Post-Imputation Report

Patrick Monnahan

27 April, 2020

This report contains summary information of the process that was used to convert imputed data from the TOPMed Imputation Server into a PLINK-formatted dataset that is ready for association analysis or admixture inference. In brief, one or more sets of gzipped VCF files are first run through CrossMap, which converts coordinates from the GRCh38 reference genome to GRCh19. Then, these files are converted to PLINK format and variants are filtered for missingness, duplicates, and indels (are removed). For each chromosome, we then merge these resulting files across datasets. Only variants that have been retained across all datasets are included in this merged dataset. Rare alleles are then filtered from this merged dataset. The DAG representing this workflow is provided at the end of this document, although it may be difficult to view. Also, see the config.yml in the workflow directory for full list of parameter inputs and settings.

The following datasets were used as input:

Dataset	Directory
cog9906	/home/pmonnaha/shared/impute_prep/cog9906/
cog9904_9905	/home/pmonnaha/shared/impute_prep/cog9904_9905/
stjude	/home/pmonnaha/shared/impute_prep/stjude/
aall0232	/home/pmonnaha/shared/impute_prep/aall0232/
aric	/home/pmonnaha/shared/impute_prep/aric/

and the pipeline was carried out using the following singularity image:

```
## [1] "/home/pmonnaha/pmonnaha/AncestryInference.sif"
```

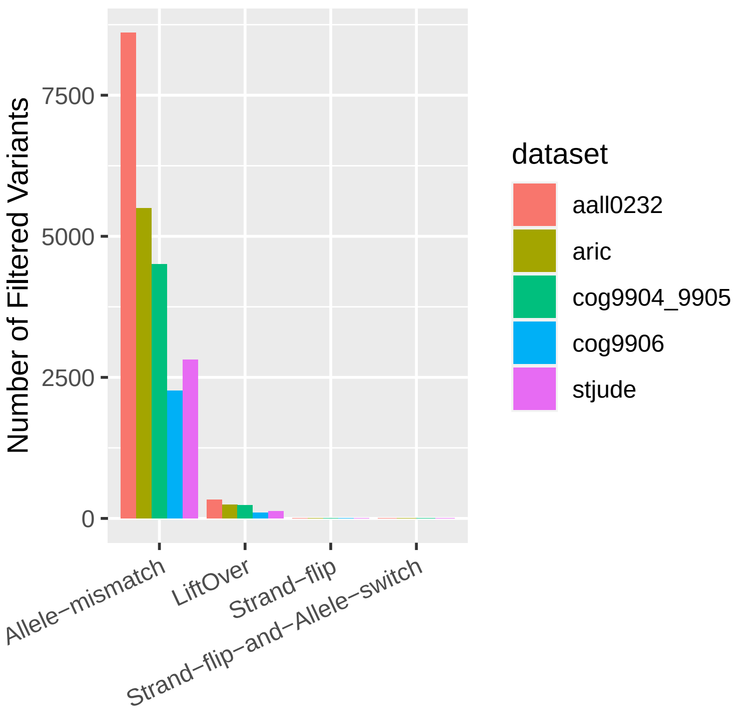
Imputation Summary

The TOPMed Imputation Server is based on the Michigan Imputation Server technology and thus implements a series of filters on the input datasets (see here for details) prior to imputation.

Input Filtering

Variant Exclusion

For each variant, the server will attempt to convert coordinates to hg38 (if necessary) via the LiftOver tool. If this is successful, it will determine if the variant matches a variant in the reference dataset along with whether the alleles in the query dataset match those in the reference. If a mismatch is found, the type of mismatch (flip, swap, or flip+swap) is determined and the variant is removed. The plot below summarizes the number of variants that were excluded subdivided by reason for exclusion. Note that these unstandardized totals will likely depend heavily on the total number of variants in the input query dataset.



Chunk Exclusion

The chromosomes are then divided into 20Mb chunks, and these chunks are excluded from imputation if: 1.) there are fewer than 3 SNPs, 2.) if <50% of sites in query dataset are found in reference dataset, 3.) any samples has a callrate <50%.

