GINGER On-site Training
Day 2: Introduction to Plink

GINGER Program 2022

University of Cape Town

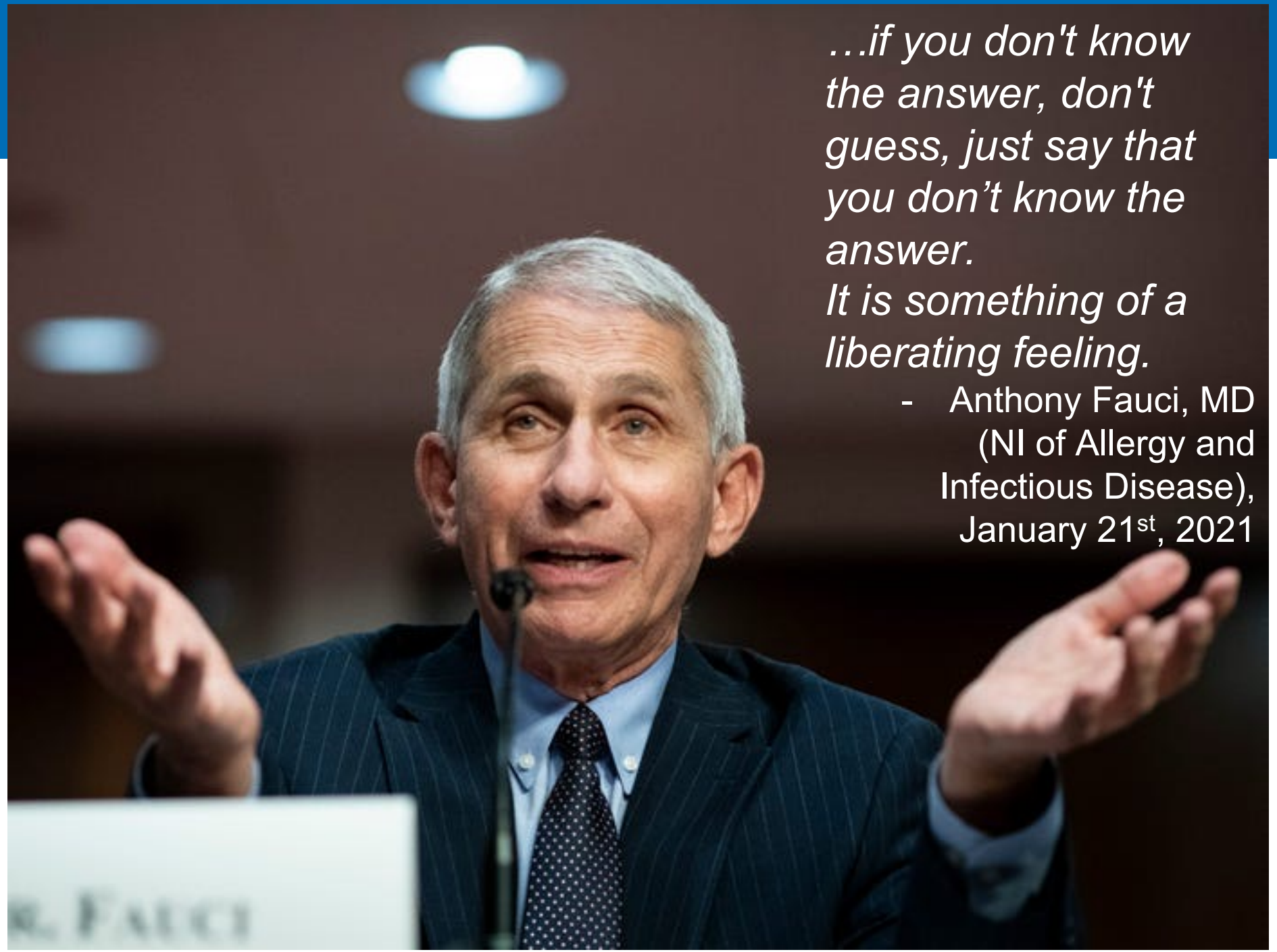
Teaching Fellows:

Kumar Veerapen, Ph.D.
Senior Expert I Data Science
Kumar.Veerapen@Novartis.com

Carla Marquez-Luna, Ph.D.
Postdoctoral Research Fellow
carlamarquezluna@gmail.com

	Monday, April 4	Tuesday, April 5	Wednesday, April 6	Thursday, April 7	Friday, April 8
9:00-10:30	**Training Welcome and Introduction** **9:00-9:30 - Training Overview** **9:30-10:00 - Professor Dan Stein Welcome** **10:00-11:00 - Begin Kampala Refresher**	**Plink Tutorial**	**Excursion to Robben Island (weather dependent)**	**9:00-10:00 am TBD** **10:00 am NeuroGAP Site Visit**	**Step-by-Step GWAS**
10:30-10:45	Tea Break	Tea Break			Tea Break
10:45-1:00	**Kampala Refresher continued**	**11:00-12:00 - Intro to Plink** **12:00-1:00 - Professor Collet Dandara**			**11:00 - Guest Lectures: Drs. Shareefa Dalvie and Nastassja Koen** **12:00-1:00 - Step-by-Step GWAS**
1:00-2:00	Lunch	Lunch	Lunch at the V&A Waterfront (weather dependent)	Lunch	Lunch
2:00-3:30	**Intro to UNIX** Fundamental Commands Genetic Data Formats and Conversion	**Plink Tutorial**	**Group Project Work**	**Step-by-Step GWAS**	**Step-by-Step GWAS**
3:30-3:45	Tea Break	Tea Break		Tea Break	Tea Break
3:45-5:00	**GINGER group projects intro**	**Plink Tutorial** **Step-by-Step GWAS**		**Step-by-Step GWAS**	**4:00-4:30 - Step by Step GWAS** **4:30-5:00 - Group Project Presentations**

GWAS all the time*


…if you don't know the answer, don't guess, just say that you don’t know the answer.
It is something of a liberating feeling.
- Anthony Fauci, MD (NI of Allergy and Infectious Disease), January 21st, 2021

Word of Advice

How do you code?

- There are many ways to code.
- Beginners, don't worry about how short your code is. As long as IT WORKS!
- Later, get someone who is more experienced to do code review with you.

Outline

- **Module 3.1.1:**

  PLINK Overview

- **Module 3.1.2:**

  Data Manipulation and Conversion

BROAD INSTITUTE

Module 3.1.1: PLINK Overview

Learning objectives:

- Navigating PLINK usage.
- Defining basic notation and commands.

