Comparative genomics

on a good day

on a bad day

The good

- **Long read assemblies unmask complex genomic regions**
- **Long read assemblies have high contiguity**
- **Long read assemblies improve structural variant detection**

Gordon *et al*. Science, 2015

The bad

- DNA alignment breaks down in complex regions
- Most software for comparing genomes was built in the early 2000s
- Some common file formats (e.g. BAM) do not support large alignments

Outline

- Quickly comparing two genomes
  - Minimap
  - Layout plots
- Structural variant calling from whole-genome alignments
  - Alignment methods
  - Tools
  - Example

Outline

- **Quickly comparing two genomes**
  - **Minimap**
  - **Layout plots**
- Structural variant calling from whole-genome alignments
  - Alignment methods
  - Tools
  - Example

Minimap a very fast pairwise aligner

fork: https://github.com/zeeev/minimap

Heng Li, arxiv. 2015

Minimap “sketches” genomes using “minimizers”

ATGCGGCAGGGTTCNACGCGCGCAAA

ATGCG
TGCGG
GCGGC
CGGCA
GGCAG
GCAGG

binary & hash →

324,567
285,432
999,999
545,123
111,111 ★

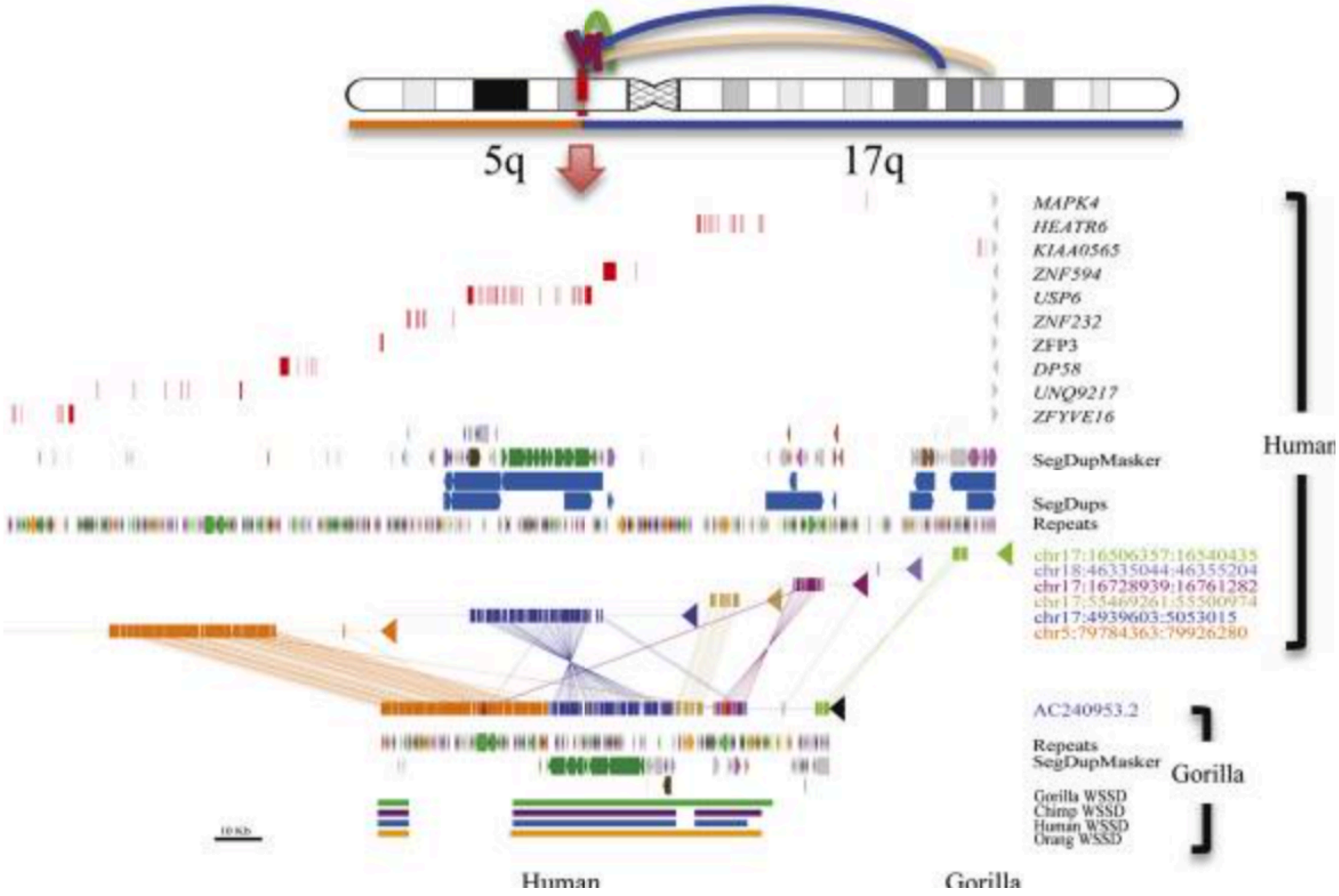
W: save every w kmer: 5
S: kmer width: 6
R = {I,P,K}: seqid, position, kmer

Heng Li, arxiv. 2015

Minimap produces a Pairwise Alignment Format (PAF)

Col	Type	Description
1	string	Query sequence name
2	int	Query sequence length
3	int	Query start (0-based)
4	int	Query end (0-based)
5	char	Relative strand: "+" or "-"
6	string	Target sequence name
7	int	Target sequence length
8	int	Target start on original strand (0-based)
9	int	Target end on original strand (0-based)
10	int	Number of residue matches
11	int	Alignment block length
12	int	Mapping quality (0-255; 255 for missing)

minimap example 1



Aligning a gorilla chromosome to GRCh38

Ventura, *Genome Res*. 2011

Let’s start with an interactive jobs

```
ls /share/biocore/workshops/Genome-Assembly-
Workshop/examples/minimap

srun?

minimap -l -t 1 -f 0.01 index/genome.mmi
gorilla-chr17.fasta > gorilla-chr17.paf
```

