Using group theory to study the inversion distance problem in bacterial genomics

Attila Egri-Nagy

School of Computing and Mathematics
University of Western Sydney

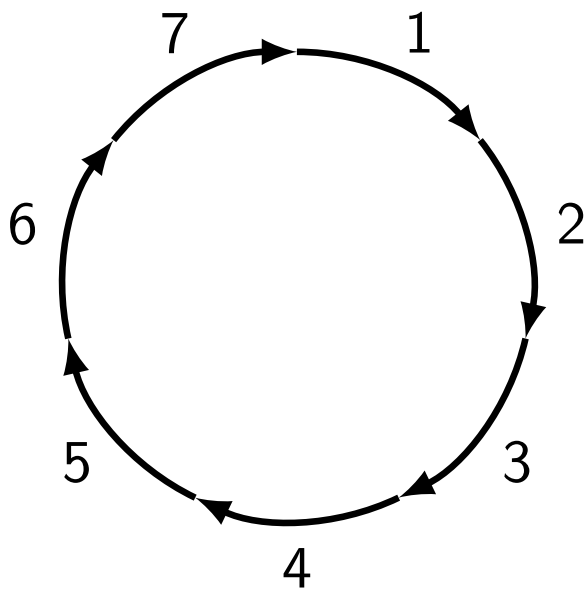
November 2011

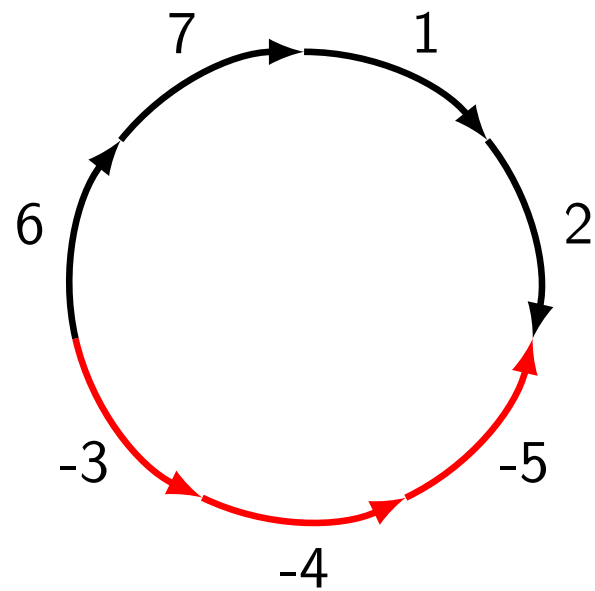
Aims of this talk

- Exploiting more group theory in combinatorial genomics
- Creating a more biologically plausible model of bacterial genomes by considering fixing the terminus

Inversions (reversals) as signed permutations

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & -5 & -4 & -3 & 6 & 7 \end{pmatrix}$$





Distance between Genomes

We measure the distance between two genomes by the minimal number of inversion events needed for going from one genome to the other one.

With group theory terminology: finding shortest paths in the Cayley graph of the group.

Problems:

- The Cayley graph is way too big to search exhaustively.
- The distance depends on the generators.

Fixing the terminus

The replication process starts from the origin, proceeds in two directions and ends at the terminus. So there is time penalty if the terminus is not in the middle relative to the origin.

Two approaches:

- Studying the subgroup of the hyperoctahedral group which fixes the terminus. The subgroup is the stabilizer of the terminus, which is an index n subgroup.
- Restricting the size of the inversions.

