Introduction to the H3ABioNet GWAS workflows

Scott Hazelhurst

H3ABioNet
Pan African Bioinformatics Network for H3Africa

1 Introduction

`http://www.bioinf.wits.ac.za/gwas/`

Genome Wide-Association Studies important application area – complex computing requirements

- hardware requirements – could probably manage with quad-core, 8GB RAM
  - e.g., 10k samples, 2m chip, individual steps would take 1-2 days
  - modest-size cluster *very* helpful
- Software requirements complex and heterogeneous

Need to build pipeline for GWAS

GWAS

- complex – several programs
- multiple steps, software dependencies
- have multiple parameters

Constraints of good scientific practice – **needs to be**

- re-rerun often to understand data
- reproducible by others
- portable

Workflow/Pipeline
Packaging of steps in a complex analysis

- user runs the package
- automate the steps
- not black-box – user needs to understand

Use appropriate software technology to support this

- Nextflow
- Containerisation (Docker/Singularity)

2 H3A GWAS Pipeline

H3A GWAS pipeline

`https://github.com/h3abionet/h3agwas`

Three mature workflows

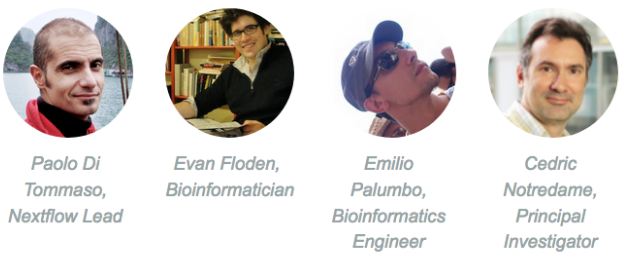
- `call2plink` Generating PLINK files from genotype report
- `qc` Quality control on PLINK data
- `assoc` Association testing

Reflects different phases of work

Work on `img2call` – workflow converting images to PLINK has started.

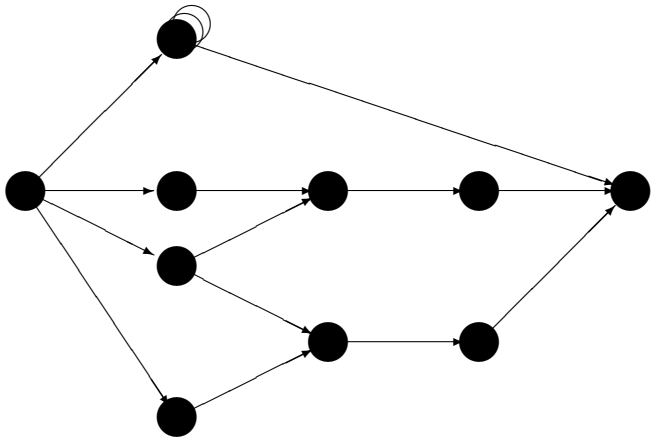
3 Nextflow

Nextflow developed at the Comparative Genomics Group at the Centre for Genomic Regulation in Barcelona



Nextflow is a language/system to coordinate individual steps of workflow

- Special purpose language with high level support for coordination of work
- Individual steps : written in more general purpose language, or call tools to be used



Features

- Detects dependencies, parallelism in workflow
- Schedules tasks when ready — maps to computational resources available
- Supports partial resumption
- Execute locally, on head node of cluster, cloud computing
- Supports Docker and Singularity

Installing Nextflow

Requires

- Java 8
- Nextflow

Instructions for installing Nextflow: Nextflow can be installed easily on Linux versions and on MacOS. The instructions here are for Ubuntu 18.04, but very similar instructions should work for other versions of Linux using the appropriate package manager (e.g. yum). Nextflow requires Java 8 – we'll use the standard version of Java that comes with Ubuntu – OpenJDK. But, if you prefer the Oracle version, you can download that manually and install that.
A video showing these instructions can be found in
`http://www.bioinf.wits.ac.za/gwas/videos/10-nextflow.mp4`

```
sudo apt-get install openjdk-8-jdk
curl -s https://get.nextflow.io | bash
sudo mv nextflow /usr/local/bin/
sudo a+rx /usr/local/bin/nextflow
sudo chmod  a+rx /usr/local/bin/nextflow
apt-get install python3
```

The only time I have had problem with these instructions is on systems which do not have curl installed on them. If you don't have *curl* and can't install it, then use *wget*, like this

```
wget --no-check-certificate  https://get.nextflow.io -O nextflow-install
bash nextflow-install
sudo mv nextflow /usr/local/bin
chmod a+rx /usr/local/bin/nextflow
```

4 Software dependencies

GWAS requires many different pieces of software

- Install them yourself

  Requires work but gives you flexibility

- Use Docker or Singularity containers

  Packages all dependencies for you

  Requires Docker or Singularity to be on your system.

