Hands-on analysis of DArTseq data for linkage mapping

Jessen Bredeson
and
Jess Lyons

NSF

IITA

Research to Nourish Africa

NATIONAL ROOT CROPS RESEARCH INSTITUTE

NRCRI

UMUDIKE

Points of order

- Please be on time and minimize disruptions
- GitHub site is home base for everything pertaining to the course
  - https://github.com/bredeson/HandsOnDArT
  - Current agenda, resources, exercises, *etc*.
  - Check there first!
- Course goals
  - Understand DArTseq data, and use it for mapping
  - Strengthen UNIX skills in an applied context
  - Teach teachers

On your index card:

Name, and preferred name

Research focus

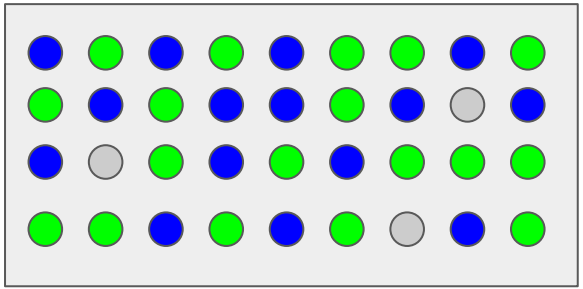
What will you be using DArTseq data for?

Experience working with genotyping data/mapping

GBS/DArTseq overview

DArT genotyping platforms

1. DArT (traditional): Oligo hybridization array chip
- Restriction digestion + bacterial cloning
- Relative fluorescent color signal
- Presence/Absence
- 100s–1,000s of markers



2. DArTseq: Reduced representation sequencing
- Restriction enzyme double-digest
- Methylation sensitivity
- Markers biased to genic portions of genome
- 1,000s–10,000s of markers

GBS/DArTseq library construction

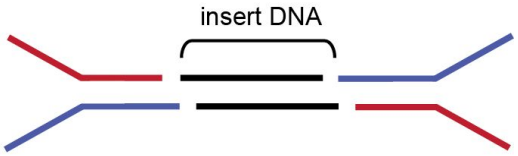
Restriction digest

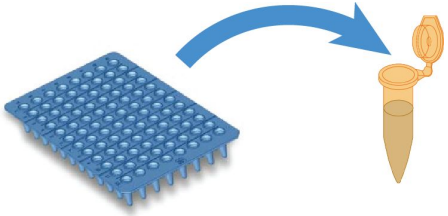
ApeKI: G|C W G C
C G W C|G

W= A or T

Ligate barcoded adapters

5′ ACACTCTTTCCCTACACGACGCTCTTCCGATCTbarcode 3′
3′ GAGCCGTAAGGACGACTTGGCGAGAAGGCTAGAbarcodeGWC 5′



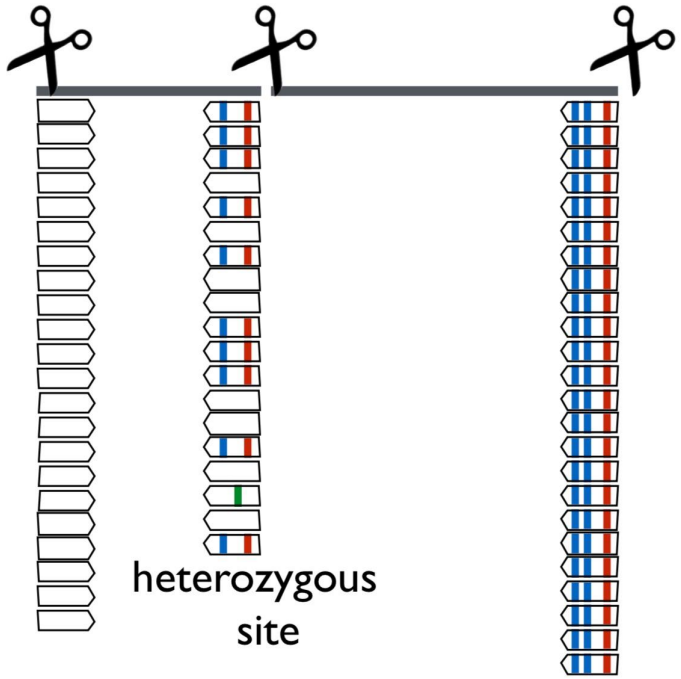


Pool samples
PCR

TbarcodeCWG CWGbarcodeA
AbarcodeGWC GWCbarcodeT

Sequencing

GBS/DArTseq variant sampling



Diversity Arrays Technology

IGSS Africa

DArTseq data types

1. SNP: Nucleotide differences observed in tag sequences

- “Co-dominant” genotypes:

```
Aa x aa => 1 Aa : 1 aa
Aa x Aa => 1 AA : 2 Aa : 1 aa
AA x Aa => 1 AA : 1 Aa
```

0	aa
1	Aa
2	AA
-	No data

2. SilicoDArTs:

- Presence/absence variation
- “Dominant” genotypes:

G^1 x G^0 : Aa x aa => 1 Aa : 1 aa => 1 G^1 : 1 G^0
: AA x aa => 2 Aa : 0 aa => 2 G^1 : 0 G^0

0	A allele not present
1	A allele present
-	No data

DArT bioinformatic analysis

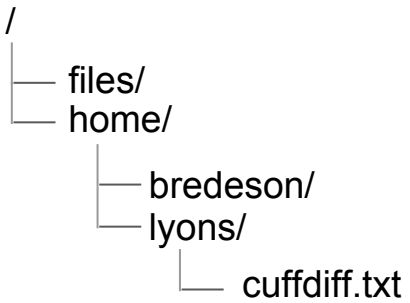
- Data generation:
  - Libraries sequenced using Illumina HiSeq 2000/2500
  - Sequencing reads filtered for 90% confidence over half of read
- Genome-independent variants and genotyping calling
  - Reads clustered to form ~77 bp marker sequence tags
  - SNPs called as differences between tags
- Allows markers to be mapped/re-mapped to a genomic sequence
  - No need to re-map all reads and re-call SNPs

A brief UNIX review

Useful UNIX commands for working with files

Tools specifically useful for looking at files/directories:

1. Navigation: ls, cd, pwd
   relative vs. absolute paths
2. File viewing: less, more, head, tail, od, column
3. File manipulation: mv, cp, rm, mkdir, grep, cat, cut, tr, sed
4. File compression and archiving: gzip, bzip2, zip, tar
5. File transfer: scp, wget, curl
6. Getting help: man, apropos



Absolute: `ls /home/lyons/cuffdiff.txt`
Relative: `ls ./cuffdiff.txt`

Common cross-platform file issues

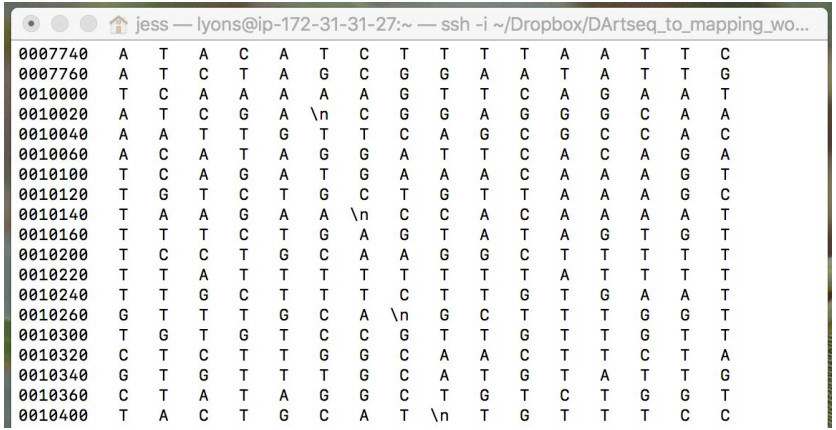
- Invisible characters:
  - CR/LF/CRLF
  - Tabs vs spaces
- Special characters:
  - File encoding (ASCII vs. Unicode)