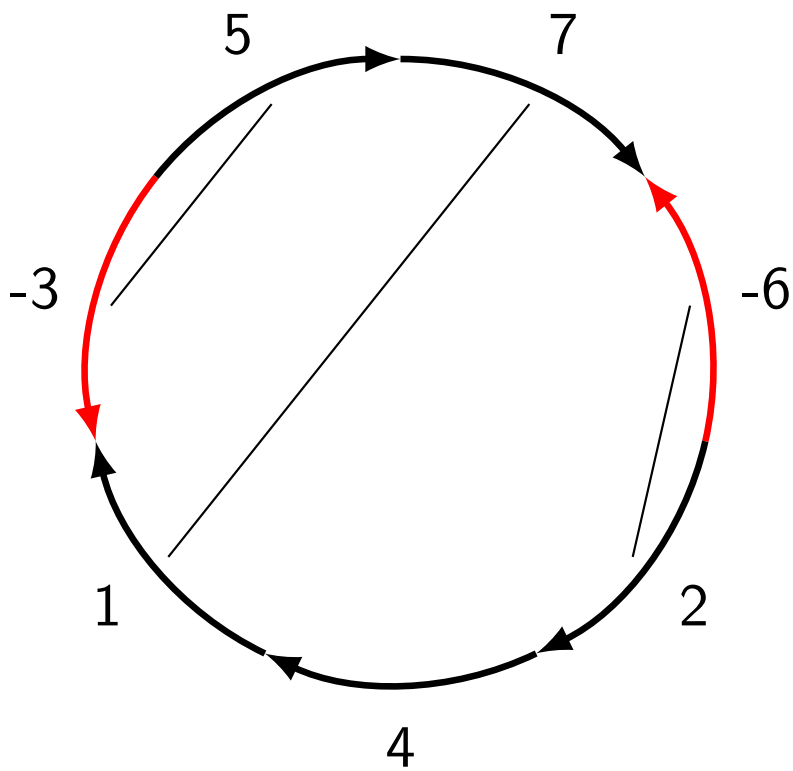
An algorithm

By fixing the terminus we reduce the group size from $2^n \cdot n!$ to $2^n \cdot (n-1)!$, which is a very small change.

Honours student Terence Bowers came up with the following algorithm:

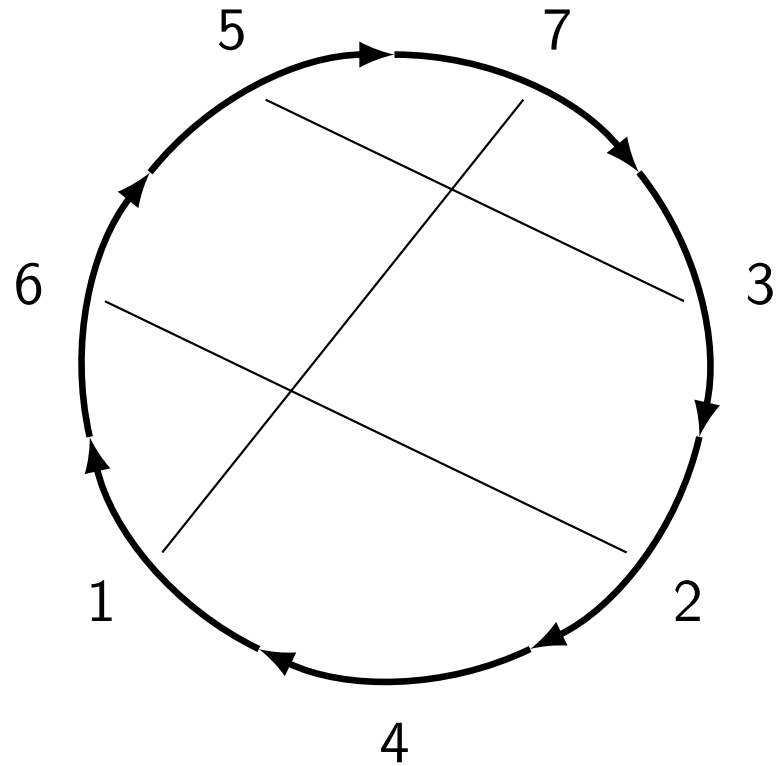
1. Separating complementary pairs
2. Ordering regions on the sides
3. Moving regions to the correct sides
4. Orienting regions

