



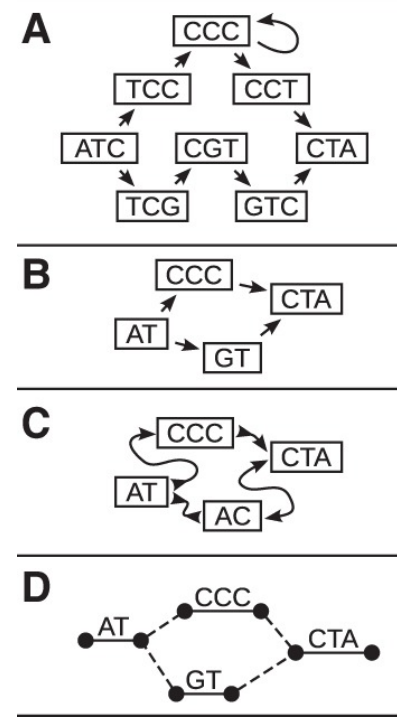
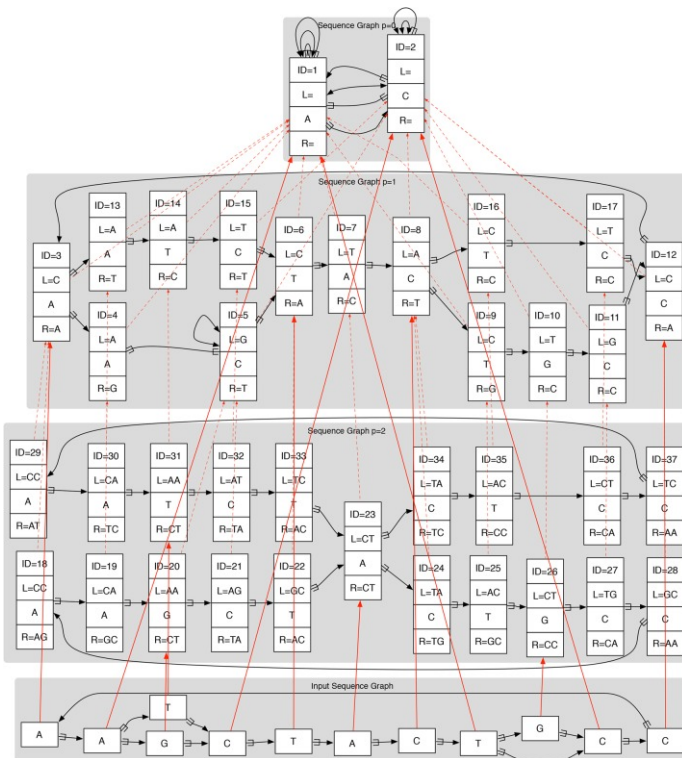
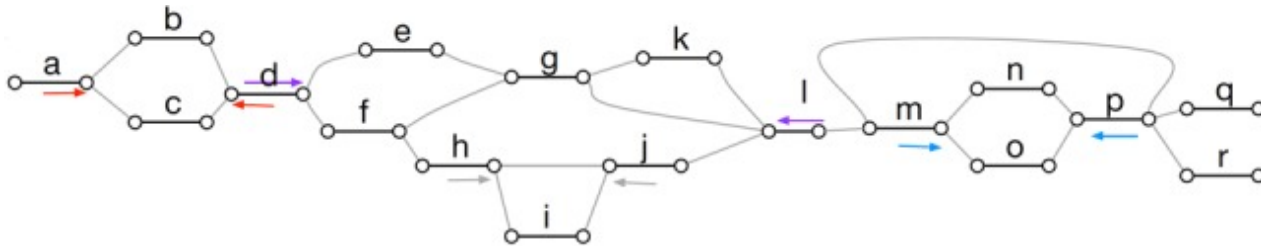
United States  
Department of  
Agriculture

Agricultural  
Research  
Service

# Inferring Genotypes from Skim Sequence using a Graph-Based Approach: The Practical Haplotype Graph

Peter Bradbury, Terry Casstevens, Dan Ilut, Lynn Johnson, Zachary  
Miller, Ramu Punna, Maria Cinta Romay, Edward Buckler

# Genome Graphs Represent Diversity



Figures taken from a review by Paten et al.  
 Genome Res. 2017 May; 27(5): 665–676.  
 doi: 10.1101/gr.214155.116

# Using a Pan-Genome

- Aligning to a pan-genome is better than aligning to a single reference
- Representing a pan-genome as a graph works
- There are a variety of approaches to graphical genomes
- Creating and using pan-genome graphs is complex

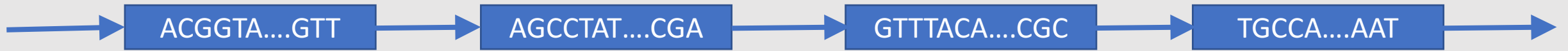
# Finding a Practical Method of Using Multiple Genomes

The maize genome consists of

- Conserved genes (and other elements)
  - ~38,000 anchored intervals
  - ~8-10% of the genome
- Non-conserved intergenic intervals (highly variable)
- Architecture similar across many species