minimap example 2

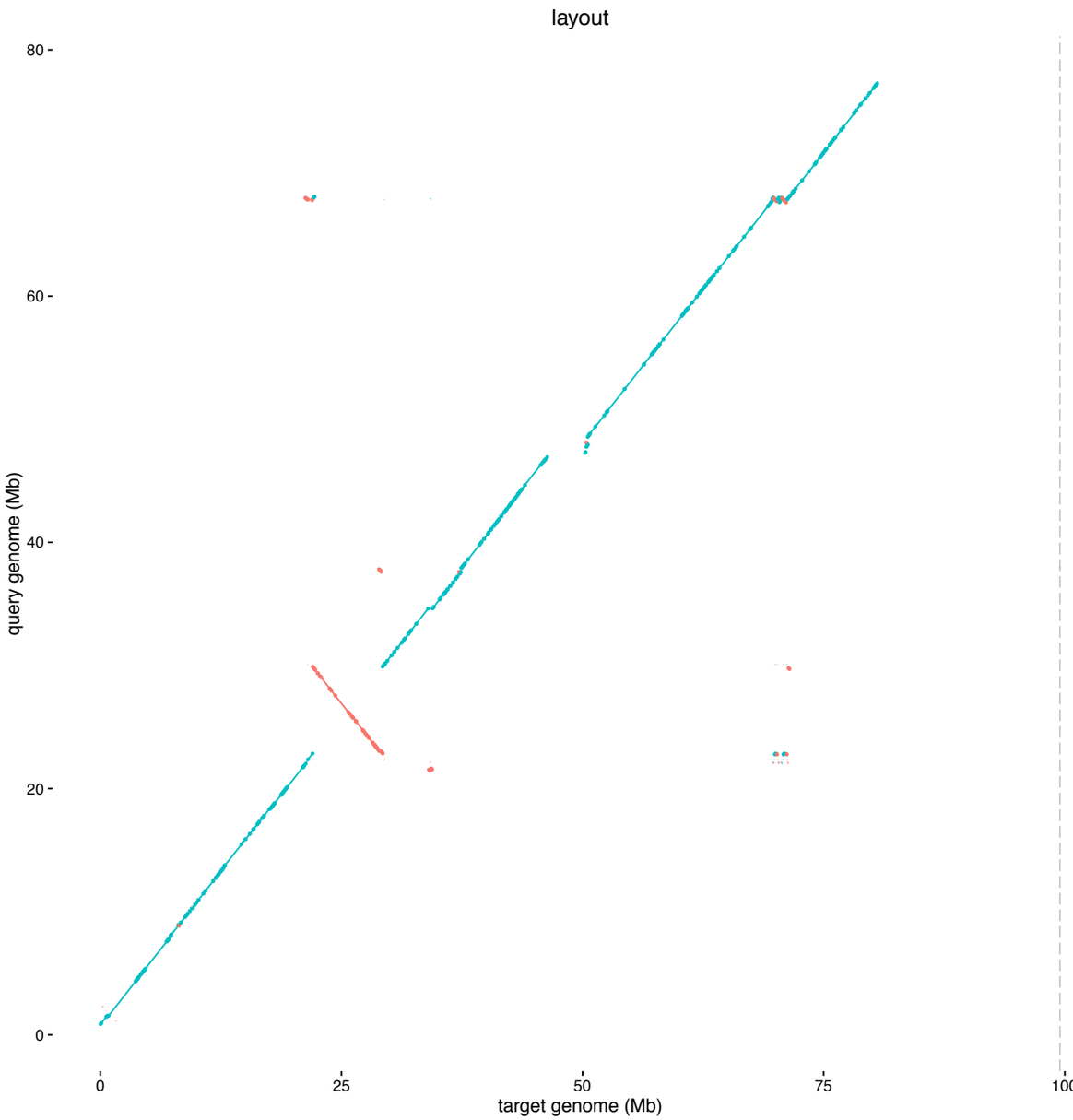
Plotting minimap alignments

```
perl -lane '$_ =~ /i:(.*)?/; print $_ if $1 > 200 '
gorilla-chr17.paf   | ../../software/minimap/utils/
bin/layout -i chr17  > gorilla-chr17.layout.txt

perl -lane '$_ =~ /i:(.*)?/; print $_ if $1 > 200 '
gorilla-chr17.paf   | ../../software/minimap/utils/
bin/layout -i chr5  > gorilla-chr17.layout.txt
```

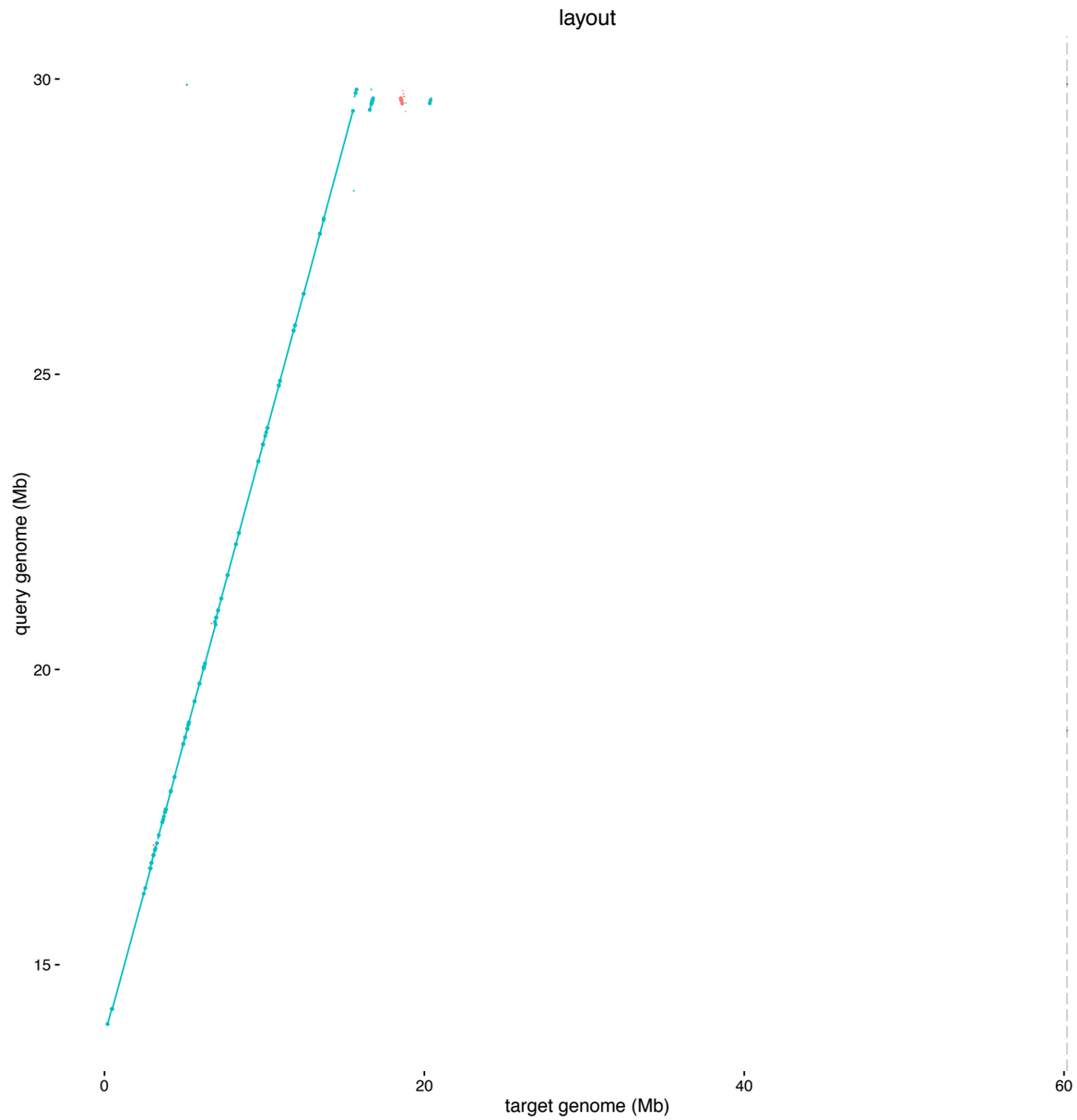
```
Rscript --vanilla ../../software/minimap/utils/plot/
plotLayout.R -f gorilla-chr5.layout.txt -p
gorilla-5-layout.pdf

 Rscript --vanilla ../../software/minimap/utils/
plot/plotLayout.R -f gorilla-chr17.layout.txt -p
gorilla-17-layout.pdf
```