Adapted from Bärbel Maus and Elizabeth Atkinson

Why use PLINK?

- A software to analyse phenotype/genotype data
- It is run from the command line
- The most common tool to analyse genome-wide genotyping data
- Free and open source
- Designed to perform a wide range of basic, (fairly) largescale analyses in (mostly) computationally efficient manner
- Can be used on several platforms

PLINK Citations

REPORT

PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses

Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel A. R. Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul I. W. de Bakker, Mark J. Daly, and Pak C. Sham

Whole-genome association studies (WGAS) bring new computational, as well as analytic, challenges to researchers. Many existing genetic-analysis tools are not designed to handle such large data sets in a convenient manner and do not necessarily exploit the new opportunities that whole-genome data bring. To address these issues, we developed PLINK, an open-source C/C++ WGAS tool set. With PLINK, large data sets comprising hundreds of thousands of markers genotyped for thousands of individuals can be rapidly manipulated and analyzed in their entirety. As well as providing tools to make the basic analytic steps computationally efficient, PLINK also supports some novel approaches to whole-genome data that take advantage of whole-genome coverage. We introduce PLINK and describe the five main domains of function: data management, summary statistics, population stratification, association analysis, and identity-by-descent estimation. In particular, we focus on the estimation and use of identity-by-state and identity-by-descent information in the context of population-based whole-genome studies. This information can be used to detect and correct for population stratification and to identify extended chromosomal segments that are shared identical by descent between very distantly related individuals. Analysis of the patterns of segmental sharing has the potential to map disease loci that contain multiple rare variants in a population-based linkage analysis.

Purcell et al. 2007 AJHG, 18,882 citations*

Chang *et al. GigaScience* (2015) 4:7
DOI 10.1186/s13742-015-0047-8

(GIGA)^n SCIENCE

TECHNICAL NOTE Open Access

Second-generation PLINK: rising to the challenge of larger and richer datasets

Christopher C Chang[1,2*], Carson C Chow[3], Laurent CAM Tellier[2,4], Shashaank Vattikuti[3], Shaun M Purcell[5,6,7,8] and James J Lee[3,9]

Abstract

Background: PLINK 1 is a widely used open-source C/C++ toolset for genome-wide association studies (GWAS) and research in population genetics. However, the steady accumulation of data from imputation and whole-genome sequencing studies has exposed a strong need for faster and scalable implementations of key functions, such as logistic regression, linkage disequilibrium estimation, and genomic distance evaluation. In addition, GWAS and population-genetic data now frequently contain genotype likelihoods, phase information, and/or multiallelic variants, none of which can be represented by PLINK 1's primary data format.

Findings: To address these issues, we are developing a second-generation codebase for PLINK. The first major release from this codebase, PLINK 1.9, introduces extensive use of bit-level parallelism, $O(\sqrt{n})$-time/constant-space Hardy-Weinberg equilibrium and Fisher's exact tests, and many other algorithmic improvements. In combination, these changes accelerate most operations by 1-4 orders of magnitude, and allow the program to handle datasets too large to fit in RAM. We have also developed an extension to the data format which adds low-overhead support for genotype likelihoods, phase, multiallelic variants, and reference vs. alternate alleles, which is the basis of our planned second release (PLINK 2.0).

Conclusions: The second-generation versions of PLINK will offer dramatic improvements in performance and compatibility. For the first time, users without access to high-end computing resources can perform several essential analyses of the feature-rich and very large genetic datasets coming into use.

Keywords: GWAS, Population genetics, Whole-genome sequencing, High-density SNP genotyping, Computational statistics

Chang et al. 2015 *GigaScience*, 2,312 citations*

*according to 1/23/20 google scholar!

What is PLINK?

- PLINK has numerous useful features for managing and analyzing genetic data

- Data management
  - Read data in a variety of formats
  - Recode and reorder files
  - Merge two or more files
  - Extracts subsets (SNPs or individuals)
  - Flip strand of SNPs
  - Compress data in a binary file format

What other things can PLINK do with your genomic data? A WHOLE lot!

- **Summary statistics for quality control**
  - Allele, genotypes frequencies, HWE tests
  - Missing genotype rates
  - Inbreeding, IBS and IBD statistics for individuals and pairs of individuals
  - non-Mendelian transmission in family data
  - Sex checks based on X chromosome SNPs
  - Tests of non-random genotyping failure

- **Basic association testing**
  - Case/control
  - Standard allelic test
  - Fisher's exact test
  - Dominant/recessive and general models
  - Model comparison tests (e.g. general versus multiplicative)
  - Quantitative traits, association and interaction
  - Multimarker predictors, haplotypic tests
  - Copy number variant analysis
  - …

What other things can PLINK do with your genomic data? A WHOLE lot!

- **Gene-based tests of association**
- **Screen for epistasis**
- **Gene-environment interaction with continuous and dichotomous environments**
- **Meta-analysis**
  - Automatically combine several generically-formatted summary files, for millions of SNPs

Do I need to be afraid of PLINK?

BROAD INSTITUTE

How do I get PLINK?

- Download page: https://www.cog-genomics.org/plink/2.0/

PLINK 2.00 alpha

PLINK 2.0 alpha was developed by **Christopher Chang**, with support from **GRAIL, LLC** and **Human Longevity, Inc.**, and substantial input from Stanford's Department of Biomedical Data Science. (More detailed credits.) (Usage questions should be sent to the **plink2-users Google group**, not Christopher's email.)

Binary downloads

	Build	
Operating system	**Development (28 Mar 2022)**	Alpha 2.3 final (24 Jan 2020)
Linux AVX2 Intel[1]	**download**	download
Linux 64-bit Intel[1]	download	download
Linux 32-bit	download	download
macOS AVX2[2]	**download**	download
macOS 64-bit[2]	download	download
Windows AVX2	**download**	download
Windows 64-bit	download	download
Windows 32-bit	download	download