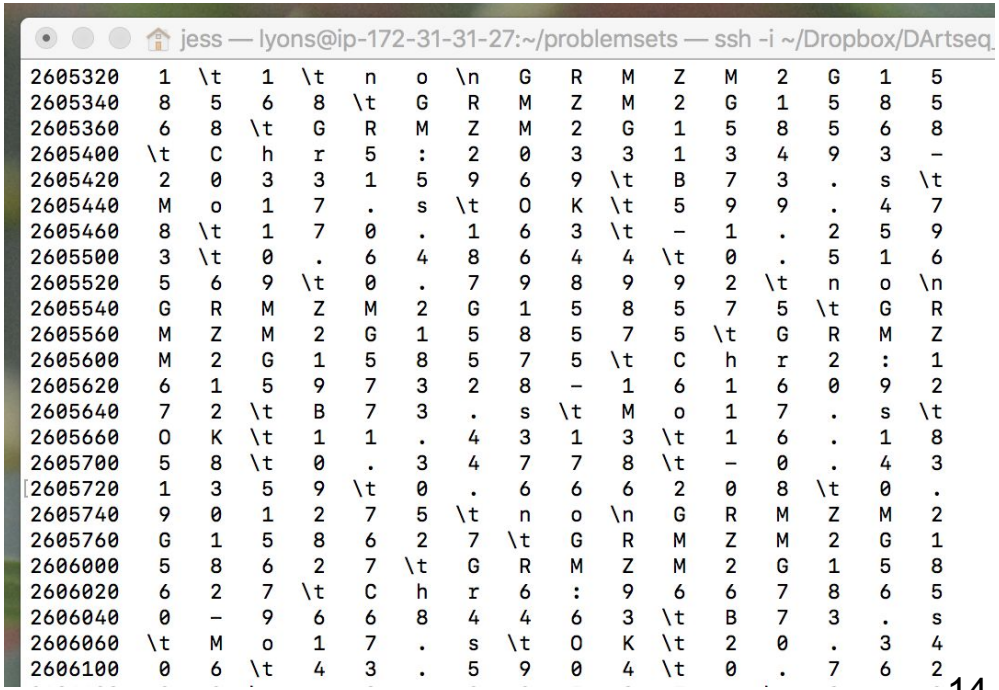
Carriage Return (CR)	Old Mac OS versions	\r
Line Feed (LF)	MacOS X, UNIX, LINUX	\n
CRLF	Windows	\r\n

!!!

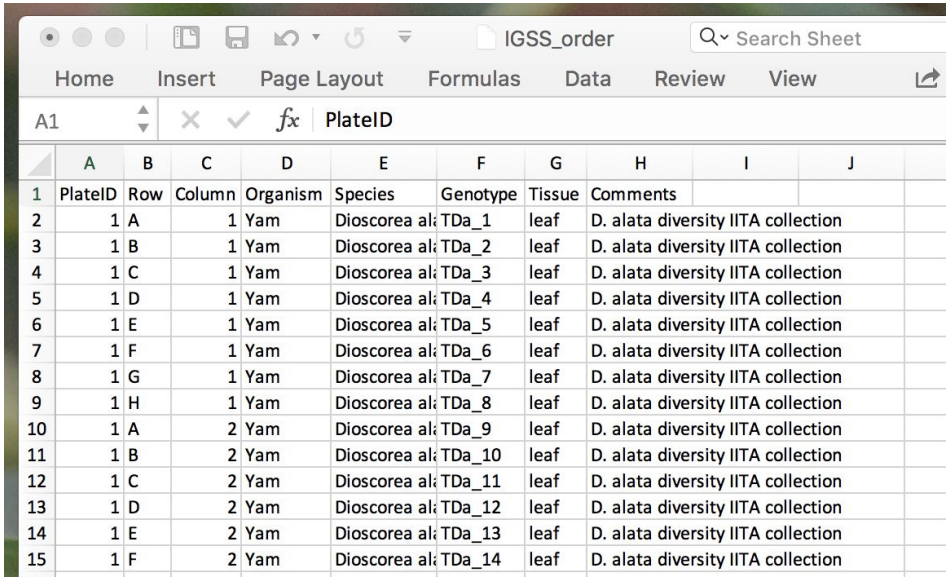
Invisible characters

od -c is your friend!





Invisible characters



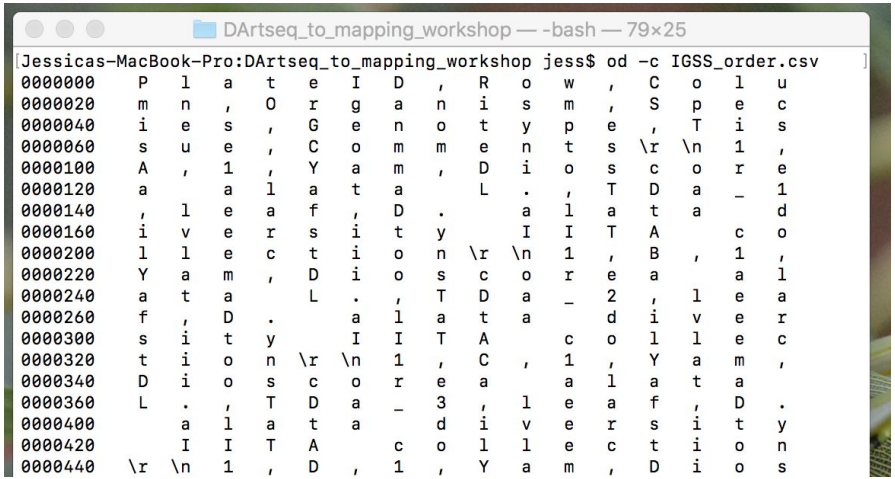
IGSS_order

Search Sheet

Home Insert Page Layout Formulas Data Review View

A1 fx PlateID

	A	B	C	D	E	F	G	H	I	J
1	PlateID	Row	Column	Organism	Species	Genotype	Tissue	Comments		
2	1	A	1	Yam	Dioscorea al	TDa_1	leaf	D. alata diversity IITA collection		
3	1	B	1	Yam	Dioscorea al	TDa_2	leaf	D. alata diversity IITA collection		
4	1	C	1	Yam	Dioscorea al	TDa_3	leaf	D. alata diversity IITA collection		
5	1	D	1	Yam	Dioscorea al	TDa_4	leaf	D. alata diversity IITA collection		
6	1	E	1	Yam	Dioscorea al	TDa_5	leaf	D. alata diversity IITA collection		
7	1	F	1	Yam	Dioscorea al	TDa_6	leaf	D. alata diversity IITA collection		
8	1	G	1	Yam	Dioscorea al	TDa_7	leaf	D. alata diversity IITA collection		
9	1	H	1	Yam	Dioscorea al	TDa_8	leaf	D. alata diversity IITA collection		
10	1	A	2	Yam	Dioscorea al	TDa_9	leaf	D. alata diversity IITA collection		
11	1	B	2	Yam	Dioscorea al	TDa_10	leaf	D. alata diversity IITA collection		
12	1	C	2	Yam	Dioscorea al	TDa_11	leaf	D. alata diversity IITA collection		
13	1	D	2	Yam	Dioscorea al	TDa_12	leaf	D. alata diversity IITA collection		
14	1	E	2	Yam	Dioscorea al	TDa_13	leaf	D. alata diversity IITA collection		
15	1	F	2	Yam	Dioscorea al	TDa_14	leaf	D. alata diversity IITA collection		