Gorilla against GRCh38

Chr5

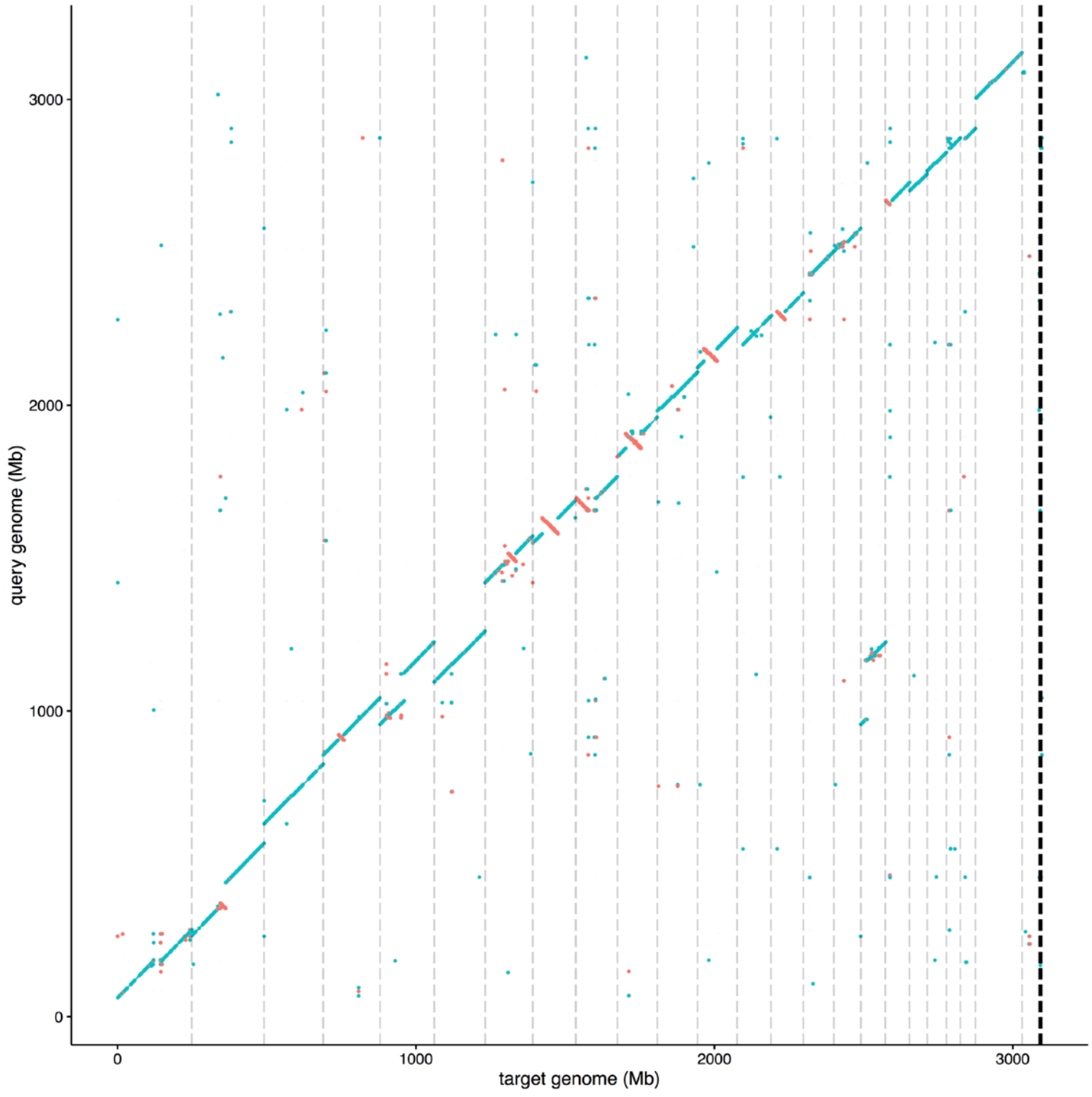


Chr17



Plotting the whole genome

Instructions on github



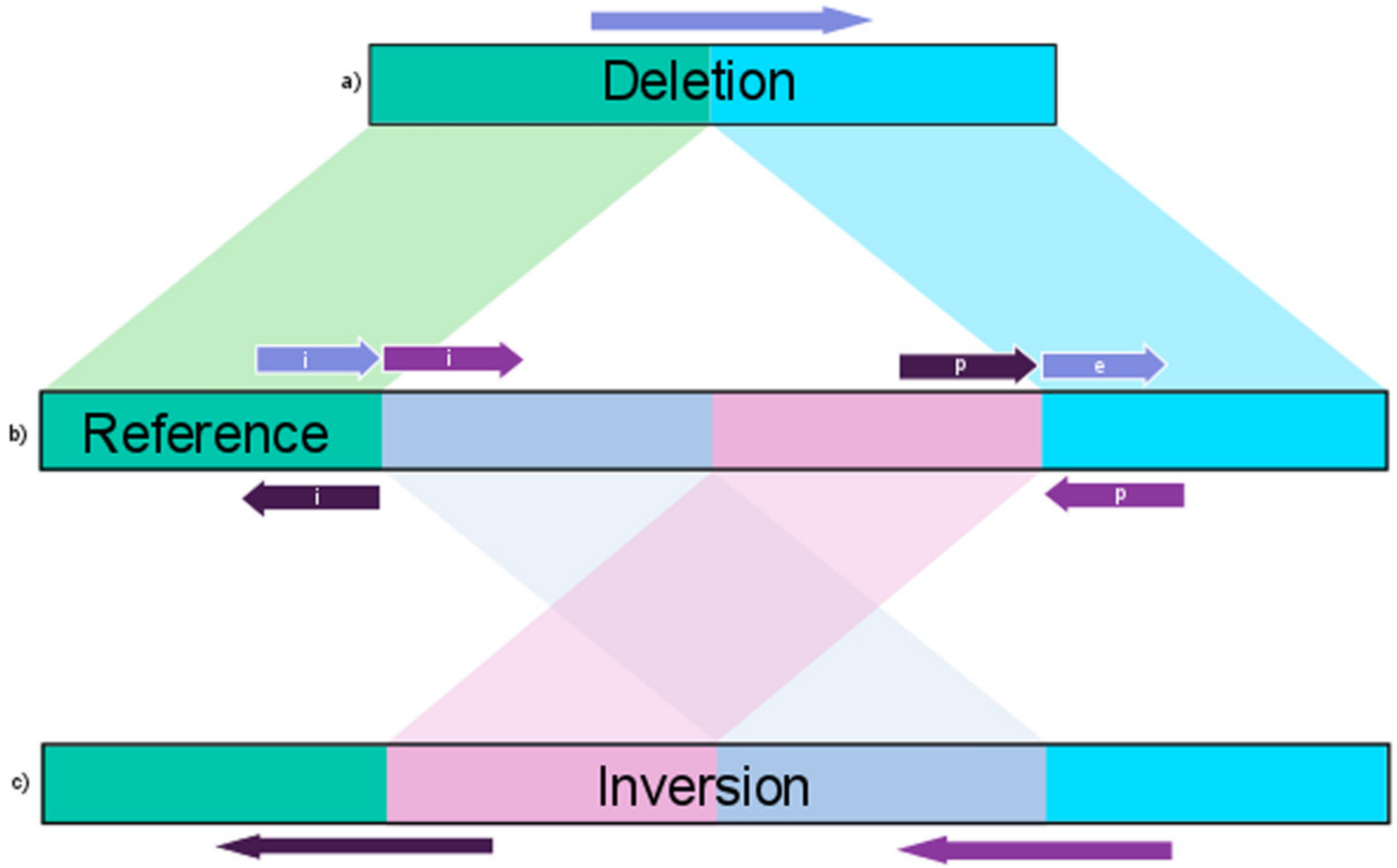
Side note: detecting scaffolding errors with synteny

CHM13 scaffolding attempt v0

Outline

