

Cloud Computing

Stephanie Gogarten

Not powerful enough for WGS data



- Memory
- CPU
- Disk space


Single server OK for smaller data sets



Large WGS data sets belong on a cluster



What is a Cluster?

- Hardware
 - Many computers (instances) each with
 - Multiple processors (cores)
 - Own shared memory
 - Shared file system
 - Network connectivity
- Software
 - Linux OS
 - Queuing system (SGE) 
 - Jobs execute independently
- Pros: Many cores and lots of memory
- Cons: Responsible for managing parallelism

* Standard distributed-memory

Where to get a cluster?

- Owning is expensive, so rent (Cloud)
- Pros
 - No/low infrastructure costs
 - Pay per use model
 - Scalable with increasing data set sizes
 - Variety of computers (RAM, CPU, disk, GPU)
 - Minimal management
 - Automatic software updates
 - Reliability and disaster recovery



Google Cloud



Where to get a cluster?

- Owning is expensive, so rent (Cloud)
 - Cons
 - Ongoing monthly costs
 - Pay for debug runs, failed runs, instances left running
 - You are your own IT person (or still need one)
 - Manage much of your own security
 - Extra effort to minimize costs
 - Cloud vendor lock-in
- } Unless you use a managed genomics platform

Where to get a cluster?

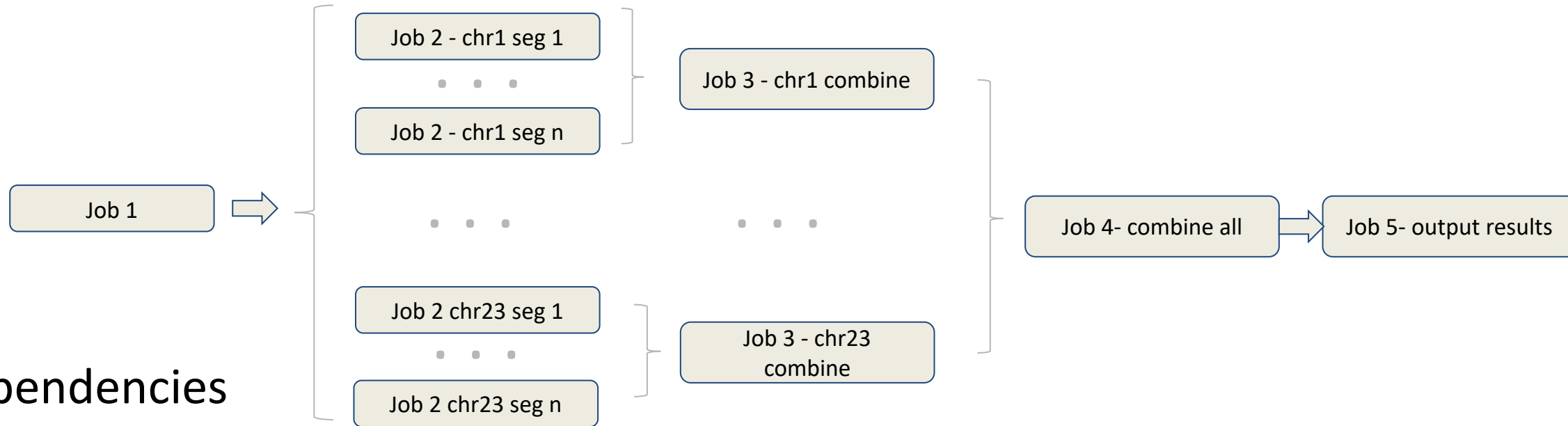
- Cloud-based genomics platforms
- Pro: ease of use
- Con: apps/workflows may be platform-specific



DNAneXus



Managing Pipeline Parallelism



- Dependencies
- Synchronization
- Heterogeneity
- Autoscaling
- Retry

Cloud environments
add cost complexity

Managing Pipeline Parallelism

- Explicit management (command line tools)
 - Python, JSON
 - AWS Batch
- Embedded in a genomics application (GUI)
 - Seven Bridges, DNAnexus, Galaxy, Terra
 - Mitigate complexity
 - Centralize data access

WGS major computational need

- Run one time
 - VCF to GDS file conversion
- Run a few times
 - Relatedness analysis
- **Run many times**
 - **Association testing**

