Tools for whole genome alignment & phylogeny

Rebecca Dikow
SIBG Postdoctoral Fellow

Why genome alignment?

- Find primary homologies not dependent on annotation
- Assess collinearity of entire genomes
- Design markers (*e.g.* UCEs)

The sequence alignment problem

- Multiple sequence alignment algorithms rely on dynamic programming (e.g. Smith & Waterman) or hashing (identify exact kmer matches, e.g. BLAST) to break the problem down into sub-problems.
- As sequence length increases, most dynamic programming does not scale well $O(n^2)$ while hashing techniques can often be O(n).

Why is genome alignment different than multiple sequence alignment?

- Very long sequences -> very long analyses
- Even very closely related genomes have been subject to rearrangement
  - Multiple sequence aligners (*e.g.* MAFFT, MUSCLE) assume (force) collinearity

Genome alignment programs

- MUMmer (http://mummer.sourceforge.net)
- Harvest: ParSNP (http://harvest.readthedocs.org/en/latest/)
- LASTZ (http://www.bx.psu.edu/~rsharris/lastz/)
- progressiveCactus (https://github.com/glennhickey/progressiveCactus)
- progressiveMauve (http://darlinglab.org/mauve/mauve.html)

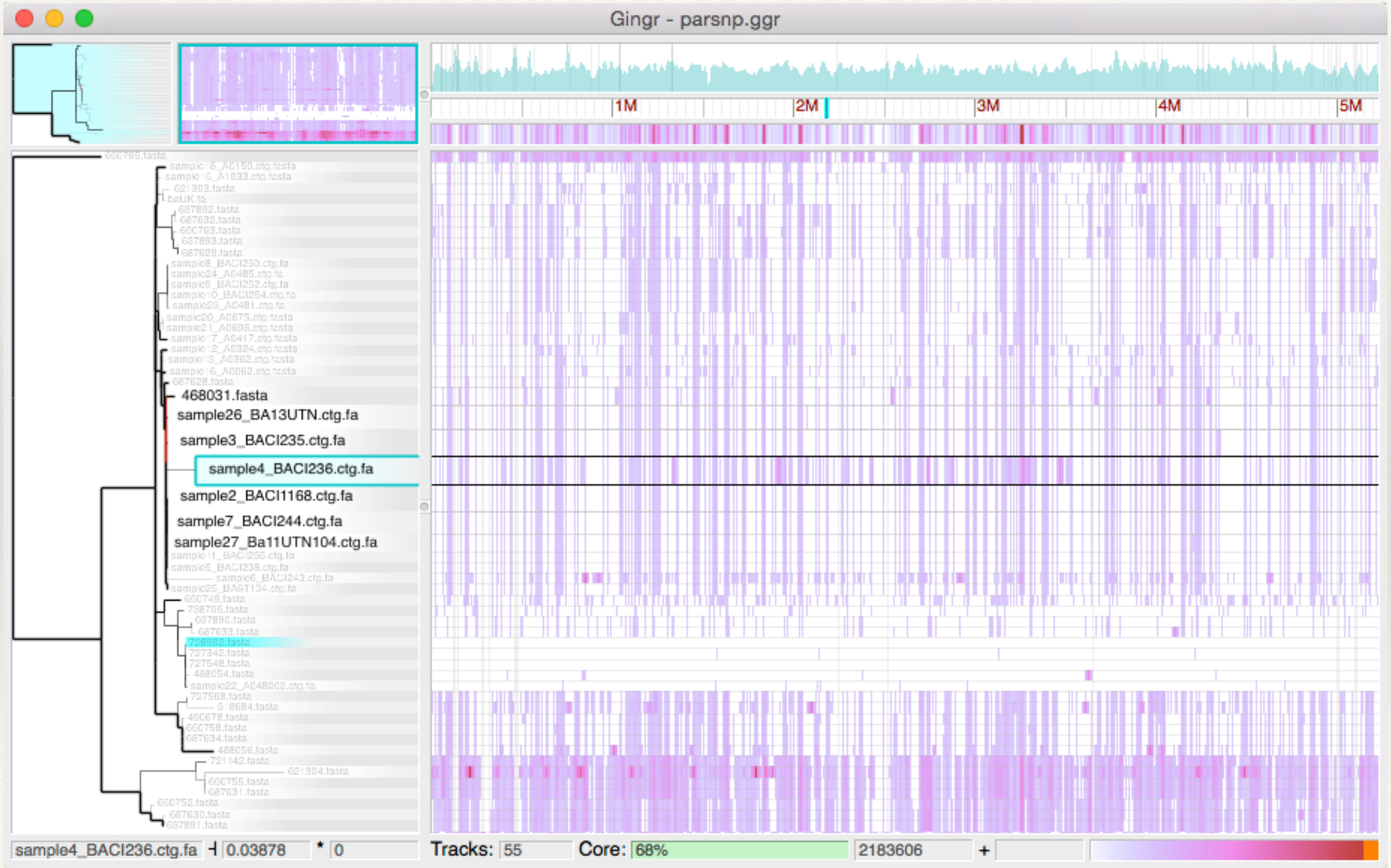
MUMmer (Delcher et al. 1999)

