

Genome Rearrangements

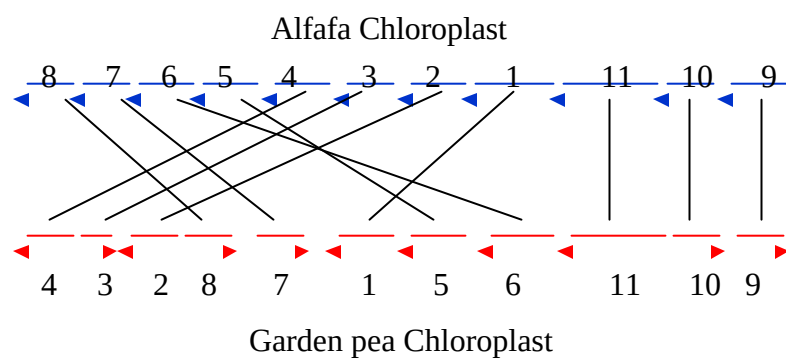
CS638

Thomas Anantharaman

tsa@biostat.wisc.edu

October 2000

Biological Background



Block Reversals

- Evolution is certain genomes (chloroplast and mitochondria) are dominated by Block reversals:



Reconstructing Block Reversals

- Problem of **Block Reversal Sorting**
- Given : Two genomes and their relationship in terms of DNA blocks that are oriented and/or permuted in different ways.
- Find : The minimum number of Block Reversals that could cause one genome to evolve into the other genome.

Formalizing the problem

- Number the blocks in one genome from left to right and label their orientation as left to right:

$$(L, \vec{1}, \vec{2}, \vec{3}, \vec{4}, \vec{5}, \vec{6}, R)$$

- Orient and number the blocks of the other genome relative to the first one
(Extended Oriented Permutation)

$$(L, \vec{2}, \vec{3}, \vec{1}, \vec{6}, \vec{5}, \vec{4}, R)$$

Reversal Operation

- Given an **Oriented Permutation** of size k and an interval $(i \dots j)$:

$$(L, a_1, a_2, \dots, \underline{a_i \cdots a_j}, \dots, a_k, R)$$

- **Reversal Operation** (i, j) changes this to:

$$(L, a_1, a_2, \dots, \underline{\bar{a}_j \cdots \bar{a}_i}, \dots, a_k, R)$$

Reversal Sorting Outline

- Use **Breakpoints** to simplify problem into smaller one.
- Construct **Reality-Desire** diagram.
- Find cycles in **RD** diagram and classify into **Good Cycles** and **Bad Cycles**.
- Find **Interleaving Graph** of Cycles and determine its connected **Components**
- Classify **Components** into 4 classes: **Good**, **NonHurdle**, **Hurdle**, **Superhurdle**.
- Apply Algorithm to find next Reversal.

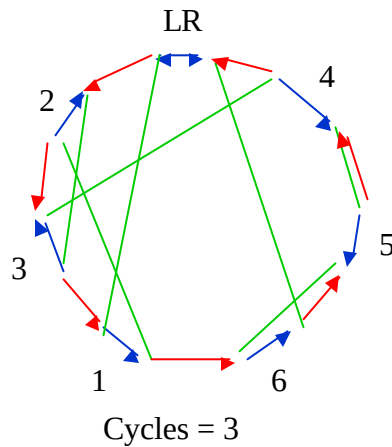
Breakpoints

- Definition : A **Breakpoint** exists between two blocks x, y of an **Oriented Permutation** if neither x, y nor $\bar{y}, \bar{x}$ are in the final order and orientation
- Example: $(L, \vec{2}, \vec{3}, \vec{1}, \vec{6}, \vec{5}, \vec{4}, R)$

- Consecutive blocks without a breakpoint can be treated as a single block.