	Gene1		Gene2
B73	AGCGT	ACGAGT - - - - CATGA	CGTAA
Mo17	ACCGT	ACGNNTAAAACATGA	CGCAA
Oh43	AGTCT	ACGAGTAA - - CANNA	CGCAA

# A chromosome is a sequence of haplotypes



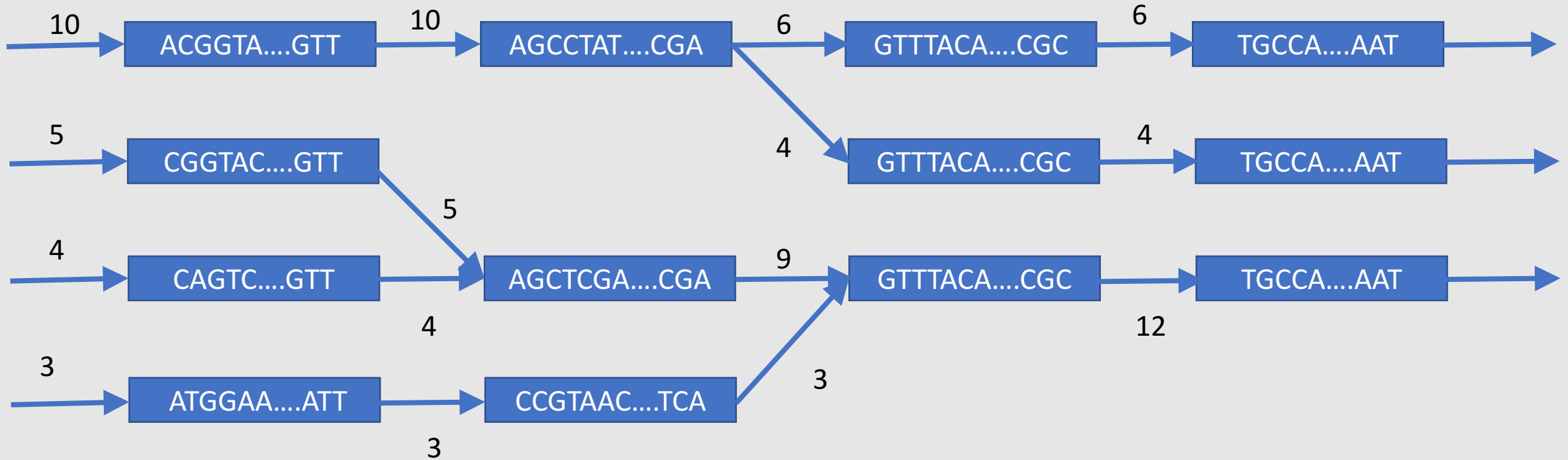
Node

A node is a haplotype