What influences cloud costs

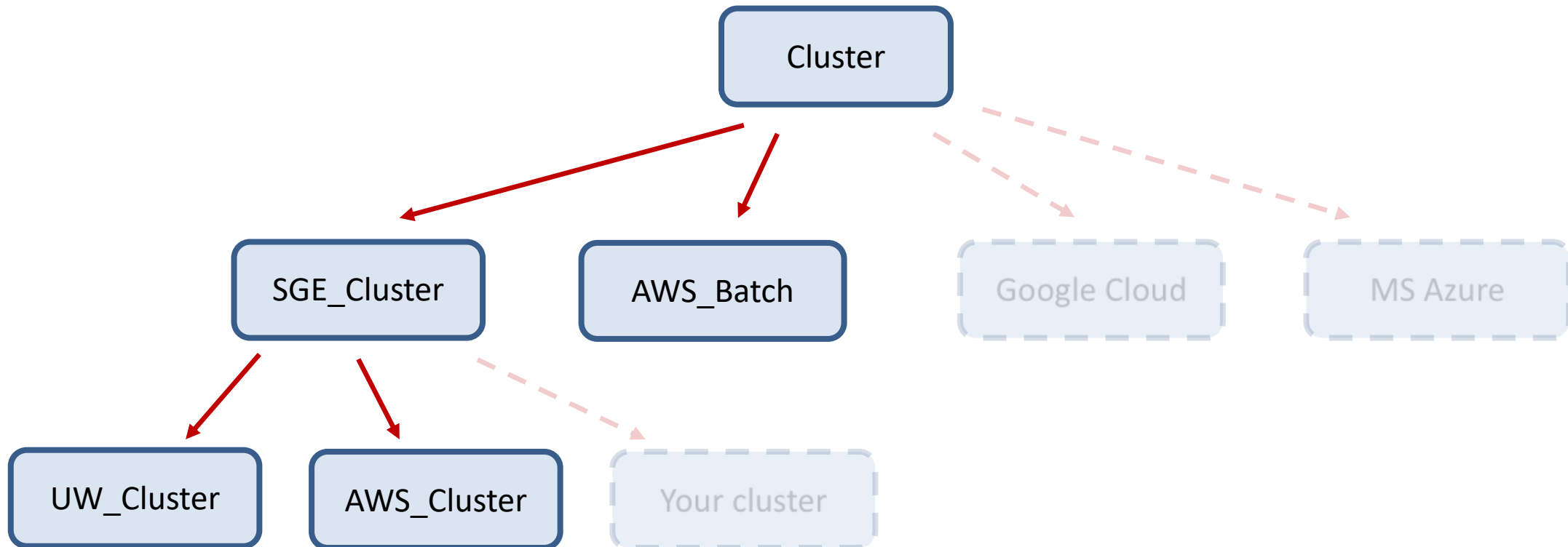
- Number of samples
- Number of variants & filtering
- Number of variants per aggregation unit
- Algorithm: Single variant, Aggregate
- Implementation: sparse matrices, fastSKAT
- Cloud hardware used (cores, RAM, disk)