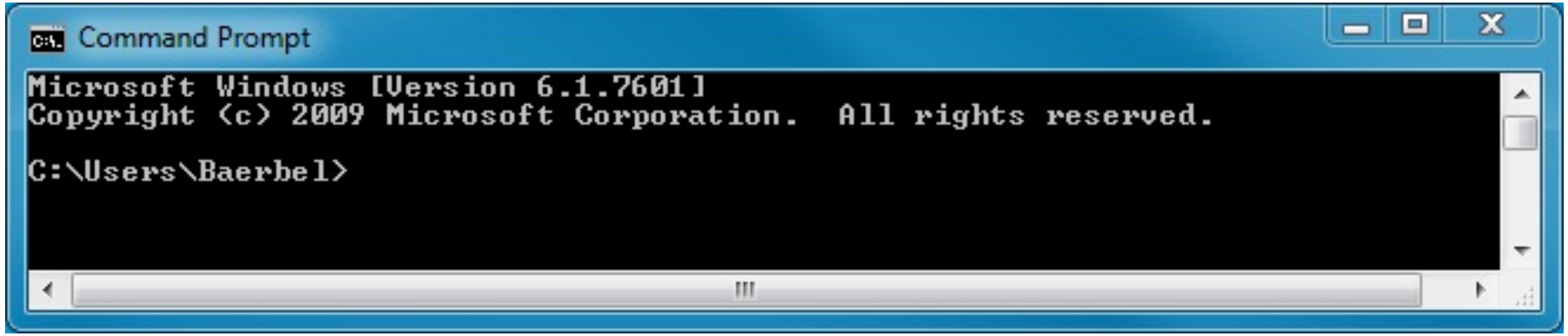
1: These builds can still run on AMD processors, but they're statically linked to Intel MKL, so some linear algebra operations will be slow. We will try to provide an AMD Zen-optimized build as soon as supporting libraries are available.
2: You need to have Rosetta 2 installed to run this on M1 Macs.

How do I get PLINK?

- Decompress the zip file into directory, e.g., `C:\Program Files\plink-1.07-dos`

- You should be ready to go!

- Open command prompt or other command line program



What if I am lost at using PLINK?

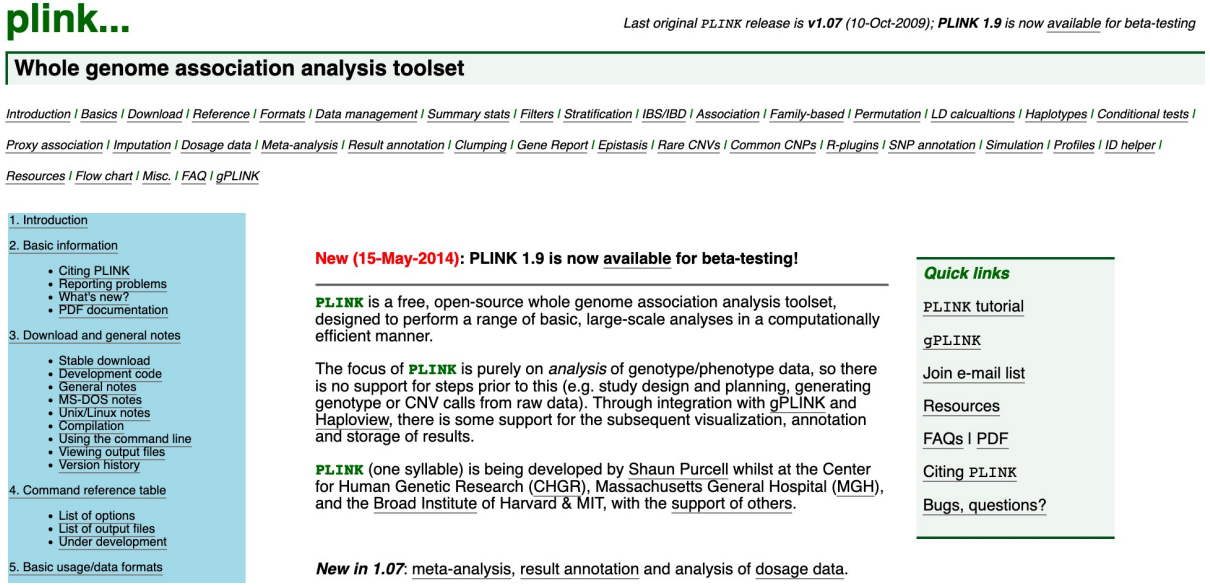
- There are 2 primary websites where you can find PLINK documentation
  - If there is a command for what you want to do
  - How to use commands
  - How to interpret output from particular flags
  - What is the correct syntax

Main website:
http://zzz.bwh.harvard.edu/plink/index.shtml

Updates/new flags in newer versions of PLINK:
https://www.cog-genomics.org/plink2

PLINK documentation

PLINK 1.07 website (bring valuable extra info)



plink...

*Last original PLINK release is **v1.07** (10-Oct-2009); **PLINK 1.9** is now available for beta-testing*

Whole genome association analysis toolset

Introduction / Basics / Download / Reference / Formats / Data management / Summary stats / Filters / Stratification / IBS/IBD / Association / Family-based / Permutation / LD calcualtions / Haplotypes / Conditional tests / Proxy association / Imputation / Dosage data / Meta-analysis / Result annotation / Clumping / Gene Report / Epistasis / Rare CNVs / Common CNPs / R-plugins / SNP annotation / Simulation / Profiles / ID helper / Resources / Flow chart / Misc. / FAQ / gPLINK

1. Introduction
2. Basic information
   - Citing PLINK
   - Reporting problems
   - What's new?
   - PDF documentation
3. Download and general notes
   - Stable download
   - Development code
   - General notes
   - MS-DOS notes
   - Unix/Linux notes
   - Compilation
   - Using the command line
   - Viewing output files
   - Version history
4. Command reference table
   - List of options
   - List of output files
   - Under development
5. Basic usage/data formats

New (15-May-2014): PLINK 1.9 is now available for beta-testing!

PLINK is a free, open-source whole genome association analysis toolset, designed to perform a range of basic, large-scale analyses in a computationally efficient manner.

The focus of **PLINK** is purely on *analysis* of genotype/phenotype data, so there is no support for steps prior to this (e.g. study design and planning, generating genotype or CNV calls from raw data). Through integration with gPLINK and Haploview, there is some support for the subsequent visualization, annotation and storage of results.

PLINK (one syllable) is being developed by Shaun Purcell whilst at the Center for Human Genetic Research (CHGR), Massachusetts General Hospital (MGH), and the Broad Institute of Harvard & MIT, with the support of others.