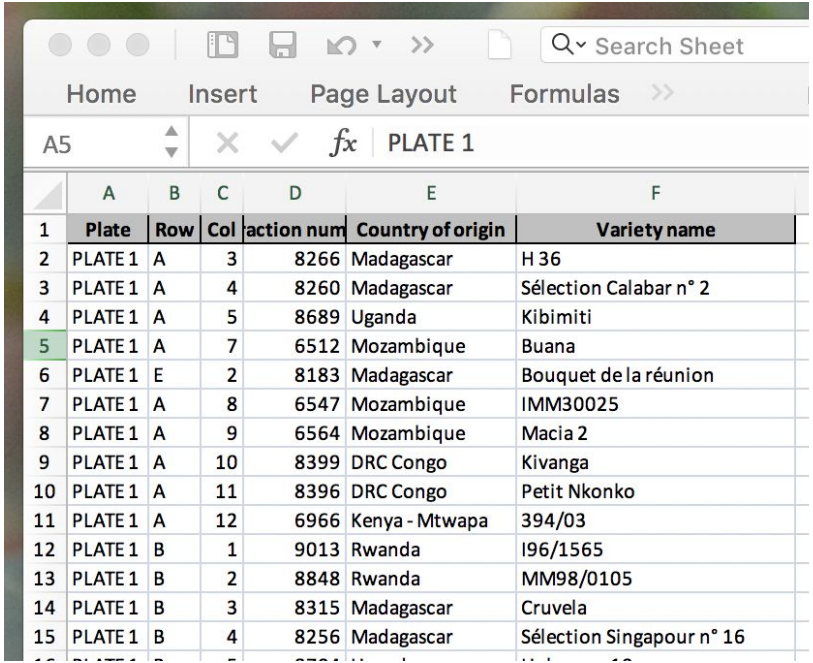


DArtseq_to_mapping_workshop — -bash — 79×25

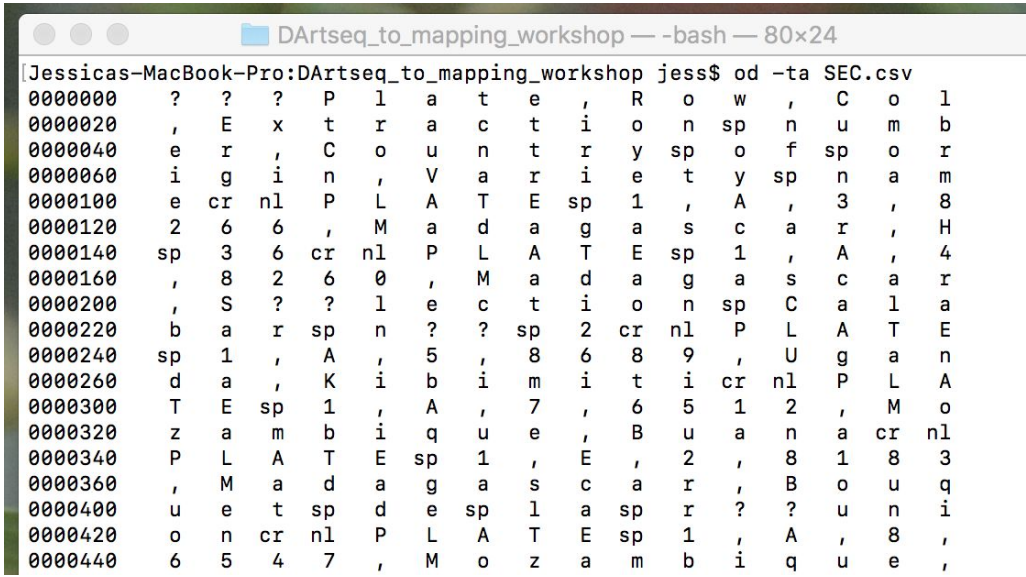
```
[Jessicas-MacBook-Pro:DArtseq_to_mapping_workshop jess$ od -c IGSS_order.csv ]
0000000   P   l   a   t   e   I   D   ,   R   o   w   ,   C   o   l   u
0000020   m   n   ,   O   r   g   a   n   i   s   m   ,   S   p   e   c
0000040   i   e   s   ,   G   e   n   o   t   y   p   e   ,   T   i   s
0000060   s   u   e   ,   C   o   m   m   e   n   t   s  \r  \n   1   ,
0000100   A   ,   1   ,   Y   a   m   ,   D   i   o   s   c   o   r   e
0000120   a       a   l   a   t   a       L   .   ,   T   D   a   _   1
0000140   ,   l   e   a   f   ,   D   .       a   l   a   t   a       d
0000160   i   v   e   r   s   i   t   y       I   I   T   A       c   o
0000200   l   l   e   c   t   i   o   n  \r  \n   1   ,   B   ,   1   ,
0000220   Y   a   m   ,   D   i   o   s   c   o   r   e   a       a   l
0000240   a   t   a       L   .   ,   T   D   a   _   2   ,   l   e   a
0000260   f   ,   D   .       a   l   a   t   a       d   i   v   e   r
0000300   s   i   t   y       I   I   T   A       c   o   l   l   e   c
0000320   t   i   o   n  \r  \n   1   ,   C   ,   1   ,   Y   a   m   ,
0000340   D   i   o   s   c   o   r   e   a       a   l   a   t   a
0000360   L   .   ,   T   D   a   _   3   ,   l   e   a   f   ,   D   .
0000400       a   l   a   t   a       d   i   v   e   r   s   i   t   y
0000420       I   I   T   A       c   o   l   l   e   c   t   i   o   n
0000440  \r  \n   1   ,   D   ,   1   ,   Y   a   m   ,   D   i   o   s
```

.csv file made on a mac

Special characters



	A	B	C	D	E	F
1	Plate	Row	Col	action num	Country of origin	Variety name
2	PLATE 1	A	3	8266	Madagascar	H 36
3	PLATE 1	A	4	8260	Madagascar	Sélection Calabar n° 2
4	PLATE 1	A	5	8689	Uganda	Kibimiti
5	PLATE 1	A	7	6512	Mozambique	Buana
6	PLATE 1	E	2	8183	Madagascar	Bouquet de la réunion
7	PLATE 1	A	8	6547	Mozambique	IMM30025
8	PLATE 1	A	9	6564	Mozambique	Macia 2
9	PLATE 1	A	10	8399	DRC Congo	Kivanga
10	PLATE 1	A	11	8396	DRC Congo	Petit Nkonko
11	PLATE 1	A	12	6966	Kenya - Mtwapa	394/03
12	PLATE 1	B	1	9013	Rwanda	I96/1565
13	PLATE 1	B	2	8848	Rwanda	MM98/0105
14	PLATE 1	B	3	8315	Madagascar	Cruvela
15	PLATE 1	B	4	8256	Madagascar	Sélection Singapour n° 16



Non-ASCII characters are not read properly

Common HTS data formats

Common HTS formats you may encounter

- FASTA
- FASTQ
- VCF
- SAM

FASTA sequence file format

"Greater-than" symbol (>) signifies a new sequence record

Sequence ID
(required)

Sequence description*
(optional)

(both on one line)

```
>AF140613.1 Manihot esculenta N-hydroxylating cytochrome P450 (CYP79D1) mRNA, complete cds
GTTCAGGGCATATCAATATGGCCATGAACGTCTCCACCACCATCGGTTTACTTAACGCCACCTCCTTCGC
CTCCTCCTCCTCCATCAACACGGTCAAGATCTTGTTCGTCACCCTCTTTATTTCCATTGTTAGTACTATT
GTAAAACTTCAAAAGAGTGCTGCTAACAAGGAAGGTAGCAAGAAACTCCCACTCCCTCCTGGCCCTACTC
CATGGCCACTCATCGGAAACATCCCGGAAATGATCCGGTACAGACCCACGTTTCGGTGGATTCACCAACT
CATGAAGGACATGAACACTGATATTTGTCTCATTCGTTTTGGAAGAACTAACTTTGTTCCTATAAGCTGT
CCTGTTCTTGCTCGTGAAATACTAAAAAAGAATGACGCTATCTTCT
```

