
        → Apply a reversal that decreases $b(\pi)$ by the largest amount, preferring reversals that leave a decreasing strip.
    end while
end procedure
```

Approximation Algorithms

- 2-Approximation [Kececioglu and Sankoff, 1993].
- 7/4-Approximation [Bafna and Pevzner, 1996]
- 3/2-Approximation [Christie, 1998]
- 1.375-Approximation [Berman et al., 2002]

References I

Bafna, V. and Pevzner, P. (1996). Genome Rearrangements and Sorting by Reversals. SIAM Journal on Computing *25*, 272–289.

Berman, P., Hannenhalli, S. and Karpinski, M. (2002). 1.375-Approximation Algorithm for Sorting by Reversals.

Christie, D. A. (1998). A 3/2-approximation Algorithm for Sorting by Reversals. In Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms SODA '98 pp. 244–252, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Kececioglu, J. and Sankoff, D. (1993). Exact and approximation algorithms for the inversion distance between two chromosomes. In Combinatorial Pattern Matching vol. 684/1993, pp. 87–105. Springer Berlin / Heidelberg.

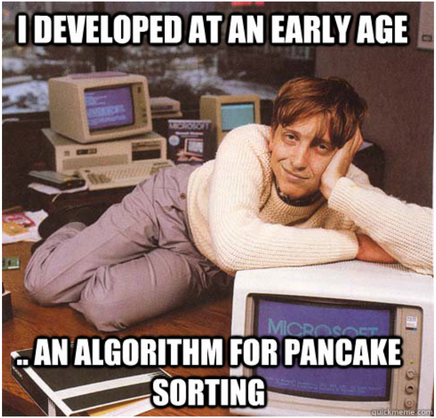
Pancake Flipping Problem

Given a stack of pancakes of different sizes, how can we flip the pancakes to arrange them from smallest to largest?

This is called the **Prefix Reversal Distance**.

Pancake Flipping Problem

Christos Papadimitriou and Bill Gates proposed an approximation algorithm in a paper from 1979, that was only improved recently, in 2008.


I DEVELOPED AT AN EARLY AGE
.. AN ALGORITHM FOR PANCAKE SORTING
MICROSOFT
quickmeme.com

Burnt Pancake Flipping

Similar to the previous problem, but now the bottom of each pancake is burnt, and the sort is completed only when all pancakes are ordered and the burnt side is down.

This is called the **Signed Prefix Reversal Distance**.

`http://www.sciencedirect.com/science/article/pii/0166218X94000093`

One of the authors, David S. Cohen, is a writer for Simpsons and Futurama.