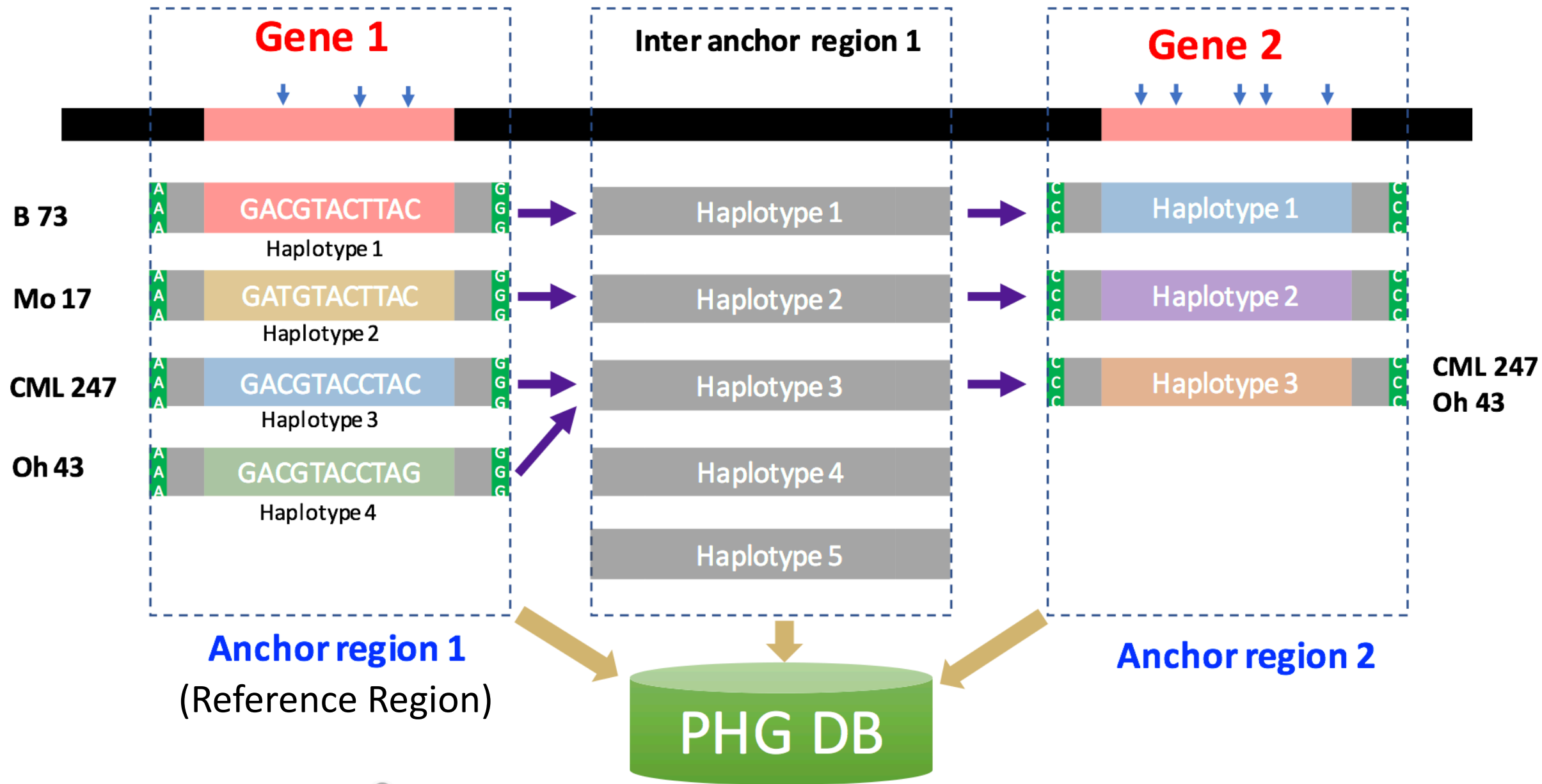
Edge

An edge is the connection between two nodes

# Population of chromosomes



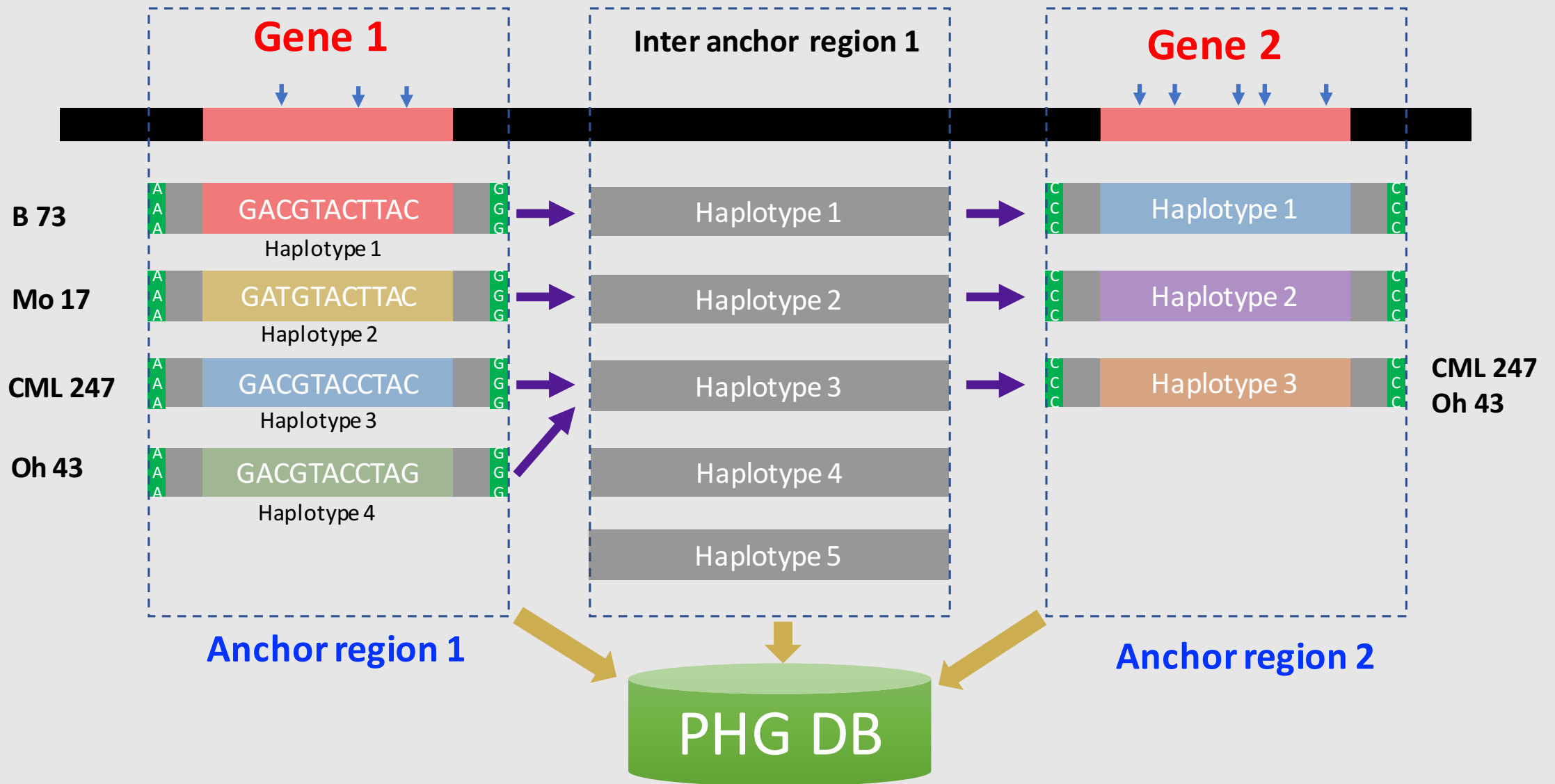
Edges are weighted by number of times observed in population