- Quickly comparing two genomes
  - Minimap
  - Layout plots
- **Structural variant calling from long read technology**
  - **Tools**
  - **Alignment methods**
  - **Example**

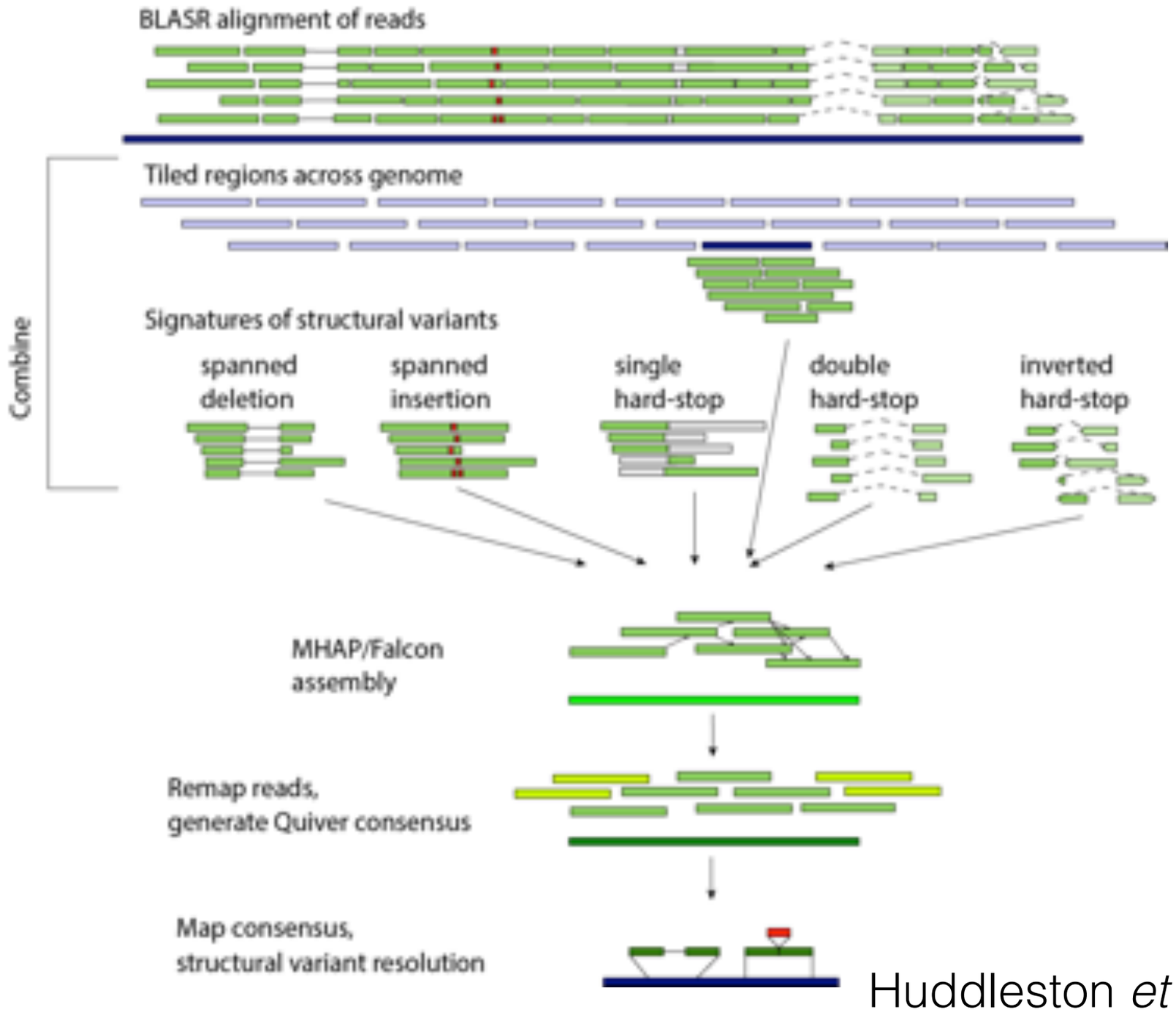
PBHoney:
Read mapping/clustering based SV caller