UW-GAC 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

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



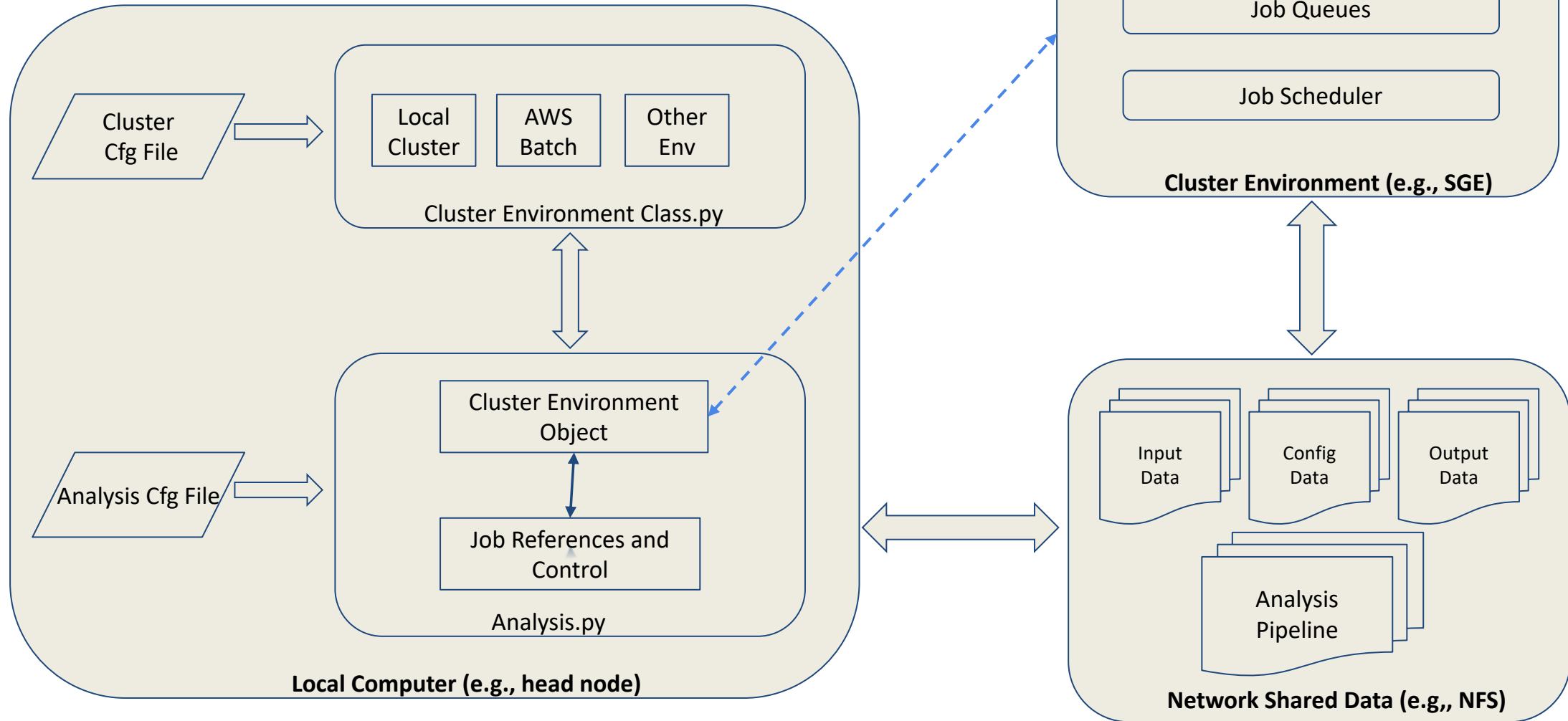
Analysis configuration

- 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/assoc_window_burden.config`):

```
out_prefix "test"
gds_file "testdata/1KG_phase3_subset_chr .gds"
phenotype_file "testdata/1KG_phase3_subset_annot.RData"
null_model_file "testdata/null_model.RData"
null_model_params "testdata/null_model.params"
variant_include_file "testdata/variant_include_chr .RData"
alt_freq_max "0.1"
test "burden"
test_type "score"
genome_build "hg19"
```