# Create Consensus Haplotypes

- Cluster haplotypes within each reference range
- Reduce memory footprint
- 308 inbred line pangenome → 8 GB RAM
- Increase haplotype coverage → better quality





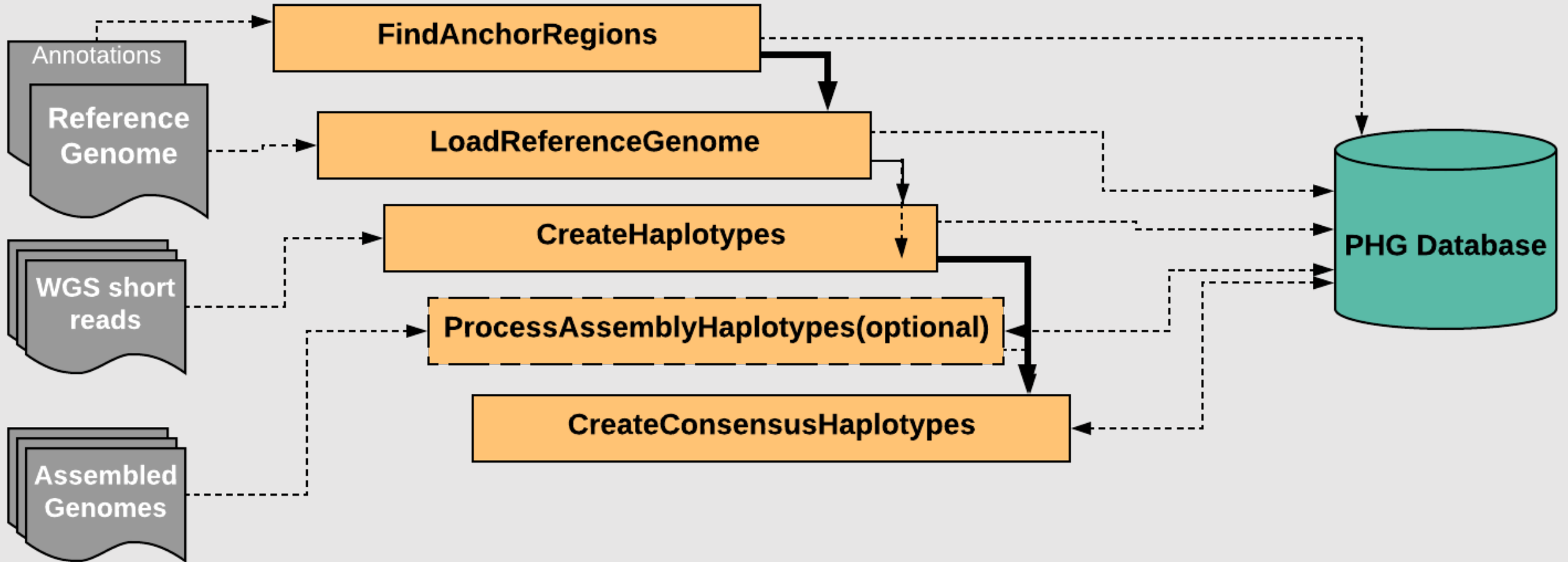
**Key elements:**

- Coordinates of reference genome attached to anchor nodes
- Anchor node have nearly 100% conserved start and ends

# What is a Practical Haplotype Graph?

- A graph-based representation of multiple genomes
- A graph-based representation of the variation present in a population
- A computational framework
- A database
- Used to impute variants from skim sequence