A random genome with 7 regions



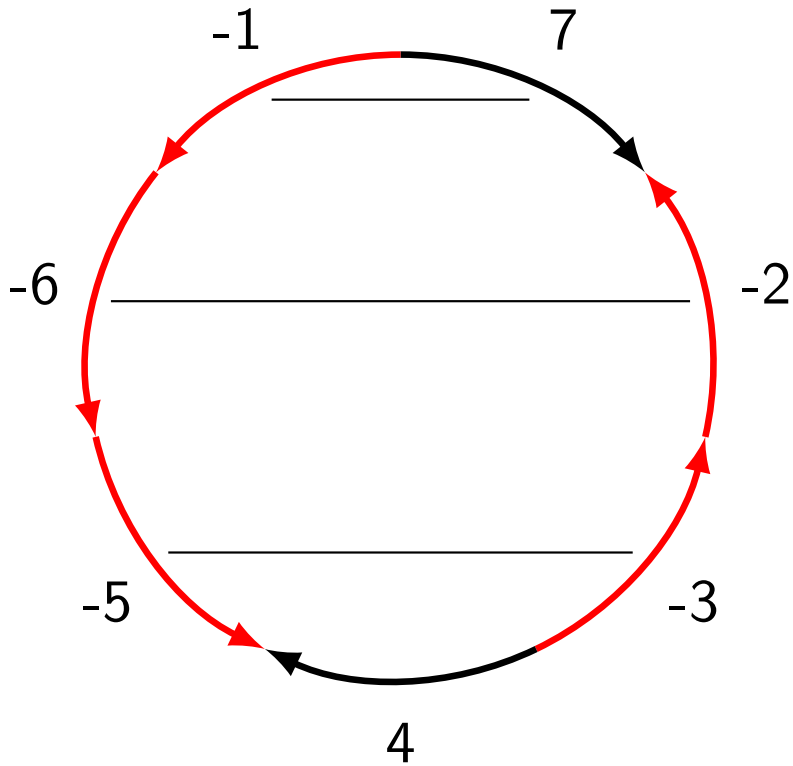
Complementary pairs separated

Top-down, maximum $n - 2$ inversions needed.



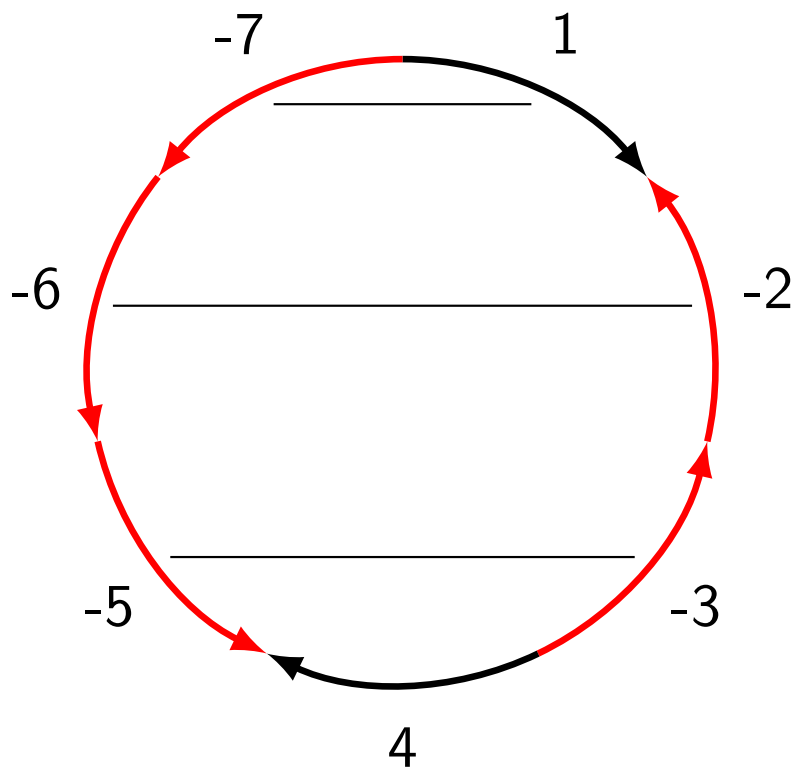
Sides sorted

Simple sorting on both sides. Again maximum $n-2$ steps.