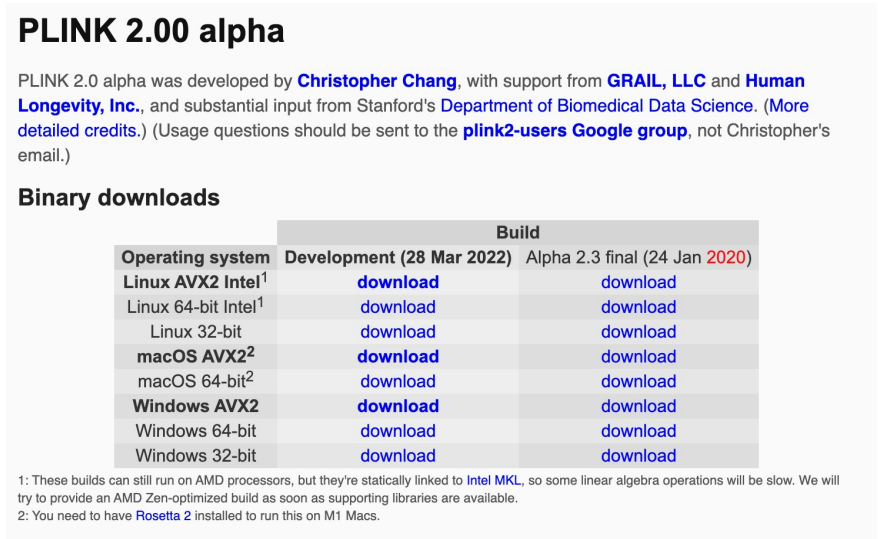
New in 1.07: meta-analysis, result annotation and analysis of dosage data.

Quick links

PLINK tutorial
gPLINK
Join e-mail list
Resources
FAQs | PDF
Citing PLINK
Bugs, questions?

http://zzz.bwh.harvard.edu/plink/

PLINK 2.0 website



PLINK 2.00 alpha

PLINK 2.0 alpha was developed by **Christopher Chang**, with support from **GRAIL, LLC** and **Human Longevity, Inc.**, and substantial input from Stanford's Department of Biomedical Data Science. (More detailed credits.) (Usage questions should be sent to the **plink2-users Google group**, not Christopher's email.)

Binary downloads

	Build	
Operating system	**Development (28 Mar 2022)**	Alpha 2.3 final (24 Jan 2020)
Linux AVX2 Intel[1]	**download**	download
Linux 64-bit Intel[1]	download	download
Linux 32-bit	download	download
macOS AVX2[2]	**download**	download
macOS 64-bit[2]	download	download
Windows AVX2	**download**	download
Windows 64-bit	download	download
Windows 32-bit	download	download

1: These builds can still run on AMD processors, but they're statically linked to Intel MKL, so some linear algebra operations will be slow. We will try to provide an AMD Zen-optimized build as soon as supporting libraries are available.
2: You need to have Rosetta 2 installed to run this on M1 Macs.

https://www.cog-genomics.org/plink/2.0/

Use documentation to know:

* if there is a command for what you want to do
* how to interpret output from particular flags
* how to use commands
* what is the correct syntax

PLINK Files

- There are two standard file types for PLINK:
  - ped files, e.g., filename.ped
    - contain information about the family, phenotype, and genotype calls
  - map files, e.g., filename.map
    - contain information about the genetic markers

Chr	Marker	cM	Position
1	rs9729550	0	11352421
1	rs6603788	0	1218086

Example Map file, chr1.map:

- Genotypes are stored in: chr1.ped
- The markers map is described in: chr1.map

Input Files: ped file

- Pedigree File - the first six columns are mandatory, followed by the genotypes:
  - Col1: Family ID
  - Col2: Individual ID
  - Col3: Paternal ID
  - Col4: Maternal ID
  - Col5: Sex (1=male; 2=female; other=unknown)
  - Col6: Phenotype

Child! 

```
test.ped:
  1 1 0 0 1  0  G G  2 2  C C
  1 2 0 0 2  0  A A  0 0  A C
  1 3 1 2 1  2  0 0  1 2  A C
  2 1 0 0 1  0  A A  2 2  0 0
  2 2 0 0 2  2  A A  2 2  0 0
  2 3 1 2 1  2  A A  2 2  A A
```

Input Files: ped file

•For most options, PLINK needs two plain text files:

1) a file with family ID, individual ID, father ID, mother ID, sex (1=male, 2 = female, other = unknown), phenotype and genotypes in columns for each individual , with the extension .ped. This file has NO HEADER so looks like this:

```
FAM001 1 0 0 1 2 A A G G A C
FAM001 2 0 0 1 2 A A A G 0 0
...
```

Input Files: map file

2) a file with marker information, including chromosome number, the SNP name, the position in morgans on the chromosome, and the position in base pairs on the chromosome, with the extension .map. Again, this file has NO HEADER so looks like this:

```
1 rs123456 0 1234555
1 rs234567 0 1237793
1 rs224534 0 1237697
1 rs233556 0 1337456
...
```

HINT!

It is easiest if these files have the same name (e.g. sheep.ped and sheep.map).

The historical text format (map/ped)

file.map

```
1   rs12 0.12 123123
1   rs34 0.14 123456
1   rs56 0.15 123789
…
23 rs78 0.12 100000
```

Chromosome (23=chr X, 24=chr Y)

Variant ID

Position (in cM)

Position (in bp)

VARIANT INFO

file.ped

```
FID1 IID1     0     0 1 1 A A 1 1 0 0 ... A T
FID1 IID2     0     0 2 1 A C 1 1 1 2 ... T T
FID1 IID3 IID1 IID2 2 2 A C 1 2 2 2 ... T T
FID1 IID4 IID1 IID2 2 0 0 0 1 2 0 0 ... A T
```

rs12 rs34 rs56 rs78

1st col: Family ID
2nd col: Individual ID
3rd col: Father ID
4th col: Mother ID
5th col: Gender (1=male, 2=female)
6th col: Pheno (1=control, 2=case);

0 or -9 = missing data

can be quantitative

INDIVIDUAL INFO

(here 2 parents and their 2 kids)

GENETIC DATA

The historical text format (map/ped)

file.map

```
1   rs12 0.12 123123
1   rs34 0.14 123456
1   rs56 0.15 123789
…
23 rs78 0.12 100000
```

Chromosome (23=chr X, 24=chr Y)

Variant ID

Position (in cM)

Position (in bp)

VARIANT INFO

file.ped

```
FID1 IID1    0    0 1 1 A A 1 1 0 0 ... A T
FID2 IID2    0    0 2 1 A C 1 1 1 2 ... T T
FID3 IID3    0    0 2 2 A C 1 2 2 2 ... T T
FID4 IID4    0    0 2 0 0 0 1 2 0 0 ... A T
```

rs12 rs34 rs56 rs78

1st col: Family ID
2nd col: Individual ID
3rd col: Father ID
4th col: Mother ID
5th col: Gender (1=male, 2=female)
6th col: Pheno (1=control, 2=case);

0 or -9 = missing data

can be quantitative

INDIVIDUAL INFO

(here 4 “unrelated” individuals)

GENETIC DATA

PLINK File Types

- Genotype data as a text file
  - Pedigree file (.ped)
  - Map file (.map)

- Genotype data as a compressed binary file
  - Fam File (.fam)
  - Bim file (.bim)
  - Bed file (.bed)

PLINK Commands

```
> plink --file filename --options
```

filename — without extension, PLINK will look for filename.ped and filename.map

options — various kind of options, see the following slides and documentation

Several options can be combined and position of options is not fixed! For example:

```
> plink --noweb --file ibdrelease5_QCI --remove related.indiv.txt
```

BUT the order in which commands are executed is fixed and may not correspond to the order in which they are entered

Output

All results are written to files with specific suffices, depending on the type of the performed operation(s).