- MUMmer was one of the first genome alignment programs and many others use its concepts and algorithm.
- First example was two strains of *Mycoplasma tuberculosis*.
- Method:
  - 1. Find all **MUMs: maximal unique matches**. A MUM is a subsequence that occurs exactly once in Genome A and once in Genome B and is not contained in any longer such sequence (both strands).
  - 2: Sort the matches found in the MUM alignment according to their position in Genome A and extract the longest possible set of matches that occur in the same order in both genomes.
  - 3: Close gaps: Gaps are interruptions in the MUM-alignment which can be a SNP, insertion, highly polymorphic region, or a repeat.

Harvest: Parsnp (Treangen et al. 2014)

- Parsnp was designed to align the core genomes of hundreds to thousands of bacterial genomes within a few minutes to few hours. Input can be both draft assemblies and finished genomes, and output includes variant (SNP) calls, core genome phylogeny and multi-alignments.
- **Core-genome alignment is inherently more scalable because it ignores subset relationships.**
  - **i.e. only considers genes contained in all input taxa.**
- Method:
  - 1: Identifies MUMs.
  - 2: Uses MUMs to both recruit similar genomes and anchor the multiple alignment.
- **Parsnp is designed for intraspecific alignments and requires input genomes to be highly similar (for example, within the same subspecies group or > =97% average nucleotide identity).**

Parsnp (Treangen et al. 2014)



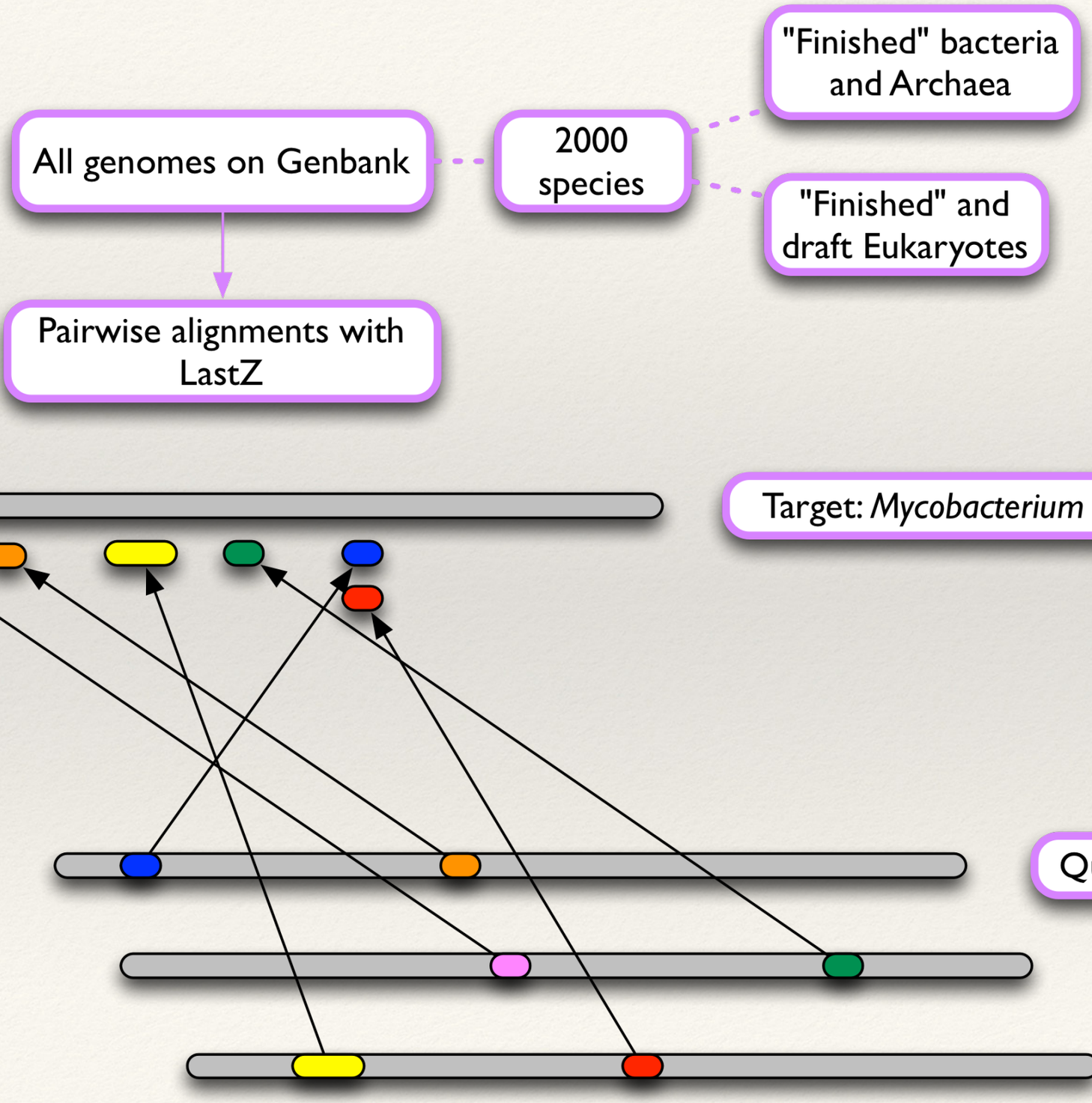
LASTZ (Harris, 2007)

