Graphs for Genetic Variation

EN 500.111 Week 7

Course Logistics - Presentations

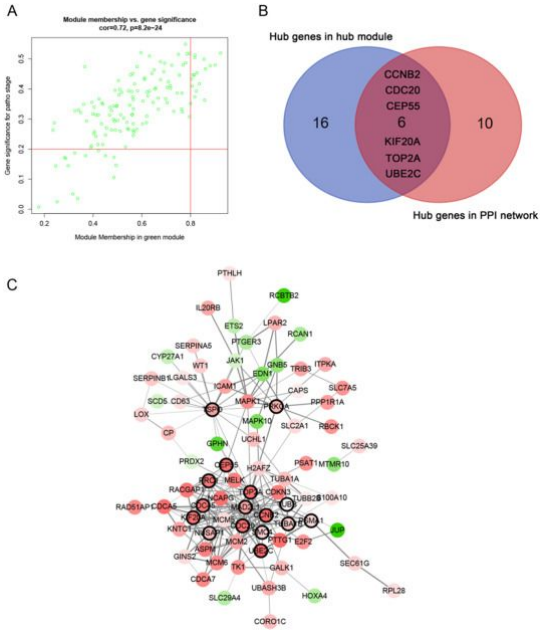
- ➔ Time will be given at the start of next class to discuss with groups
- ➔ Schedule posted here: https://github.com/mkirsche/TAG2020/blob/master/presentation/guidelines.md
  - ◆ Feel free to switch time slots if both groups agree - just have someone from each group email me!

Outline

- ➔ Gene Interaction Networks
- ➔ Assembly Review
- ➔ Genome Graphs
- ➔ Graphs for Structural Variant Comparison

Gene Interaction Networks

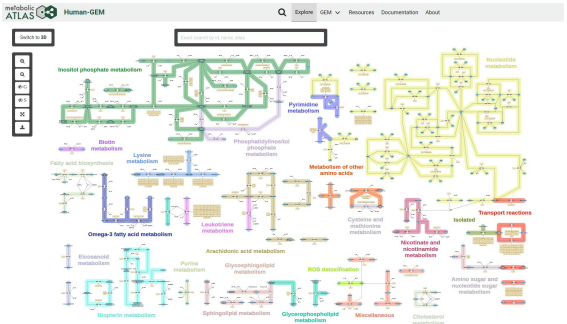
- ➔ While all genes in an individual have the same genome and code for the same proteins, environmental factors affect how much each protein is produced
- ➔ By sequencing RNA instead of DNA, it is possible to measure the extent to which a cell is translating different proteins (i.e., expressing the genes which code for them)
- ➔ By correlating these genes' expressions with one another, it's possible to infer pathways, identify functionally significant genes, and classify cell types

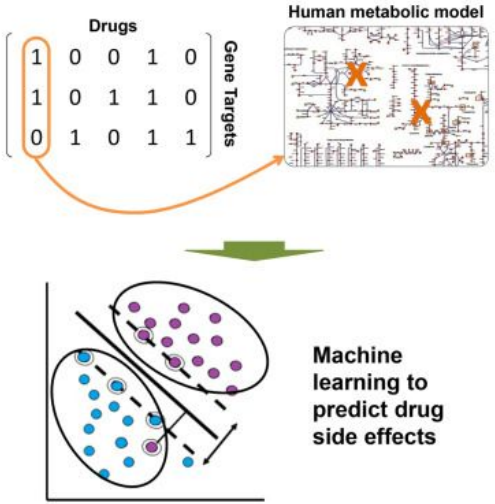


Yuan, Lushun et. al. Co-expression network analysis identified six hub genes in association with progression and prognosis in human clear cell renal cell carcinoma (ccRCC). Genomics Data, 2017.

Using metabolic pathway models

➔ Once pathways have been predicted from gene expression data or other experiments, they can be combined with machine learning techniques to help inform drug response

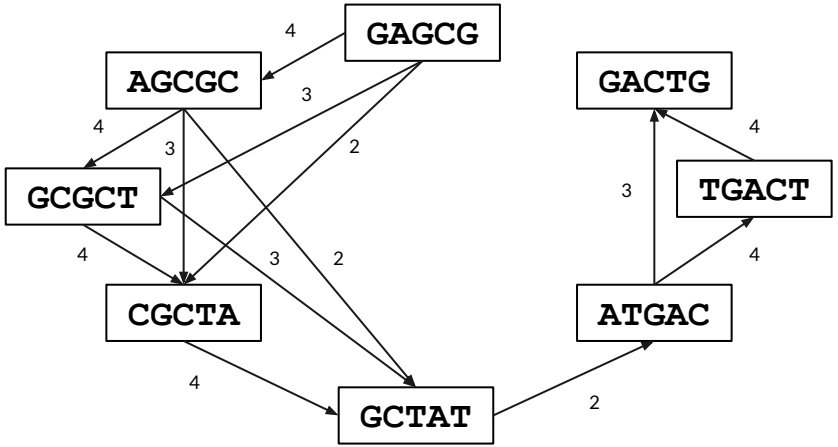


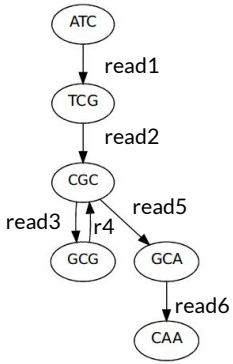


Shaked, Itay et. al. Metabolic Network Prediction of Drug Side Effects. Cell Systems, 2016.

Review - Assembly

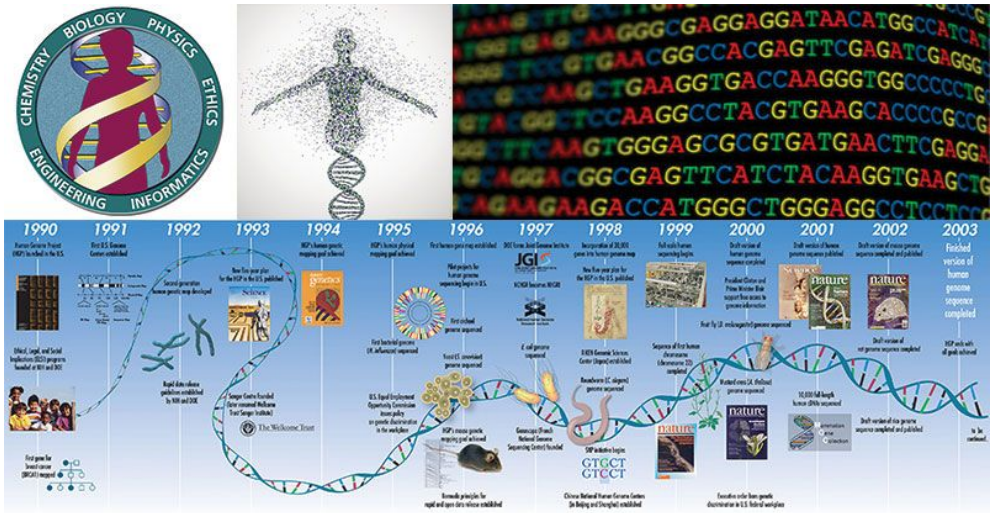
- Genome consists of pairs nucleotides (~3 billion for humans)
- Genomic sequencing gives us small pieces of about 500 basepairs
- Assembly is the problem of inferring the overall genome from many tiny pieces
- Overlap graphs and de bruijn graphs are some strategies for this
- Both require lots of sequencing data and computational resources





Human Genome Project

- ➔ Completed in 2003
- ➔ Started in 1990 and cost about $3 billion
- ➔ Goal: to build the first complete map of the human genome sequence



https://microbenotes.com/the-human-genome-project/

Whose genome is it?

- ➔ Consists of parts from 5 individuals from a pool of >20 who responded to their newspaper ad looking for volunteers
- ➔ Over 70% is from one person, RP-11, an African-American male

WANTED

20 Volunteers

to participate in the

Human Genome Project

a very large international scientific research effort.

The goal is to decode the human hereditary information (*human blueprint*) that determines all individual traits inherited from parents. The outcome of the project will have tremendous impact on future progress of medical science and lead to improved diagnosis and treatment of hereditary diseases.

Volunteers will receive information about the project from the Clinical Genetics Service at Roswell Park, and sign a consent form before participating.

No personal information will be maintained or transferred.

Volunteers will provide a one-time donation of a small blood specimen. A small monetary reimbursement will be provided to the participants for their time and effort.

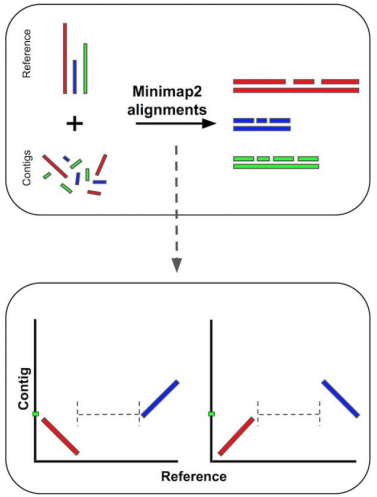
Individuals must be at least 18 years of age.
Persons who have undergone chemotherapy are not eligible.

ROSWELL PARK CANCER INSTITUTE

For more information please contact the
Clinical Genetics Service
845-5720 (9:00 am - 3:00 pm)
March 24 - 26, 1997

What can you do with reference genomes?