Reality-Desire Diagram

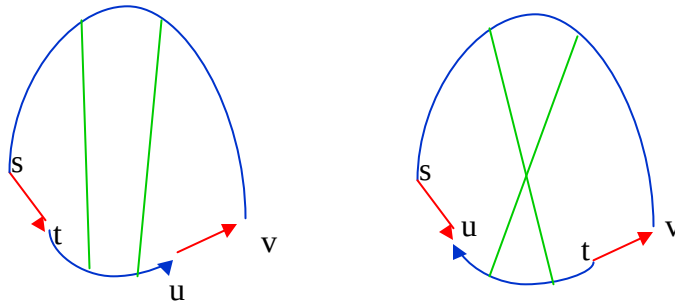
$(L, \vec{2}, \vec{3}, \vec{1}, \vec{6}, \vec{5}, \vec{4}, R)$



Effect of Reversal on RD Diag.

- Ends of each reversal are **Reality** edges in **RD Diagram**, eg (s,t) and (u,v)
- Remove **Reality** edges (s,t) and (u,v), and add **Reality** edges (s,u) and (t,v)
- Reverse the section of the circle involved without changing any edges.

Effect of Reversal on RD Diag.

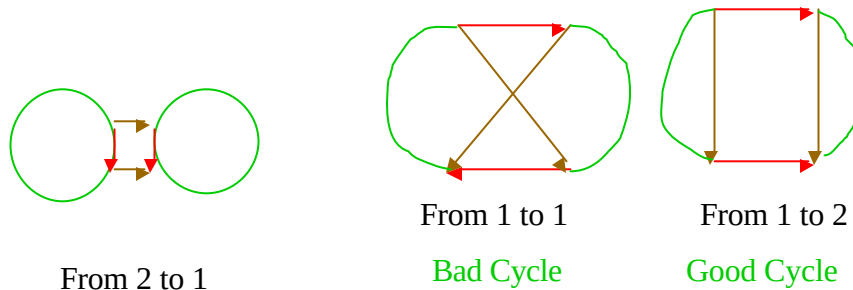


Effect of Reversal on RD Circles

Theorem : Let a Reversal remove **Reality** edges **f** and **g** . Then

1. If **f** and **g** were in different circles, number of circles decrease by 1.
2. If **f** and **g** were in same circle with same orientation, number of circles are unchanged.
3. If **f** and **g** were in same circle with opposite orientation, number of circles increase by 1.

Effect of Reversal on RD Circles



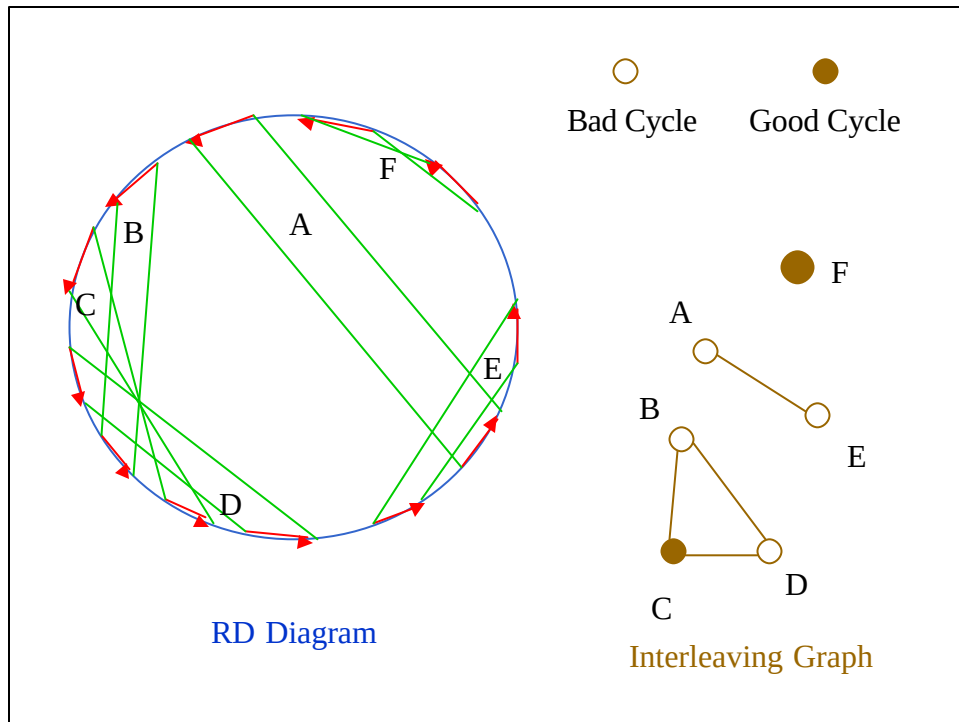
Cycles change by at most 1 per reversal

Hence minimum reversals is :