Sequence string
May be "wrapped flush" to an arbitrary width, or not at all

*Everything after the first whitespace (one or more spaces or tabs) is interpreted as description

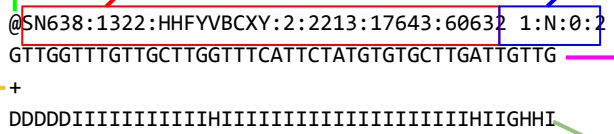
FASTQ sequence file format

"At" symbol (@) signifies a new sequence record (but may also be observed in the quality string

Sequence ID
(required)

(both on one line)

Sequence description*
(optional; Paired-end Illumina sequences will have 1 or 2 in the leading colon-separated fields)



Sequence string
Not wrapped

Quality string
Not wrapped; Characters represent ASCII+*offset* transformed numeric values; see https://en.wikipedia.org/wiki/FASTQ_format for more info

Quality ID
Must start with a '+';
Typically left blank

*Everything after the first whitespace (one or more spaces or tabs) is interpreted as description

VCF: Variant Call Format

Tab-separated, columnar file

Declarative Meta information header line(s)
Starts with two pound symbols (##)

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="PASSes all quality filters">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##contig=<ID=Chr1,length=26803966>
##contig=<ID=Chr2,length=24424175>
#CHROM  POS     ID  REF  ALT    QUAL      FILTER  INFO  FORMAT  Sample
Chr1    187607  .   A    G      34149.20  PASS    .     GT      1/1
Chr1    1597049 .   ACT  A,AC   21739.90  PASS    .     GT      0/2
```

File body header line
Starts with one pound symbol (#)

SNP locus

Indel locus

Homozygous Alternate allele

Heterozygous

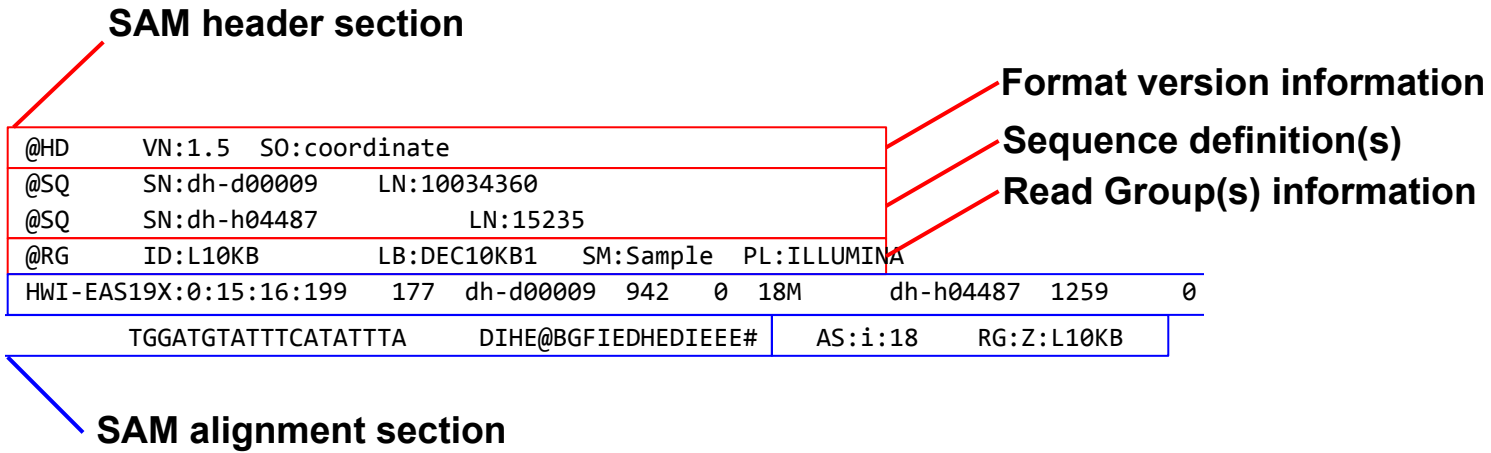
Alternate alleles indices = 1,2,...

Reference allele index = 0

Please visit https://samtools.github.io/hts-specs/VCFv4.2.pdf for more detailed information.

SAM: Sequence Alignment/Map format

Tab-separated, columnar file



Please visit https://samtools.github.io/hts-specs/SAMv1.pdf for more detailed information.

UNIX & File Format Exercises

1. If you have not already completed the exercises on the PFB2017 website, please do so first. If you have, proceed to the next item.
2. Log into the AWS server
3. Make a directory called “exercises” and navigate into it
4. Copy all files starting with “data” from the /files directory into your exercises directory.
5. Examine the contents of data1 (hint: use less).
    a. What format of file is it? Rename the file with the appropriate file extension.
    b. Is the contents of the file single-end or paired-end?
6. Examine the contents of data2.
    a. What format of file is it? Rename the file with the appropriate file extension.
    b. What is incorrect about its formatting?

UNIX & File Format Exercises

7. Examine the contents of data3 (hint: use man to look at the -S option).
    a. What format of file is it? Rename the file with the appropriate file extension.
    b. How many loci are there? How many SNPs? How many Indels?
8. Examine the contents of data4.
    a. What format of file is it?
    b. What is incorrect about this file (hint: see od)?
    c. From which operating system might it have come from?
    d. Correct the file using UNIX commands (hint: see tr)
9. Examine the contents of data5.
    a. The file is tab-delimited, but what is incorrect about this file?
    b. Correct the file using UNIX commands.

UNIX & File Format Exercises Solutions

1. If you have not already completed the exercises on the PFB2017 website, please do so first. If you have, proceed to the next item.

   The solutions to these may be found at:
   https://github.com/srobb1/pfb2017/blob/master/problemsets/answers/JVB_Unix_01_problemset.md
   Ignore the solutions to #1 and #2; they are not specific to your machine or our course.

2. Log into the AWS server

   Instructions are here:
   https://github.com/bredeson/HandsOnDArT#logging-into-the-aws-server

3. Make a directory called “exercises” and navigate into it

   From your home directory:

   ```
   mkdir exercises
   cd exercises
   ```

UNIX & File Format Exercises Solutions

4. Copy all files starting with “data” from the /files directory into your exercises directory.

```
cp /files/data* .
```

5. Examine the contents of data1 (hint: use less).

```
less data1
```

   a. What format of file is it? Rename the file with the appropriate file extension.

   FASTQ

```
mv data1 data1.fastq
```

   b. Is the contents of the file single-end or paired-end?

   paired-end

UNIX & File Format Exercises Solutions

6. Examine the contents of data2.

```
less data2
od -c data2
```

   a. What format of file is it? Rename the file with the appropriate file extension.

      FASTA

      ```
      mv data2 data2.fasta
      ```

   b. What is incorrect about its formatting?

      Extra line break, between Sequence ID and Sequence description, putting the description on its own line. The sequence description should be on the same line as the sequence ID, with the sequence line immediately following on the next line.

UNIX & File Format Exercises Solutions

7. Examine the contents of data3 (hint: use man to look at the -S option).

```
less -S data3
```

   a. What format of file is it? Rename the file with the appropriate file extension.

      VCF

```
mv data3 data3.vcf
```

   b. How many loci are there? How many SNPs? How many Indels?

```
grep -v -e# data3.vcf | wc -l
```

      41 loci
      All are SNPs
      No Indels

UNIX & File Format Exercises Solutions

8. Examine the contents of data4.

```
less -S data4
```

   a. What format of file is it?

      CSV

   b. What is incorrect about this file (hint: see od)?

      ```
      od -c data4
      ```

      Carriage returns (\r) instead of line feeds (\n)

   c. From which operating system might it have come from?

      e.g. Legacy Mac OS

   d. Correct the file using UNIX commands (hint: see tr)

      ```
      cat data4 | tr '\r' '\n' > data4.fixed
      ```

UNIX & File Format Exercises Solutions

9. Examine the contents of data5.

