PHG Workflow (PHG 0.1.x):

**Create Default Directory**

**Executing using the image phg\_latest.sif**

1. Singularity shell phg\_latest.sig
2. /tassel-5-standalone/run\_pipeline.pl -debug -MakeDefaultDirectoryPlugin -workingDir /home/tkosfeld/PHG/phg/ -endPlugin
3. [Create a PHG database](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/MakeInitialPHGDBPipeline.md)

**You will need a valid intervals file for this step. (see step 4)**

**Note liquidbase directory stored in image is delicate in pathing here.**

singularity exec phg\_22.simg /tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters /home/tkosfeld/new\_phg/phg/initial\_config.txt -MakeInitialPHGDBPipelinePlugin -endPlugin Create bed file to define genome intervals

Get anchor file from b73 genomes: faidx -i bed genome.fa > out.bed

1. */tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters phg/config.txt -CreateValidIntervalsFilePlugin -intervalsFile phg/inputDir/reference/original\_intervals.bed -referenceFasta phg/inputDir/reference/Ref.fa -mergeOverlaps true -generatedFile phg/validBedFile.bed -endPlugin*

**Step 2 Populate. (Config.txt was updated here /home/tkosfeld/test\_phg/phg/gvcf\_config.txt)**

1. ~~/tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters /home/tkosfeld/PHG/phg/config.txt -PopulatePHGDBPipelinePlugin -endPlugin~~

**OLD PHG (did not need to run again with new PHG as we already had .gvcf files)**

singularity exec phg\_latest.sif /tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters /home/tkosfeld/PHG/phg/anchor\_config.txt -AssemblyMAFFromAnchorWavePlugin -endPlugin

singularity run phg\_latest.sif /tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters phg/maft\_to\_gvcf\_config.txt -MAFToGVCFPlugin -endPlugin

**NEW PHG**

Relocate and edit CreateHaplotypesFromGVCF.groovy to /home/tkosfeld/test\_phg/CreateHaplotypesFromGVCF.groovy to change hardcoded file pathing from “/phg/” to “phg/”.

singularity exec phg\_test.simg /home/tkosfeld/test\_phg/CreateHaplotypesFromGVCF.groovy -config /home/tkosfeld/test\_phg/phg/gvcf\_config.txt

singularity exec phg\_test.simg /tassel-5-standalone/run\_pipeline.pl -Xmx500G -debug -configParameters phg/imputation\_config.txt -HaplotypeGraphBuilderPlugin -configFile phg/imputation\_config.txt -methods assembly\_by\_anchorwave -includeVariantContexts true -endPlugin -RunHapConsensusPipelinePlugin -referenceFasta phg/inputDir/reference/Ref.fa -dbConfigFile phg/consensus\_config.txt -collapseMethod assembly\_by\_anchorwave -collapseMethodDetails assembly\_by\_anchorwave -rankingFile phg/ranking\_file.txt -mxDiv 0.0001 -clusteringMode kmer\_assembly -isTestMethod true -endPlugin

**Step 3. Imputation. (Config file: /home/tkosfeld/test\_phg**/**phg/imputation\_config.txt)**

1. singularity exec phg\_test.simg /tassel-5-standalone/run\_pipeline.pl -Xmx200G -debug -configParameters phg/imputation\_config.txt -ImputePipelinePlugin -imputeTarget pangenome -endPlugin > createPangenome\_output.txt
2. singularity exec phg\_test.simg /tassel-5-standalone/run\_pipeline.pl -Xmx200G -debug -configParameters phg/imputation\_config.txt -ImputePipelinePlugin -imputeTarget path -endPlugin
3. singularity exec phg\_test.simg /tassel-5-standalone/run\_pipeline.pl -Xmx200G -debug -configParameters phg/imputation\_config.txt -ImputePipelinePlugin -imputeTarget pathToVCF -endPlugin

**Common errors:**

* Jblob errors and heap errors have the same root problem. The amount of memory allocated to the heap needs to be increased. “-Xmx200G” specifies 200gb for the heap, increase if necessary. If you cannot increase the heap parameter, then reducing the number of allocated threads in a config file has the same effect.
* You’re getting an error that the sqlite db is locked (SQLITE\_BUSY).  Remove the hung transactions is to create a backup that doesn’t have lock, then copy the backup db to the original.  This should preserver the original, but remove the hung transactions.  
    
  Assume you have sqlite3 loaded, and assume you db is named phg\_db\_name.db, you would enter sqlite3 and create a backup by running these commands: (“>” is the system prompt)  
    
  > sqlite3 phg\_db\_name.db  
  sqlite> .backup main phgdb\_backup.db  
  sqlite> .exit  
    
  This would give you a file name “phgdb\_backup.db” without locks.  Then do the following to write the backup to the original db:  
  > mv phgdb\_backup.db phg\_db\_name.db
* Null errors are often caused by an inconsistency between the pathing parameters and the pangenome index parameters. Make delete and remake all pangenome data after changing parameters.
* Keyfiles are EXTREMLY finicky, make sure each keyfile matches the current working directory or database construction will be very flawed. Please keep in mind that these flaws might not manifest as errors during construction.