Below is the full list of excluded chunks along with the number of datasets that they were excluded from and the reasons for exclusion.

chunk	Low.SNP.Number	Low.Ref.Overlap	Bad.Sample	Num.DataSets
chunk_14_0000000001_0020000000	0	3	0	3
chunk_15_0000000001_0020000000	2	0	2	4
chunk_9_0040000001_0060000000	5	3	0	5

Experience has shown that there should only be a few excluded chunks, which tend to be relatively consistent across datasets (e.g. chromosome 9 and 14)

Number of excluded chunks in each dataset, further classified by the reason for which they were filtered. Note that a single chunk may have failed multiple filters.

dataset	Low.SNP.Number	Low.Ref.Overlap	Bad.Sample	Total
aall0232	1	2	0	2
aric	1	1	1	2
cog9904_9905	1	2	1	3
cog9906	2	1	0	3
stjude	2	0	0	2

The plot below shows the number of imputed variants after excluding the SNPs and chunks listed above. If few chunks were excluded, then these numbers should be very consistent across datasets. Note that numbers are slightly artificially inflated due to the coding of multiallelic variants. These are represented on multiple lines, one for each alternative allele. These multiallelic variants as well as the indels are removed, subsequently, and generally make up a small portion (5-6%) of the total number. Also, note that the majority of these imputed variants are likely fixed for one allele and will ultimately be removed (see 'Removing rare alleles' below)