```
od -c data5
od -ta data5
```

   a. The file is tab-delimited, but what is incorrect about this file?
      The first line has spaces instead of a tab

   b. Correct the file using UNIX commands.

```
cat data5 | tr -s ' ' '\t' > data5.fixed
```

Tuesday, 24 July 2018

- Quick review of Problem Set solutions
- DArTseq file formats
- Anchoring genotypes to a genome assembly
- Marker filtering

DArT file formats

DArTseq files

Report_DY17-1234/

- Report_DY17-1234_SNP_singlerow.csv
- Report_DY17-1234_SNP.csv
- Report_DY17-1234_SilicoDArTs.csv
- metadata.json

SNP (singlerow)

```
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,DY17-2486,DY17-2486,DY17-2486,DY17-2486,DY17-248
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,910917261001,910917261001,910917261001,910917261
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,A,A,A,A,A,A,A,A,A,A,A,A,B,B,B,B,B,B,B,B,B,B,B,B,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,1,10,11,12,2,3,4,5,6,7,8,9,1,10,11,12,2,3,4,5,6,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,Progeny,Progeny,Progeny,Progeny,Progeny,Progeny,
AlleleID,AlleleSequence,TrimmedSequence,Chrom_Assembly,ChromPos_Assembly,AlnCnt_Assembly,A
100072732|F|0-42:C>G-42:C>G,TGCAGCATCATGTACAATCAACAAATAGAACTGAACCACATACCATATGCCCACCTACAAAG
100049426|F|0-59:A>T-59:A>T,TGCAGCAGTCCACAAAACTTGTCAGCACTTCTAAATAATTAAATCAAGTGTTCCAAACAAAT
100014258|F|0-32:C>A-32:C>A,TGCAGACAGAATATAACTTCAACTACTTTTAGCAAGACTCATGTGTAAATAACTGCTTACAT
100069025|F|0-28:G>T-28:G>T,TGCAGTCAATTCATTTTAAAGACACAAGGTAAAGGAAGGGAGATTACAGATCGGAAGAGCGG
100041372|F|0-35:G>A-35:G>A,TGCAGCTGGGAGCTGGGAGGATGTCAATTTAATCAGATCAGAGATTACAGATCGGAAGAGCG
```

SNP (standard/two-row)

```
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,DY17-2486,DY17-2486,DY17-2486,DY17-2486,DY17-2
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,910917261001,910917261001,910917261001,9109172
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,A,A,A,A,A,A,A,A,A,A,A,A,B,B,B,B,B,B,B,B,B,B,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,1,10,11,12,2,3,4,5,6,7,8,9,1,10,11,12,2,3,4,5,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,Progeny,Progeny,Progeny,Progeny,Progeny,Progen
AlleleID,AlleleSequence,TrimmedSequence,Chrom_Assembly,ChromPos_Assembly,AlnCnt_Assembly
100072732|F|0--42:C>G,TGCAGCATCATGTACAATCAACAAATAGAACTGAACCACATACCATATGCCCACCTACAAAGTTAT
100072732|F|0-42:C>G-42:C>G,TGCAGCATCATGTACAATCAACAAATAGAACTGAACCACATAGCATATGCCCACCTACAA
100049426|F|0--59:A>T,TGCAGCAGTCCACAAAACTTGTCAGCACTTCTAAATAATTAAATCAAGTGTTCCAAACAAATAACT
100049426|F|0-59:A>T-59:A>T,TGCAGCAGTCCACAAAACTTGTCAGCACTTCTAAATAATTAAATCAAGTGTTCCAAACAT
100014258|F|0--32:C>A,TGCAGACAGAATATAACTTCAACTACTTTTAGCAAGACTCATGTGTAAATAACTGCTTACATCACA
100014258|F|0-32:C>A-32:C>A,TGCAGACAGAATATAACTTCAACTACTTTTAGAAAGACTCATGTGTAAATAACTGCTAAC
100069025|F|0--28:G>T,TGCAGTCAATTCATTTTAAAGACACAAGGTAAAGGAAGGGAGATTACAGATCGGAAGAGCGGTTCA
```

SilicoDArTs

```
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,DY17-2486,DY17-2486,DY17-2486,DY17-2486,DY17-2
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,910917261001,910917261001,910917261001,9109172
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,A,A,A,A,A,A,A,A,A,A,A,A,B,B,B,B,B,B,B,B,B,B,B,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,1,10,11,12,2,3,4,5,6,7,8,9,1,10,11,12,2,3,4,5,
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,Progeny,Progeny,Progeny,Progeny,Progeny,Progen
CloneID,AlleleSequence,TrimmedSequence,Chrom_Assembly,ChromPos_Assembly,Al
100018673,TGCAGCCAAATACTAAGCTTTAGTTCCTACGATCAACCAAGATGCTAACCTTGAATCTTGATGA
100062117,TGCAGAACTTGAATTTATCAAGAAATCATTTATCAGGACAACTTTCAGAAAAAATCGGCCAGCT
100070803,TGCAGCAATTGAGAACACACTGTCAACTAAGCTAGAGGATGCTCGGCAGCTATTGCAGTTACAG
100000052,TGCAGCTGGTGAATTTGTTATATTTATCTTTGCTGTCATCTACTGAAGGATTTGAAAGTAGTTT
100069972,TGCAGAACACTAATGCTGTTCAAACCACAGAGAATGTAATACAGATCTCATTGGACCATGTAAG
100022056,TGCAGGCTAGCTTCTTGGATGATAGAAAGAATGAGCCATTGGATCTTCATATTTAGTAAGCTGC
```

JSON: **J**ava**S**cript **O**bject **N**otation

metadata.json