Analysis Pipeline

General Architecture



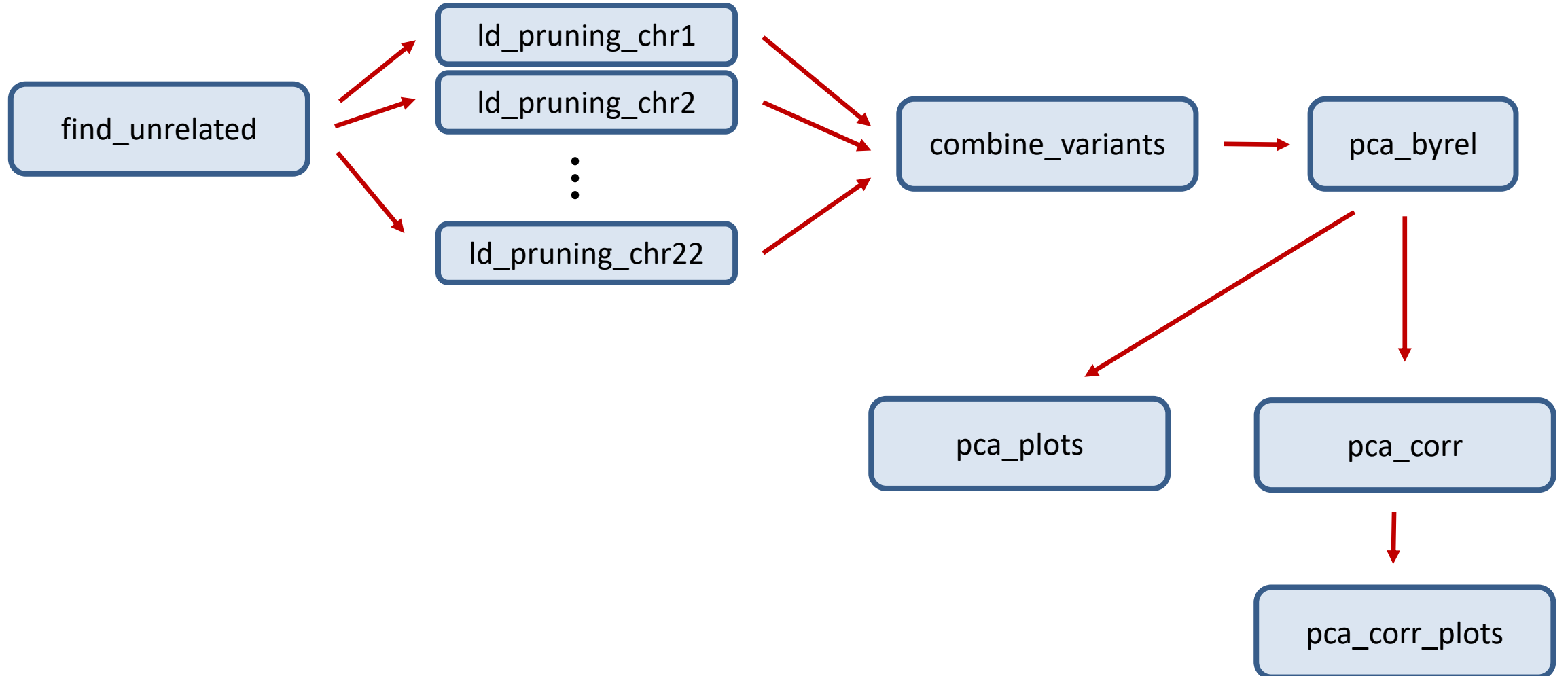
Parallelization

- 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 multithreading, where the user can request the job be divided among N cores

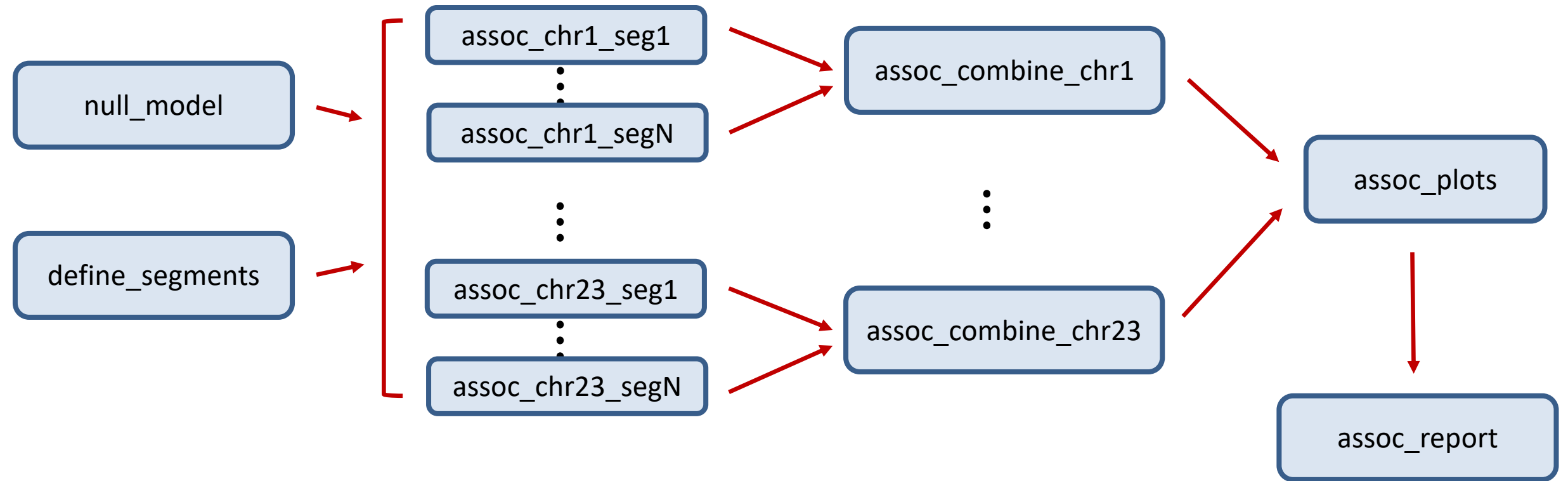
Available scripts

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

Flow chart: pcair.py



Flow chart: assoc.py



Managing software dependencies

- R compiled with [Intel MKL](#)
- Bioconductor packages
 - SeqArray
 - SeqVarTools
 - SNPRelate
 - GENESIS
- CRAN packages
 - argparse (argument parsing for R scripts)
 - dplyr, tidyr (data frame manipulation)
 - ggplot2, GGally (plotting)
- Python 2.7
- Command-line software
 - bcftools
 - plink
 - king



What is Docker?