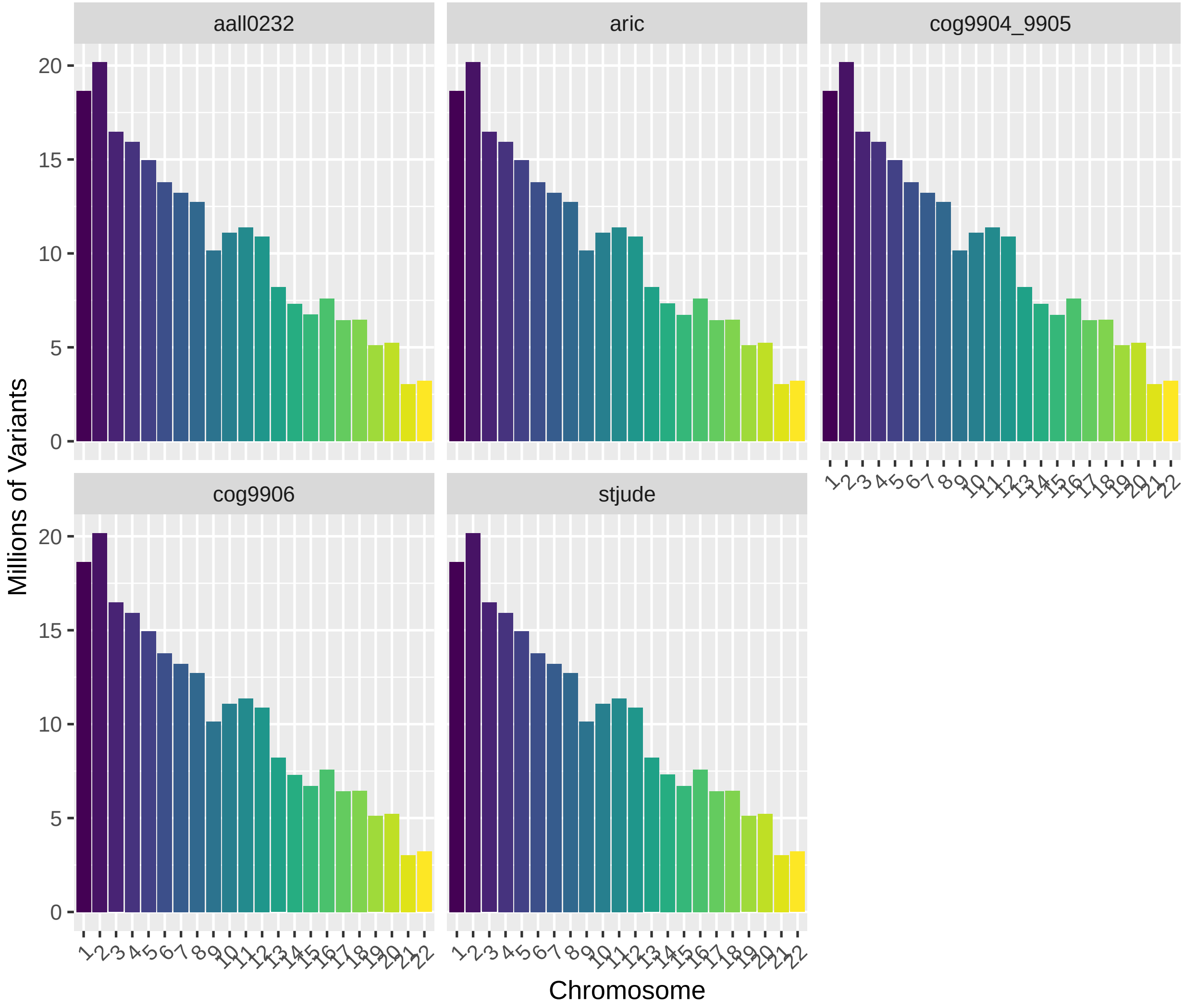


Figure S1: Total number of imputed variants (SNPs and Indels)

Coordinate Conversion

The program CrossMap was used to convert coordinates from GRCh38 to GRCh19. The reference fasta and 'chain' files (key linking coordinates across chromosomes) were taken from:

Reference Fasta

```
## [1] "/home/spectorl/pmonnaha/misc/hg19.fa"
```

Chain File

```
## [1] "/home/spectorl/pmonnaha/misc/GRCh38_to_GRCh37.flt.chain.gz"
```