Adam English *et al*. G.R. 2014

Sniffles:
Read mapping based SV caller

SMRT-SV:
local assembly SV caller



Huddleston *et al*. G.R. 2016

Assembly based SV calling

Method	DNA diff	DASVC	Smartie-SV
Aligner	AMUMMERA3BL MUMMER 3+ TMUMMER.3DR	*LastZ*	*Blasr*
Paper	Kurtz *et al*. G.B. 2004	Z. Kronenberg, unpublished	

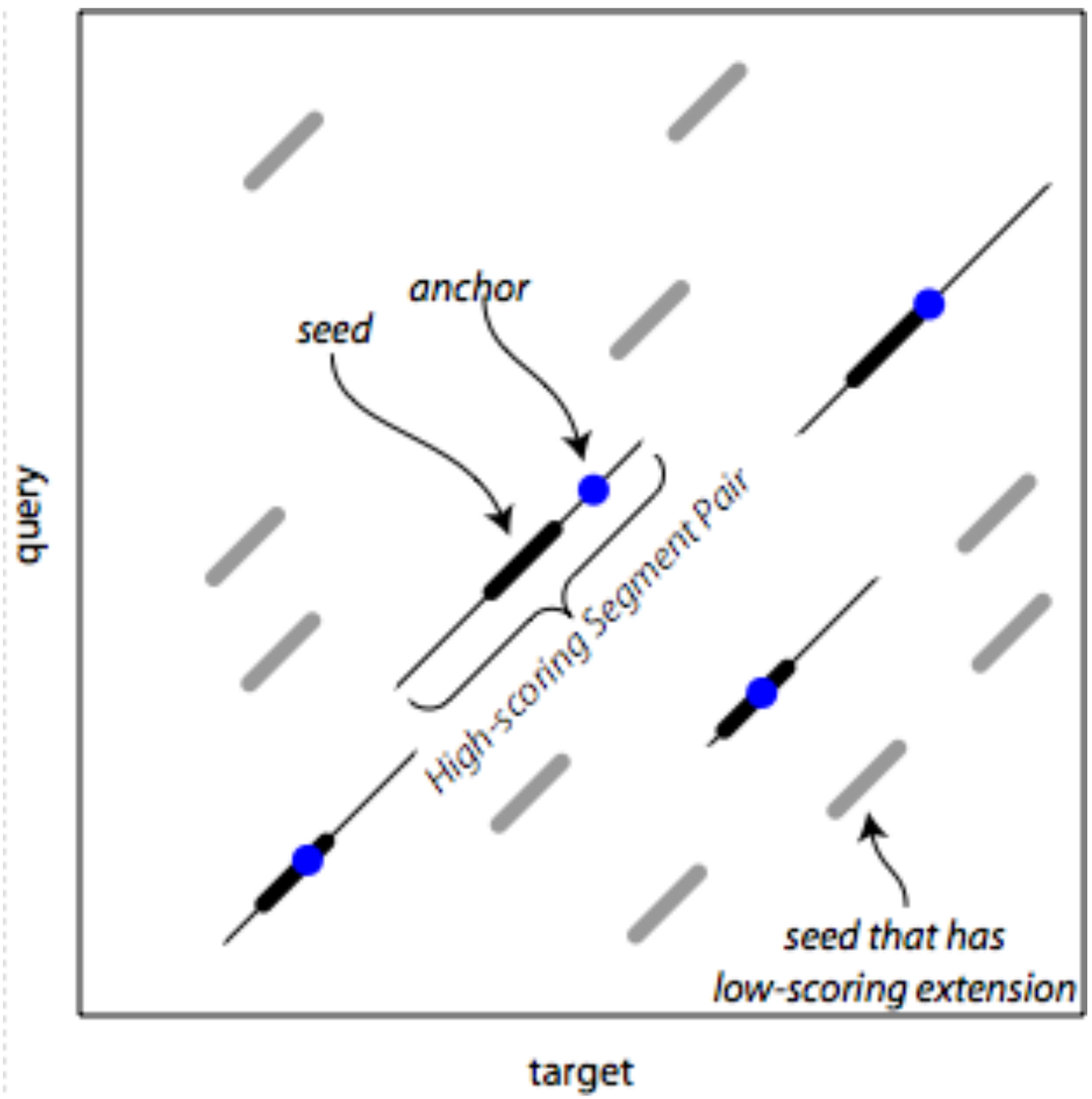
De novo Assembly Structural variant caller (DASVC)