$$n + 1 - C(\alpha)$$

Interleaving Graph

- Two circles Interleave if their **Desire** edges intersect anywhere.
- The Interleaving Graph shows which circles Interleave.
- The Interleaving Graph breaks into connected **Components**



Types of Components

- Good Component : Includes at least one Good Cycle
- Bad Component : No Good Cycle
 - Non-Hurdle : Separates two other Bad Components
 - Super Hurdle : Unique Bad Component on one side of a Non-Hurdle.
 - Simple Hurdle : Any other Bad Component
- Fortress Condition: There exist an odd number of Super Hurdles and no Simple Hurdles

Minimum Number of Reversals

Theorem :

$$d(\alpha) = n + 1 - c(\alpha) + h(\alpha) + f(\alpha)$$

Minimum
Reversals
needed

Number of
hurdles

1 IFF
Fortress
Condition
is true

Good Cycle

- A Reversal based on the two divergent **Reality Edges** of a **Good Cycle** will reduce number of circles, but may turn its **Component** into a **Bad Component**.
- No effect on other **Components**

Theorem : If there are any **Good Cycles** there exists at least one pair of divergent **Reality Edges** whose Reversal will not create a **Bad Component**

Bad Components

A Simple Theorem:

- **Bad Components** contain at least 2 **Bad Cycles**.
- A Reversal in a **Bad Cycle** will twist at least one other cycle in the same **Component** turning it into a **Good Cycle**
- No effect on other **Components**

Bad Component

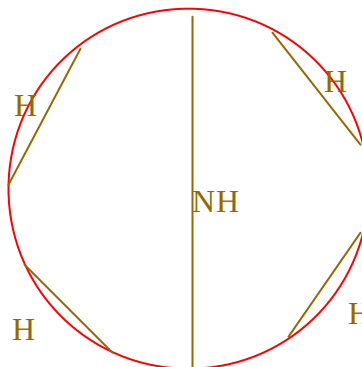
- A Reversal in a **Bad Component** will not change the number of cycles, but if we choose a **Hurdle**, this will become a **Good Cycle** and reduce $h(\alpha)$
- However if it is a **Super-Hurdle** another **Hurdle** is created.
- Also we might end up creating a **Fortress**, going from 4 to 3 **Hurdles**.
- Solution : Pick a **Simple-Hurdle** when $h(\alpha)$ is odd

Hurdles and Non-Hurdles

- Another Simple Theorem:
- A reversal using **Reality Edges** from two different **Bad Components A and B** will twist them and any **Non-Hurdle C**, that separates **A and B** turning **A,B and C** into a good components.
- This increase Cycles by 1 but decreases $h(\alpha)$ by 2.

Hurdles and Non-Hurdles

- Problem : Changing Two **Hurdles** into **Good Components** can turn a **Non-Hurdle** into a hurdle.



Hurdles and Non-Hurdles

- Merge only **Opposite Hurdles**:
 - Number of Hurdles is even.
 - Pick Hurdles on opposite sides of **RD** circle.

Sorting Reversal Algorithm

- Goal : Given a Oriented Permutation α find the next Reversal :
- If there is a good **Component** in $RD(\alpha)$
 - Try every reversal for pair of divergent **Reality Edges** in a **Good Cycle** of this **Component** : If no new bad component is created, return this reversal.
- If $h(\alpha)$ is even use **Edges** of two **Opposite** hurdles.
- If $h(\alpha)$ is odd and there is a simple hurdle pick any two **Edges** from any simple hurdle.
- Fortress : use **Edges** from any two hurdles.