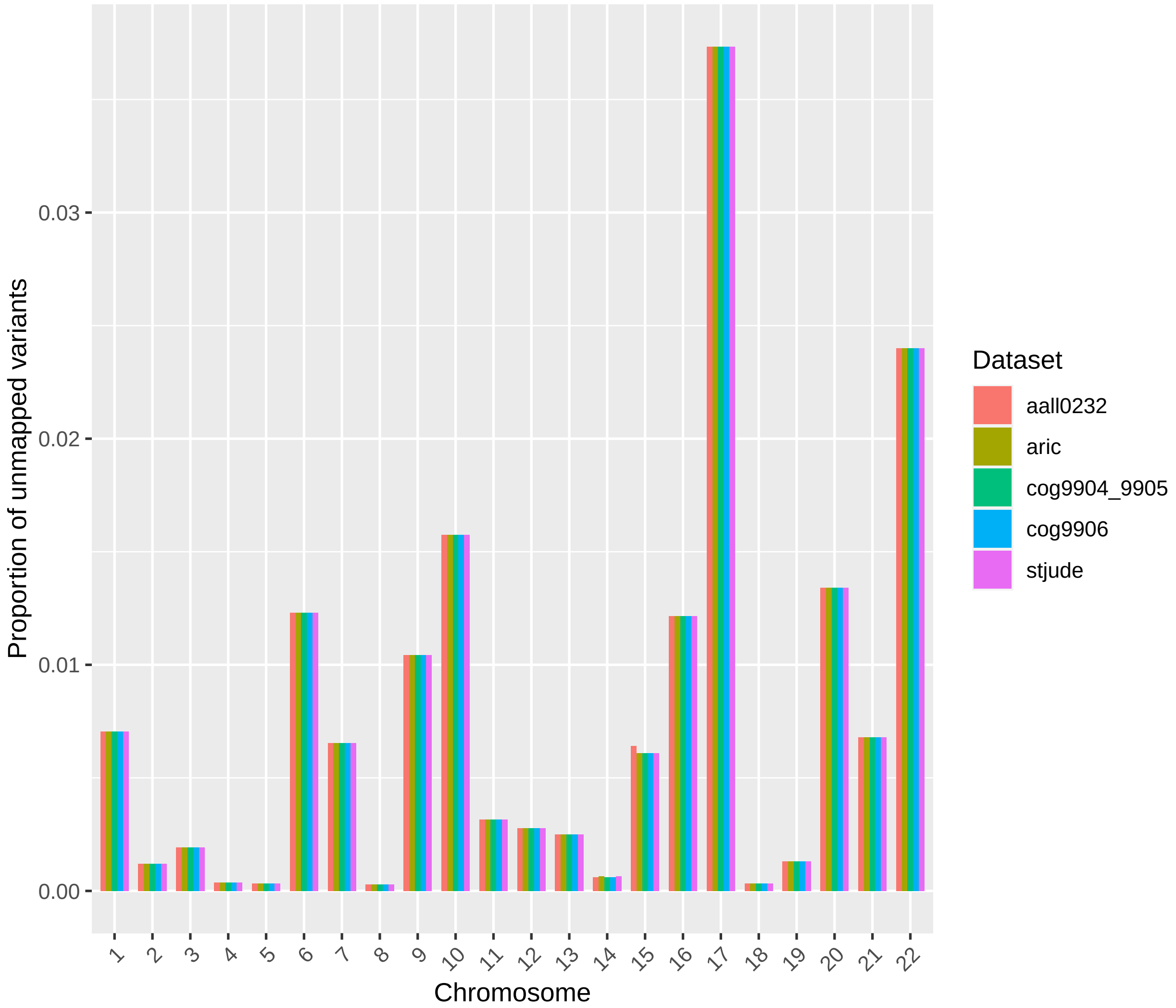


Figure S2: Proportion of variants whose coordinates were not successfully cross-mapped

The proportion here is calculated for each chromosome as: #unmapped / (#mapped + #unmapped). Unmapped variants were removed from subsequent steps.

PLINK Conversion and initial QC

Following coordinate conversion, the VCFs for each chromosome are converted to plink format. During this conversion, poorly imputed genotypes are filtered out. That is, if the probability of the most probable genotype falls below the following threshold, then the genotype for this sample is set to missing.

```
## [1] "0.85"
```

Thus, the missingness filter discussed below also filters for imputation 'quality'.

Variants are then filtered for missingness and duplicates and indels are removed. The threshold for maximum proportion of missing samples for a given variant is:

```
## [1] "0.05"
```

This missingness criterion is applied after first excluding samples that exceeded the following rate of missingness across variants:

```
## [1] "0.1"
```