Complementary pairs exchanged

Maximum $\frac{n}{2}$ steps needed.



The final orientation needs n inversions in the worst case.

Evaluation

Advantages

- Linear time algorithm.
- It shows that there is a normal form TSTS for any permutation.

Disadvantages

- The generator set consists of inversion of any size.
- It is not yet known how good is this distance measure.

The first step does not scale

The upper bound for the number of required inversions is $\frac{7n}{2} - 4$. So the first contributes roughly $1/3$ of the distance value, while it turns out that for bigger n the genomes with nonseparated pairs tend to dominate the search space (the group).

Group	Size	#nonseparated	ratio
FT_3	16	0	0%
FT_5	768	256	$1/3 \approx 33\%$
FT_7	92160	55296	$3/5 = 60\%$
FT_9	20643840	15925248	$27/35 \approx 77\%$
FT_{11}	7431782400	6488064000	$55/63 \approx 87\%$

The idea

What is the 'distance' between 391352 and 18781? We do not see the answer immediately, but we have an algorithm for calculating the answer on the coordinatized form of these numbers.

$$\begin{array}{rcccccc} & (3, & 9, & 1, & 3, & 5, & 2) \\ - & (0, & 1, & 8, & 7, & 8, & 1) \\ \hline & (3, & 7, & 2, & 5, & 7, & 1) \end{array}$$

Our decimal number notation system uses modulo 10 counters (cyclic groups) forming a hierarchical structure.

Algorithms that are in accordance with group structure

Hierarchical coordinatizations (Lagrange decomposition)

	Natural Numbers	**Permutation Groups**
Building Blocks	Primes	Simple Groups
Composition	Multiplication	(Sub)Wreath Product
Precision	Equality	Isomorphism
Uniqueness	Unique	Different Decompositions

Also known as Krasner-Kaloujnine embeddings, or simply the monomial map.

Lagrange Coordinates

Theorem (Lagrange Coordinatization for Transitive Actions)

Let G act on X transitively. Let $G = G_1 > \ldots > G_n = H$ be a subgroup chain for G, where H is the stabilizer of some element of X. Then (X, G) admits the following coordinatization

$$(X, G) \cong \bigsqcup_{\substack{\mathcal{L} \\ 1 \leq i < n}} (G_i/G_{i+1}, G_i/\mathrm{Core}_{G_i}(G_{i+1}))\,.$$

Coordinatized distance

By giving subgroup chains (relatively easy by using stabilizers) we can create a coordinatization. In case the chain is a subnormal chain the components are guaranteed to be smaller, so we can do the distance calculation in a smaller search space.

```
Lagrange decomposition of:
C2 x ((((C2 x C2 x C2 x C2 x C2) : A6) : C2) : C2)
1 2 C2
2 |-360 A6
3 |-|-2 C2
4 |-|-|-16 C2 x C2 x C2 x C2
5 |-|-|-|-2 C2
6 |-|-|-|-|-2 C2
```

Coordinatized distance – example

```
gap> g1 := Random(FT7);
(1,9,14,5)(2,10,13,6)(3,12,4,11)
gap> SignedPermutation(g1);
[ 5, -6, 1, 4, -7, 2, -3 ]
gap> Perm2CascadedState(ld,g1);
C(1,161,2,13,1,2)
gap> g2 := Random(FT7);
(1,4,2,3)(5,12,10,13,6,11,9,14)(7,8)
gap> SignedPermutation(g2);
[ -2, 1, -6, -4, -7, 5, -3 ]
gap> Perm2CascadedState(ld,g2);
C(1,317,1,14,1,1)
```

Another possibility – short inversions

Instead of fixing the terminus artificially we can examine to what extent the minimal words move regions that are already in place.

For inversions of size 2 the situation is similar to the bubble sort algorithm.

Thank You!

Group decomposition software:

`http://sgpdec.sf.net`