17 Non-collinear alignment

This exposition is based on:

1. Darling, A.E., Mau, B., Perna, N.T. (2010) *progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement.* PLoS One 5(6):e11147.
2. Darling, A.C.E., Mau, B., Blattner, F.R., Perna, N.T. (2004) *Mauve: multiple alignment of conserved genomic sequence with rearrangements.* Genome Res 14:1394–1403.

We will introduce the concept of non-collinear alignment and utilize the approach implemented in the program progressiveMauve (pMauve) as an example.

17.1 Motivation

Non-collinear alignment allows us to compare the organization and structure of genomes in addition to local changes.

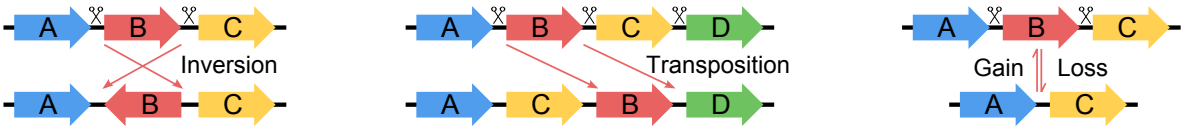
The genomic structure provides information about large-scale evolutionary events. Such events cause identical regions to be fragmented and reodered in the comparison of two or more genomes, and some fragments are possibly missing or occur in multiple copies.

The large-scale evolutionary events include:

Inversion (= reversal): Two breakpoint operation. Replace a fragement by its reverse complement. This mutation can happen through erroneous repair of a loop.

Transposition (= rearrangement): Three breakpoint operation. Move a fragment to a different position.

Gain and loss : Insertion and deletion of larger fragments. Gain is a result of horizontal transfer. Loss can also happen through erroneous repair of a loop.



If we allow large-scale changes when comparing genomes, the "alignment" will be split into several *blocks*, often called synteny blocks or locally collinear blocks (LCBs).

Synteny: (greek: syn = together, taenia = ribbon) physical co-localization of genes on the same chromosome / conserved gene order among different species / collinearity

Non-collinear alignment approaches combine the two tasks of synteny mapping (= identifying the ends of blocks) and detailed alignment of the blocks.

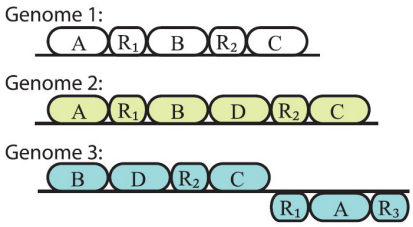
Similar to collinear genome alignment, the approaches commonly use an anchoring heuristic, i.e. start by computing local alignments of the input sequences. But in contrast to collinear alignment, removal of spurious matches by computation of a trace is not possible because of inversions and transpositions. The central step of non-collinear alignment is to choose the subset of local alignments that is most probable to display all homologies of the genomes.

Different types of non-collinear multiple genome alignment approaches exist:

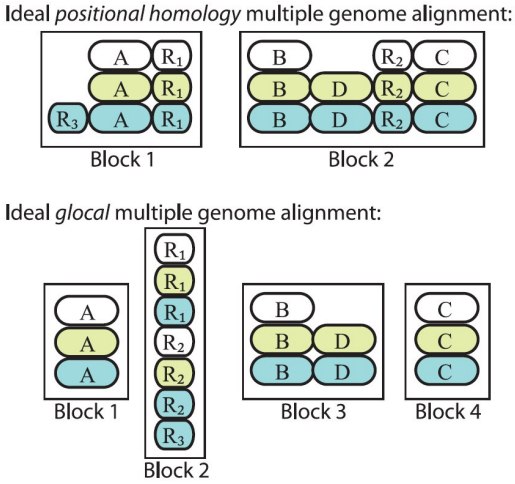
Reference-guided: One genome is selected as a reference. Detects homologies only if conserved in the reference genome, e.g. a human-mouse-rat alignment with human as a reference will not display homologies among mouse and rat that are not conserved in humans. The blocks are usually ordered according to their order in the reference genome.