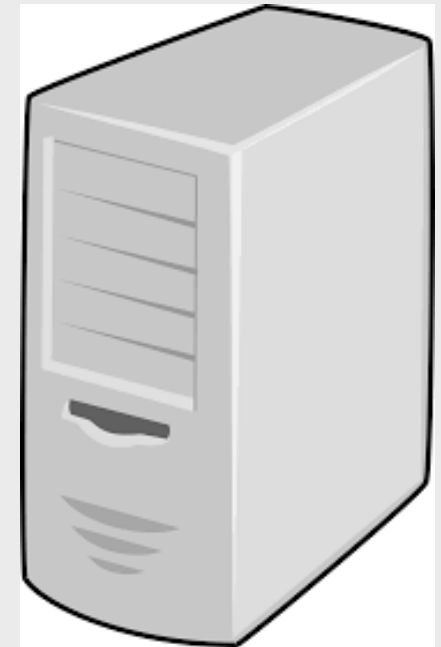
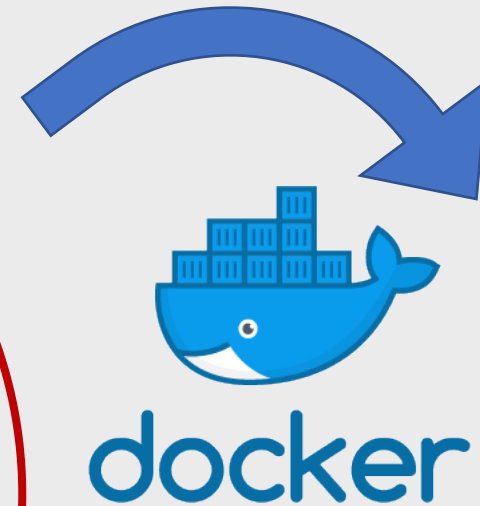
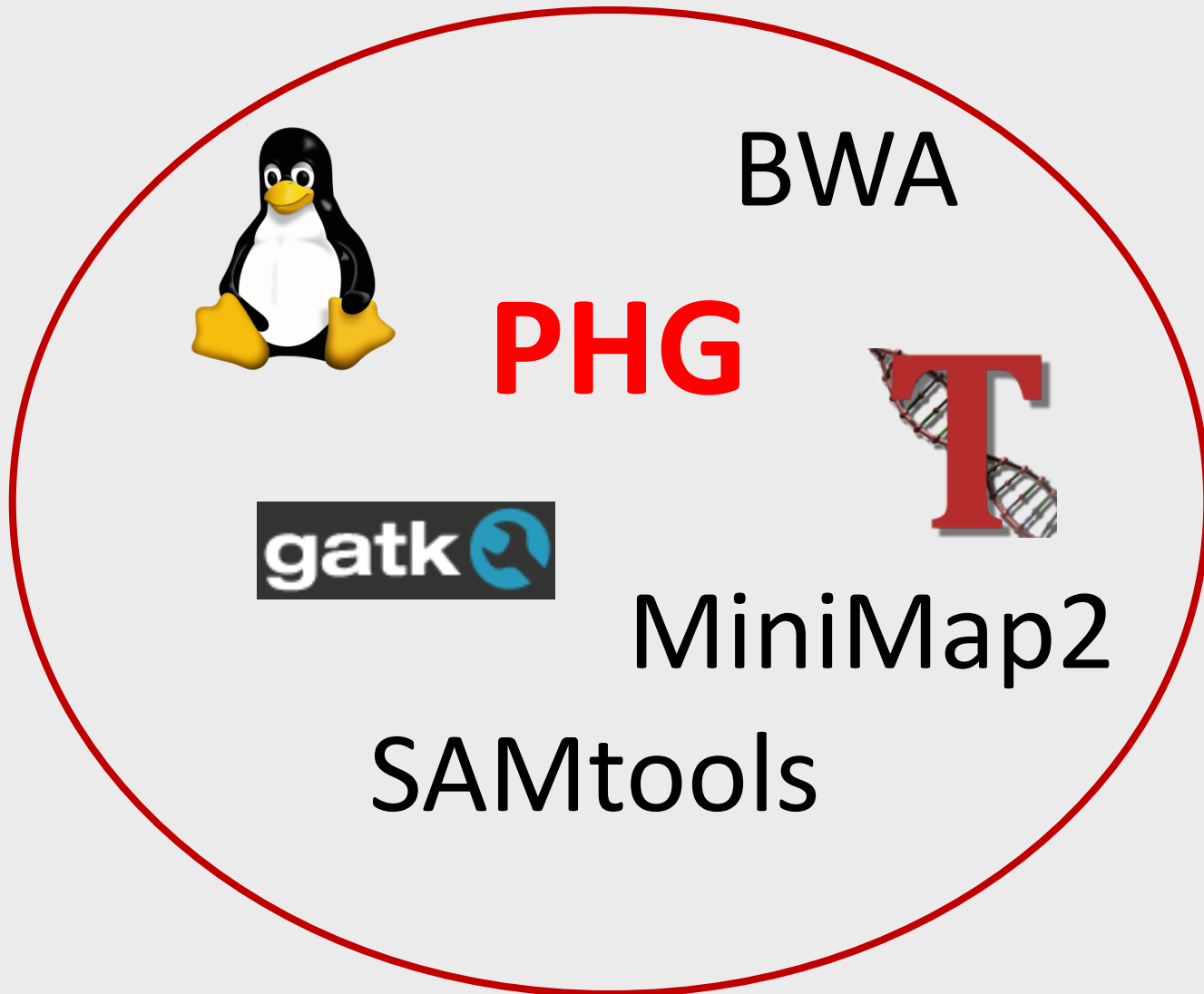
# Populating the PHG Database