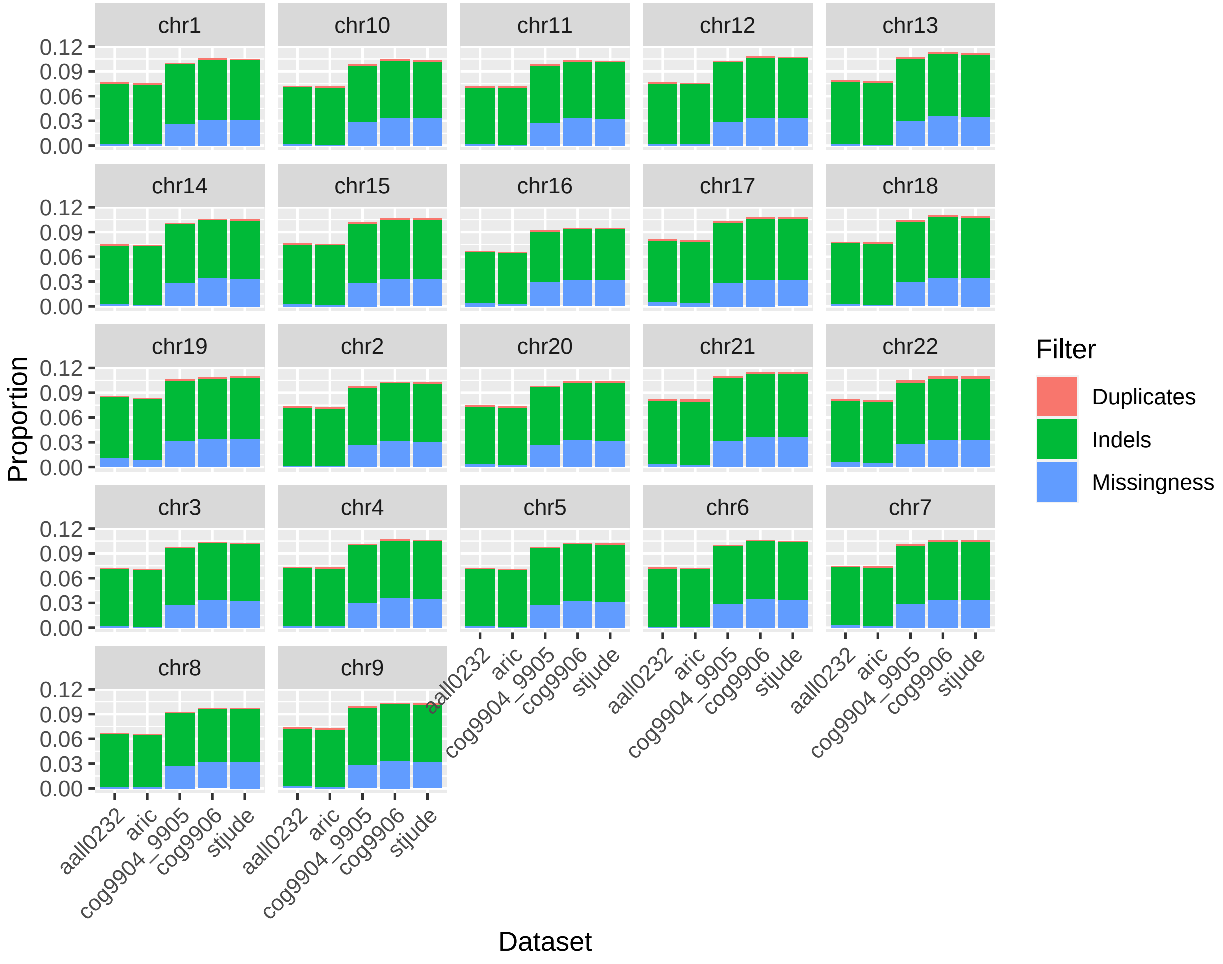


Figure S3: Proportion of total imputed sites removed by each filter.

Note: Indels includes multiallelic variants as well, which are the vast minority.

Filtering Merged Data

Overlap filtering

If multiple imputed datasets were provided as input, the next step would be, for each chromosome, to merge the genotypes across datasets. Importantly, only variants that are still present in all datasets (i.e. have not been filtered in any single dataset) will be retained. This way, if a variant imputed poorly in one dataset for whatever reason, it would be removed entirely from the merged dataset.

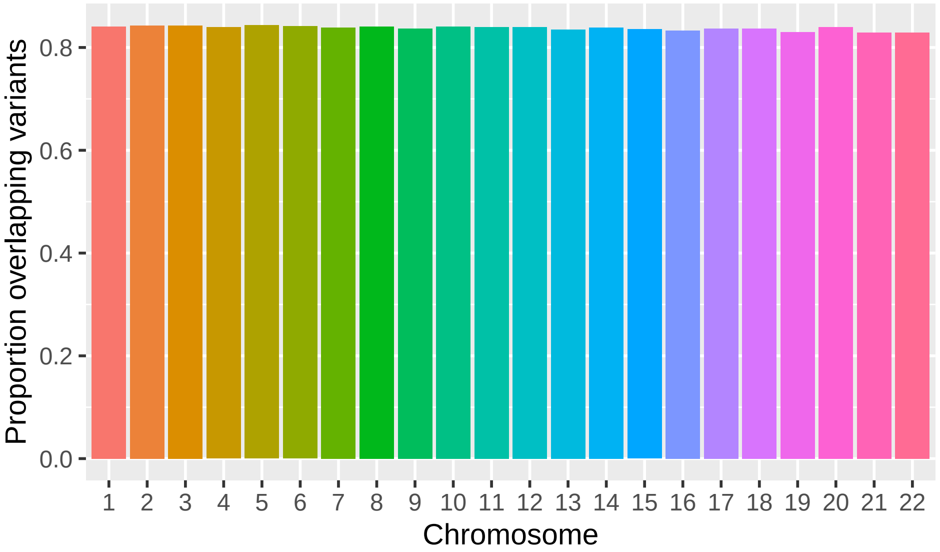


Figure S4: Proportion of variants found in all component datasets for each chromosome