Examples for standard suffixes for PLINK output:

Types of Operation	**Suffix**
Association	plink.assoc
Logistic regression model	plink.assoc.logistic
Hardy-Weinberg test statistics	plink.hwe

… and many, many more (see documentation)

Specify root name (this replaces ‘plink’ in filenames; suffix unchanged):

```
> plink --out name
```

Creating Binary PLINK files

With the PLINK files myfile.ped and myfile.map, the PLINK command

```
> plink --file myfile --make-bed --out myfile
```

generates the following three files:

- `Myfile.bed`
- `Myfile.bim`
- `Myfile.fam`

These files contain the same information as the standard map/ped files, but are a more compact representation of the data, which saves space and speeds up subsequent analysis and future portability. --make-bed tells the output to be in binary format (binary ped)

PLINK Binary Files

- The .fam and .bim files are still plain text files: these can be viewed with a standard text editor, or in the command line.

- Do not try to view the .bed file however: it is a compressed file and you'll only see lots of strange characters on the screen...

- To tell plink that the input data are in binary format, as opposed to the normal text PED/MAP format, just use the --bfile option instead of --file.

```
plink.bed      ( binary file, genotype information )
plink.fam      ( first six columns of mydata.ped )
plink.bim      ( extended MAP file: two extra cols = allele names)
```

The binary format (bim/fam/bed)

file.map

```
1   rs12 0.12 123123
1   rs34 0.14 123456
1   rs56 0.15 123789
…
23  rs78 0.12 100000
```

file.ped

```
FID1 IID1 0 0 1 1 A A 1 1 0 0 ... A T
FID2 IID2 0 0 2 1 A C 1 1 1 2 ... T T
FID3 IID3 0 0 2 2 A C 1 2 2 2 ... T T
FID4 IID4 0 0 2 0 0 0 1 2 0 0 ... A T
```

file.bim

```
1   rs12 0.12 123123 C A
1   rs34 0.14 123456 2 1
1   rs56 0.15 123789 1 2
…
23  rs78 0.12 100000 A T
```

Minor allele Major allele

file.fam

```
FID1 IID1 0 0 1 1
FID2 IID2 0 0 2 1
FID3 IID3 0 0 2 2
FID4 IID4 0 0 2 0
```

file.bed

COMPRESSED GENETIC DATA

VARIANT INFO **INDIVIDUAL INFO** **GENETIC DATA**

Module 3.1.2:
Data Manipulation and Conversion

Learning objectives:

Describing steps for simple data manipulation and conversion.
Providing a hands-on example; details in homework.

Adapted from Bärbel Maus and Elizabeth Atkinson

Need any help during your analysis?

```
> PLINK --help
> PLINK --bfile --help
```

Tips for working with PLINK

- Type all options on a single line
- If necessary, add an “\” to include more options
- Ensure exact syntax and spelling!
- Always check the logfile! Can help troubleshoot errors
- Check if PLINK options can really be combined
- Check the order in which PLINK options are executed, see documetation for this

Obtaining Allele Frequency Information (--freq)

```
> PLINK \
  --bfile $BFILE \
  --freq \
  --out data_clean
```

← Name header of PLINK inputs

← Actions you want PLINK to do

← Name header of PLINK outputs

Output from --freq flag

```
-bash:login02:/broad/GINGER/ginger_vc_c1_year3/class_01.29.20 1054 $ $PLINK \
>   --bfile $BFILE \
>   --freq \
>   --out NeuroGAP_pilotData_clean
PLINK v1.90b6.12 64-bit (28 Oct 2019)           www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to NeuroGAP_pilotData_clean.log.
Options in effect:
  --bfile /broad/GINGER/data/pilotData_clean/NeuroGAP_pilotData_clean
  --freq
  --out NeuroGAP_pilotData_clean

7812 MB RAM detected; reserving 3906 MB for main workspace.
Allocated 292 MB successfully, after larger attempt(s) failed.
332284 variants loaded from .bim file.
913 people (0 males, 0 females, 913 ambiguous) loaded from .fam.
Ambiguous sex IDs written to NeuroGAP_pilotData_clean.nosex .
Using 1 thread.
Before main variant filters, 913 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.999441.
--freq: Allele frequencies (founders only) written to
NeuroGAP_pilotData_clean.frq .
```

Output of the --freq flag

```
> head -5 NeuroGAP_pilotData_clean.*
==> NeuroGAP_pilotData_clean.frq <==
 CHR                  SNP   A1   A2          MAF  NCHROBS
   1          1:58814:G:A    A    G       0.1628     1824
   1         1:752721:A:G    G    A       0.4118     1826
   1         1:759036:G:A    A    G        0.046     1826
   1         1:794332:G:A    A    G       0.1703     1826

==> NeuroGAP_pilotData_clean.log <==
PLINK v1.90b6.12 64-bit (28 Oct 2019)
Options in effect:
  --bfile /broad/GINGER/data/pilotData_clean/NeuroGAP_pilotData_clean
  --freq
  --out NeuroGAP_pilotData_clean

==> NeuroGAP_pilotData_clean.nosex <==
0       201689740201_R01C01
0       201689740201_R03C01
0       201715370152_R01C01
0       201715370152_R01C02
0       201715370152_R02C01
```

1st col: chromosome
2nd col: rs ID (same as bim file)
3rd col: minor allele
4th col: major allele
5th col: minor allele frequency
6th col: # alleles observation

Record messages and events that occur during the run

List individuals with no sex

PLINK options to read and write genetic data

FORMAT	INPUT OPTION	OUTPUT OPTION
text (map/ped)	`--file`	`--recode`
binary (bim/fam/bed)	`--bfile`	`--make-bed`
imputed (sample/bgen)	`--sample --bgen *`	`NA`
sequencing (vcf)	`--vcf`	`--recode vcf`
additive (raw, for R) `NA`		`--recodeA`

* See options `--hard-call-threshold <value>` or `--hard-call-threshold random` (https://www.cog-genomics.org/plink/2.0/data#recode for more output options)

Carinal Rules and Caveats

- When using *PLINK* there are a few key points to remember.
  - Always consult the LOG file (console output)
  - *PLINK* has no memory
    - each run loads data anew, previous filters lost
  - Exact syntax and spelling is **very important**

- Not every option can be combined with every other option
    - For example, basic haplotype tests cannot take covariates
    - *PLINK* doesn't always warn you
    - LOG file often shows what has happened (or not)
  - Consult the web documentation (http://pngu.mgh.harvard.edu/purcell/plink/)

Adapted from Elizabeth Atkinson, Ph.D.