# The PHG is a computational framework

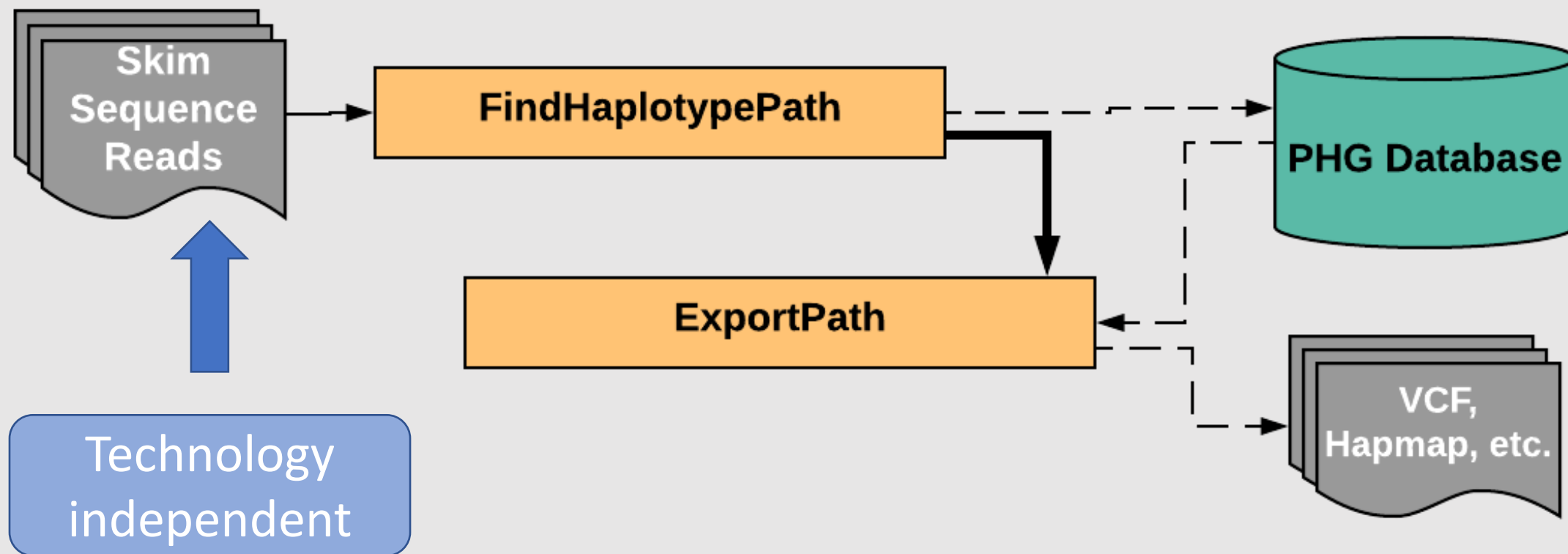
- Database
- Software
  - Populate database
  - Generate the graph in-memory from the database
  - Use the in-memory graph
- Pipeline that uses software from several sources
- Distributed as a Docker image