- Platform for developing, deploying and running applications or systems
- *A Docker image is:*
 - built containing all software necessary to run the application
 - Usually built from a base image (e.g., *ubuntu*)
 - Includes all additional software to support an application or system (e.g., *gnu C/C++, python*)
 - Typically composed of multiple layers (e.g., *ubuntu layer, development tools layer, R layer*)
 - a read-only template used to create a *Docker container*

What is Docker?

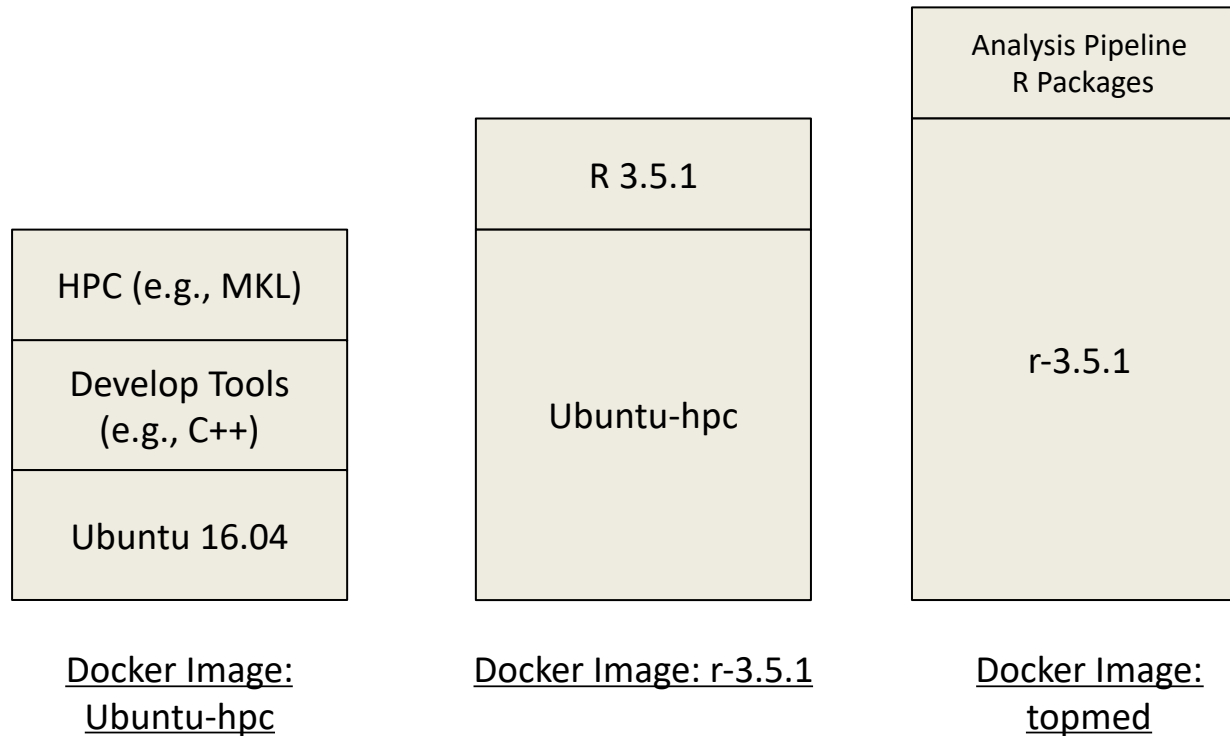
- *A Docker container is:*
 - a runnable instance of an image on a local or host computer (e.g., *Windows 10, macOS, Ubuntu*)
 - what the image becomes in memory when executed
 - runs natively on Linux
 - runs a Virtual Machine on *macOS* and *Windows*
 - the container is considered *stateless* - when the container stops all changes to code and data are discarded (except for data on local host that is mapped to the container)

What is Docker?

