DCC analysis pipeline

Stephanie Gogarten

July 20, 2018

DCC analysis pipeline

`https://github.com/UW-GAC/analysis_pipeline`

- `TopmedPipeline` R package
- R scripts for various analysis tasks
- Python scripts submit R scripts to a cluster or cloud environment
- `TopmedPipeline.py` defines cluster environments

Required software

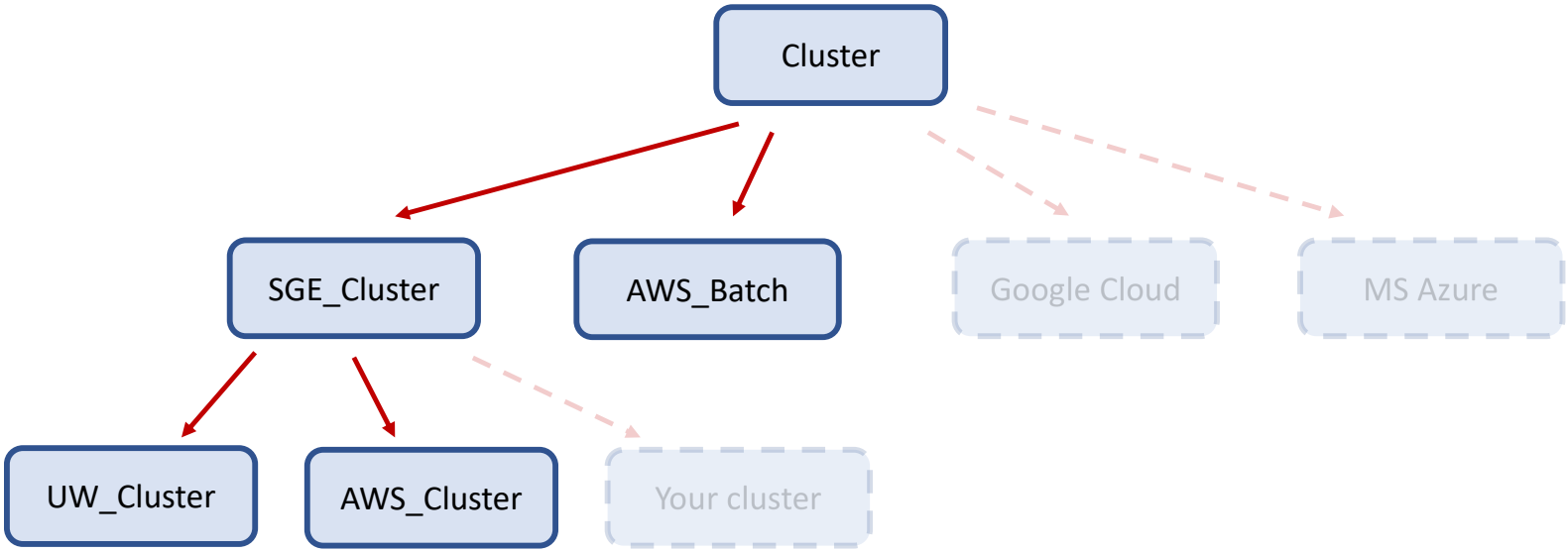
- R compiled with Intel MKL
- Bioconductor packages
  - SeqArray
  - SeqVarTools
  - SNPRelate
  - GENESIS
- CRAN packages
  - argparser (argument parsing for R scripts)
  - dplyr, tidyr (data frame manipulation)
  - ggplot2, GGally (plotting)
- Python 2.7

Docker images with software pre-installed

- `https://hub.docker.com/u/uwgac` (images)
- `https://github.com/UW-GAC/docker` (Dockerfiles to build images)

Cluster class definitions

- All Cluster objects have a `submitJob` method
- Cluster defaults set in JSON file
  - Users can create custom JSON files to override default parameters



Analysis configuration

- Every python script requires a configuration file (space-delimited plain text)
- Parameters include input and output file names, job-specific arguments
- Python scripts create intermediate config files to pass to each R script

Examples in `testdata` directory (e.g., `testdata/pcair.config`):

```
out_prefix "round1"
gds_file "testdata/1KG_phase3_subset.gds"
sample_include_file "testdata/sample_include.RData"
variant_include_file "testdata/variant_include_chr .RData"
ld_win_size 0.5
ld_threshold 0.2
king_file "data/test_ibd_king.RData"
kinship_method "king"
n_pcs 12
phenotype_file "testdata/1KG_phase3_subset_annot.RData"
group "Population"
```