- LASTZ is designed to preprocess one sequence or set of sequences (**target**) and then align several **query** sequences to it.
- Method:
  - 1: Read the target into memory, and use that to build a seed word position table to quickly map any word (kmer) in the target to all of the positions where it appears.
  - 2: Read each query sequence in turn. We examine the word starting at each base in the query and use the position table to find matches, called **seeds**, in the target.
  - 3: Extend seeds to longer matches called **HSPs** (high-scoring segment pairs) and filtered based on score.
  - 4: The HSPs are chained into the highest-scoring set of syntenic alignments, and then reduced to single locations called **anchors**.
  - 5: The anchors are then extended to local alignments (which may contain gaps) and again filtered by score, followed by back-end filtering to discard alignment blocks that do not meet specified criteria for certain traits.
  - 6: We then interpolate, repeating the entire process at a higher sensitivity in the holes between the alignment blocks.
  - 7: Then these steps are repeated with the reverse complement of the query sequence, before moving on to the next sequence in the query file.

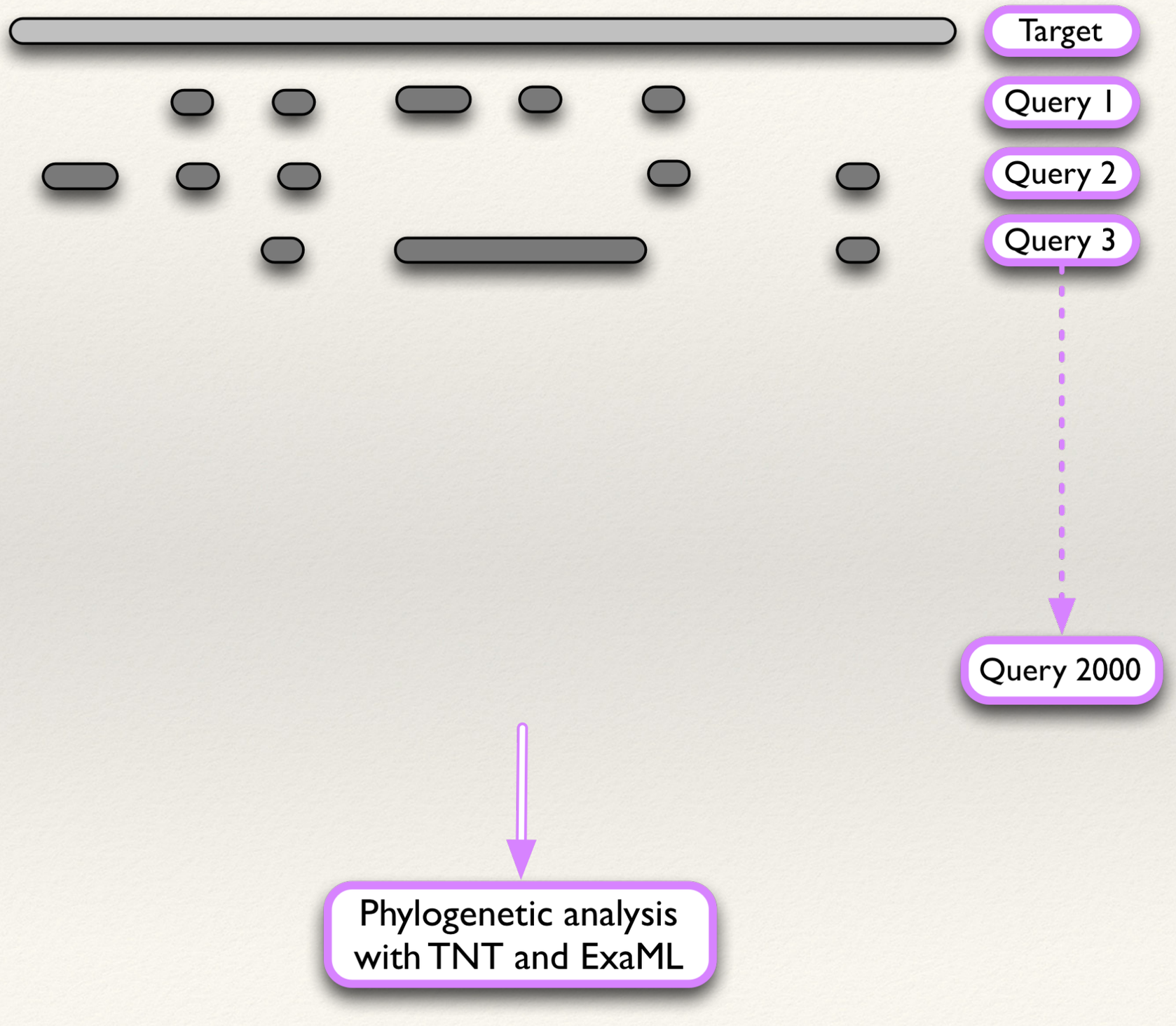
LASTZ (Harris, 2007)

- Pairwise (multiple files per target and query allowed).
- Output: MAF (multiple alignment format), SAM, others.
- Can also be used to align short reads rather than genome-genome or chromosome-chromosome.
- Geneious plug-in
- Can mask sequence before using as a target (important for Eukaryote genomes)
- Used in UCE pipeline

LASTZ application (Dikow, in prep.)