- ➔ Establish backbone for assembling other people's genomes (99.9% similar)
- ➔ Define coordinate system for marking locations of genes, known mutations, and other regions of interest



Alonge,M. Et. al. RaGOO: Fast and accurate reference-guided scaffolding of draft genomes. Genome Biology (2019)

Human Reference Shortcomings

- ➔ Only a single sequence which represents a few individuals
- ➔ A few percent of it (>100 million bp) is errors or missing sequence

Chromosome	All scaffolds
1	249,698,942
2	242,508,799
3	198,450,956
4	190,424,264
5	181,630,948
6	170,805,979
7	159,345,973
8	145,138,636
9	138,688,728
10	133,797,422

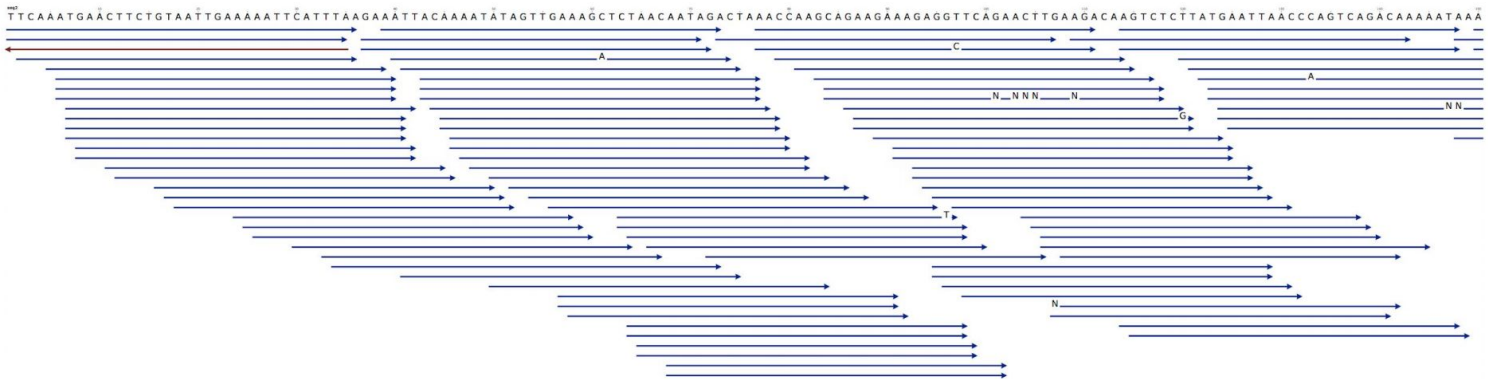
Human chromosome lengths including (left) and excluding (right) gaps in the current reference

https://www.ncbi.nlm.nih.gov/grc/human/data

Chromosome	All scaffolds
1	231,223,641
2	240,863,511
3	198,255,541
4	189,962,376
5	181,358,067
6	170,078,524
7	158,970,135
8	144,768,136
9	122,084,564
10	133,263,006

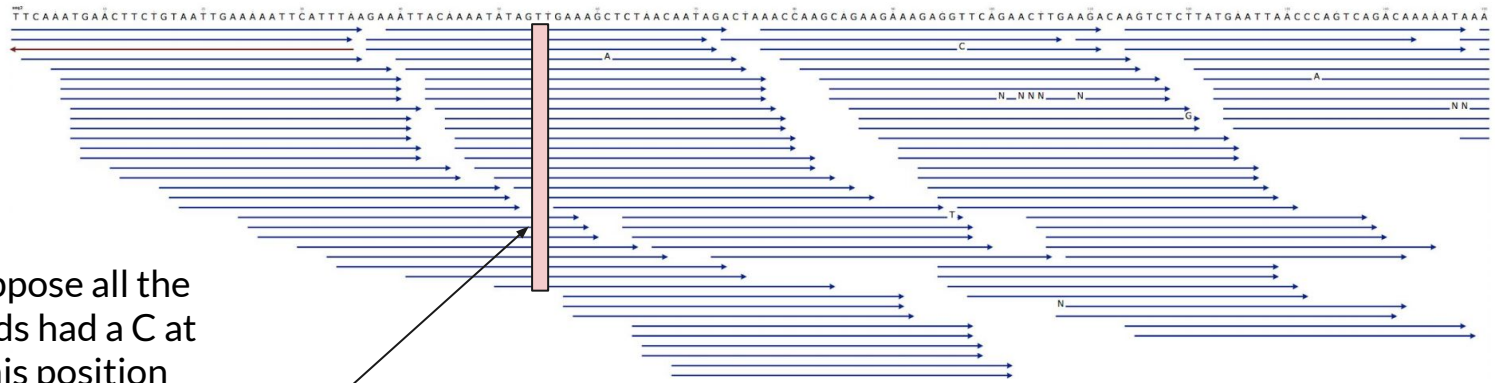
Genomic Read Alignment

- ➔ Using a reference to identify where in the genome a read came from
- ➔ Inexact pattern matching



Mismatches

➔ Read alignments can be used to identify consistent mismatches

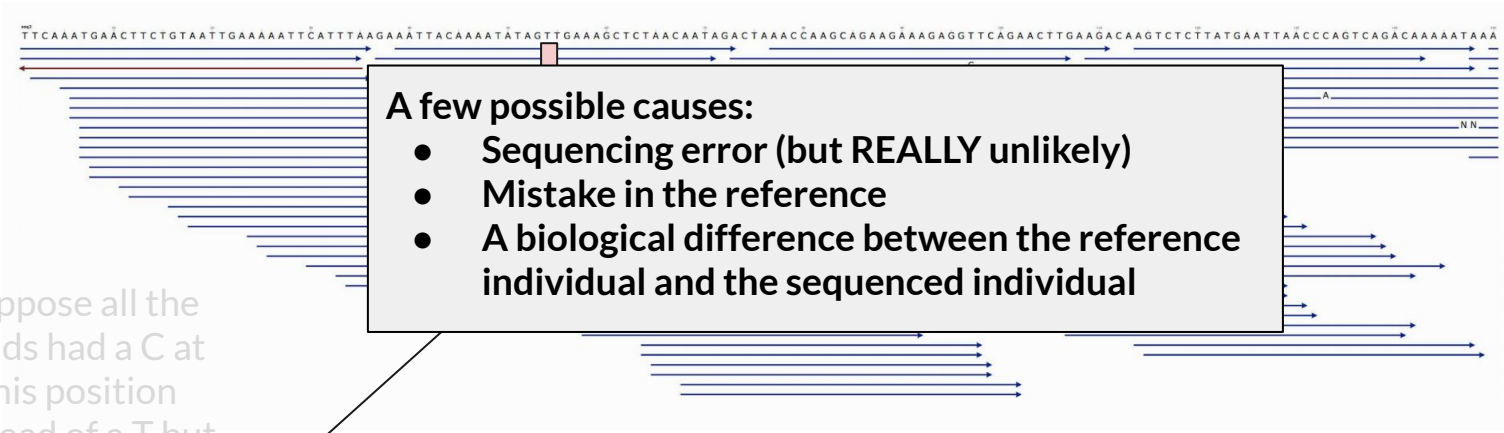


Suppose all the reads had a C at this position instead of a T but otherwise matched the reference

What could cause this?

Mismatches

➔ Read alignments can be used to identify consistent mismatches



A few possible causes:

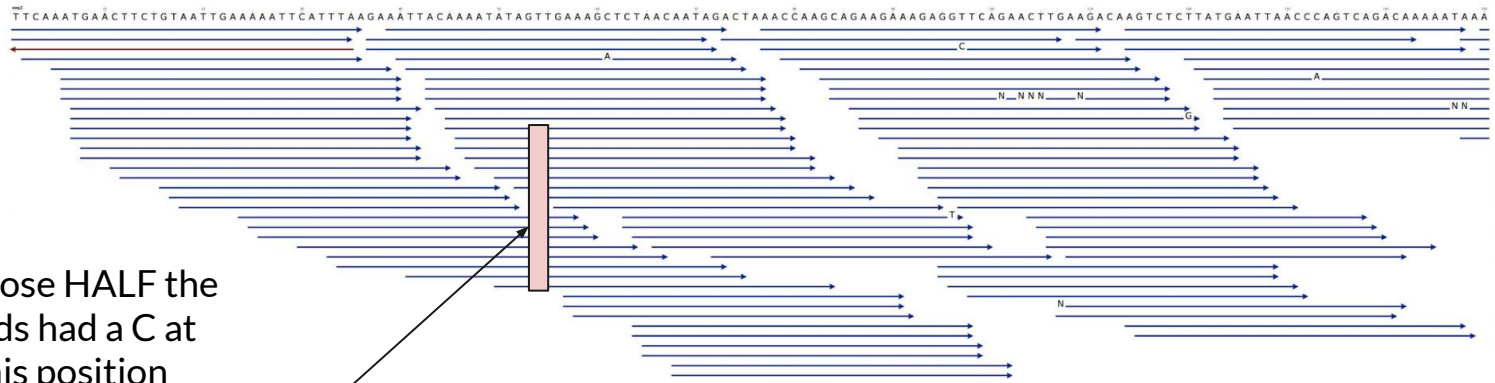
- **Sequencing error (but REALLY unlikely)**
- **Mistake in the reference**
- **A biological difference between the reference individual and the sequenced individual**

Suppose all the reads had a C at this position instead of a T but otherwise matched the reference

What could cause this?

Mismatches

➔ Read alignments can be used to identify consistent mismatches

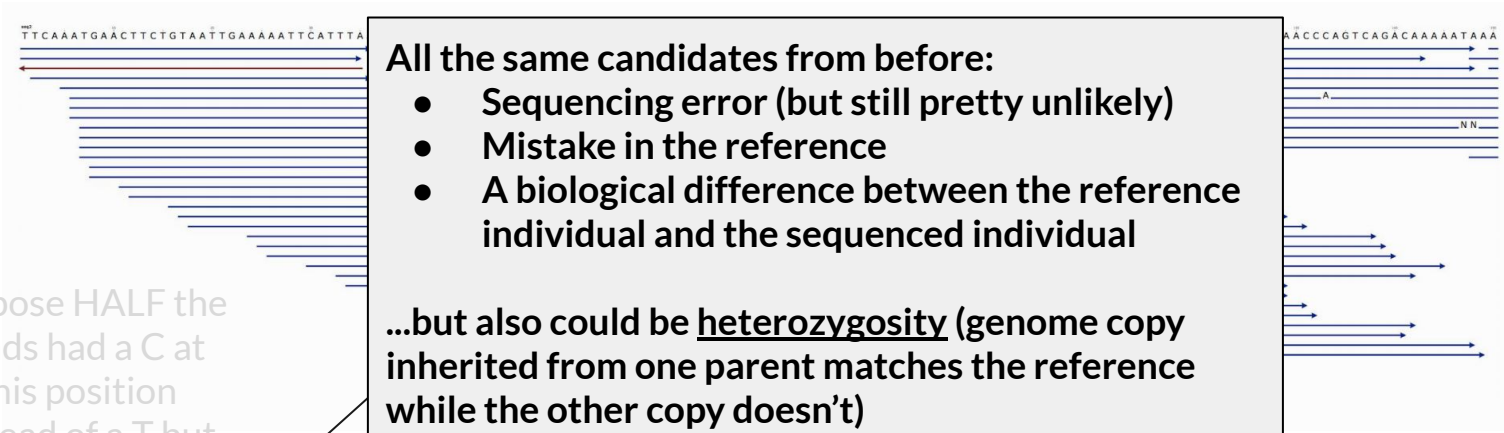


Suppose HALF the reads had a C at this position instead of a T but otherwise matched the reference

What could cause this?