4.1 Containerisation

Light-weight support for virtual machines

- a software *container* image is a package of an operating system, libraries, tools needed for an application

- can run *containers* from the image
  - each container has its own isolated set of resources
  - own file system
  - can run different OS to the host operating system
- We provide Docker images and Singularity support for our workflows
- You don't need to be an expert in Docker and/or Singularity – Nextflow manages it for you

Docker versus Singularity
Docker

- better known, better support
- not intended for multi-user computers – security issue – so probably won't find on your local university cluster
- Linux, recent MacOS, Windows 10 with MS Hyper-V

Singularity:

- Better security – can run on shared computer
- Linux or
- MacOS or Windows with extra requirements

If you've decided to use containerisation, make a choice of which you want to use. If you are installing on a multi-user system you should most likely use Singularity; if in your own machine Docker may be a better option as support is a bit better. **You will need to have root privileges to install – if you don't have root privileges and you can't persuade your administrator then you will have to install your own sources**

Installing Docker

To install on Ubuntu

```
sudo apt-get install docker.io
sudo systemctl enable docker
```

You also need to add yourself to Docker group (using my name as an example)

```
sudo usermod -a -G docker scott
```

And you will have to log out and log in the first time
To install on RHEL-like operating systems (e.g. RHEL, CentOS, Scientific Linux)

```
sudo yum install docker
sudo systemctl enable docker
sudo usermod -a -G docker scott
```

Installing Singularity

To install on RHEL/CentOS

```
yum install singularity-container
```

You probably need Singularity version 3

If you DON'T want to use Singularity or Docker then you need to install all the dependencies.

To be clear if you are using Docker or Singularity you can skip the rest of this section

Installing dependencies

1. PLINK1.9
2. Install Python pip

   ```
   apt install python3-pip
   ```

3. Install texlive:

   ```
   apt install texlive-full
   ```

4. Install Python libraries

   ```
   sudo pip3 install matplotlib pandas scipy numpy openpyxl
   ```

5. GEMMA, BOLTLMM? download it.

5 Quick-start example

There's a video – but I'll take you through it.

`http://www.bioinf.wits.ac.za/gwas/12-install-h3agwas-docker-quickstart`

There are two ways of managing the workflow software

- Nextflow itself does it
- Use git

In the Quickstart example, we use the first. Detail give later.

First, fetch the workflow – it gets saved in a hidden directory but you don't need to see it

```
nextflow pull h3abionet/h3agwas
```

You will need to do this when the workflow is updated.

Docker install if not done

If you haven't done it, install Docker (or singularity) – use your userid, not mine!

```
sudo apt-get install docker.io
sudo systemctl enable docker
sudo usermod -a -G docker scott
```

Log out and log in (group membership only updated when you log in)

Get sample data

Create a directory to work in and change directory to it. Then fetch the sample data

```
wget http://www.bioinf.wits.ac.za/gwas/sample.zip
unzip sample.zip
```

Run the QC workflow

By default, the *qc* workflow looks in the *sample* directory for data.

```
nextflow run h3abionet/h3agwas/qc/main.nf
```

By default, output goes to the *output* directory:

- Report: *out.pdf*
- PLINK files: *out.bed*, *out.bim*, *out.fam*

6 Installing

Options for installing the workflow

1. Use Nextflow to manage – will download all the code for you
2. Use git to manage
   More advanced, needed if you want to modify the workflow

6.1 Using Nextflow to manage

Using Nextflow to manage

```
nextflow pull h3abionet/h3agwas
```

Running Nextflow
If you use Nextflow to manage the workflows you'll run Nextflow as follows

- `nextflow run h3abionet/h3agwas/topbottom.nf .....`
- `nextflow run h3abionet/h3agwas/plink-qc.nf .....`
- `nextflow run h3abionet/h3agwas/plink-gwas.nf .....`

Updating the workflow
Each time we update Nextflow, the next time you run the script you'll get a message like

> *NOTE: Your local project version looks outdated - a different revision is available in the remote repository*

Update by saying: `nextflow pull h3abionet/h3agwas`

Using Git

```
git clone https://github.com/h3abionet/h3agwas
```

This creates a folder called *h3agwas* with scripts inside it. The directory structure is as follows

- The Nextflow programs for the workflows
  - topbottom.nf
  - plink-qc.nf
  - plink-assoc.nf
- nextflow.config : the default nextflow configuration file
- bin, templates: directories with code that the workflows need.
- aux: scripts that are not strictly part of the workflow but you might find it useful.