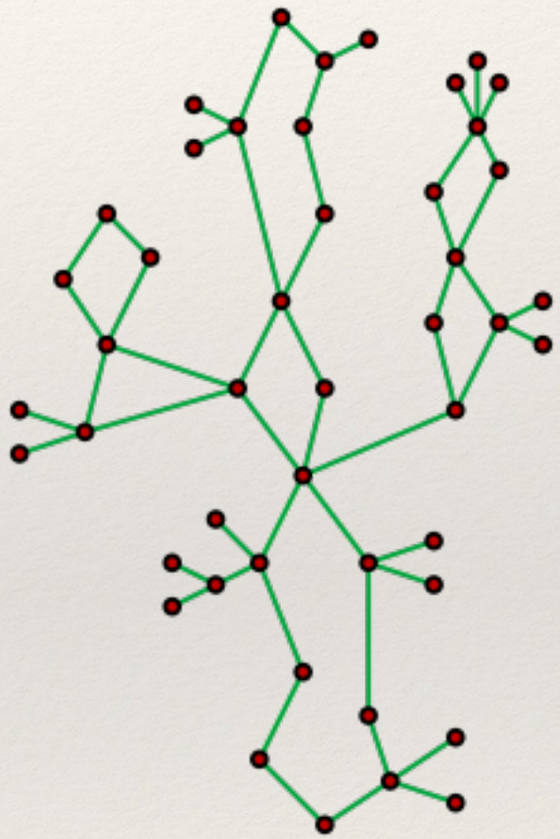


LASTZ application (Dikow, in prep.)