Mismatches

➔ Read alignments can be used to identify consistent mismatches



All the same candidates from before:

- **Sequencing error (but still pretty unlikely)**
- **Mistake in the reference**
- **A biological difference between the reference individual and the sequenced individual**

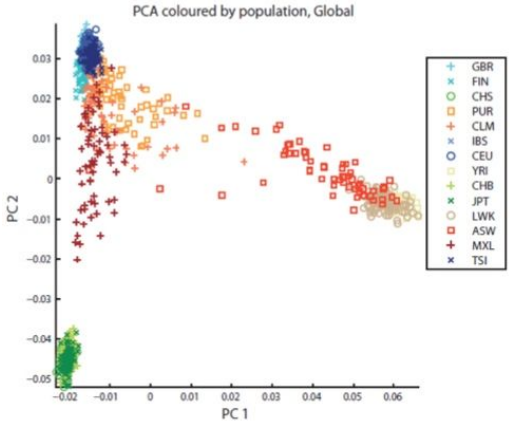
...but also could be <u>heterozygosity</u> (genome copy inherited from one parent matches the reference while the other copy doesn't)

Suppose HALF the reads had a C at this position instead of a T but otherwise matched the reference

What could cause this?

A more closely related reference is better

- ➔ Fewer true variants mean that alignment is easier
- ➔ Some regions in the sequenced individual might be entirely changed from or added to the reference



Genetic variants clustered and colored by population
http://massgenomics.org/2012/11/human-genetic-variation-1000-genomes.html

How to reduce bias?

- ➔ One common approach is to characterize variants and assemble genomes from many different populations
- ➔ For example, the Simons Genome Diversity Project sequenced 260 individuals from different populations:
  - 39 Africans
  - 23 Native Americans
  - 27 Central Asians or Siberians
  - 49 East Asians
  - 27 Oceanians
  - 38 South Asian
  - 71 West Eurasians

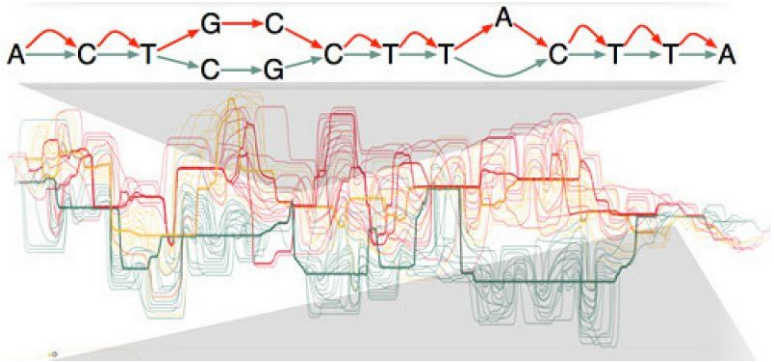
How to choose the best reference for an individual?

- ➔ Common allele representation
- ➔ Choose one based on known information about ancestry
- ➔ Or try all of them and pick the best (though computationally expensive)

In any case, individuals without a close relative in the reference panel or with mixed ancestry are likely to have less accurate results

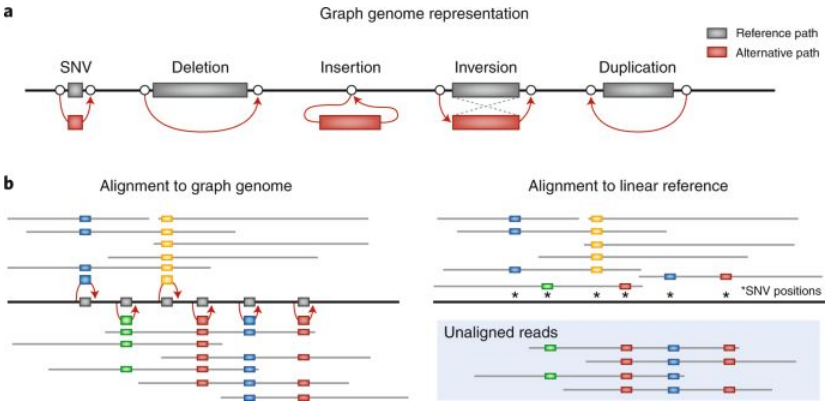
Graph Genome

- ➔ A single data structure which stores a whole collection of genomes
- ➔ Definition:
  - ◆ *Nodes*: Single nucleotides in a genome
  - ◆ *Edges*: Pairs of nucleotides which occur consecutively in at least one genome
- ➔ For a single genome, this is a line graph, with additional genomes showing up as branching paths in the regions where they differ



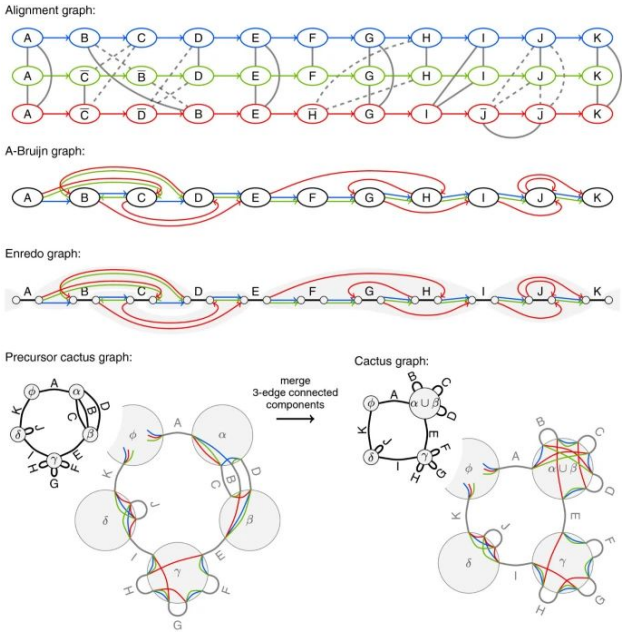
Graph Genome

- ➔ Advantages:
  - ◆ Succinct representation of many genomes
  - ◆ Allows you “pick and choose” parts of different genomes
- ➔ Disadvantages
  - ◆ Requires specialized software and algorithms to be written to deal with genomes in this format
  - ◆ Unclear concept of genomic coordinates

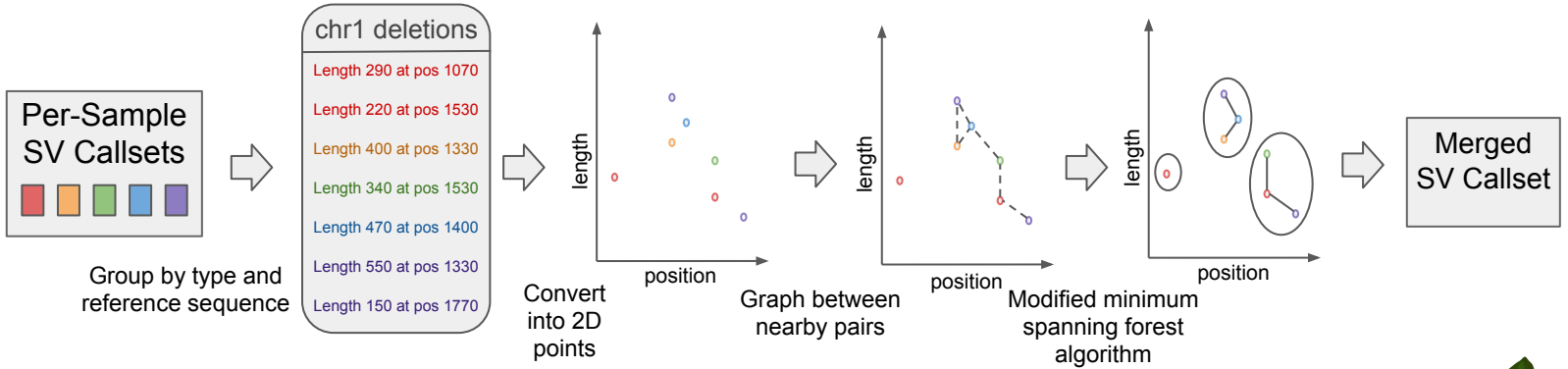


Graphs for Graph Genome Alignment

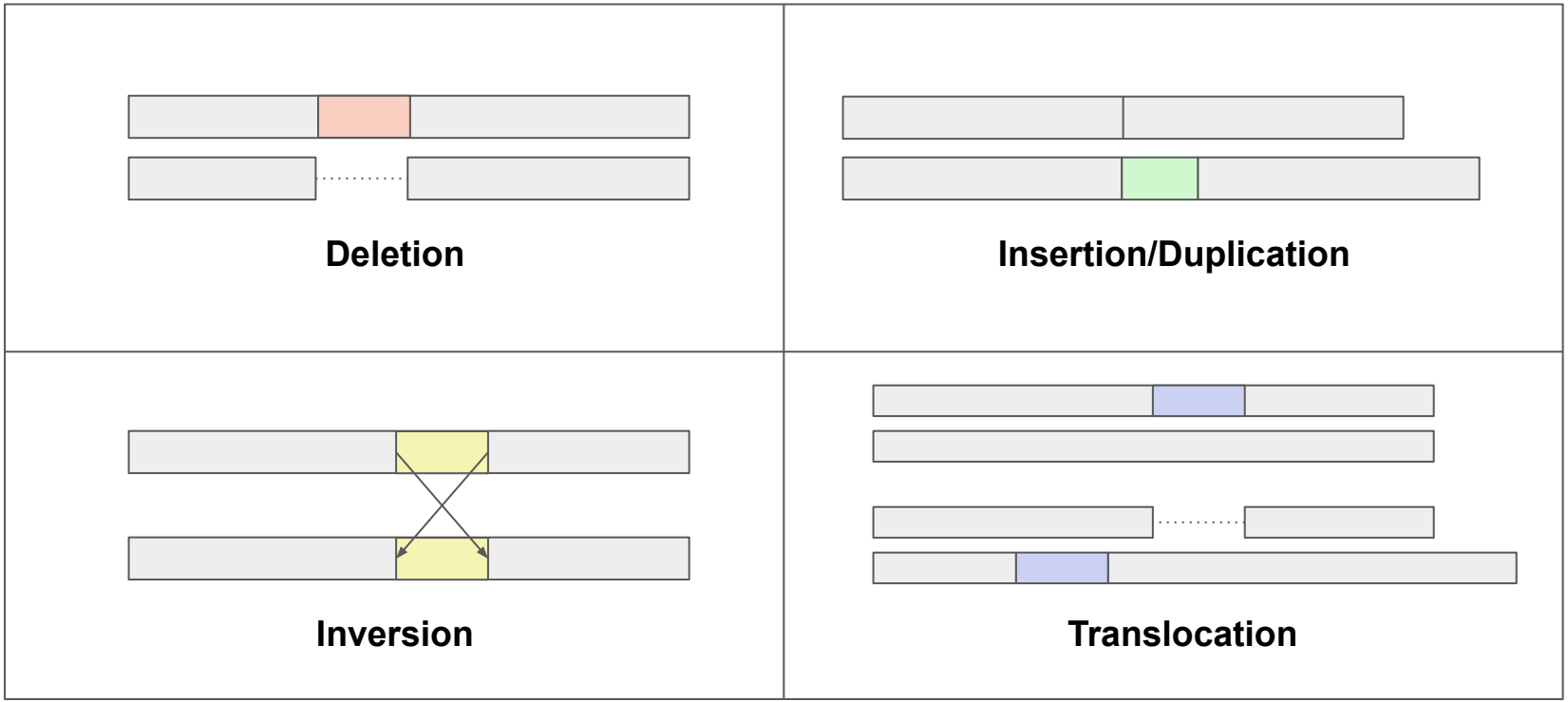
Graph genomes can be represented in many different forms to make problems such as read alignment easier.