All homologies ("glocal"): The goal is to identify all homologies – orthologs and paralogs, i.e. including all copies of a repetitive segement. The word "glocal" is a hybrid of global and local, underlining the approach of computing a global alignment that displays all significant local similarities of the genomes.

Positional homology: The objective is to align only the positionally conserved copy of repetitive segments. Paralogs, are not aligned, i.e. the alignment does not contain information about duplications.



Source: Darling *et al.* (2010)



In the following, we will describe the approach used by the program pMauve. pMauve computes positional homology alignments and uses local multiple alignments as anchors. For the central step of choosing a subset of local alignments it uses a sum-of-pairs breakpoint score to greedily eliminate breakpoints.

17.2 pMauve: The algorithm

Outline of pMauve:

1. Local multiple alignment (= computation of anchors)
2. Guide tree computation via a coverage distance matrix
3. Progressive alignment
   - Anchor selection: sum-of-pairs greedy breakpoint elimination
   - Recursive anchor search
   - Global anchored alignment
4. Split the alignment of unrelated sequence

17.3 Local multiple alignment in pMauve

In principle, any local alignment method (pairwise or multiple) that is fast enough on a genomic scale could be used to compute anchors.

Note that there is no match refinement step in pMauve, overlapping matches are trimmed. This suggests to apply a method that identifies only maximal unique matches, which is also justified by the goal of computing positional homology alignments.

The original program Mauve used multiMUMs (maximual unique matches among all input sequences) as anchors, which is very fast but less sensitive.

pMauve computes local alignments of multiple sequences by identifying and extending q-grams that are common to ≥ 2 and unique within the sequences. Families of gapped, palindromic q-grams are used, the extension is ungapped.

17.4 Guide tree computation in pMauve

pMauve computes a Neighbor Joining tree from a distance matrix that is based on the genomes' coverages by the initial set of local multiple alignments.

The *projection of a multiple alignment* onto a pair of genomes g_i and g_j is an alignment that consists of the two alignment rows i and j reduced by all gap-only columns.

A genomic position is considered *covered* with respect to another genome if it is contained in the pairwise projection of a local alignment and identical in the other genome.

The values in the coverage distance matrix are the fractions of uncovered positions in a pair of genomes.

17.5 Progressive alignment in pMauve

For each internal node of the guide tree pMauve performs the following three steps:

Anchor selection: Selection of a subset of local alignments that maximizes a sum-of-pairs breakpoint score. pMauve iteratively removes local alignments in order to eliminate breakpoints from the alignment.

Recursive anchor search: Local alignment with more sensitive parameter settings between anchors within blocks, and outside blocks.

Global anchored alignment: The selected local alignments can be combined to compute global collinear alignments of the blocks. By using the local alignments as anchors for global alignment the search space is much smaller.

Anchor selection:

We will first look at the sum-of-pairs breakpoint score.

Given a set of local multiple alignments A on a set of genomes $g_1, \ldots, g_n$. The *projection of A* onto a pair of genomes g_i and g_j is a the subset of alignments from A that contain a row for genomes g_i and g_j, projected onto g_i and g_j. We denote this projection by $\pi_{ij}(A)$.

A *pairwise block* (locally collinear block, LCB) is a subset $L_{ij} \subseteq \pi_{ij}(A)$ of projected local alignments that occur in the same order and orientation in the pair of genomes g_i and g_j. Let $\mathcal{L}_{ij} = \{L_{ij}^1, \ldots, L_{ij}^K\}$ be the minimal partition of a projection $\pi_{ij}(A)$ into disjoint pairwise blocks.

Note that for partitioning into disjoint pairwise blocks, local alignments cannot overlap.