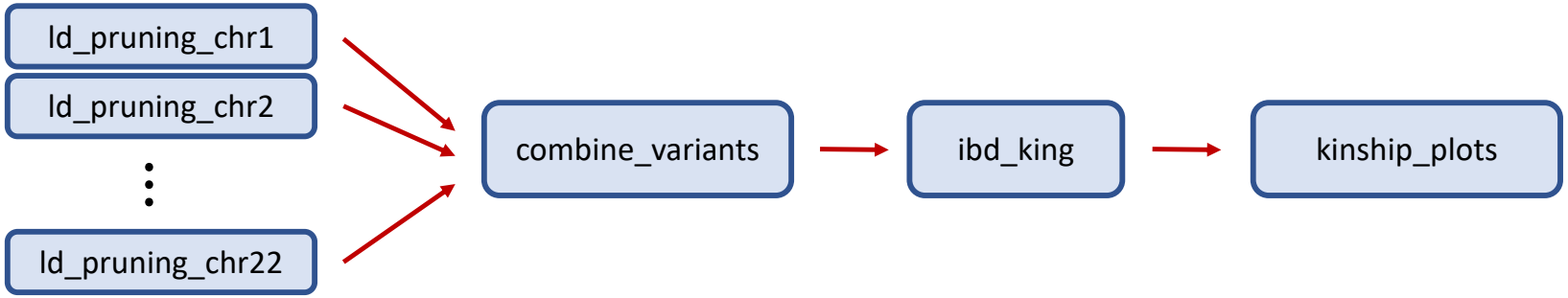
Parallelization

- By chromosome
- By segment
  - The genome is divided into segments based on length or number of requested segments
  - Default segment length is 10 Mb
  - Each chromosome spawns a job per segment
  - Segments are combined into one file per chromosome
- Multithreading
  - Some jobs allow mutithreading, where the user can request the job be divided among N cores

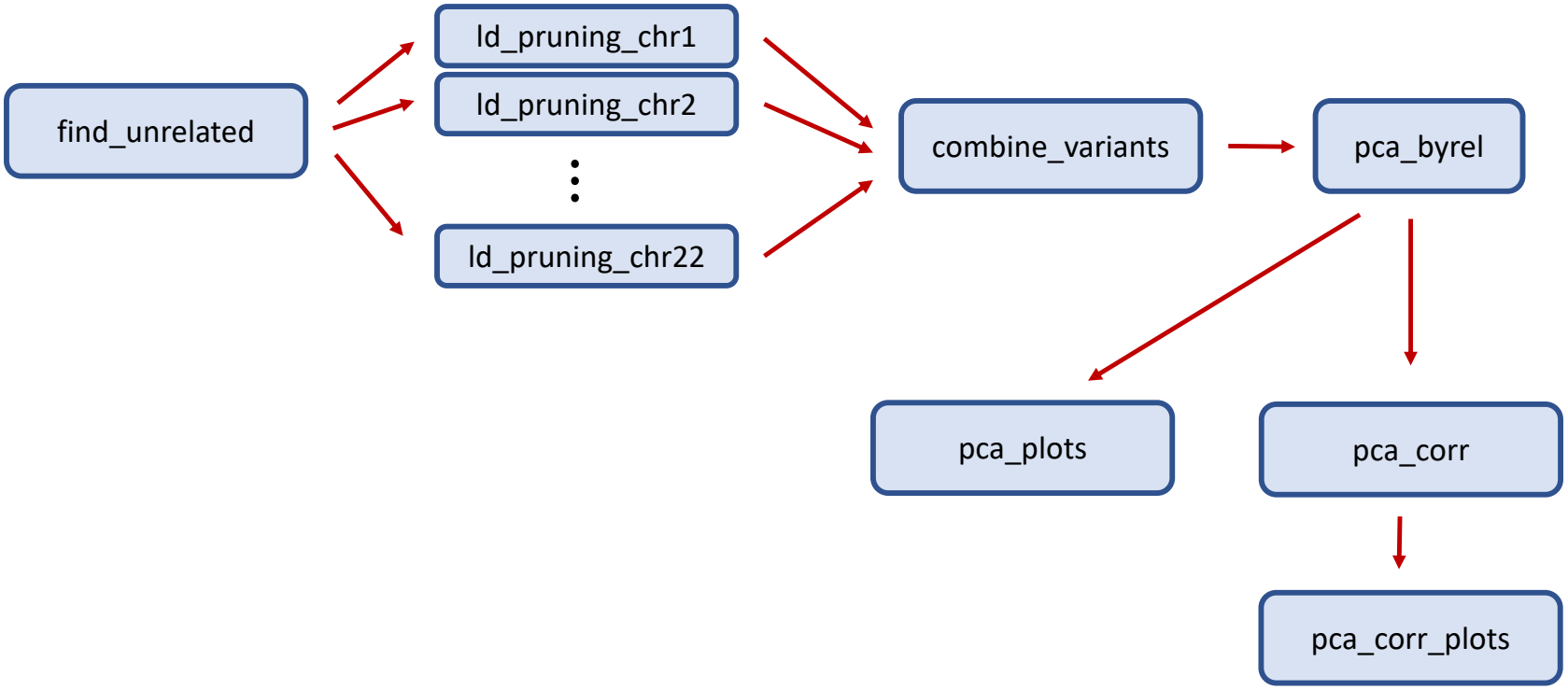
Available scripts

- Conversion to GDS
  - vcf2gds.py
- Relatedness and Population structure
  - grm.py
  - king.py
  - pcair.py
  - pcrelate.py
- Association tests
  - assoc.py
  - locuszoom.py

Flow chart: `king.py`



Flow chart: `pcair.py`



Flow chart: `assoc.py`