Jasmine: Graphs for Structural Variant Comparison

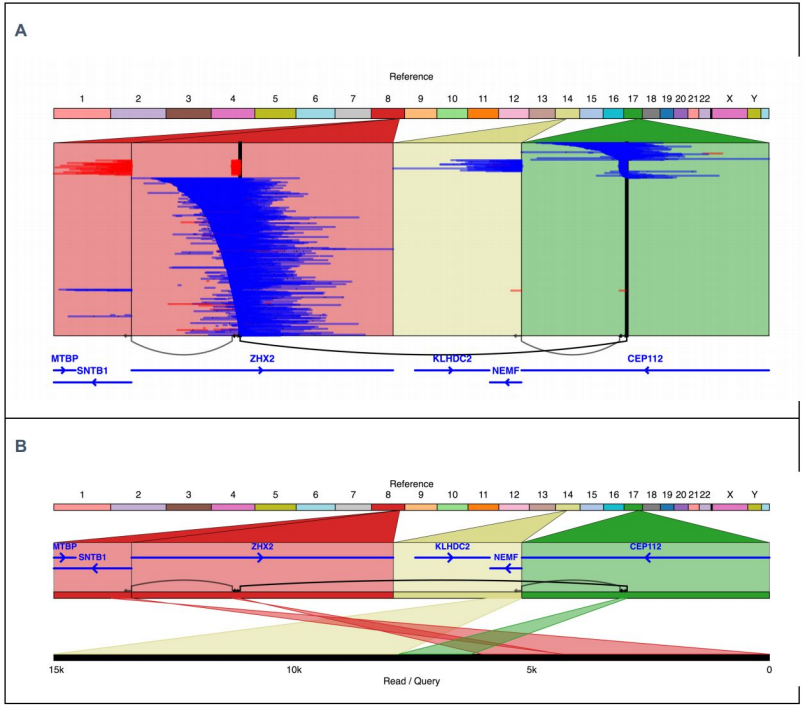


Main Classes of Structural Variants (SVs)



Why do we care about SVs?

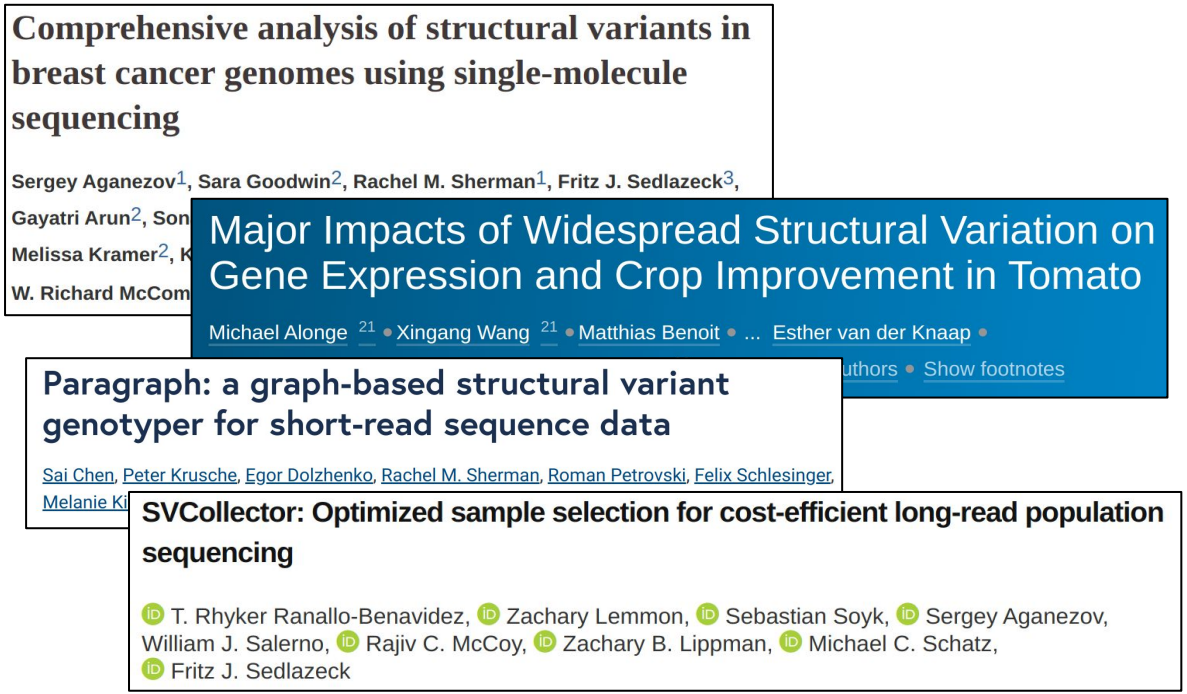
- ➔ Responsible for more divergent base pairs than any other type of variation
- ➔ Known examples of SVs with functional and/or disease impact
- ➔ Commonly introduced during evolution of cancer cells



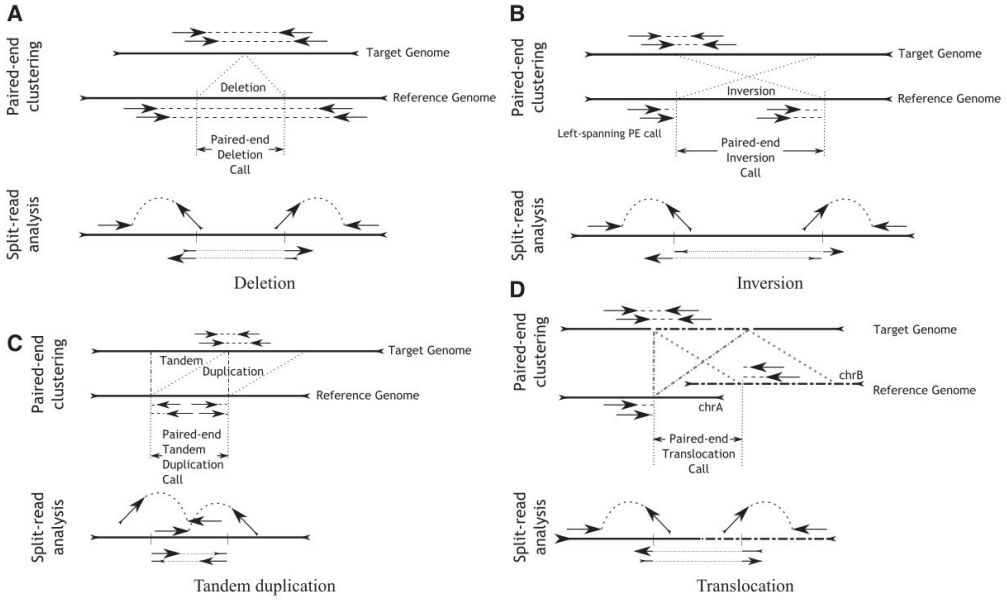
Supplementary Figure 18. Ribbon plot of "3-hop" KLHDC2-SNTB1 gene fusion captured by long reads. This is a "3-hop" gene fusion in SK-BR-3 created by a series of three variants (A). These variants are captured together in several individual SMRT sequencing reads, one of which is shown in (B).

Maria Nattestad, Sara Goodwin, Karen Ng, Timour Baslan, Fritz J. Sedlazeck, Philipp Rescheneder, et al. "Complex rearrangements and oncogene amplifications revealed by long-read DNA and RNA sequencing of a breast cancer cell line". Genome Res. 2018 Aug; 28(8): 1126–1135. doi: 10.1101/gr.231100.117

SV Processing is one of the major focuses of the Schatz lab



Many SV callers look for distinct signatures



Examples of SV signatures in short-read alignments

*(*Rausch et. al.*, DELLY: structural variant discovery by integrated paired-end and split-read analysis,* Bioinformatics 2018*)*

Consensus SV Calling

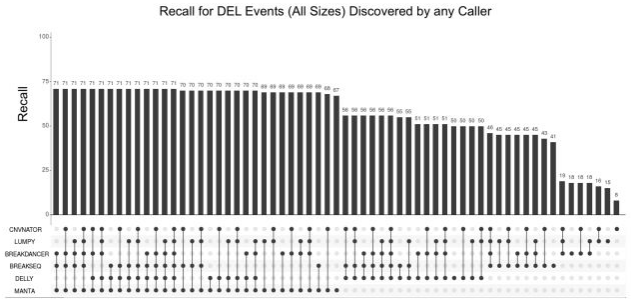


Figure 6. Recall of Parliament2 for DEL Events for an Event Found in any Sample

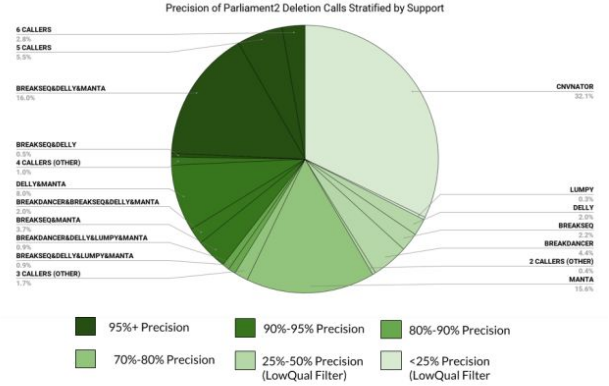
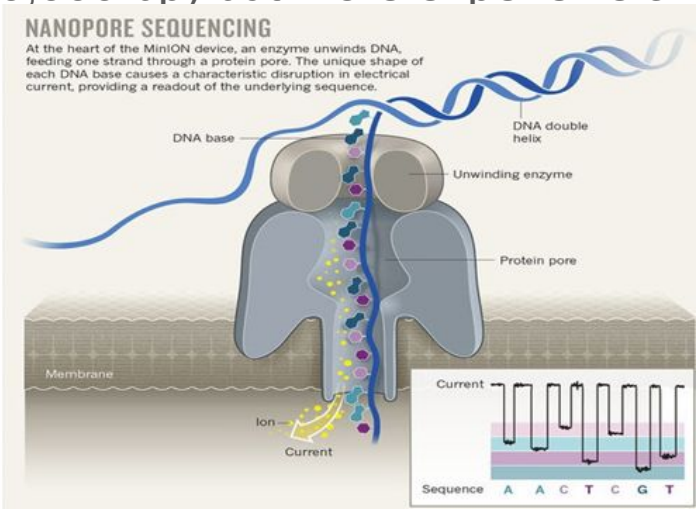