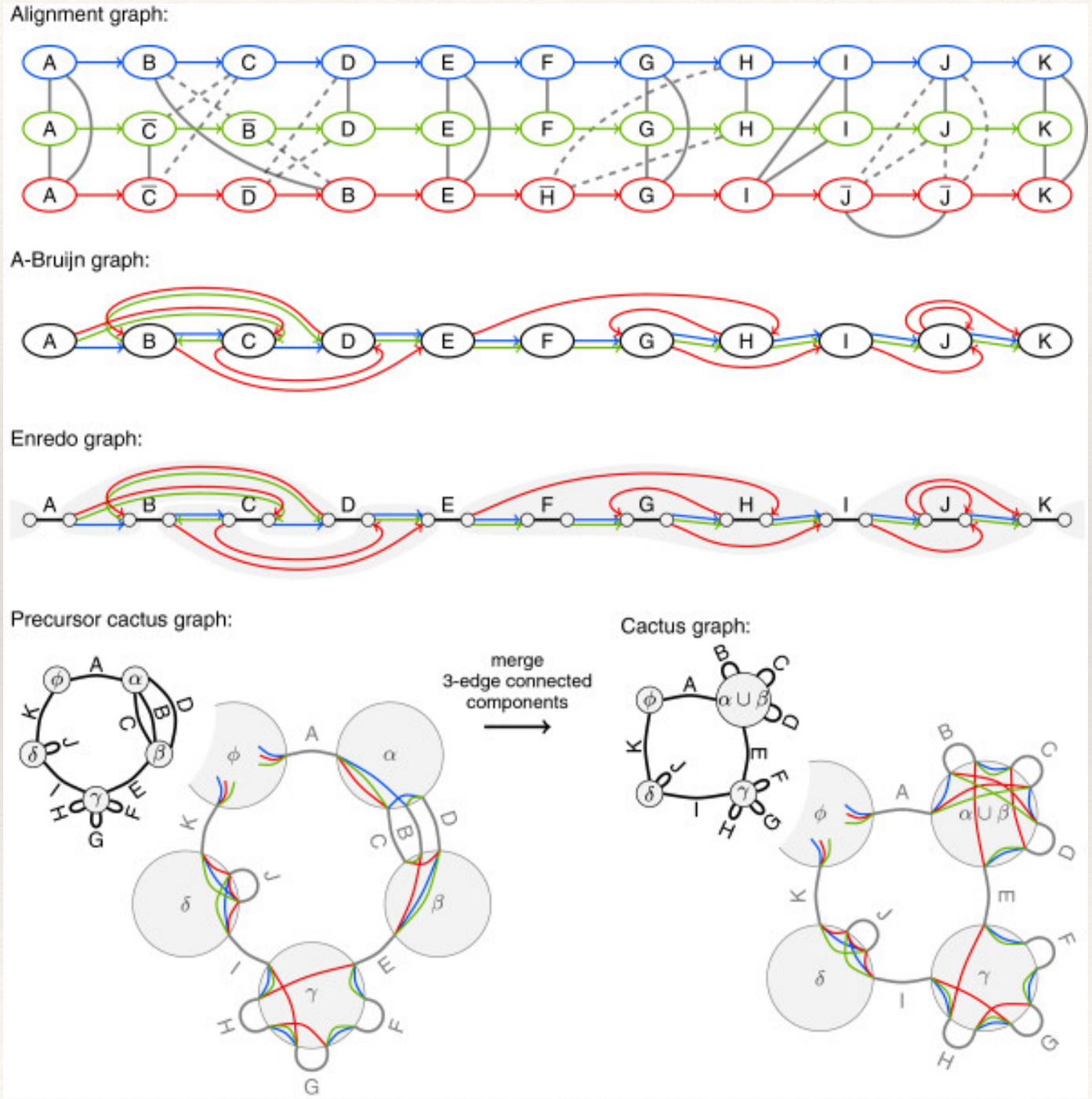


Progressive Cactus (Paten et al. 2011)

- Cactus graph: a graph in which any edge is a member of at most one simple cycle, to construct a genome alignment. Allows for arbitrary types of rearrangement and duplication, but favors alignments that create 'chains' of aligned bases.
- 150GB+ of memory on at least one machine when aligning mammal-sized genomes; less memory is needed for smaller genomes.
- Outputs in the HAL format - graph based (HDF5).
- SGE supported.
- Allows multiple genomes.



Progressive Cactus (Paten et al. 2011)



Progressive Cactus (Paten et al. 2011)

Align genomes locally.

Build a graph.

Characterize substructures.

Detect substructures.

Eliminate substructures.

Find a segmentation.

Genome alignment

Progressive Cactus (Paten et al. 2011)

- 5 Mammal genomes: (Total Runtime: 20h11m51s) and just under 100 CPU days per genome aligned (70weeks2days1h20m56s / 5 ~= 98 days). Run on a shared compute cluster with 1000 CPUs (actual usage was generally lower than 1000) and, for the large memory jobs, a machine with 64 CPUs and 1TB of RAM. The largest Target used around 100GB of ram, and total peak memory usage on the large memory machine was ~250GB of ram.
- In terms of asymptotic scaling, progressive cactus will scale linearly in the number of input genomes, provided a phylogenetic tree is provided. If no tree is provided, or the tree is poorly resolved (e.g. a near star tree) then scaling is quadratic in the number of input genomes.

progressiveMauve (Darling et al. 2004, 2010)

- Originally designed for microbial genomes.
- Finds **LCBs** (locally collinear blocks).
- Finds **subset sequences.**
- Uses a guide tree and iterative refinement, allowing multiple genomes.
- Scales cubically in the number of genomes to align, making it unsuitable for datasets containing more than 50-100 bacterial genomes.
- Geneious plug-in available.

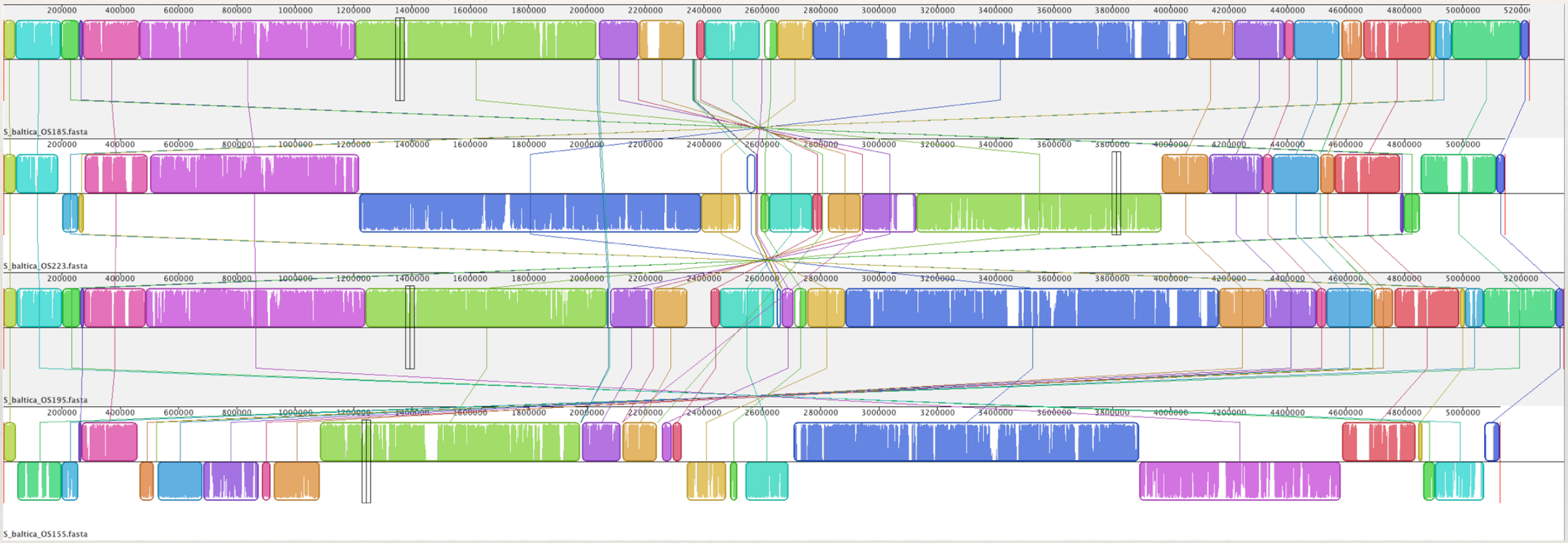
progressiveMauve (Darling et al. 2010)