# A Docker Image Captures The Computing Software Environment

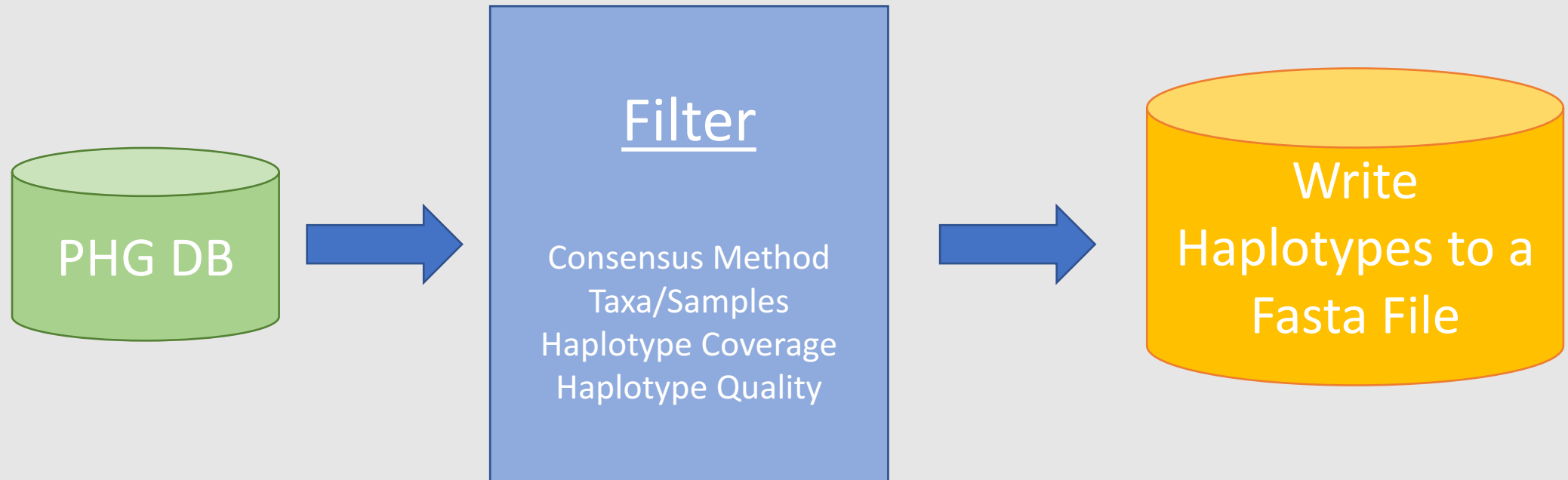


Any Supported OS

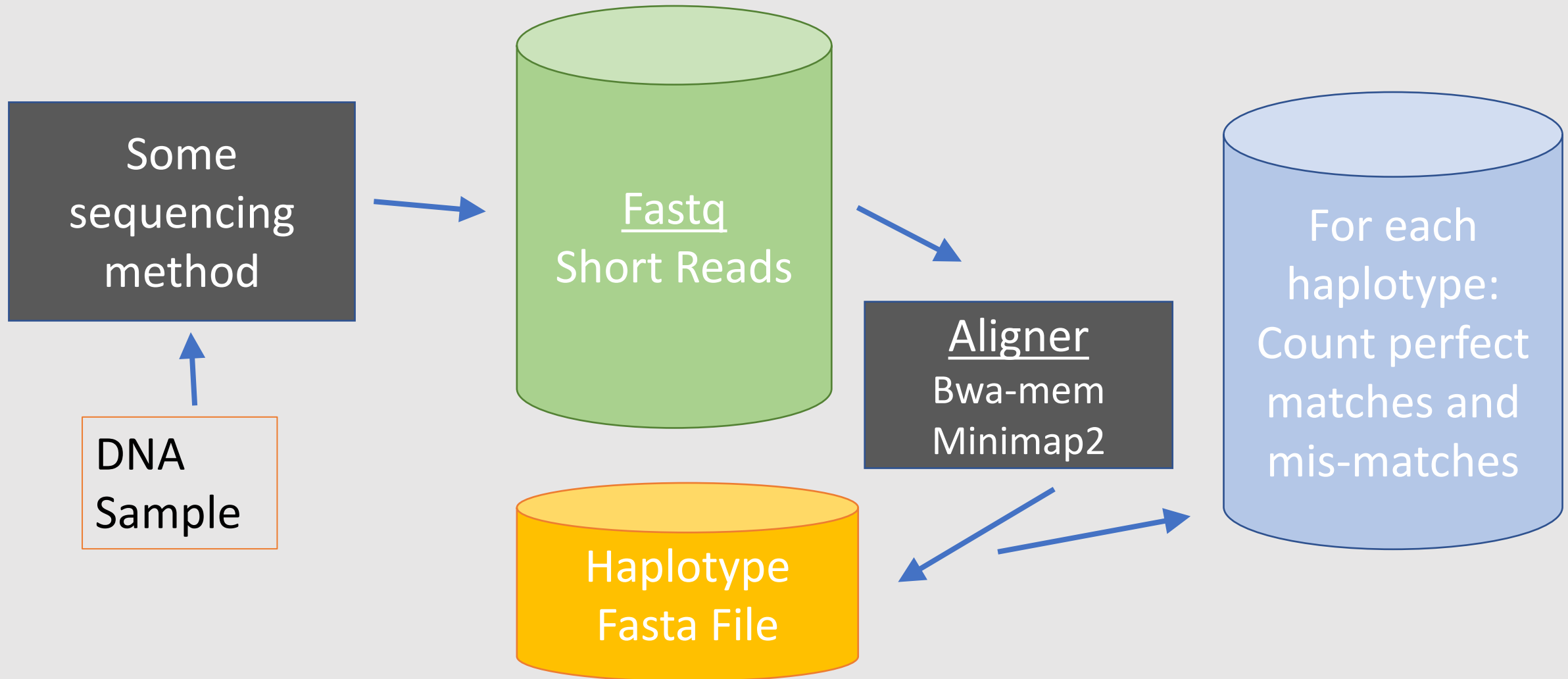
# Genotyping Using a PHG



# Filter and write PHG to Fasta

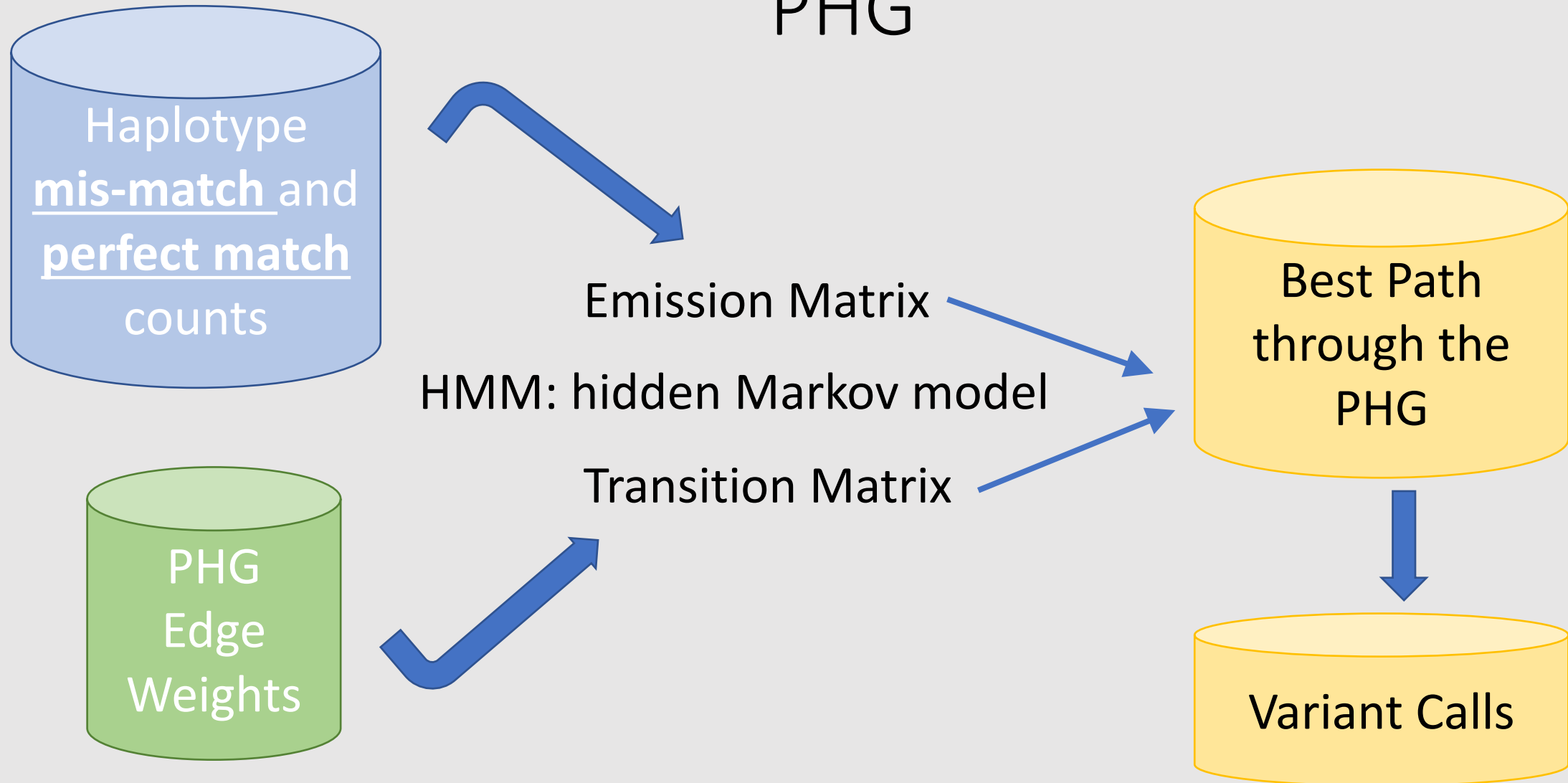


# Align Short Sequence Reads To Haplotype Fastas





# Use an HMM to find the best path through a PHG



## **Current status:**

- **Maize - 308 taxa**
  - **Sorghum - 140 taxa (under development)**
  - **Cassava - 348 taxa (under development)**
- 
- **Tested using W22 GBS sequence**
  - **Pathway: 85% of nodes called correctly**
  - **Error rate calling SNPs – 2% (compared to Axiom array)**
- 
- **GBS reads for 10K taxa (CIMMYT) – processed through PHG**
    - **3.7 M SNPs in anchor regions**
    - **Used for genomic selection**

# Limitations

- Still under active development
- The current genotyping application targets breeding programs
  - Populations with a limited number of founders
- Testing to date has been done with inbred lines

# Key Points

- PHG is a simplified pan-genome graph
- A single reference genome plus moderate coverage WGS is sufficient
- PHG software will be relatively easy to run
- Both haplotype finding and genotyping can combine sequence from a variety of technologies

# Acknowledgements



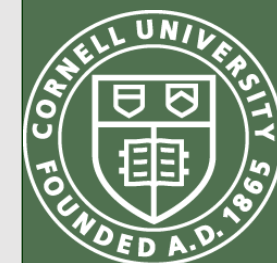
## Ed Buckler Lab

Lynn Johnson  
Terry Casstevens  
Zack Miller  
Ramu Punna  
Cinta Romay  
Dan Ilut  
Sara Miller



## Michael Gore Lab

Dan Ilut



# For more information

- PHG Wiki: documentation, and source code at
- <https://bitbucket.org/bucklerlab/practicalhaplotypegraph>
- Docker image available in March 2018

Poster 114  
Lynn Johnson



bucklerlab / InfraStructure / PracticalHaplotypeGraph

Wiki

PracticalHaplotypeGraph / Home

View History Edit