Removing rare alleles

Removal of rare alleles is the final and likely most consequential step in the post-imputation QC pipeline. We wait to filter rare alleles until after merging in case there are fixed differences across datasets. Such variants, although rare in each individual dataset, may be intermediate in the merged dataset. The reason that this filter will likely remove the largest number of variants is due to the fact that the majority of imputed variants are, in fact, non-variant. The

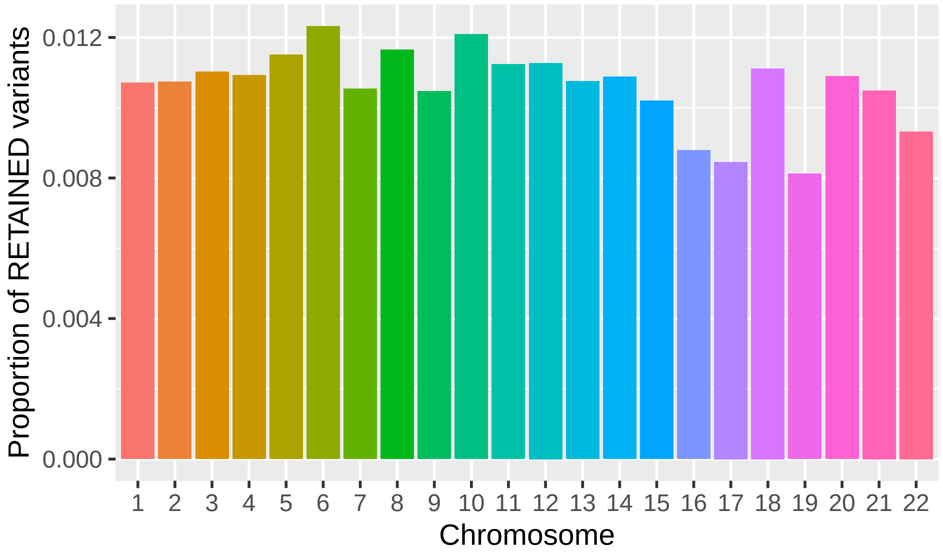


Figure S5: Proportion of variants that were RETAINED following removal of rare SNPs

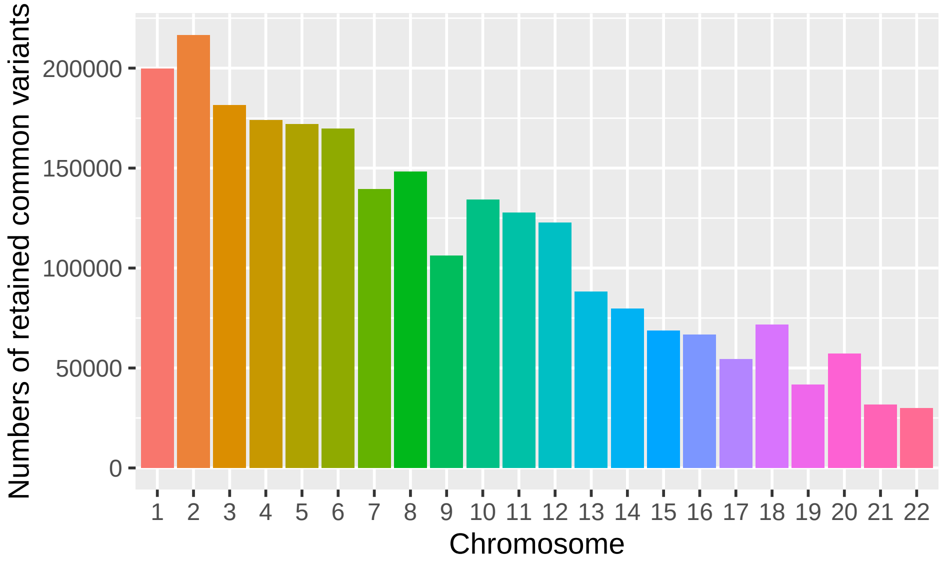


Figure S6: Total number of imputed variants remaining in the final QC'ed dataset.