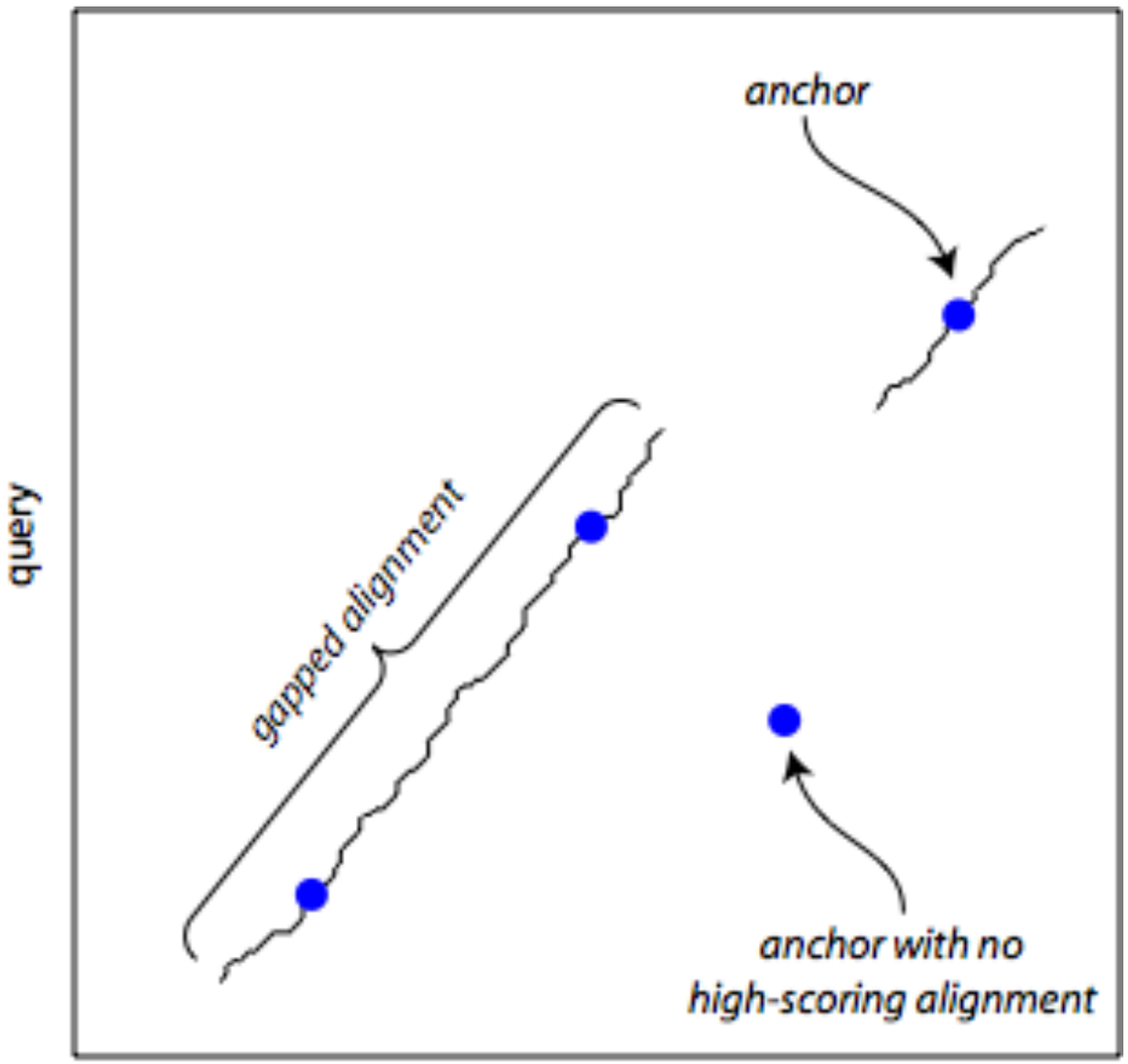
- Start with lastZ alignments of query to target
- Uses UCSC's chaining and netting
- Converts net files into SAM alignments
- Looks within and between alignments for SVs

De novo Assembly Structural variant caller (DASVC)

- **Start with lastZ alignments of query to target**
  - Uses UCSC's chaining and netting
  - Converts net files into SAM alignments
  - Looks within and between alignments for SVs

LastZ starts with exact kmer matching





Harris, R.S. (2007) Improved pairwise alignment of genomic DNA. Ph.D. Thesis

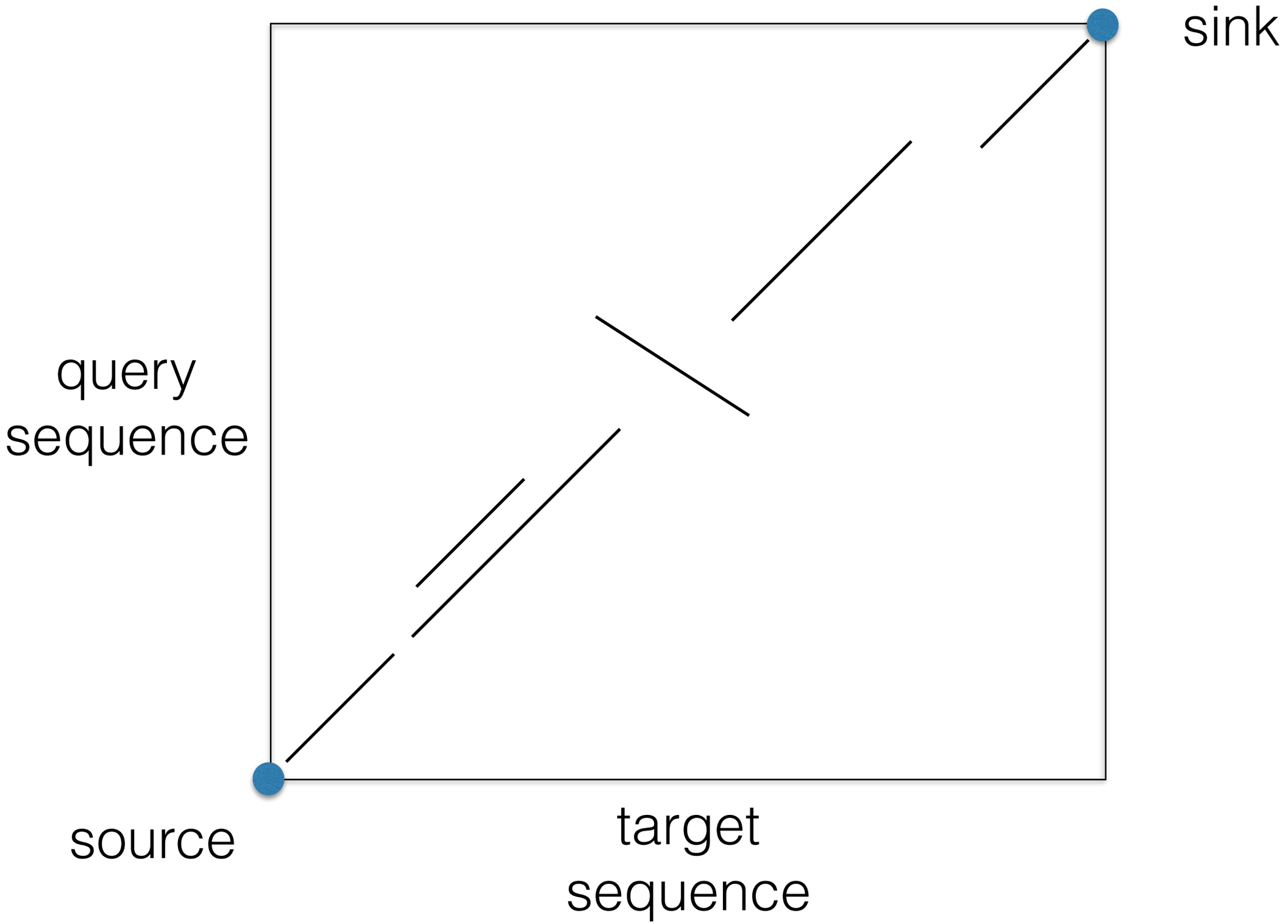
De novo Assembly Structural variant caller (DASVC)