```
"orders" : {
   "DY17-1234" : {
      "clientemail" : "your@address.org",
      "sequencing" : {
         "1" : {
            "HiSeq 2500" : "2017-06-02 16:14:10"
         },
         "2" : {
            "HiSeq 2500" : "2017-06-23 14:47:38"
         }
      },
      "orderdatetime" : "2017-05-05 03:19:36",
      "clientname" : "Your Name",
      "productname" : "DArTseq (1.0)"
   }
},
```

Mapping SNP tags to a genome

MapTK: Introduction to the UI

MapTK is a tool kit for linkage mapping

- Many useful tools/commands
- Every tool has its own usage page, options, and arguments

Not compatible with Windows machines--yet!

Note: Non-required and required arguments: [arg] vs. <arg>

ICGMC. 2015. G3 Journal. doi: 10.1534/g3.114.015008
https://bitbucket.org/rokhsar-lab/gbs-analysis/src/master/

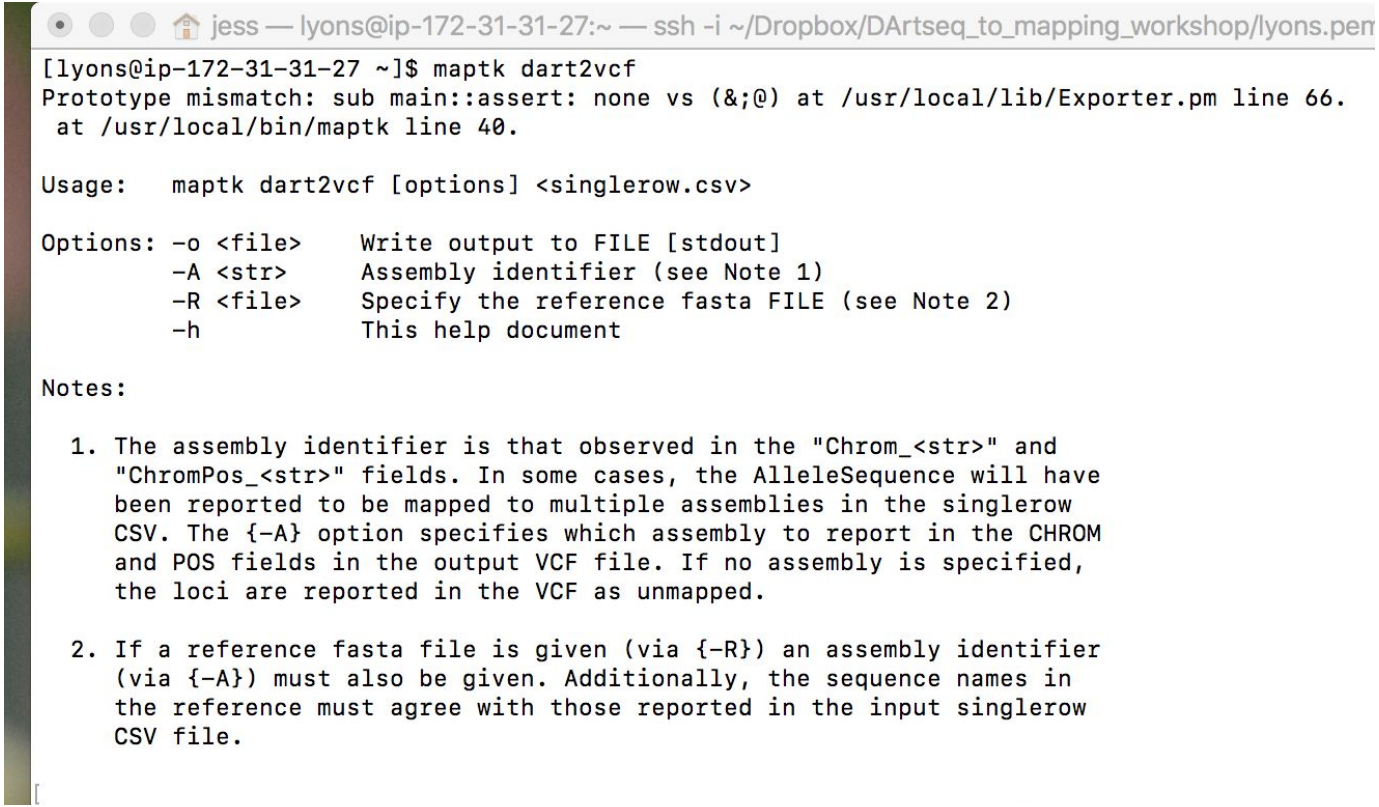
MapTK UI: list of commands

jess — lyons@ip-172-31-31-27:~ — ssh -i ~/Dropbox/DArtseq_to_mapping_workshop/lyons.pe

```
[lyons@ip-172-31-31-27 ~]$ maptk
Prototype mismatch: sub main::assert: none vs (&;@) at /usr/local/lib/Exporter.pm line 66.
 at /usr/local/bin/maptk line 40.

Program: maptk (Utilities for working with mapping data)
Version: 0.1.0
Contact: Jessen Bredeson <jessenbredeson@berkeley.edu>

Usage:   maptk <command> [options]

Command: dart2vcf  Convert DArT singlerow CSV format to VCF
         maptags   Extract and re-map marker tags to an assembly
         chif1     Calc. Mendelian segregation stats for F1 crosses
         mtvr      Calc. Mendelian transmission violation rate
         mclust    Model-based clustering of relatedness2 output
         vcf2loc   Convert VCF to JoinMap .loc format for mapping
         vcf2raw   Convert VCF to OneMap .raw format for mapping

         dotplot   Plot the genetic distance of one map against another
         h2k       Convert from Haldane to Kosambi centiMorgan scale
         nn        Est marker genetic position by nearest neighbors
         predict   Est marker genetic position by physical position
         paint     Color a genetic map by component contig
         rescale   Rescale a map by regressing onto a second map
         rev       Reverse some or all LG orientations

         anchor    Anchor and orient contigs onto a genetic map
         break     Break scaffolds at user-defined locations
         join      Join scaffolds with user-defined join information
```

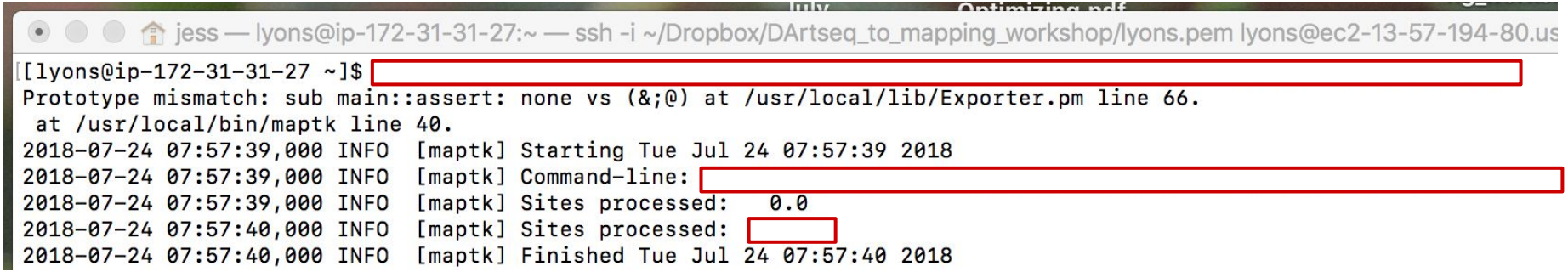
```
Notes:

  1. Input .map files are required to adhere to the PLINK MAP file spec
     (http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#map).

  2. Many of the above tools include a '-M' option which, when enabled,
     requires marker IDs to be formatted as a concatentation of each
     marker's contig ID and physical position separated by any one of
     the characters in the set [:/], e.g. "Contig1:1000". This marker
     ID format is intended to allow these tools to reference a marker's
     coordinate in both sequenced contig space and chromosome space
     should they differ (as when anchoring scaffolds onto a map to form
     pseudomolecular representations of the chromosome).
```

MapTK dart2vcf command



MapTK dart2vcf command

How do you know if dart2vcf ran correctly?



MapTK maptags command

jess — lyons@ip-172-31-31-27:~ — ssh -i ~/Dropbox/DArtseq_to_mapping_workshop/lyons.pen

```
[lyons@ip-172-31-31-27 ~]$ maptk maptags
Prototype mismatch: sub main::assert: none vs (&;@) at /usr/local/lib/Exporter.pm line 66.
 at /usr/local/bin/maptk line 40.

Usage:   maptk maptags [options] <in.vcf> <genome.fasta> <out-prefix>

Options: -p <str>   INFO tag representing variant position in tag [SnpPosition]
         -s <str>   INFO tag representing variant tag sequence [AlleleSequence]
         -q <uint>  Minimum mapping quality for confidently mapped reads [60]
         -h         This help document