Output files to .csv:

.headers on

.mode csv

.output filename.extension

Select haplotypes\_id,gamete\_grp\_id,ref\_range\_id,genome\_file\_id,gvcf\_file\_id,asm\_contig,asm\_start\_coordinate,asm\_end\_coordinate,seq\_len,method\_id from haplotypes;

Select \* from gamete\_haplotypes;

Deprecated (PHG 0.0.2):

[*Step 0: Download PHG Docker*](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/CreatePHG_step0_main.md)

singularity pull docker://maizegenetics/phg

singularity pull docker://maizegenetics/phg\_liquibase

[*Step 1: create an initial PHG with reference data*](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/CreatePHG_step1-2_main.md)

1. [Create the default directory structure](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/MakeDefaultDirectory.md)

Make the following directories and files in WORKING\_DIR.

**${**WORKING\_DIR**}**/config.txt

**${**WORKING\_DIR**}**/load\_asm\_genome\_key\_file.txt

**${**WORKING\_DIR**}**/load\_wgs\_genome\_key\_file.txt

**${**WORKING\_DIR**}**/readMapping\_key\_file.txt

**${**WORKING\_DIR**}**/inputDir/

**${**WORKING\_DIR**}**/inputDir/assemblies/

**${**WORKING\_DIR**}**/inputDir/loadDB/

**${**WORKING\_DIR**}**/inputDir/loadDB/bam/

**${**WORKING\_DIR**}**/inputDir/loadDB/bam/temp/

**${**WORKING\_DIR**}**/inputDir/loadDB/bam/dedup/

**${**WORKING\_DIR**}**/inputDir/loadDB/bam/mapqFiltered/

**${**WORKING\_DIR**}**/inputDir/loadDB/fastq/

**${**WORKING\_DIR**}**/inputDir/loadDB/gvcf/

**${**WORKING\_DIR**}**/inputDir/loadDB/temp/

**${**WORKING\_DIR**}**/inputDir/reference/

**${**WORKING\_DIR**}**/inputDir/reference/load\_genome\_data.txt

**${**WORKING\_DIR**}**/outputDir/

**${**WORKING\_DIR**}**/outputDir/align/

**${**WORKING\_DIR**}**/outputDir/align/gvcfs/

**${**WORKING\_DIR**}**/tempFileDir/

**${**WORKING\_DIR**}**/README.txt

singularity exec phg\_latest.sif /tassel-5-standalone/run\_pipeline.pl -debug -MakeDefaultDirectoryPlugin -workingDir /home/tkosfeld/PHG/phg/ -endPlugin

1. [Create a PHG database](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/MakeInitialPHGDBPipeline.md)

**You will need a valid intervals file for this step. (see step C)**

**Config.txt was updated here.**

**Note liquidbase directory stored in image is delicate in pathing here.**

singularity exec phg\_latest.sif /tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters /home/tkosfeld/PHG/phg/config.txt -MakeInitialPHGDBPipelinePlugin -endPlugin

Fill out load\_genome\_data.txt

Edit config.txt to build directory.

Make Initial PHG DB Pipeline Plugin

singularity exec phg\_latest.sif \

/tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters phg/config.txt \

-CreateValidIntervalsFilePlugin -intervalsFile phg/inputDir/reference/original\_intervals.bed \

-referenceFasta phg/inputDir/reference/Ref.fa \

-mergeOverlaps true \

-generatedFile phg/validBedFile.bed -endPlugin

1. [Create bed file to define genome intervals](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/CreatePHG_step1_bedfile.md)

Get anchor file from b73 genomes:

faidx -i bed genome.fa > out.bed

singularity exec phg\_latest.sif \

/tassel-5-standalone/run\_pipeline.pl -Xmx100G -debug -configParameters phg/config.txt \

-CreateValidIntervalsFilePlugin -intervalsFile phg/inputDir/reference/original\_intervals.bed \

-referenceFasta phg/inputDir/reference/Ref.fa \

-mergeOverlaps true \

-generatedFile phg/validBedFile.bed -endPlugin

D. [Optionally, set up additional groups of PHG intervals](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/CreatePHG_step1_groupGenomeIntervals.md)

[*Step 2: add haplotypes*](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/CreatePHG_step1-2_main.md)

Step 2.5: Optional (see above) [*Update PHG database schema*](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/UpdatePHGSchema.md)

[*Step 3: impute variants or haplotypes*](https://bitbucket.org/bucklerlab/practicalhaplotypegraph/wiki/UserInstructions/ImputeWithPHG_main.md)