Figure 3. Precision of Parliament2 for DEL events stratified by support from specific callers

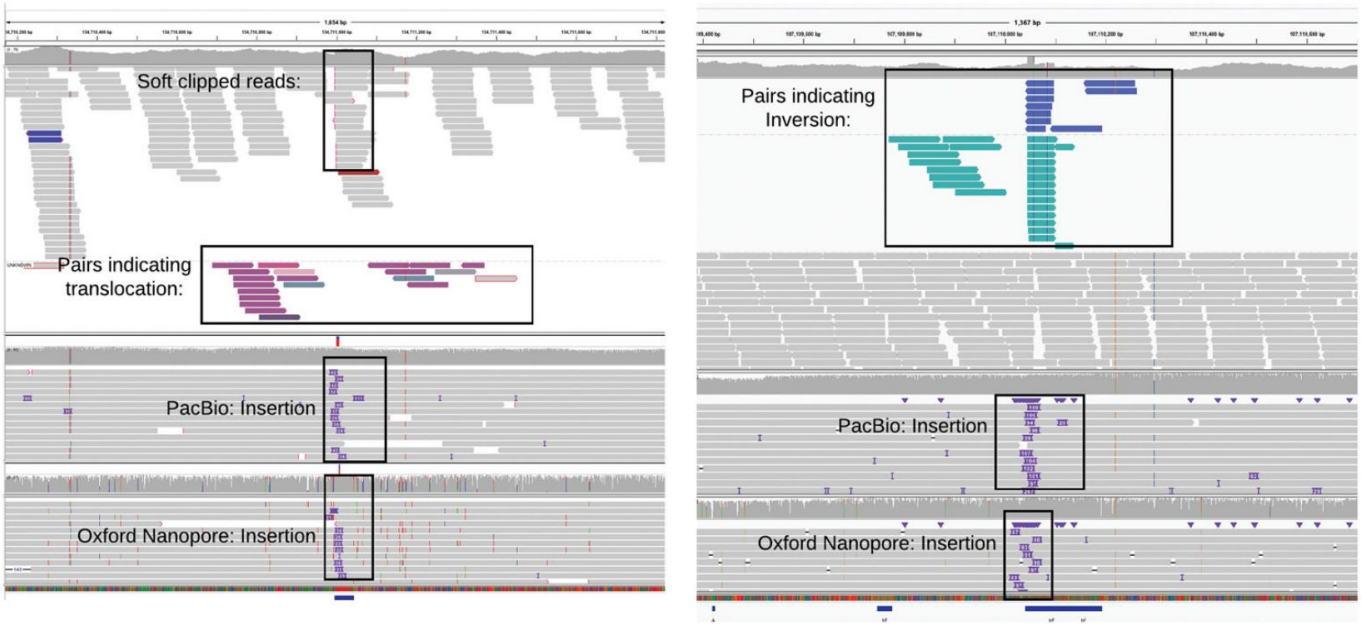
- Due to the lower cost of short-read sequencing, many methods exist for calling SVs from short reads
- These callers have low accuracy on their own, but consensus methods combine them to achieve greater accuracy

Different Read Technologies

- ➔ Last week we talked about 500 bp Illumina sequencing reads which are 99.9% accurate
- ➔ In more recent years, other sequencing technologies have been developed which produce longer (> 50,000 bp) but more expensive and less accurate (~95% accuracy) reads



Long Reads Enable Better SV Detection



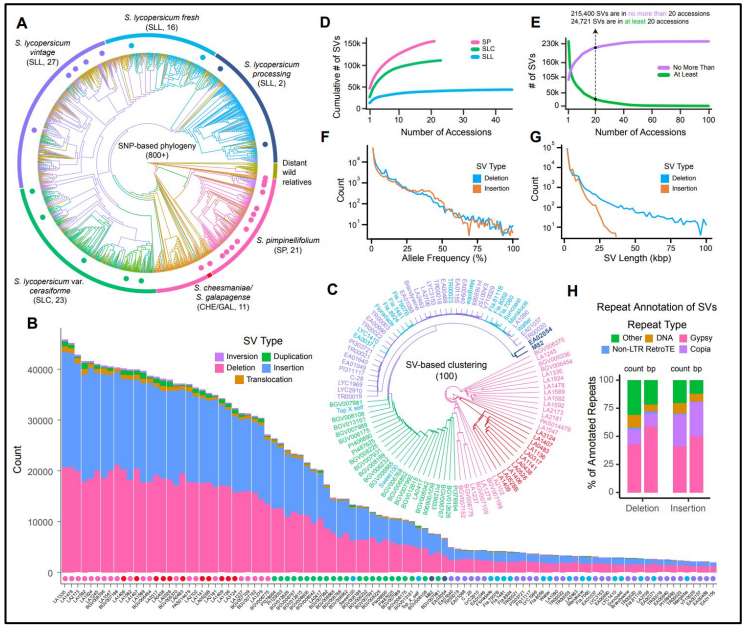
Examples of instances where short-read alignments mistake insertions for translocations or inversions

*(*Sedlazeck, Rescheneder et. al.*, Accurate detection of structural variations using single molecule sequencing,* Nature Methods 2018*)*

Population SV Studies

- ➔ As long-read sequencing becomes more prevalent, studies of SVs with multiple individuals are becoming more common
- ➔ Enables better understanding of functional impact through association with phenotype or phenotype-correlated small variants

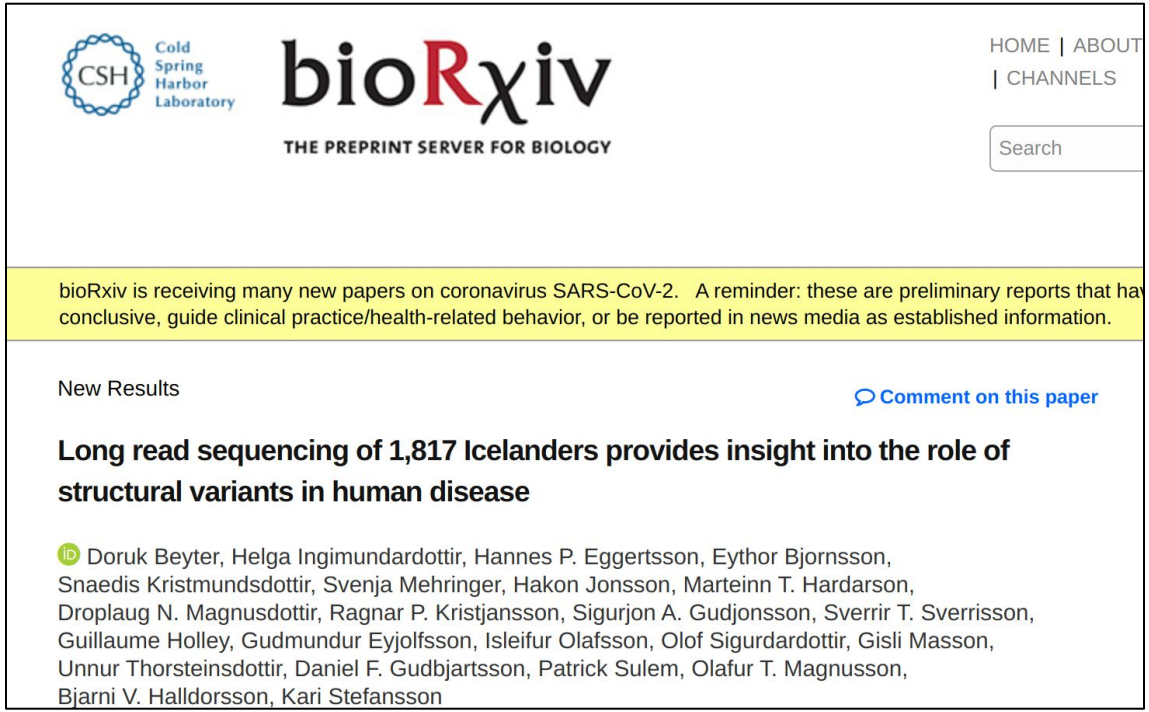
SVs in 100 Tomato Accessions



Alonge, Michael, Xingang Wang, Matthias Benoit, Sebastian Soyk, Lara Pereira, Lei Zhang, Hamsini Suresh, et al. 2020. "Major Impacts of Widespread Structural Variation on Gene Expression and Crop Improvement in Tomato." *Cell* 182 (1): 145–61.e23.