Notes:  Arguments to {-p} and {-s} are case sensitive
```

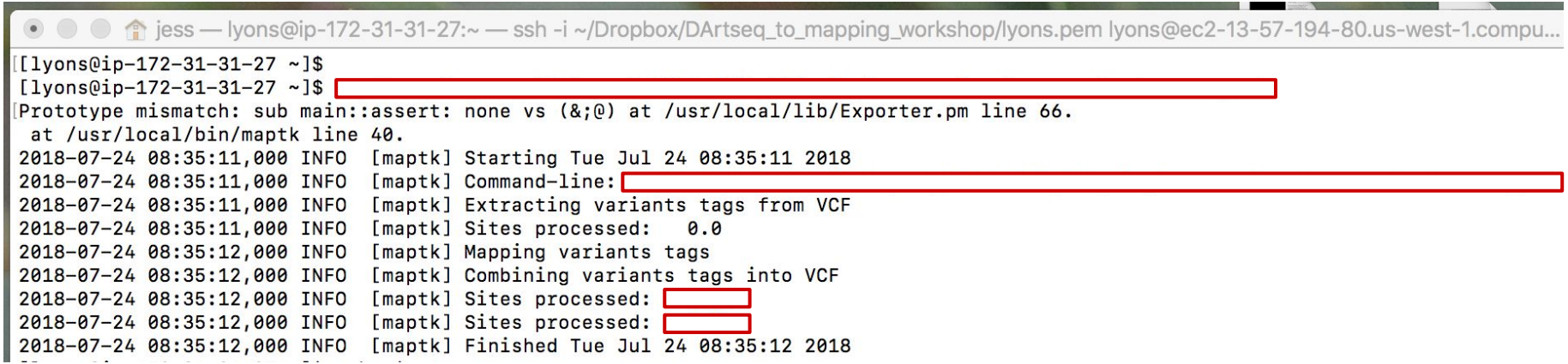
Our reference genome: /files/reference.fasta

Sample DArT data: /files/Report_DY17-1234

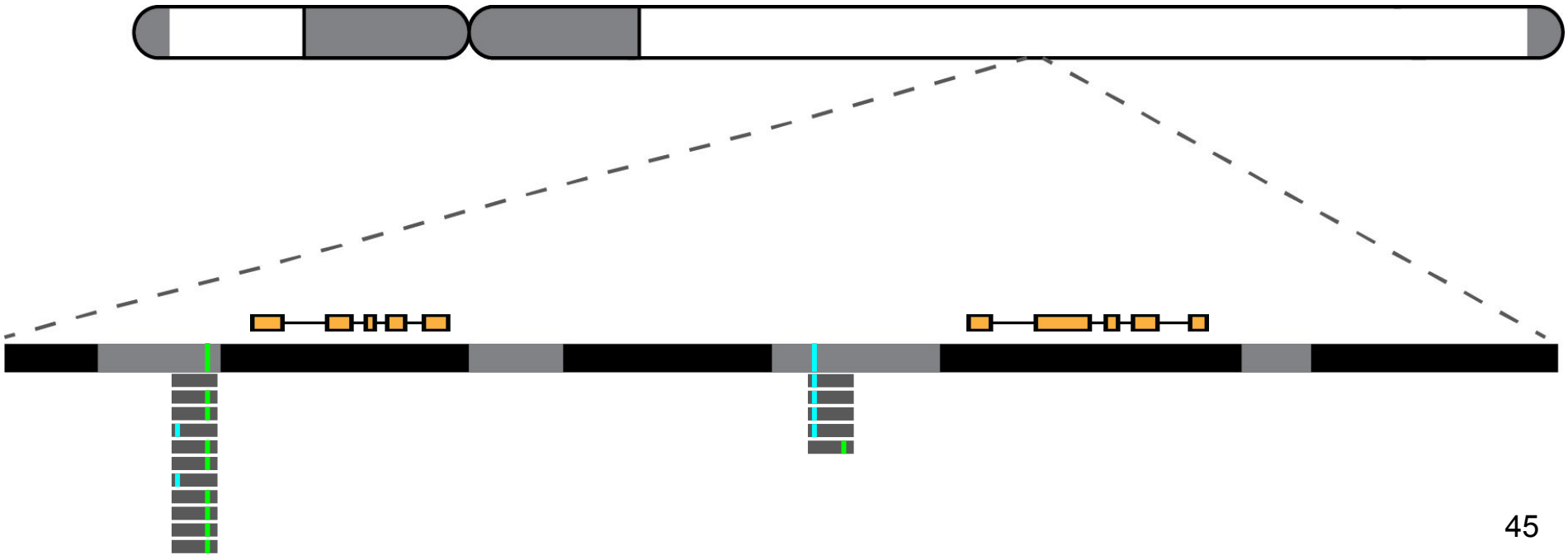
Copy to your home directory

MapTK maptags command

How do you know if maptags ran correctly?



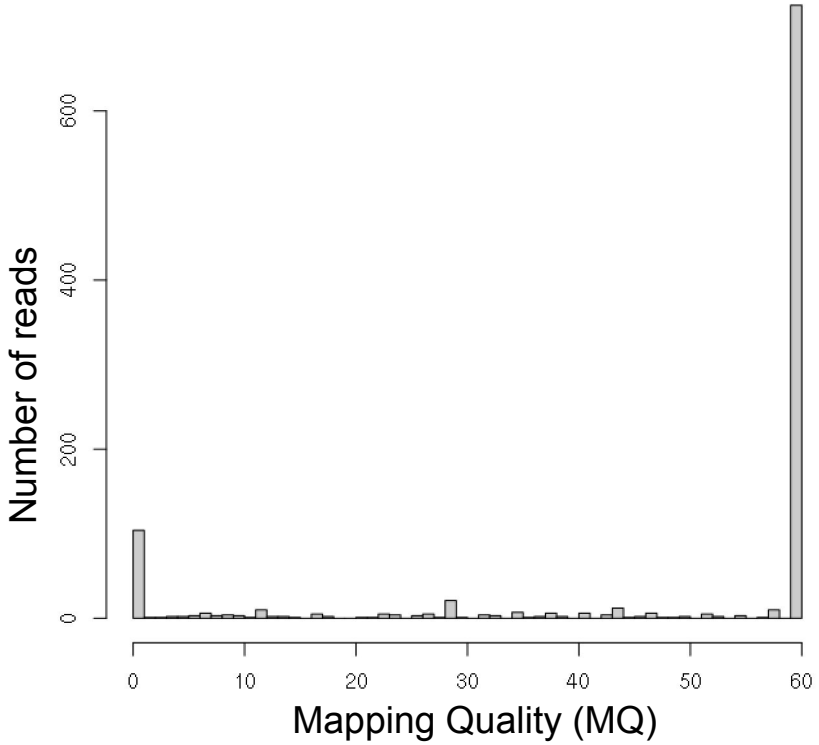
Genomic repeats frustrate proper sequence mapping



Mapping Quality: Scoring mapped sequences

- Range: 0–60
- MQ = 60: Read uniquely mapped
- MQ < 60: Has a sub-optimal mapping