- Start with lastZ alignments of query to target
- **Uses UCSC's chaining and netting**
- Looks within and between alignments for SVs

Chaining maximizes the number of aligned bases in the query and target

- Synteny: co-linear sequences in increasing order
- Rules:
  - No overlapping alignments
  - No changes in strand
  - Do while : alignments remain

Chaining: a dynamic programming solution



Overlay chains to form nets

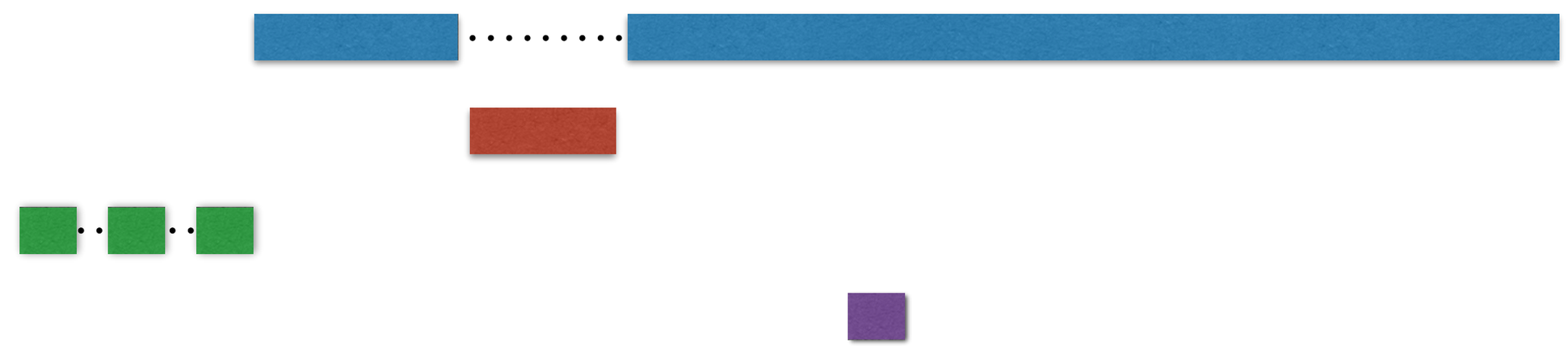
Target Genome

chain 1

chain 2

chain 2

chain 3



resulting net:

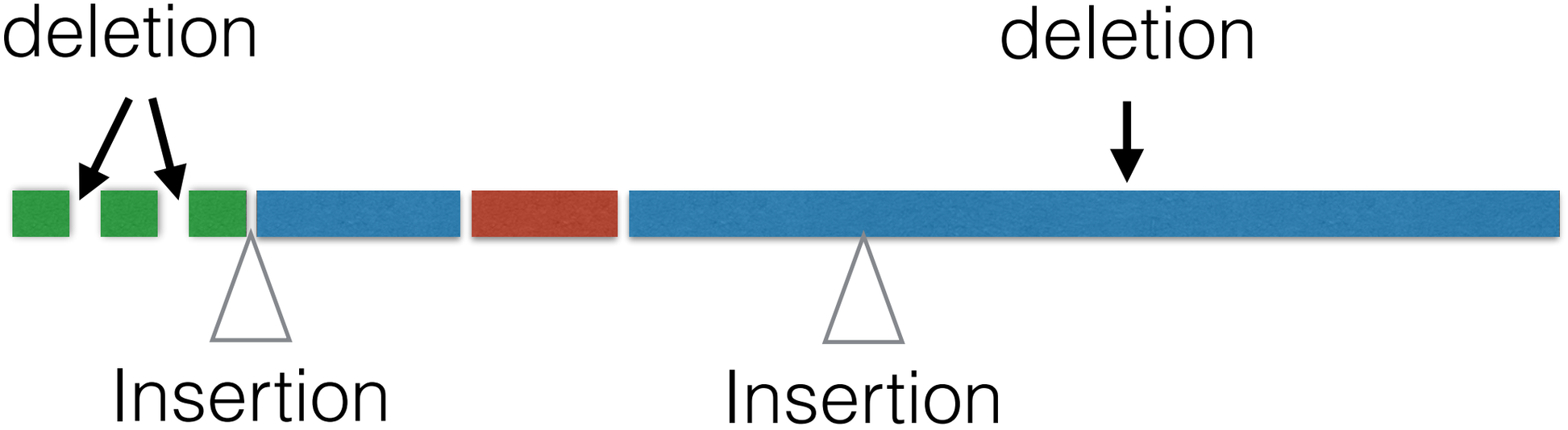
**

**bad fills do occur

De novo Assembly Structural variant caller (DASVC)

- Start with lastZ alignments of query to target
- Uses UCSC's chaining and netting
- **Looks within and between alignments for SVs**

DASVC identifies SVs within and between alignment blocks



DASVC pipeline summary

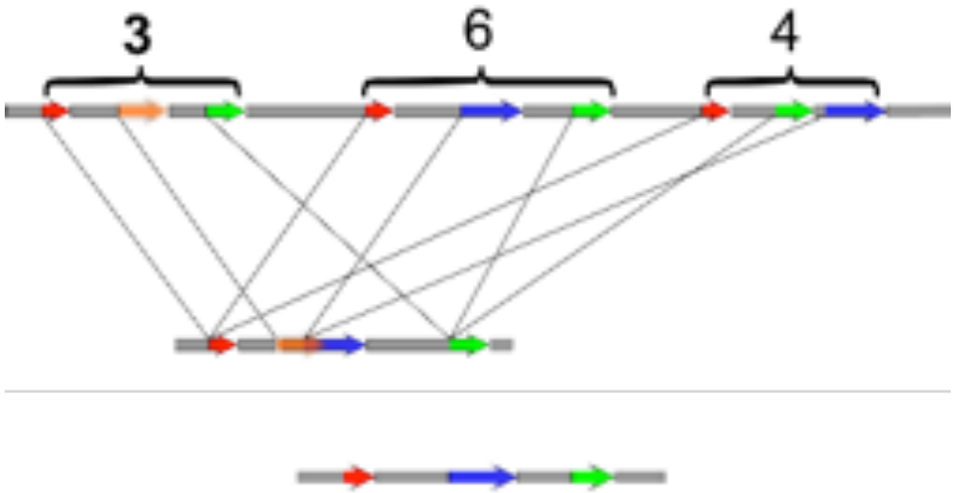
- First iteration of my SV caller
- Pros:
    - **Generates UCSC nets and chains**
        - Has generally good sensitivity and specificity across divergent species
- Cons:
    - Nets are subject to bad fills
    - Calling SV between alignment blocks is tricky.