Long-read datasets are only getting bigger

- ➔ Bigger cohorts (e.g., Iceland paper)
- ➔ Higher coverage
- ➔ Better SV detection methods



Short-read genotyping methods enable us to leverage long-read calls to revisit short-read datasets through a more powerful lens

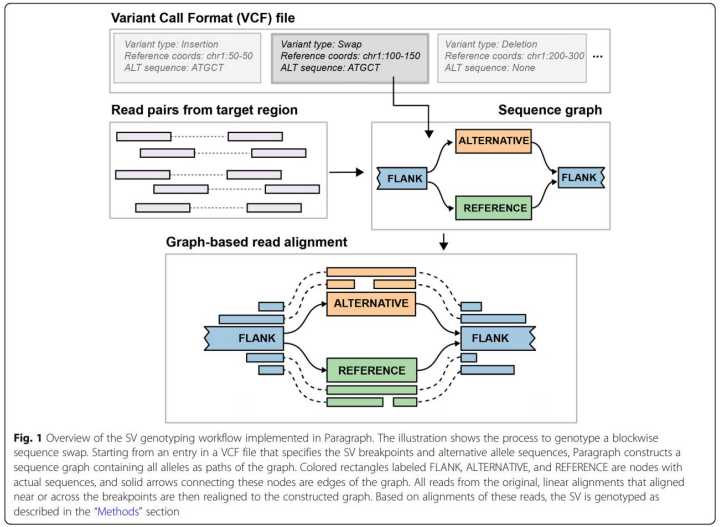
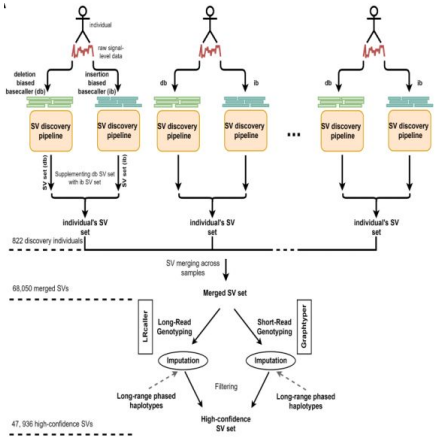


Fig. 1 Overview of the SV genotyping workflow implemented in Paragraph. The illustration shows the process to genotype a blockwise sequence swap. Starting from an entry in a VCF file that specifies the SV breakpoints and alternative allele sequences, Paragraph constructs a sequence graph containing all alleles as paths of the graph. Colored rectangles labeled FLANK, ALTERNATIVE, and REFERENCE are nodes with actual sequences, and solid arrows connecting these nodes are edges of the graph. All reads from the original, linear alignments that aligned near or across the breakpoints are then realigned to the constructed graph. Based on alignments of these reads, the SV is genotyped as described in the "Methods" section

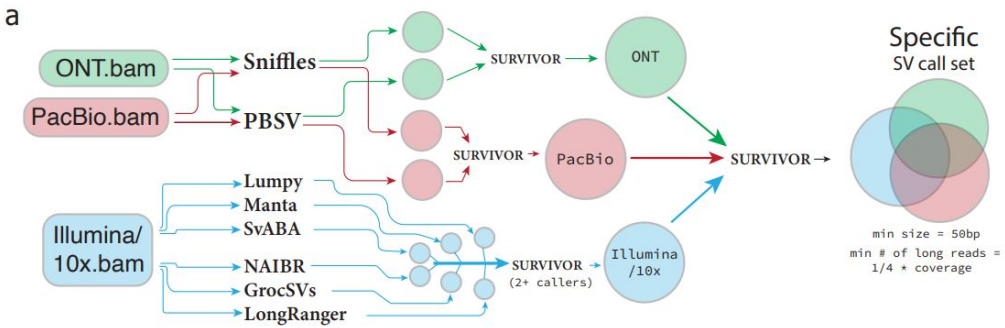
Chen S, Krusche P, Dolzhenko E, Sherman RM, Petrovski R, Schlesinger F, Kirsche M, Bentley DR, Schatz MC, Sedlazeck FJ, Eberle MA. Paragraph: A graph-based structural variant genotyper for short-read sequence data. bioRxiv. 2019;24:635011. https://doi.org/10.1101/635011.

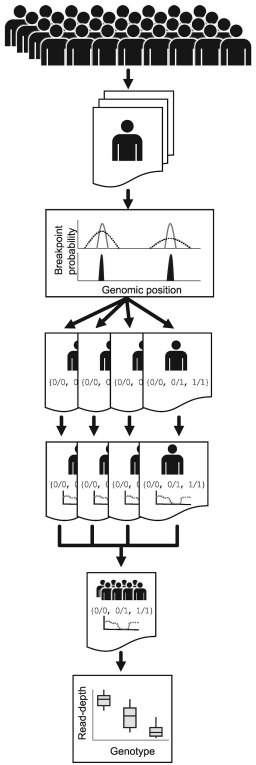
SV calling datasets and methods are heterogeneous



Potential differences in:

- Sequencing technology
- Coverage
- Aligner
- Variant Caller
- Pre- and post-processing





Thousands of WGS samples

Parallel SV discovery in each sample
`speedseq sv` and `lumpy`

SV merging and breakpoint refinement
`svtools lsort` and `svtools lmerge`

Re-genotyping using SVTyper
`svtools genotype`

Copynumber annotation with CNVnator
`svtools copynumber`

Cohort-level SV VCF creation
`svtools vcfpaste`
`svtools afreq`
`svtools prune`

Variant type reclassification
`svtools classify`

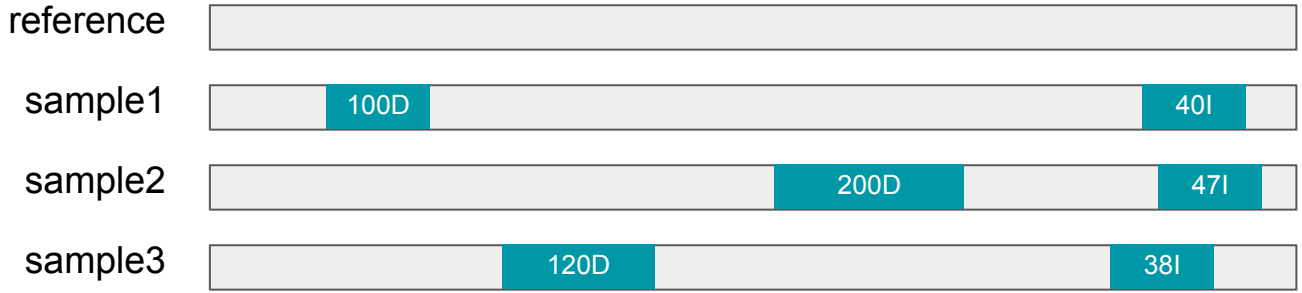
Goal: Identify interesting or disease-relevant SVs

A number of methods:

- ➔ Presence in disease individuals but absent from healthy individuals (association)
- ➔ Proximity to genes
- ➔ Other genomic functional indicators like effect on reading frame
- ➔ Incorporating expression data
- ➔ Gene editing
- ➔ Other population genetics methods

Population-Scale Structural Variant Comparison

When can two or more variants be called the same?



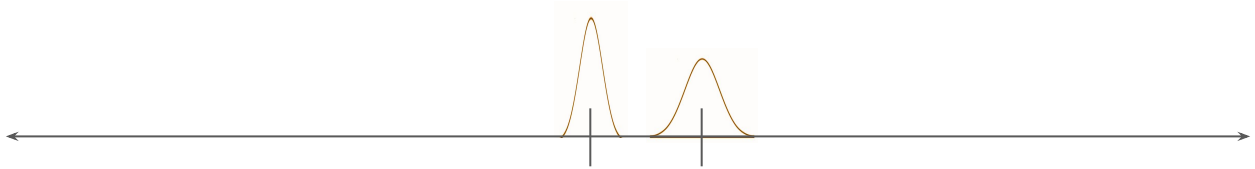
Depends on technology used, sequencing error, basecalling, alignment, variant calling, normalization, true differences in the phenotypes, and more

Representing Structural Variants

1-dimensional points

Breakpoint Intervals

Breakpoint Distributions



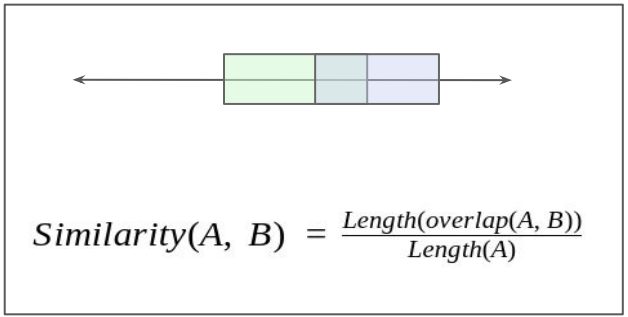
Note: Definition usually also includes chromosome and type

Representing Structural Variants

Representation	Pros	Cons
Single Point	• Very lightweight • Natural ordering • Simple distance metric	• Ignores size of SVs
Interval (2-D Point)	• Simple distance metric	• Doesn't take into account confidence of breakpoints
Distribution	• Accounts for less precise/confident SV calls	• Requires a lot of time/memory • Difficult to define distance between variants • Not always reported by callers