- What about accessing data on local host?
 - Data is typically not included in the *Docker image*
 - Data accessible on the local host can be mapped¹ (or *bind mounted*) to the *Docker container*
 - Any changes to data that is mapped to the local host is persisted when the *Docker container* stops

¹On macOS, file sharing is specified in the Docker Preferences

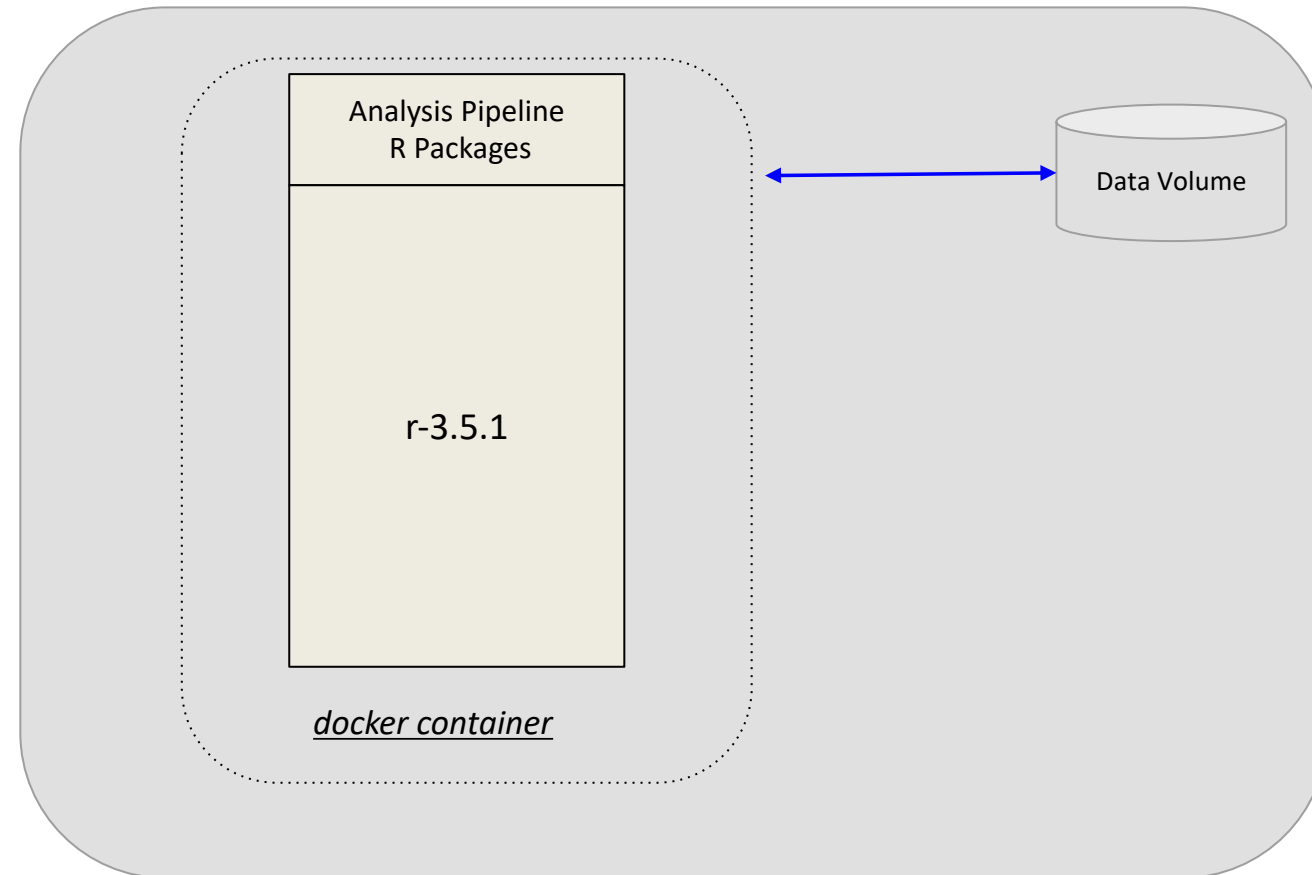
Docker images



<https://hub.docker.com/u/uwgac> (images)

<https://github.com/UW-GAC/docker> (Dockerfiles to build images)

Docker container



Linux, macOS or Windows
computer

Docker on the cloud