Data structure importance

1. Reproducibility (for yourself and others)
2. Collaboration
3. Reusability
4. Writing up
5. Debugging

Thank you, TJ Singh!

Script organization

Scripts should be named, ordered and organized externally (relative to each other), and internally

1. Loading libraries and data (see dependencies)
2. Functions, classes, methods (refactoring)
3. QC, computing and model fitting (expensive things)
4. Visualization, presenting, narrative
5. Communication

Easier to keep track of how everything is done

Run all the costly/expensive stuff first, and then move on to more nimble model fitting and interpretation

Thank you, TJ Singh!

File structure

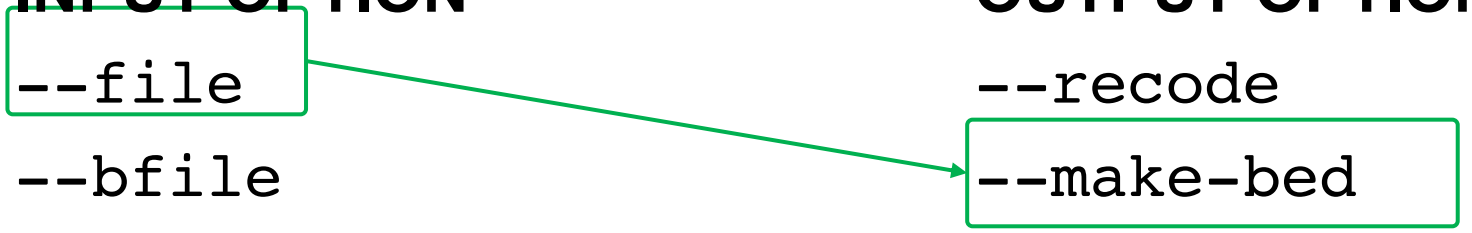
`/analysis` : final reports (write up, figures etc. output)
`/src` : scripts (source files here, reusable)
`/output` : data artifacts (munging outputs)
`/notebook` : where notebooks are
`/data` : raw source data
`/doc` : documentation

`/ext` : external files

Thank you, TJ Singh!

Example 1
Creating a binary file (deprecated!!!)

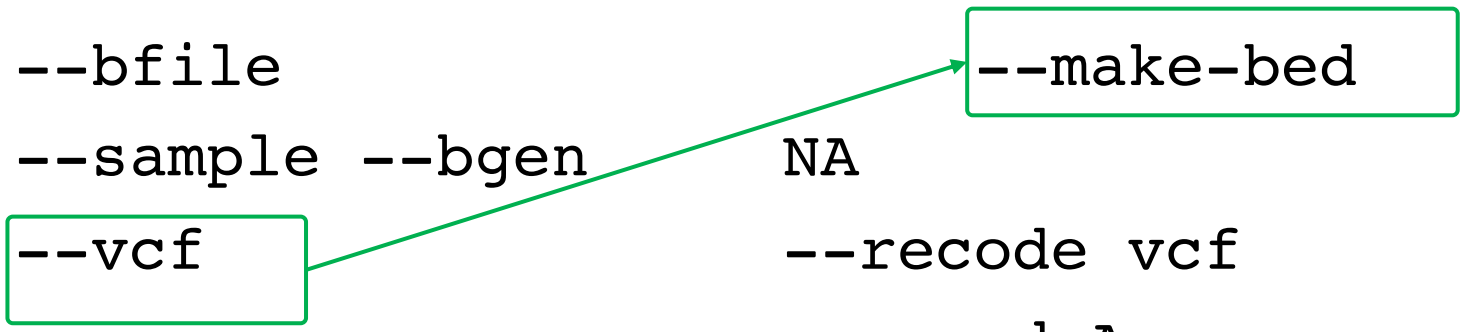
FORMAT	INPUT OPTION	OUTPUT OPTION
text (map/ped)	`--file`	`--recode`
binary (bim/fam/bed)	`--bfile`	`--make-bed`
imputed (sample/bgen)	`--sample --bgen`	`NA`
sequencing (vcf)	`--vcf`	`--recode vcf`
additive (raw, for R) `NA`		`--recodeA`



(see https://www.cog-genomics.org/plink/1.9/data#recode for more output options)

Example 2
Converting a vcf file to a binary file

FORMAT	INPUT OPTION	OUTPUT OPTION
text (map/ped)	`--file`	`--recode`
binary (bim/fam/bed)	`--bfile`	`--make-bed`
imputed (sample/bgen)	`--sample --bgen`	`NA`
sequencing (vcf)	`--vcf`	`--recode vcf`
additive (raw, for R) `NA`		`--recodeA`



(see https://www.cog-genomics.org/plink/1.9/data#recode for more output options)

Example 2
Converting a vcf file to a binary file

```
$PLINK \
  --vcf $VCF \          # input: $VCF
  --make-bed \
  --out chr22           # outputs: chr22.bim, chr22.fam, …
```

Example 3
Keeping/removing individuals based on a file with FID and IID

```
# obtain list of IIDs from .fam file
cut -f1,2 $BFILE.fam | head > listiid.txt
```

```
# keep indiv. in listiid.txt
$PLINK \
  --bfile $BFILE \
  --keep listiid.txt \
  --make-bed \
  --out listiid_in
```

```
# remove indiv. in listiid.txt
$PLINK \
  --bfile $BFILE \
  --remove listiid.txt \
  --make-bed \
  --out listiid_out
```

Example 4
Obtaining Hardy-Weinberg Test Statistic

```
> plink --bfile filename --hardy
```

Creates text file plink.hwe (use less, or a text editor to open)