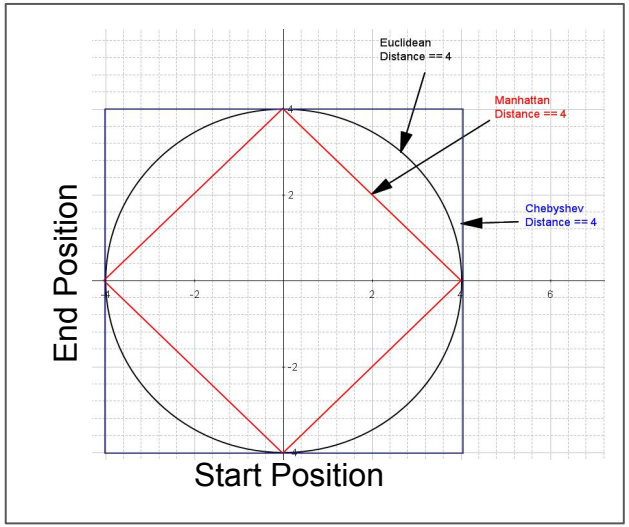
Similarity/Distance Metrics

Interval Overlap Similarity

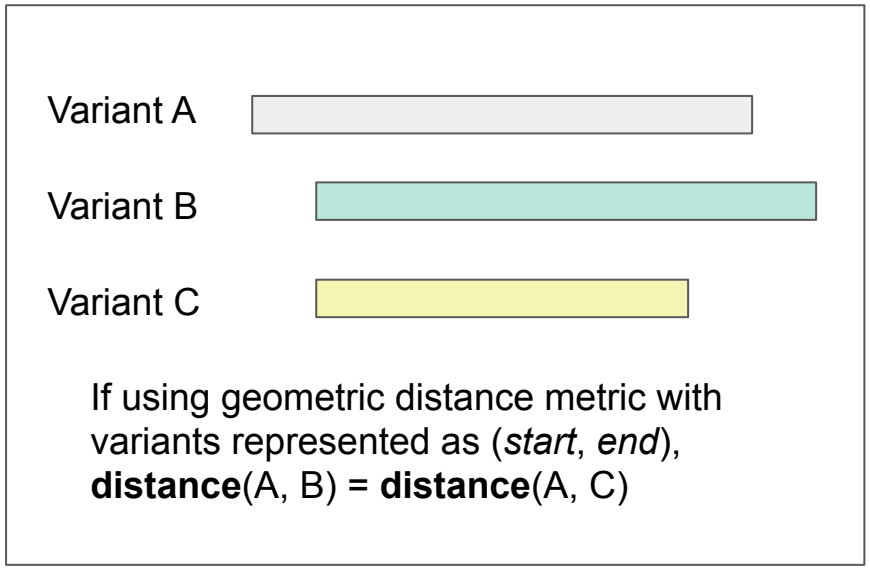


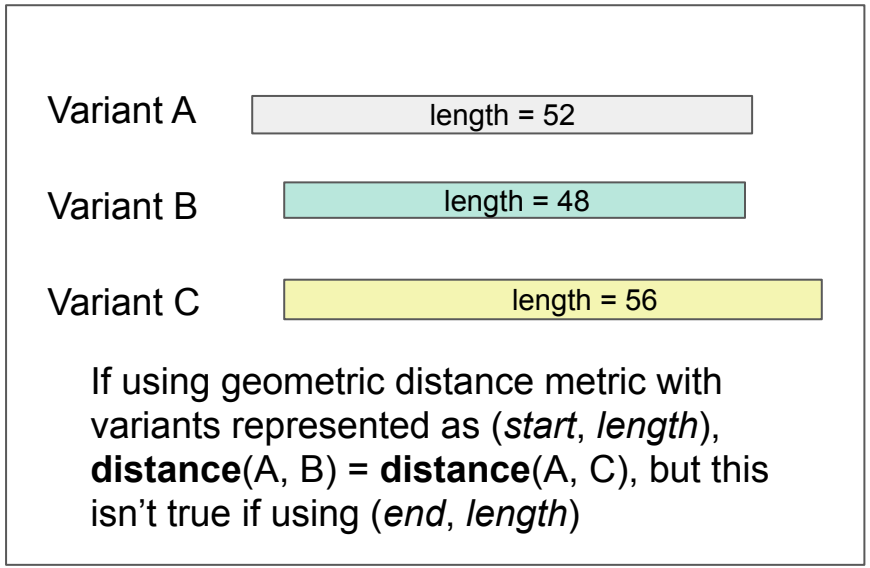
$$Similarity(A,\ B) = \frac{Length(overlap(A,\ B))}{Length(A)}$$

Geometric Distances



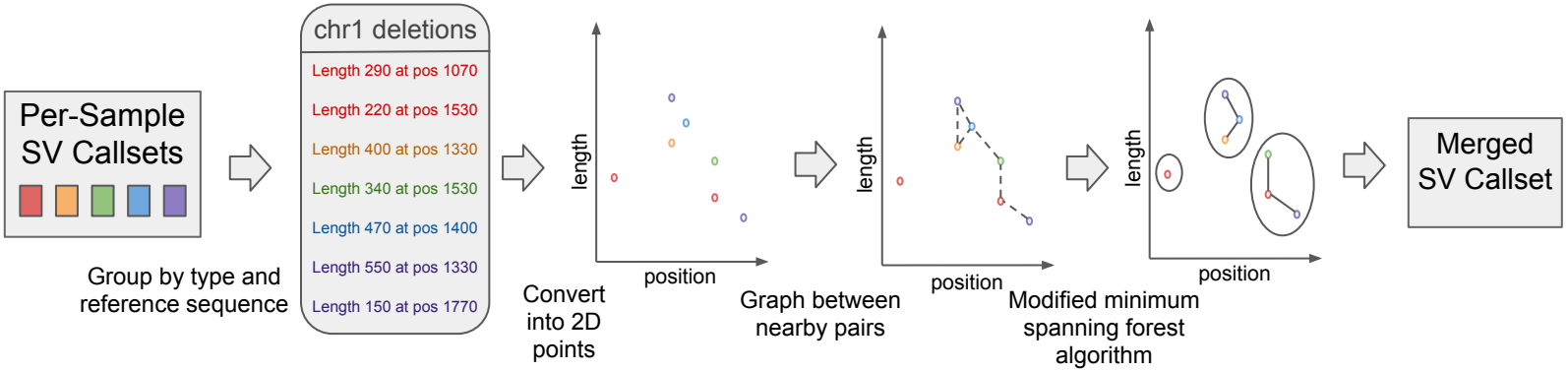
Variant Similarity is Complicated





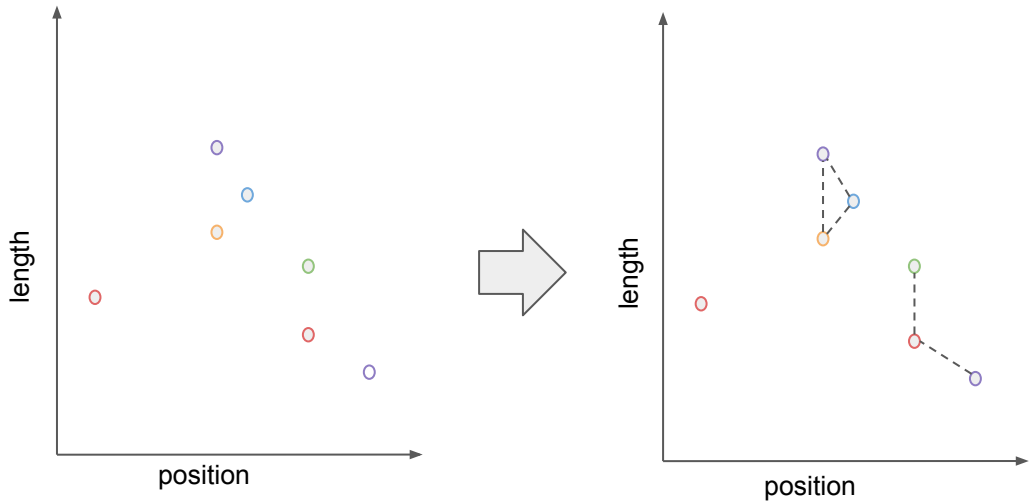
Variant B will be closer to Variant A if using (*start*, *length*) as the representation instead!

Jasmine Methods



By using an algorithm similar to Kruskal's algorithm for minimum spanning trees, Jasmine guarantees that it performs merging in a way which minimizes the sum of the distances of joins it performs.

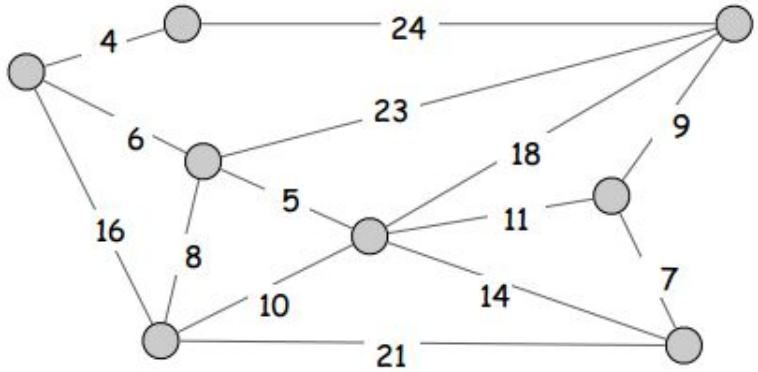
Graph Representation of Equivalent SV Pairs



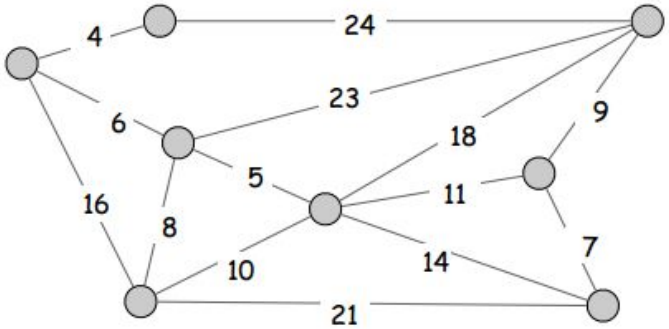
The entire graph is way too large to store in memory, but different merging algorithms use different parts or approximations of it to decide what to merge.

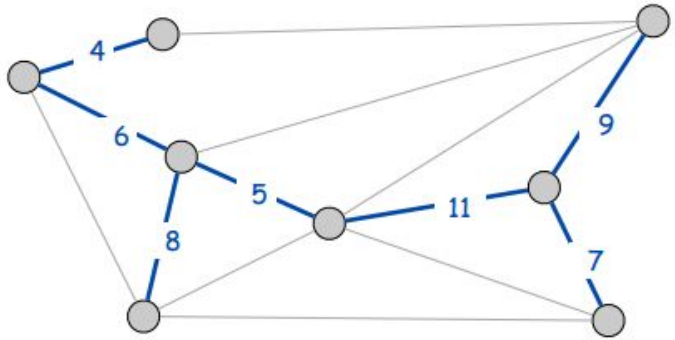
Related Problem: Minimum Spanning Tree

Problem: Find a set of edges which are enough to connect the graph into a single connected component,with the smallest total edge weight possible.