- Uses the anchored alignment technique to rapidly align genomes. Unlike most genome alignment methods however, Mauve allows the order of alignment anchors to be rearranged in each genome permitting identification of genome rearrangements.
- The current Mauve release uses inexact, ungapped matches as alignment anchors. The inexact matches are found using a seed-and-extend method, where each seed match conforms to a pattern of matching nucleotides.

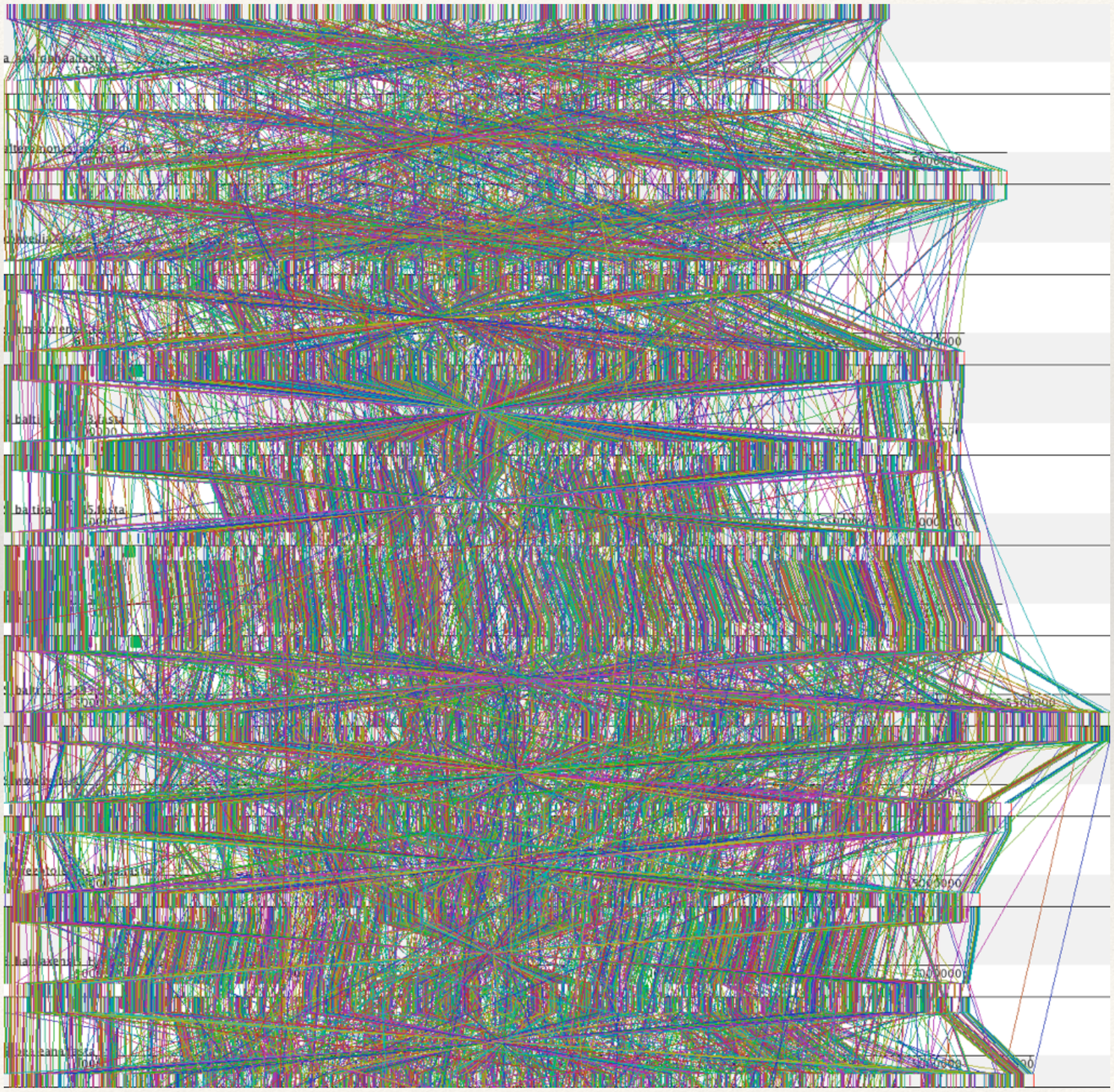
progressiveMauve (Darling et al. 2010)

- progressiveMauve builds up genome alignments progressively according to a guide tree (similar to MSA algorithms).
  - The guide tree is computed based on an estimate of the shared gene content among each pair of input genomes. For a pair of input genomes, g.x and g.y, shared gene content is estimated by counting the number of nucleotides in gx and gy aligned to each other in the initial set of local multiple alignments.
  - Neighbor joining is then applied to the matrix of distance estimates to yield a guide tree topology.