CHR	SNP	TEST	A1	A2	GENO	O(HET)	E(HET)	P
1	snp1	ALL	A	C	1/2/3	0.3333	0.4444	1
1	snp1	AFF	A	C	0/1/2	0.3333	0.2778	1
1	snp1	UNAFF	A	C	1/1/1	0.3333	0.5	1
1	snp2	ALL	G	T	1/3/2	0.5	0.4861	1
1	snp2	AFF	G	T	0/1/2	0.3333	0.2778	1
1	snp2	UNAFF	G	T	1/2/0	0.6667	0.4444	1

```
> plink --bfile filename --hardy --out filename2
```

Creates text file filename2.hwe

Example 5
Extracting SNPs using rsIDs from a dataset

```
> plink --bfile filename --extract mysnps.txt --recode
```

Creates file plink.ped and plink.map (use text editor, e.g., notepad to open) – did not specify output file name!

```
> plink --file filename --extract mysnps.txt --recode --
out filename2
```

Creates file filename2.ped and filename2.map

Example 5.1
Extracting SNPs using rsIDs from a dataset

```
cut -f2 $BFILE.bim | head > listrs.txt
```

```
# extract SNPs in listrs.txt
> PLINK \
  --bfile $BFILE \
  --extract listrs.txt \
  --make-bed \
  --out listrs_in
```

```
# exclude SNPs in listrs.txt
> PLINK \
  --bfile $BFILE \
  --exclude listrs.txt \
  --make-bed \
  --out listrs_out
```

Example 5.1
Extracting SNPs using rsIDs from a dataset

```
PLINK v1.90b6.12 64-bit (28 Oct 2019)          www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to listiid_out.log.
Options in effect:
  --bfile /broad/GINGER/data/pilotData_clean/NeuroGAP_pilotData_clean
  --extract listrs.txt
  --make-bed
  --out listiid_out

7812 MB RAM detected; reserving 3906 MB for main workspace.
Allocated 292 MB successfully, after larger attempt(s) failed.
332284 variants loaded from .bim file.
913 people (0 males, 0 females, 913 ambiguous) loaded from .fam.
Ambiguous sex IDs written to listiid_out.nosex .
--extract: 10 variants remaining.
Using 1 thread.
Before main variant filters, 913 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.999671.
10 variants and 913 people pass filters and QC.
Note: No phenotypes present.
--make-bed to listiid_out.bed + listiid_out.bim + listiid_out.fam ... done.
```

```
PLINK v1.90b6.12 64-bit (28 Oct 2019)          www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to listiid_out.log.
Options in effect:
  --bfile /broad/GINGER/data/pilotData_clean/NeuroGAP_pilotData_clean
  --exclude listrs.txt
  --make-bed
  --out listiid_out

7812 MB RAM detected; reserving 3906 MB for main workspace.
Allocated 292 MB successfully, after larger attempt(s) failed.
332284 variants loaded from .bim file.
913 people (0 males, 0 females, 913 ambiguous) loaded from .fam.
Ambiguous sex IDs written to listiid_out.nosex .
--exclude: 332274 variants remaining.
Using 1 thread.
Before main variant filters, 913 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.999441.
332274 variants and 913 people pass filters and QC.
Note: No phenotypes present.
--make-bed to listiid_out.bed + listiid_out.bim + listiid_out.fam ... done.
```

Example 6
Keeping individuals and extracting variants at the same time

```
$PLINK \
  --bfile $BFILE \
  --extract listrs.txt \
  --keep listiid.txt \
  --make-bed \
  --out listrsiid_in
```

```
PLINK v1.90b6.12 64-bit (28 Oct 2019)          www.cog-genomics.org/plink/1.9/
(C) 2005-2019 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to listrsiid_in.log.
Options in effect:
  --bfile /broad/GINGER/data/pilotData_clean/NeuroGAP_pilotData_clean
  --extract listrs.txt
  --keep listiid.txt
  --make-bed
  --out listrsiid_in

7812 MB RAM detected; reserving 3906 MB for main workspace.
Allocated 292 MB successfully, after larger attempt(s) failed.
332284 variants loaded from .bim file.
913 people (0 males, 0 females, 913 ambiguous) loaded from .fam.
Ambiguous sex IDs written to listrsiid_in.nosex .
--extract: 10 variants remaining.
--keep: 10 people remaining.
Using 1 thread.
Before main variant filters, 10 founders and 0 nonfounders present.
Calculating allele frequencies ... done.
10 variants and 10 people pass filters and QC.
Note: No phenotypes present.
--make-bed to listrsiid_in.bed + listrsiid_in.bim + listrsiid_in.fam ... done.
```

Example 7
Extracting/excluding variants based on their chromosome

```
# keep variants from chr 20
$PLINK \
  --bfile $BFILE \
  --chr 20 \
  --make-bed \
  --out chr20
# you can also use options --chr 1-4,10 or --not-chr 20
```

Flags for extracting variants based on rs IDs or positions

```
--snp rsx                  # extract SNP rsx
--snps rsx,rsy             # extract SNPs rsx and rsy
--snps rsx-rsy             # extract SNPs from rsx to rsy
--snp rsx --window 50      # extract SNPs +/-50kb around rsx

--from-bp a --to-bp b      # extract SNPs from pos. a to b
--from-kb a --to-kb b      # extract SNPs from pos. a kb to b kb
--from-mb a --to-mb b      # extract SNPs from pos. a Mb to b Mb
```

Bonus example:
Extracting candidate variants to analyze with for R

```
> PLINK \
  --bfile $BFILE \
  --snps 1:752721:A:G,1:840753:T:C \
  --recodeA \                  # create a genotype matrix, coded 0/1/2
  --out mysnps

use R-3.5
R
data = read.table("mysnps.raw", header = T)
head(data); table(data$X1.752721.A.G_G)
```

What if you have tri-allelic sites?

- PLINK can represent tri-allelic alleles. BUT
  - Only very limited ability to analyse them
  - Same SNP may appear several times in the MAP or BIM file
  - Usually filter out tri-allelic alleles
  - Often an issue when merging data sets

Hint:
If you don't want to deal with it,
use the --max-alleles 2 flag

What if the strands in your dataset is flipped?

- Different chips or experiments may record a SNP using different strands
  - A C
  - T G

- See when merge data —appears to be multi-allelic
  - PLINK reports apparently multi-allelic SNPs
  - You can flip them – create a new data set
  - Try merge again – if multi-allelic should work
  - Filter out remaining
  - May incorrectly flip a few

- Flipping strand means changing alleles

A -> T C -> G G -> C T -> A

Hint: To flip, use the --flip merged.missnp flag where merged_missnp is a list of SNPs that you want to strand flip

so, for example, a A/C SNP will become a T/G; alternatively, a A/T SNP will become a T/A SNP (i.e. in this case, the labels remain the same, but whether the minor allele is A or T will still depend on strand).