Running Nextflow under Git management

Assuming you are in the *h3agwas* directory

- `nextflow run call2plink ....`
- `nextflow run qc  ....`
- `nextflow run assoc ....`

If not in *h3agwas* directory give the full path to where the workflow can be found

Updating the workflow under Git

To get any updates we release you'll need to go the main directory you downloaded and say:

git pull

You are, of course, welcome to update any of files as you see fit. If your updates clash with ours, when you do the *git pull* you'll have to reconcile the versions using the normal rules of Git – this is outside the scope of this thesis.

6.2 Configuration files

Nextflow runs are controlled by configuration files -

- Can have several
- Recommended – use two
  - default nextflow.config file
  - a smaller config file that redefines just those things that you need

Default config files can be found in each sub-workflow, e.g., `https://github.com/h3abionet/h3agwas/blob/master/qc/nextflow.config`

```
nextflow run h3abionet/h3agwas/qc/main.nf -c b.config
```

Will use :

- default nextflow.config file plus
- the specified config file (over-rides anything in the default)

7 Running the qc.nf workflow

- Remove duplicate SNPs
- Remove SNPs, individuals with high missingness, HWE, MAF
- Remove outliers on sample heterozygosity
- Remove relatedness
- Tests differential missingness
- Produce reports

Config file : Main components

- Input directory and file
- Output directory and file
- Batch analysis: strongly recommended
  Case-control: binary – compulsory
  By phenotype: e.g., *site*, strongly recommended
- QC cut-offs

Need phenotype file(s) with headers.

Running a QC

Controlled by a config file

```
params.input_dir  = "sample"
params.input_pat  = "sampleA"
params.output     = "test-qc"
params.output_dir = "output"
params.case_control = "sample/sample.phe"
params.case_control_col = "PHE"
params.batch = "sample/sample-batch-site.phe"
params.batch_col = "batch"
params.phenotype = "sample/sample-batch-site.phe"
params.pheno_col = "site"
params.sexinfo_available = true
params.pi_hat = 0.18
params.cut_maf= 0.05
```

Another way of writing this would be

```
params {

   case_control = "sample/sample.phe"
   case_control_col = "PHE"
   batch = "sample/sample-batch-site.phe"
   batch_col = "batch"
   phenotype = "sample/sample-batch-site.phe"
   pheno_col = "site"
   sexinfo_available = true
   pi_hat = 0.18
   cut_maf= 0.05


}
```

Batch analysis is intended for cases where the DNA from the experiment is collected, shipped or genotyped separately. In QC you need to be sure that there aren't significant differences between the different batches.

Phenotype analysis allows you to do more detailed batch analysis. Depending on the phenotype chosen, there may be overall genotype difference or not. For example, in the AWI-Gen study, we chose collection *site* as the phenotype of interest. Here we expect there to be very significant overall genotype differences because our sites are in very different parts of Africa. The point of doing phenotype analysis for us is to ensure that there are no (a) batch effects at any sites, and (b) to explore missingness and QC at the site level. You could also choose *sex* as the phenotype, in which case you would not see overall genotype difference.

The *case-control* column in compulsory – you need to choose a phenotype that is binary. We do not expect the genotypes overall to be different between cases and controls – and if there is an overall difference there's likely a QC problem. If you don't have a suitable binary case-control make up one – even randomly assign participants to different groups.

For each of the above, you need to specify the file name where the data can be found and the column (the first line of the file must have headers). Usually, it will be the same file and different column names, but you may have multiple files

Running the sample

```
nextflow run h3abionet/h3agwas -c sc.config qc/main.nf
```

This assumes that all the tools have been installed on our system. The video can be found at `http://www.bioinf.wits.ac.za/gwas/videos/20-run.mp4`.

```
N E X T F L O W  ~  version 0.30.1
Launching `../plink-qc.nf` [pedantic_kare] - revision: ac5217ecef
Sexinfo available command
[warm up] executor > local
[fe/c0582e] Submitted process > inMD5 (1)
[d4/1e3bbb] Submitted process > getDuplicateMarkers (1)
[f4/ed17d8] Submitted process > removeDuplicateSNPs (1)
[ad/6d480c] Submitted process > getInitMAF (1)
[8b/40df13] Submitted process > getX (1)
[57/ef097e] Submitted process > identifyIndivDiscSexinfo (1)
[91/59ea3c] Submitted process > generateIndivMissingnessPlot (1)
[ab/e1643d] Submitted process > generateSnpMissingnessPlot (1)
[d8/9d32e8] Submitted process > removeQCPhase1 (1)
..
..
[e5/2817fb] Submitted process > generateMafPlot (1)
[d4/e6d01c] Submitted process > produceReports (1)
The output report is called output/test-qc.pdf
```