### Practical Haplotype Graph (PHG)

With improved sequencing technology, sequencing costs are declining very rapidly. But bioinformatics challenge has increased to process the sequencing data to infer genotypes. To address this, we have developed a general, graph-based, computational framework called Practical Haplotype Graph (PHG), that can be used with a variety of skim sequencing methods to infer high-density genotypes directly from low-coverage sequence. Hypothesis behind to develop PHG is – in a given breeding program, all parental genotypes can be sequenced at high coverage and load the parental haplotypes to a relational database. Progenies can be sequenced at low coverage and infer the haplotypes/genotypes from the stored haplotypes in PHG database.

Practical Haplotype Graph: PHG is a trellis graph based representation of genic (anchors) and intergenic regions (inter anchors) which represents diversity across and between species, create custom genome for alignment, calling rare alleles, imputation and data compression. Skim sequences generated from a given taxa are aligned to consensus sequences in PHG to identify the haplotype node at a given anchor. All the anchors for a taxon are processed through Hidden Markov Model (HMM) to identify the most likely path through the graph. Path information is used to identify the variants (SNPs). Low cost sequencing technologies coupled with PHG facilitate in genotyping of large number of samples to increase the size of training population in GS models, increases selection intensity and helps in increasing prediction accuracy.

#### PHG Docker Pipeline

**Phase 1: Populate PHG Database**

FindAnchorRegions → LoadReferenceGenome → CreateHaplotypes → ProcessAssemblyHaplotype(optional) → CreateConsensusHaplotypes → PHG Database

**Phase 2: Genotyping Using PHG**

FindHaplotypePath → ExportPath → PHG Database

#### PHG Application Programming Interface (API)

```
ReferenceRange
  -> chromosome()
  -> start()
  -> end()
  -> name()

HaplotypeSequence
```