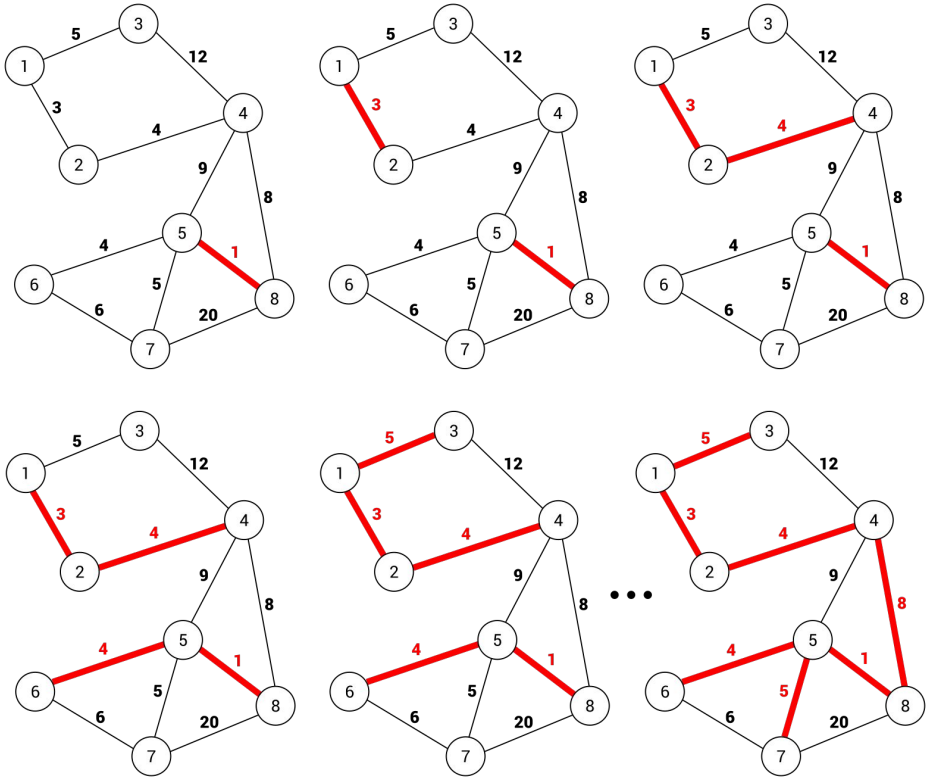
Minimum Spanning Tree





T, $\Sigma_{e \in T} c_e = 50$

Kruskal's Algorithm

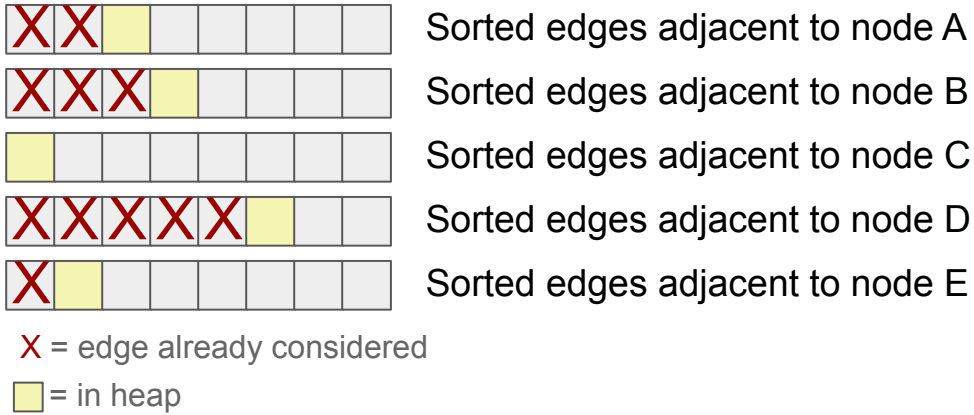


Jasmine Algorithm

1. Consider edges from smallest to largest weight
2. If an edge is valid (doesn't connect variants from the same sample, sufficient sequence similarity, etc.), merge the variants incident to it
3. When the edge weights are above merging threshold, stop

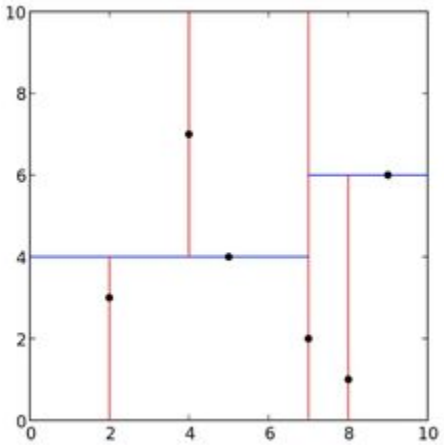
Adapting Kruskal's Algorithm

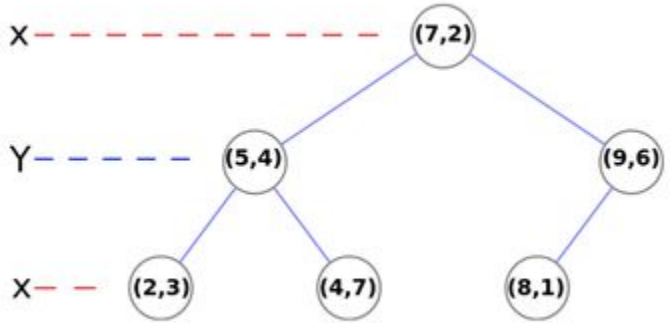
- ➔ Normally Kruskal's algorithm involves storing all of the edges of the graph in a heap or sorted list, and processing them from smallest weight to largest
- ➔ At any given time point, we'll only need to consider the smallest unprocessed edge adjacent to each vertex



Merging Algorithm

Getting smallest edge weights for each variant requires finding nearest neighbors





KD Trees allow answering K-nearest-neighbor queries in expected O(log(n) + K) time

Merging Algorithm - Storing Graph Dynamically

Nearest neighbors for each variant (initially query the KD tree for 2 of each - fetch more later if needed)

Nearest Neighbors

```
A: (B, 10), (C, 15)
B: (E,  5), (D,  7)
C: (A, 15), (B, 25)
D: (B,  7), (E, 12)
E: (B,  5), (D, 12)
```

Min-heap of the **closest neighbor for each variant** (its first entry in the nearest neighbor table).

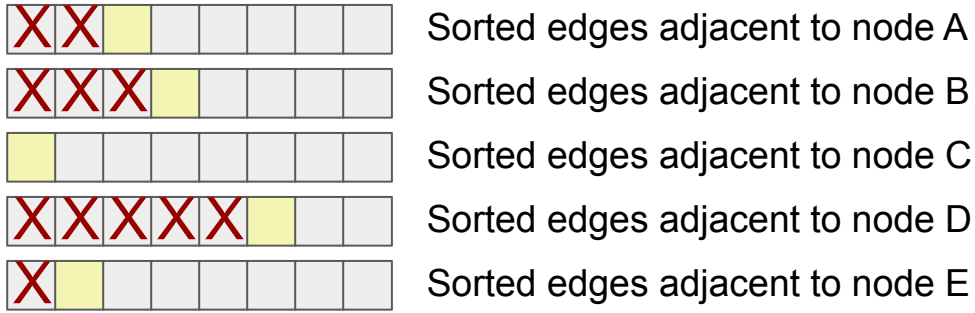
Variant	Neighbor
B	(E, 5)
E	(B, 5)
D	(B, 7)
A	(B, 10)
C	(A, 15)

This tells us the next edge in the graph to try

Merging Algorithm

Variant	Neighbor
B	(E, 5)
E	(B, 5)
D	(B, 7)
A	(B, 10)
C	(A, 15)

Next edge to consider and remove

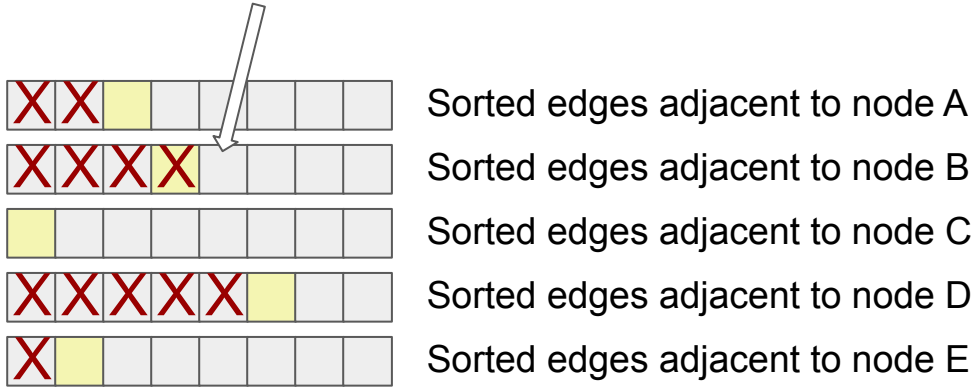


Merging Algorithm

Variant	Neighbor
~~B~~	~~(E, 5)~~
E	(B, 5)
D	(B, 7)
A	(B, 10)
C	(A, 15)

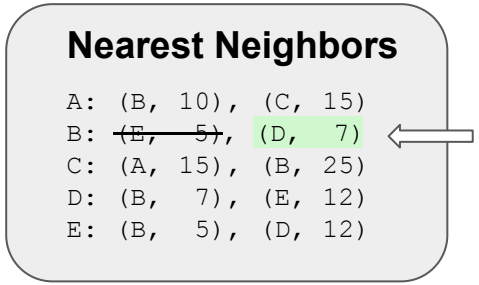
Just used this edge

But now we need to add this edge to replace the one we used



Merging Algorithm

Variant	Neighbor
E	(B, 5)
B	(D, 7)
D	(B, 7)
A	(B, 10)
C	(A, 15)



We can get B's next-smallest edge from the nearest neighbors list

Merging Algorithm

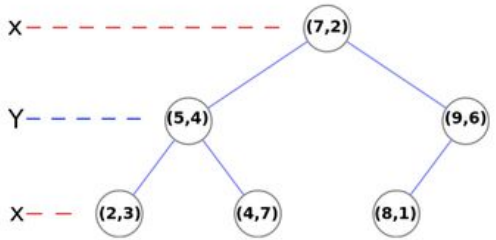
What if a variant runs out of nearest neighbors?

Nearest Neighbors

A: (B, 10), (C, 15)
B: ~~(E, 5)~~, ~~(D, 7)~~, (A, 10), (C, 25)
C: (A, 15), (B, 25)
D: ~~(B, 7)~~, (E, 12)
E: ~~(B, 5)~~, (D, 12)

Query KD-tree for 4 nearest neighbors:

1. (E, 5)
2. (D, 7)
3. (A, 10)
4. (C, 25)



But we can't afford to keep making queries to the KD-tree all the time, so query twice as many neighbors as last time (2, then 4, then 8, etc.) - guarantees at most log(n) queries per variant.

Merging Algorithm

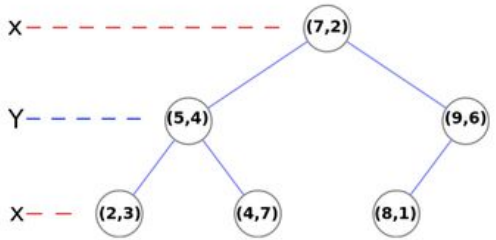
What if a variant runs out of nearest neighbors?

Nearest Neighbors

A: (B, 10), (C, 15)
B: ~~(E, 5)~~, ~~(D, 7)~~, (A, 10), (C, 25)
C: (A, 15), (B, 25)
D: ~~(B, 7)~~, (E, 12)
E: ~~(B, 5)~~, (D, 12)

Query KD-tree for 4 nearest neighbors:

1. (E, 5)
2. (D, 7)
3. (A, 10)
4. (C, 25)



But we can't afford to keep making queries to the KD-tree all the time, so query twice as many neighbors as last time (2, then 4, then 8, etc.) - guarantees at most log(n) queries per variant.