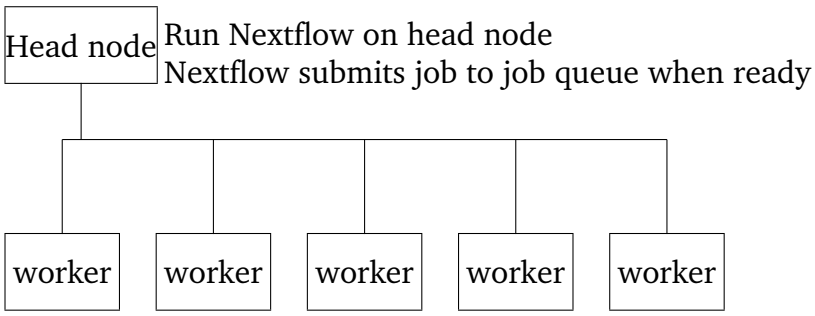
Running with Singularity

If tools have not all been installed on the system, run with Docker

```
nextflow run h3abionet/h3agwas -c sc.config qc/main.nf\
              -profile singularity
```

The first time you do this, there may be very long delays as the Docker images are fetched from their repositories.

Running on a cluster



Running with SLURM

Run on the head node of the cluster

```
nextflow run h3abionet/h3agwas -c sc.config qc/main.nf\
              -profile slurm


nextflow run h3abionet/h3agwas -c sc.config  qc/main.nf\
              -profile slurm,singularity
```

Running the workflow

To run the workflow you'll need to specify the following:–

- the name of the script to run
- The configuration file
- the mode of running

You will either use *one* of the two lines below depending on whether you have installed all the software.

```
# if you have installed Docker
nextflow run h3abionet/h3agwas/qc/main.nf -c example.config   -profile docker

# if you have installed all the software
nextflow run h3abionet/h3agwas/qc/main.nf -c ga14.config
```

Association study

plink-assoc.nf
Association workflow very experiment dependant

- data, population structure, co-variate, question
- basic workflow implemented for initial study
- can be extended

Config file

This is a template

Example config file

```
params {
   input_dir = "/data/scott/assoc/agt"
   input_pat = "t25"
   output  =  "allgemma"
   output_dir = "assocresults"
   data = "/data/scott/assoc/data.csv"

   covariates = "age,sex"
   pheno="bmi_c/np.log,wst_hip_r_c,standing_height_mm"
   gemma_num_cores = 8
   gemma           = 1
   linear          = 1
}
```

```
nextflow -c assoc.config config h3abionet/h3agwas/plink-assoc.nf
```

Running the workflow

```
nextflow run h3abionet/h3agwas/assoc/main.nf \
         -c assoc.config
```

The output will be found in the *assocresults* configuration file

- because that's specified in the *assoc.config* file

Running under SLURM/Singularity

```
nextflow run h3abionet/h3agwas/assoc/main.nf -c assoc.config \
   -profile slurm,singularity
```

Features

PLINK

- χ^2, Fisher, linear & logistic regression
- adjusted for multiple testing and permutation testing
- covariates

GEMMA

- with/without covariates

- need to take care of missing data

Sampling/testing

- thin
- chrom

Coming soon – on “awigen” branch

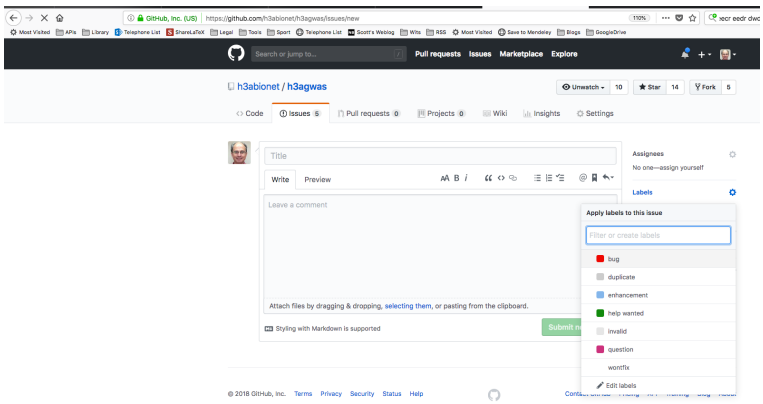
- BoltLMM, FastLMM
- Gene/Environment interaction

Asking for help

- H3ABioNet Help desk

  `https://www.h3abionet.org/support`
- On GitHub – need a GitHub account

  `https://github.com/h3abionet/h3agwas/issues`



Acknowledgements

Funded by NIH NHGRI grants U41HG006941, HG006938. Work at different institutions and individuals.

Eugene de Beste Lerato Magosi Phelelani Mpangase Rob Clucas Jean-Tristan Brandenberg Harry Noyce Ayton Meintjes Don Armstrong Fourie Joubert Gerrit Botha Sumir Panji Nicky