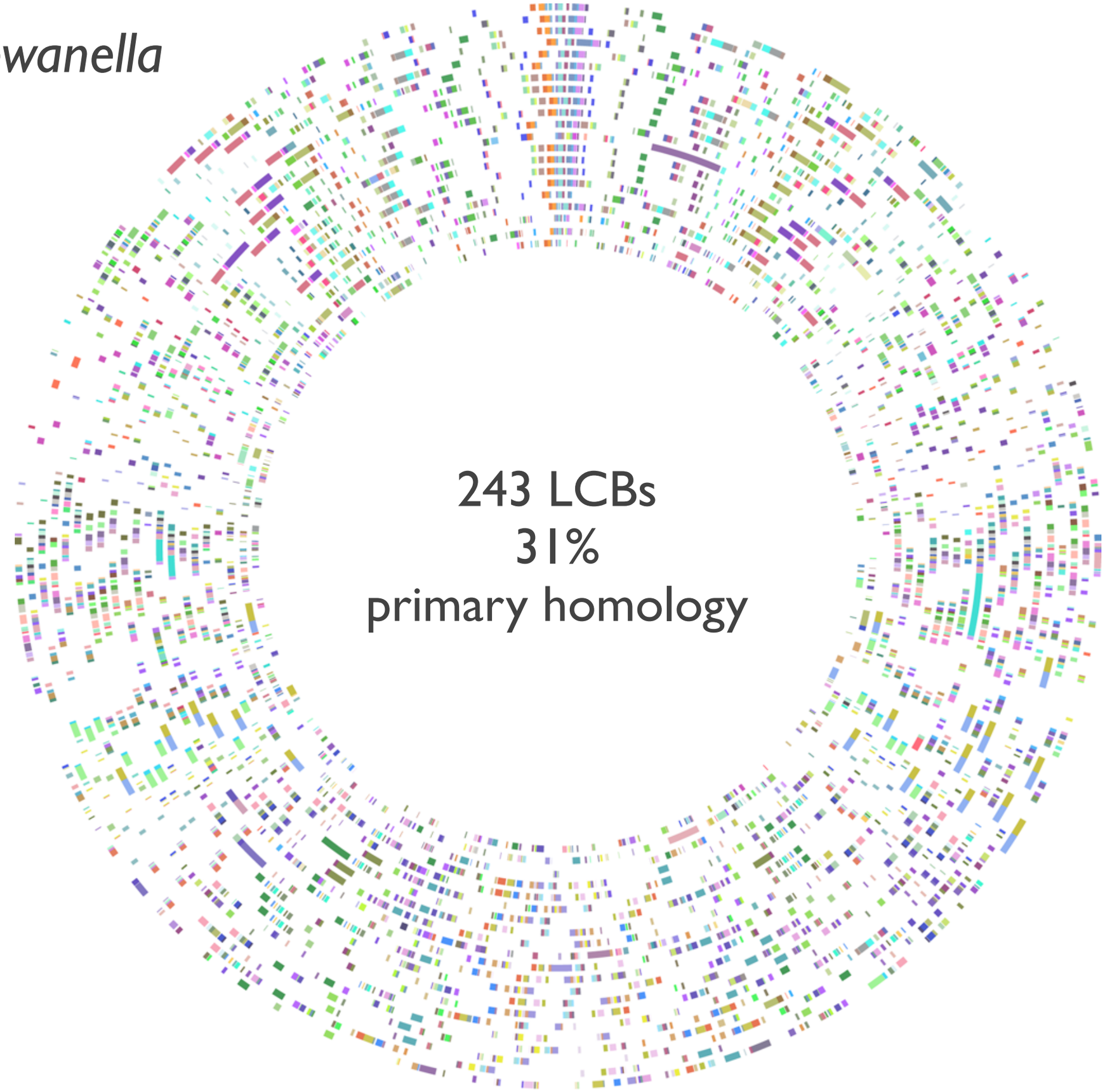
progressiveMauve (Darling et al. 2010)



progressiveMauve (Darling et al. 2010)