For the current internal node η of the guide tree, let $R(\eta)$ and $L(\eta)$ be the set of leaf nodes in the right and left subtree. The sum-of-pairs breakpoint score is

$$\mathcal{S} = \sum_{i \in R(\eta)} \sum_{j \in L(\eta)} S(\pi_{ij}(A)) - \beta \cdot |\mathcal{L}_{ij}|$$

$S(\pi_{ij}(A))$ is the sum of alignment scores of the alignments in $\pi_{ij}(A)$, β is a breakpoint penalty, and $|\mathcal{L}_{ij}|$ is the size of $\mathcal{L}_{ij}$.

$|\mathcal{L}_{ij}|$ is in fact the pairwise breakpoint distance among the genomes g_i and g_j.

For each breakpoint, pMauve applies a penalty β.

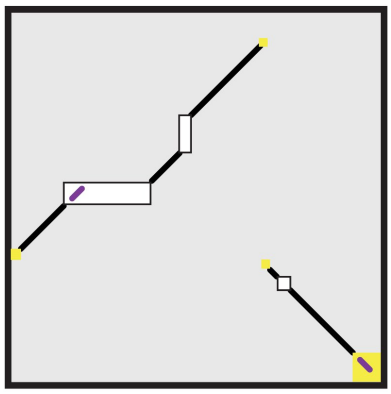
β is scaled according to the expected rates of rearrangement among the pair of genomes, i.e. accounts for variability of rearrangement rates.

pMauve applies a greedy breakpoint elimination heuristic to maximize the sum-of-pairs breakpoint score $\mathcal{S}$. It iteratively selects pairwise blocks and eliminates the corresponding local alignments:

1. Find the pairwise block L_{ij}^k that results in the biggest increase in $\mathcal{S}$ if all local alignments contained in L_{ij}^k are removed from A.
2. If such a block exists, remove the coreesponding local alignments from A.
3. Recompute $\mathcal{S}$, and go back to *1.*

The algorithm terminates if S cannot be further improved by removal of a single block.

Recursive anchor search:



pMauve iterates greedy breakpoint elimination with increasingly sensitive searches for local alignments within and outside blocks. It chooses the parameters automatically according to the size of the unaligned region.

Black lines are previous local alignments. Yellow boxes are regions outside blocks, white boxes are regions between anchors within blocks. Purple lines are newly found local alignments.

Global anchored alignment:

All blocks of the alignment are independently subjected to anchored alignment using a collinear alignment method.

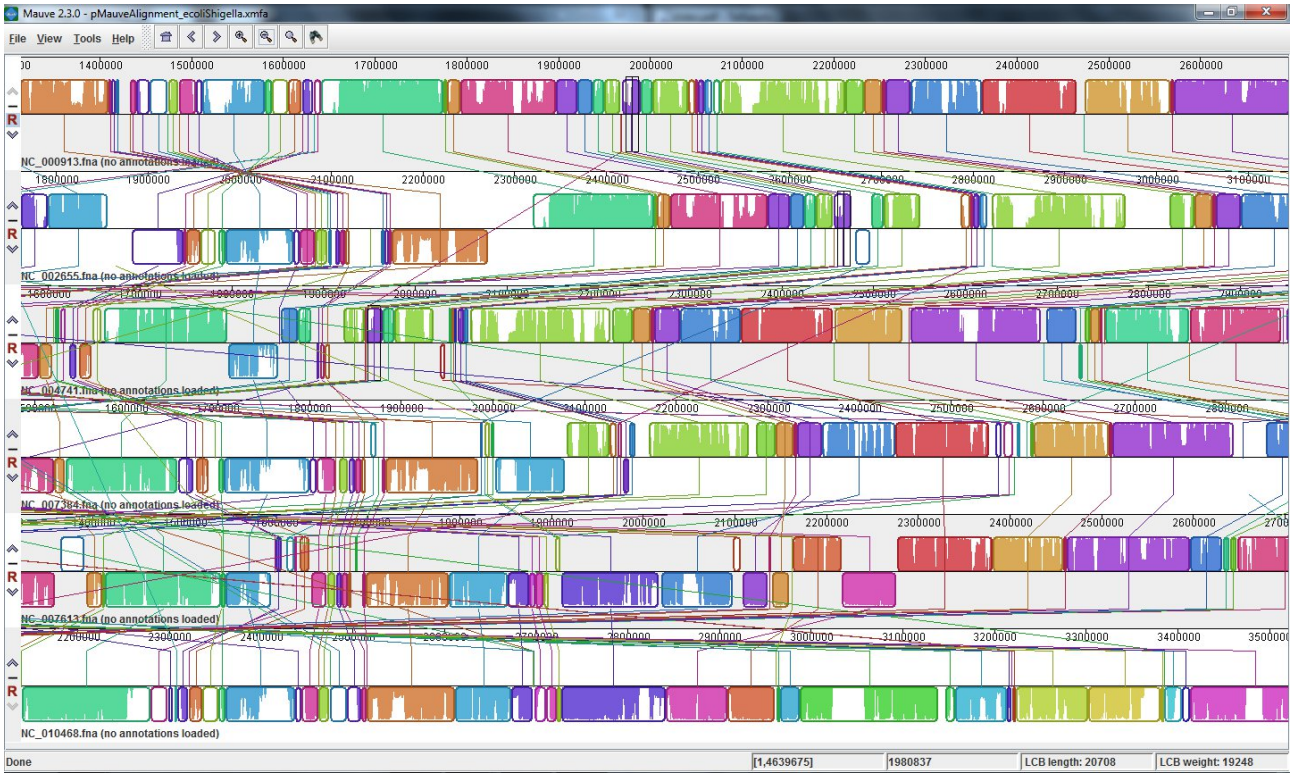
To further improve the alignments, an iterative refinement step is added in which a multitude of alternative guide trees is used.