Conclusion

- PLINK can do a lot of things by taking as inputs only binary files
- PLINK can be used for data management (keeping/removing individuals and extracting/excluding variants)
- And now to get our hands dirty!
- Don’t be afraid to ask for more explanations if something in the lab isn’t clear enough, or a command isn’t working!

Getting started with plink on your VM

- Spin up your VM from yesterday
- Make a directory named `20220405_GINGER-UCT_IntroToPlink`
- Create the following dir structure within the `20220405_GINGER-UCT_IntroToPlink` dir
  - `src`
  - `output`
  - `log`
  - `data`
- Go into `src` dir and download plink

```
curl -LO 'https://s3.amazonaws.com/plink2-assets/plink2_linux_x86_64_20220328.zip'

unzip plink2_linux_x86_64_20220328.zip

rm plink2_linux_x86_64_20220328.zip

mv plink2 plink

sudo mv plink /usr/local/bin/
```

Getting started with plink on your VM

- Go into you your `data` dir
  - Download to this dir the following files

```
gsutil cp gs://neurogap_phenos_genos/NeuroGAP_pilot_clean_grch38.* .
```

Best friends here:

```
cd ..
cd <insert dir name>
ls
pwd
```

- Start the exercises

https://www.dropbox.com/s/w30r0prys8qqv8f/GINGER2022_day2_plink.pdf?dl=0

Sometimes you need both plink versions
Carla will explain more tomorrow

Final plink tasks

- Copy your log files (plink logs and your own log document of today's scripts) to your VM

```
gcloud compute scp <files> --project "gingeriimak" <VM-id>:~/ 20220405_GINGER-UCT_IntroToPlink/log --zone "us-central1-b"
```

- Copy your entire plink VM directory to a local folder in your own machine

```
gcloud compute scp --project "gingeriimak" --recurse <VM-id>:~/20220405_GINGER-UCT_IntroToPlink/ --zone "us-central1-b" <localFolder>
```

ALWAYS check to see if your files are where they are with `ls`

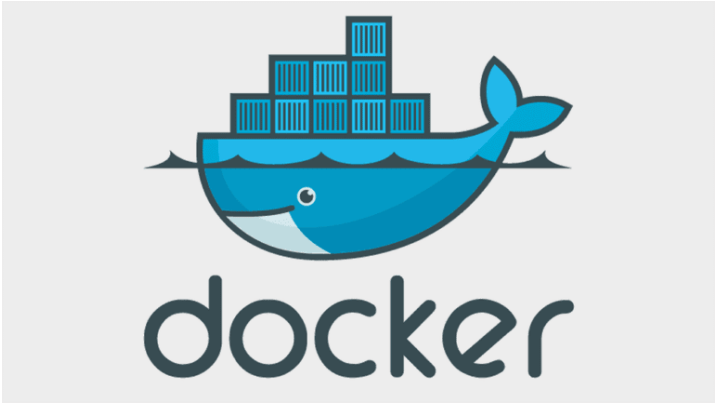
FINALLY, switch off your VM.

Changes to schedule for the rest of the week (for the moment)

	Monday, April 4	Tuesday, April 5	Wednesday, April 6	Thursday, April 7	Friday, April 8
9:00-10:30	**Training Welcome and Introduction** **9:00-9:30 - Training Overview** **9:30-10:00 - Professor Dan Stein Welcome** **10:00-11:00 - Begin Kampala Refresher**	**Plink Tutorial**	**Step-by-Step GWAS**	**9:00-10:00 am TBD** **10:00 am NeuroGAP Site Visit**	**9:00-10:30am - Step by Step GWAS** **10:30-11:00am - Group Project Presentations** **11:00am-12:00pm - Guest Lectures: Drs. Shareefa Dalvie and Nastassja Koen**
10:30-10:45	Tea Break	Tea Break	Tea Break		
10:45-1:00	**Kampala Refresher continued**	**11:00-12:00 - Intro to Plink** **12:00-1:00 - Professor Collet Dandara**	**Step-by-Step GWAS**	12-1pm - Lunch	12-1pm - Lunch
1:00-2:00	Lunch	Lunch	Lunch	**Step-by-Step GWAS**	**Excursion to Robben Island**
2:00-3:30	**Intro to UNIX** Fundamental Commands Genetic Data Formats and Conversion	**Plink Tutorial**	**Group Project Work**		
3:30-3:45	Tea Break	Tea Break		Tea Break	
3:45-5:00	**GINGER group projects intro**	**Plink Tutorial**		**Step-by-Step GWAS**	

Docker image and why we use it in data science?

- **Containers**: very small user-level virtualization that helps you build, install, and run your code. Like a cookie
- **Images**: a snapshot of your container. A cookie cutter mould
- **Dockerfile**: a yaml-based file that's used to build your image; this is what we can version control. Instructions on how to create the cookie mould.
- **Dockerhub**: GitHub for your Docker images; you can set up Dockerhub to automatically build an image anytime you update your Dockerfile in GitHub



YOUR TURN with Docker!

- Spin up your VM back up

```
sudo apt-get install docker.io
gcloud auth configure-docker
wget "https://github.com/GoogleCloudPlatform/docker-credential-gcr/releases/download/v2.1.0/docker-credential-gcr_linux_386-2.1.0.tar.gz"


tar -xvzf docker-credential-gcr_linux_386-2.1.0.tar.gz
sudo mv docker-credential-gcr /usr/bin/
sudo chmod +x /usr/bin/docker-credential-gcr


docker-credential-gcr gcr-login
#follow instructions and use Broad email
sudo chmod 666 /var/run/docker.sock

docker pull us.gcr.io/gingeriimak/ginger-uct:0.1 #pull image to you
docker run -it -v /home/kumar/:/mnt/home us.gcr.io/gingeriimak/ginger-uct:0.1 #mount your local directory to your image

cd mnt/home
```

GINGER On-site Training
Day 2: Introduction to Plink

GINGER Program 2022

University of Cape Town

Teaching Fellows:

Kumar Veerapen, Ph.D.
Senior Expert I Data Science
Kumar.Veerapen@Novartis.com

Carla Marquez-Luna, Ph.D.
Postdoctoral Research Fellow
carlamarquezluna@gmail.com