Shewanella



243 LCBs
31%
primary homology

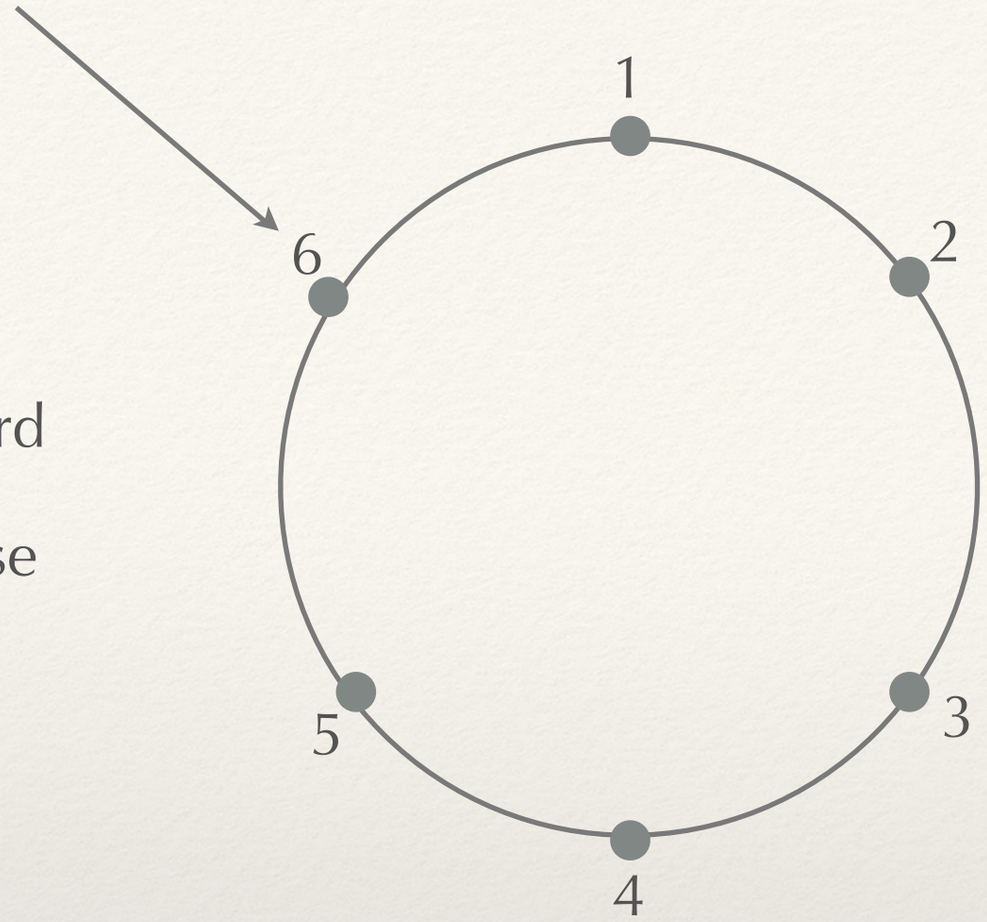
Dikow, 2011 *BMC Genomics*

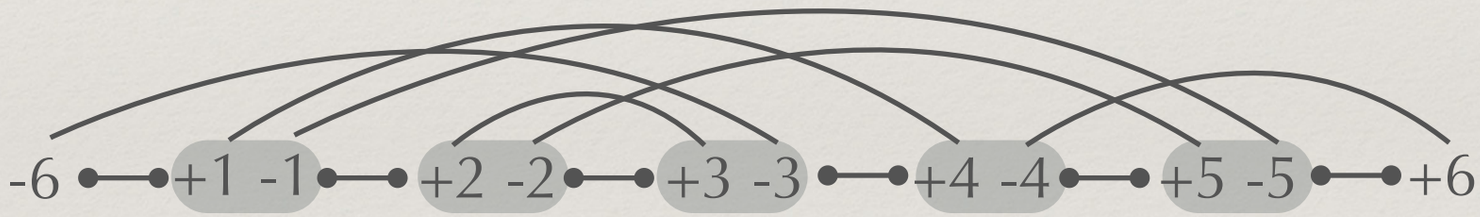
Genome rearrangement

- The path to convert one genome into another
- Can be used to hypothesize relationships

+1 -1 = forward

-1 +1 = reverse





breakpoint graph:
Xu & Sankoff 2008

+4 +1 -1 -5 +5 -2 +2 +3 -3 -6 +6 -4

genome 1 (straight lines): +1 +2 +3 +4 +5 +6

genome 2 (curves): +1 -5 -2 +3 -6 -4

Whole genome phylogeny tools

- Genome alignment produces a set of putative collinear homologs, which may or may not include subset sequences and rearrangement information.
- What phylogeny tools can handle data in the millions of base-pairs?

Whole genome phylogeny tools

- RAxML/RAxML-light (Stamatakis, 2014)
- ExaML (Exascale Maximum Likelihood (ExaML) code for phylogenetic inference on supercomputers using MPI; Kozlov et al., 2015; 3.2 X faster than RAxML-light)
- TNT (Goloboff et al., 2008)
- All of these can analyze my big matrix of 2058 taxa X 4.5 Mbp. TNT is fastest.

Whole genome phylogeny tools

- What about support?
- Gene-tree analyses?

Next installment

- Testing tools - run time, RAM use, etc. on standard datasets.
- What applications are you most interested in trying?