17.6 Splitting the alignment of unrelated sequence in pMauve

The global anchor alignment step assumes sequence between anchors to be homologous without knowing if homology exists. Therefore, it may force unrelated sequence to be aligned.

pMauve applies an HMM posterior decoder to classify alignment columns either homologous or unrelated. Those regions of the alignment that are found to be unrelated are split and removed from the alignment.

17.7 Example alignment by pMauve



17.8 Other approaches

Positional homology aligners:

Mugsy: (Anguioli & Salzberg, 2010) Computes MUMs with Nucmer, refines segment matches, applies a min-cut max-flow algorithm to partition local alignments into blocks, aligns blocks separately with Seqan::T-Coffee.

TBA: (Blanchette *et al.*, 2004) Computes pairwise local alignments with BlastZ, and constructs multiple alignments by a process of merging and filtering.

Glocal alignment approach:

S-LAGAN: (Brudno *et.al.*, 2003) Is a pairwise aligner. Uses Chaos to compute local alignments. Computes a 1-monotonic conservation map to identify blocks, identifying duplications in only one sequence. An extension to multiple alignments has been published in Dubchak *et al.*, 2009.

Graph-based approaches:

Several approaches exist that construct a graph from the input genomes and a given set of local alignments. The graphs differ from each other and have different properties, but the approaches all define modifications on the respective graph to select a subset of local alignment to be included in a global non-collinear alignment.

ABA: (Raphael *et al.*, 2004) Removes certain substructures (small cycles) in an "A-Bruijn" graph.

Enredo: (Paten *et al.*, 2008) The Enredo graph is modified by biologically motivated operations, such as removal of pseudogenes.

Cactus: (Paten *et al.*, 2011) Removes and adds local alignments iteratively in a cactus graph that is constructed from the input genomes in several steps.

17.9 Summary

- Non-collinear alignment captures large-scale evolutionary events, such as inversions, transpositions, and gain/loss, in addition to single nucleotide substitutions and small indels.
- Similar to collinear genome alignment, the first step in non-collinear alignment is usually computation of local alignments.
- The central step of non-collinear alignment approaches is choosing a subset of local alignments to be included in the global alignment of the genomes, e.g. greedy breakpoint elimination in pMauve to optimize a sum-of-pairs breakpoint score.