Smartie-SV

- Built upon ultra-long Blasr alignments
- SVs are contained with alignments
- High validation rate with BACs
- Runs genome-wide alignments

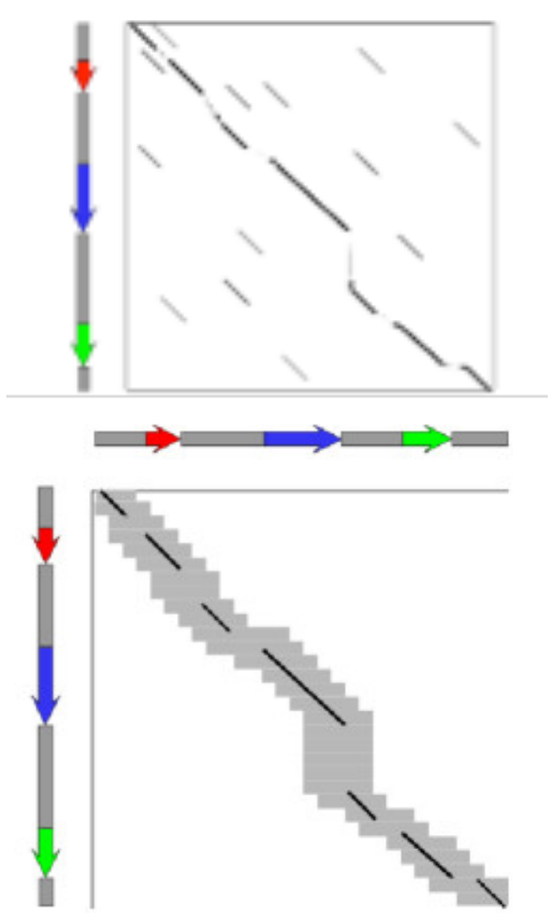
Blasr's alignment methods

1. Find clusters of exact matches using suffix array search



Motivated from HTS mapping

2. Refine cluster alignments using sparse dynamic programming

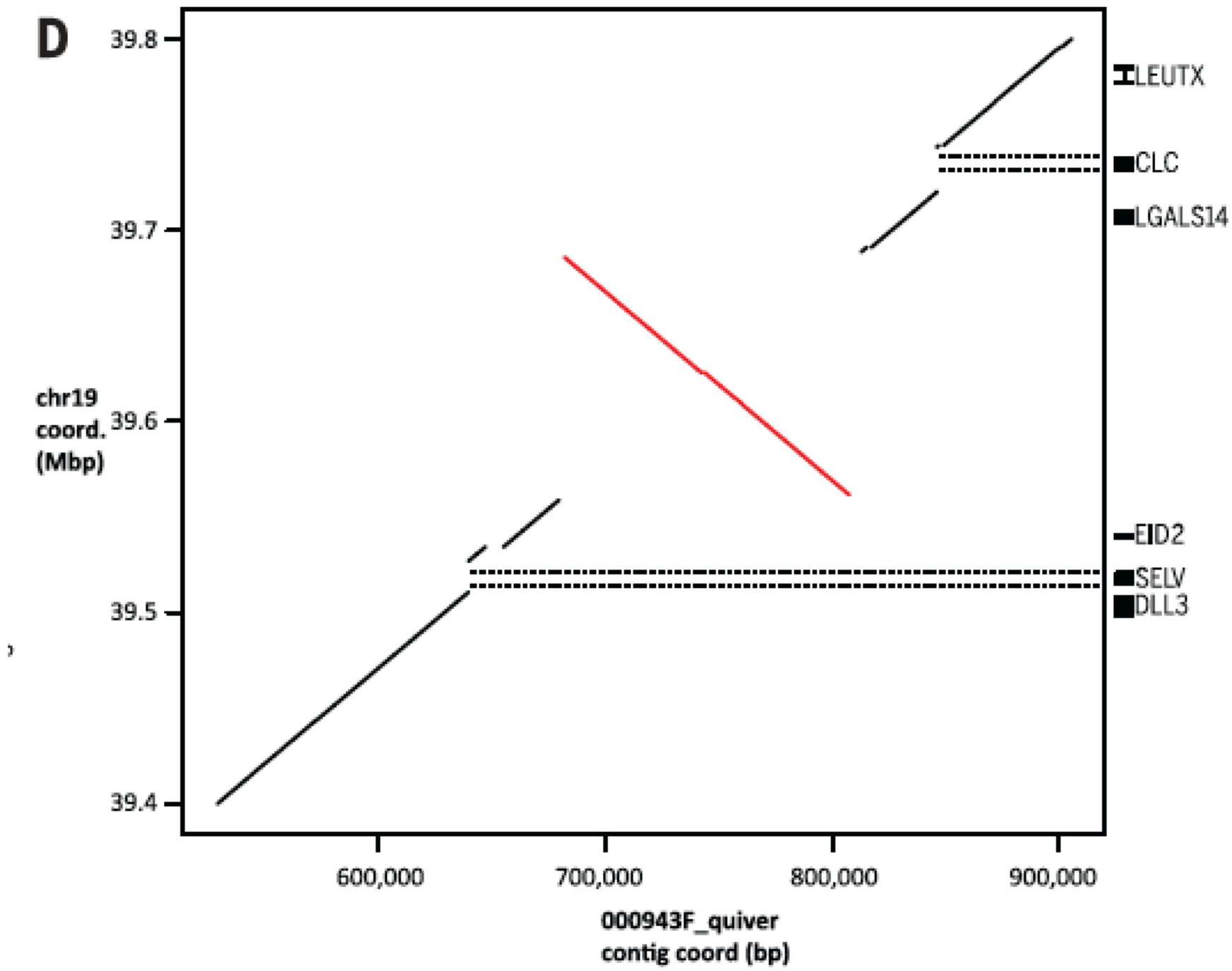


Motivated from whole-genome alignment

Chaisson and Tesler, *BMC Bioinformatics*, 2012

This is the sugar:

Smartie-SV example



Gordon *et al*. Science, 2015

Putting your skills to the test:

running the following example:

/share/biocore/workshops/Genome-Assembly-Workshop/examples/smartie-sv-gorilla/

Conclusions

- We've learned how to:
  - Quickly compare two genomes for visual inspection
  - Hopefully gained some knowledge about alignment
  - Call SVs like a pro ;-)

Acknowledgements

Questions?