Rule Graph

Below is a directed acyclic graph depicting the steps involved in this post-imputation QC pipeline. When possible, computation within each node was parallelized by dataset, chromosome, etc. The full DAG visualizing the parallel computing can be generated via:

```
snakemake --dag | dot -Tpng > jobgraph.png
```

from within the directory that the post-imputation QC was carried out. These are typically too large to fit easily in a pdf, and so were not included in this report.

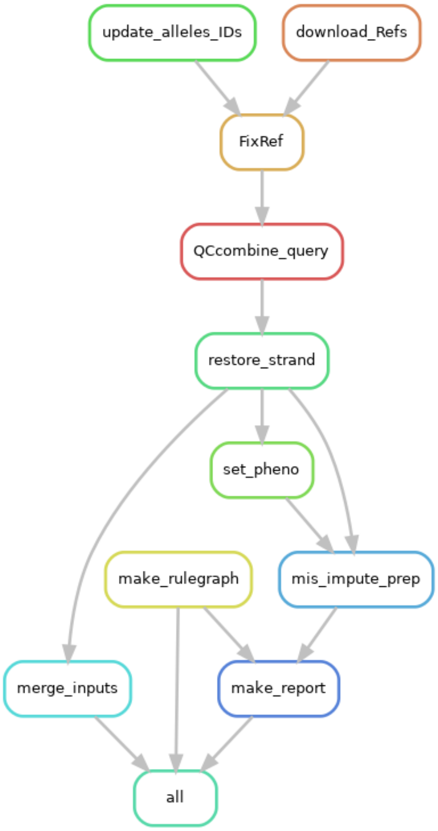


Figure S7: A rule graph showing the different steps of the bioinformatic analysis that is included in the Snakemake workflow.

Reproducibility

The code for reproducing this analysis is available here. The repo contains:

- A Snakemake workflow for running all steps.
- A collection of scripts to acheive individual steps
- A Singularity definitions file that can be used to generate the Singularity image used to run all steps. ** This image file is also directly available upon request

The code for reproducing this report is available here.

The input files for the figures produced herein are from:

```
## $chrom_file
## [1] "data/chromfilter.stats"
##
## $merge_file
## [1] "data/merge.stats"
##
## $chunk_file
## [1] "data/chunks_excluded.txt"
##
## $snp_file
## [1] "data/snps_excluded.txt"
##
## $rulegraph_file
## [1] "data/ewingsOG-rulegraph.png"
##
## $config_file
## [1] "data/config.yml"
```

Also, see the config.yml in the workflow directory for full list of parameter inputs and settings.

The results in this supplementary were generated in the following R environment:

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin18.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS/LAPACK: /usr/local/Cellar/openblas/0.3.7/lib/libopenblasp-r0.3.7.dylib
##
## locale:
## [1] en_US/en_US.UTF-8/en_US/C/en_US/en_US
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] magrittr_1.5   stringr_1.4.0  tidyr_1.0.2    dplyr_0.8.5    yaml_2.2.1
## [6] optparse_1.6.4 reshape2_1.4.3 ggplot2_3.3.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.3       highr_0.8        pillar_1.4.3     compiler_3.6.1
##  [5] plyr_1.8.6       tools_3.6.1      digest_0.6.25    viridisLite_0.3.0
##  [9] evaluate_0.14    lifecycle_0.2.0  tibble_2.1.3     gtable_0.3.0
## [13] png_0.1-7        pkgconfig_2.0.3  rlang_0.4.5      xfun_0.12
## [17] withr_2.1.2      knitr_1.28       vctrs_0.2.4      grid_3.6.1
## [21] tidyselect_1.0.0 getopt_1.20.3    glue_1.3.1       R6_2.4.1
## [25] rmarkdown_2.1    farver_2.0.3     purrr_0.3.3      scales_1.1.0
## [29] htmltools_0.4.0  ellipsis_0.3.0   assertthat_0.2.1 colorspace_1.4-1
## [33] labeling_0.3     stringi_1.4.6    munsell_0.5.0    crayon_1.3.4
```