- MQ = 0: read has two equivalently-scored mapped positions
- MQ score affected by fraction of mismatched bases in alignment

$$\text{Phred} = -10 \log_{10}(\text{Prob})$$



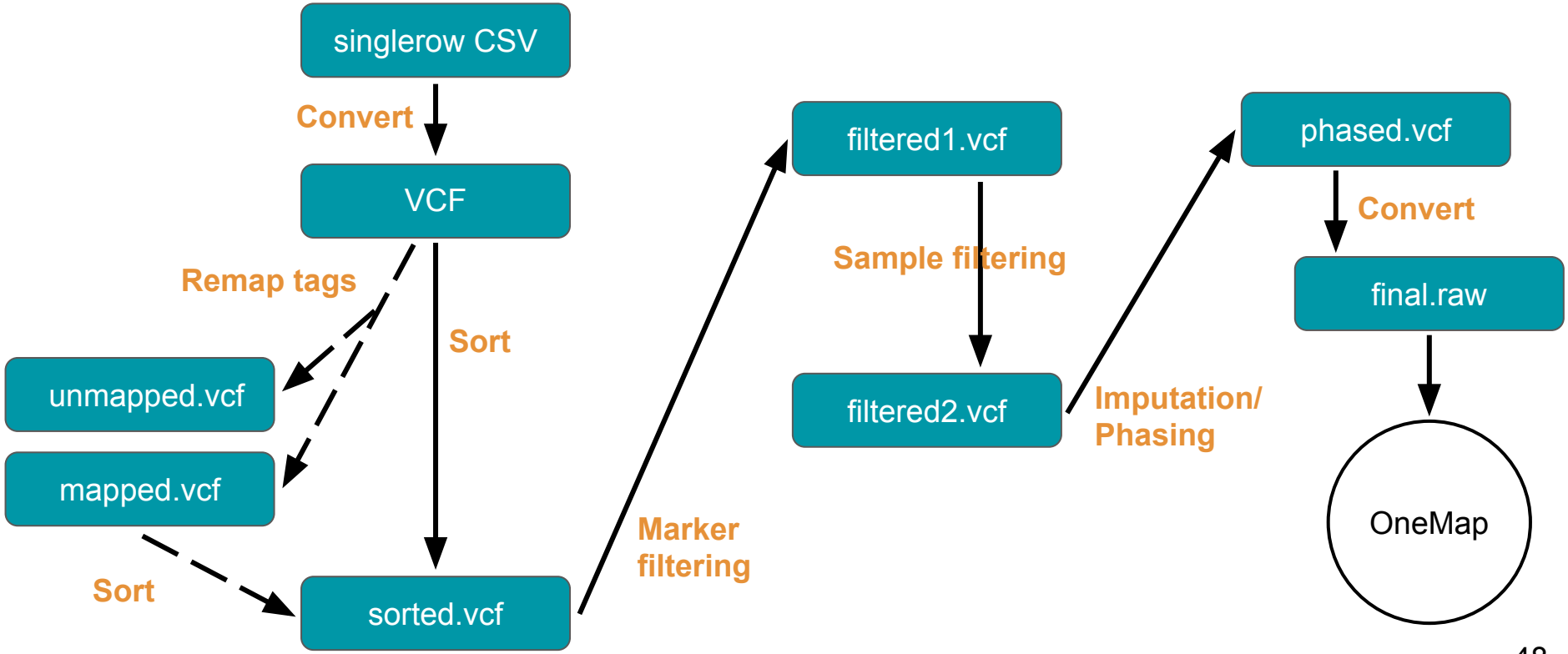
Wednesday, 25 July 2018

- Marker filtering
- Sample Filtering
- Imputation

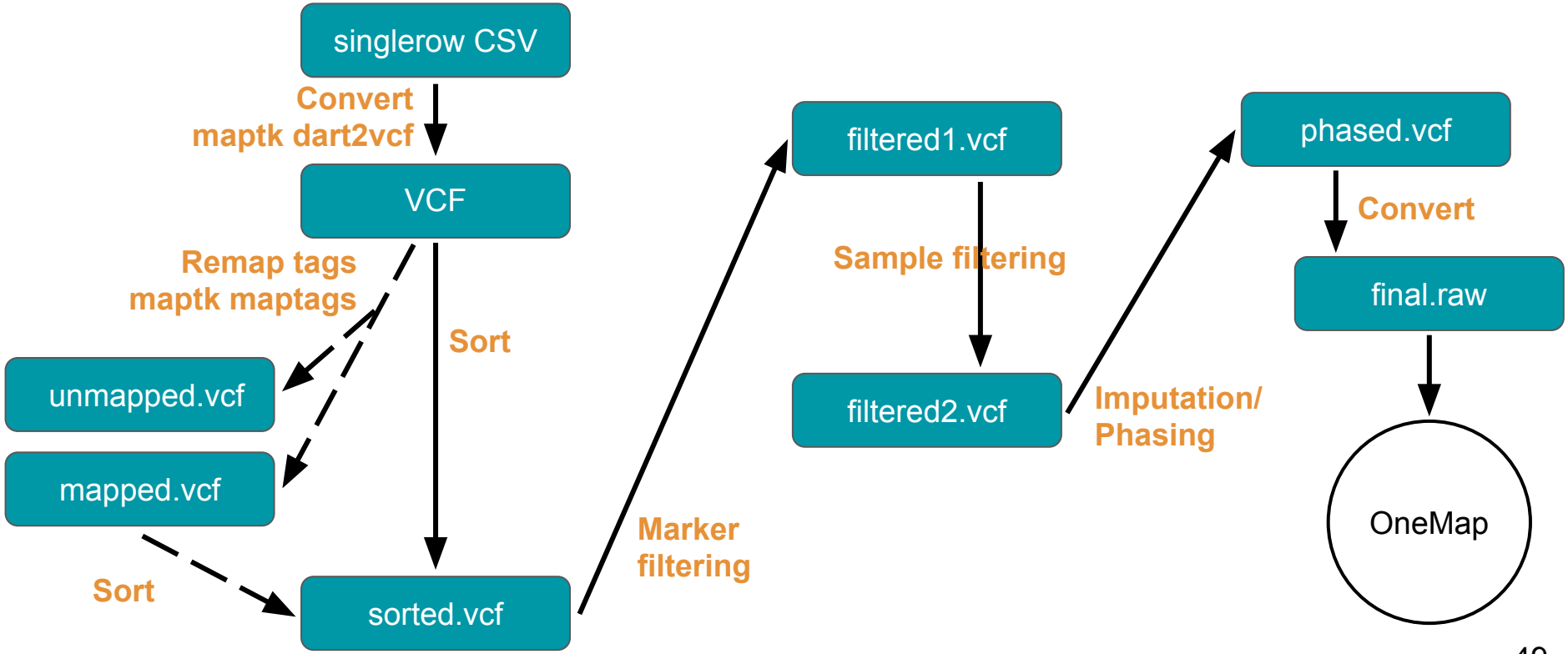
Windows users will need WinSCP today

+ Daily reminder to copy your commands for later reference :-)

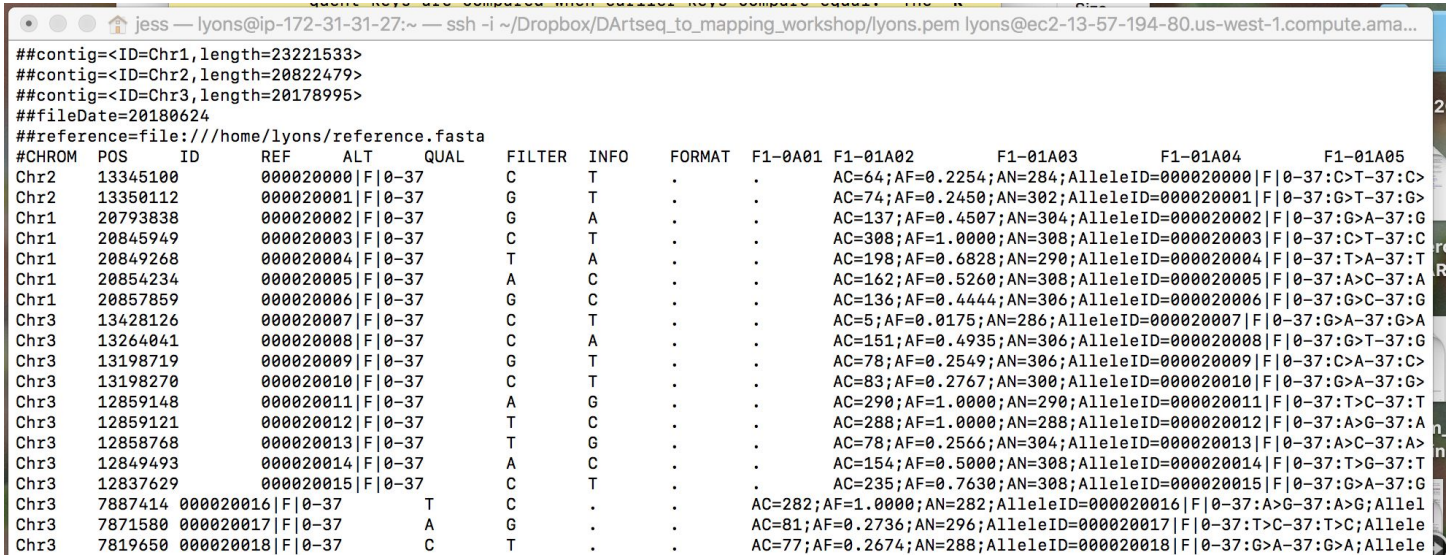
DArTseq to mapping: analysis flow chart



DArTseq to mapping: analysis flow chart



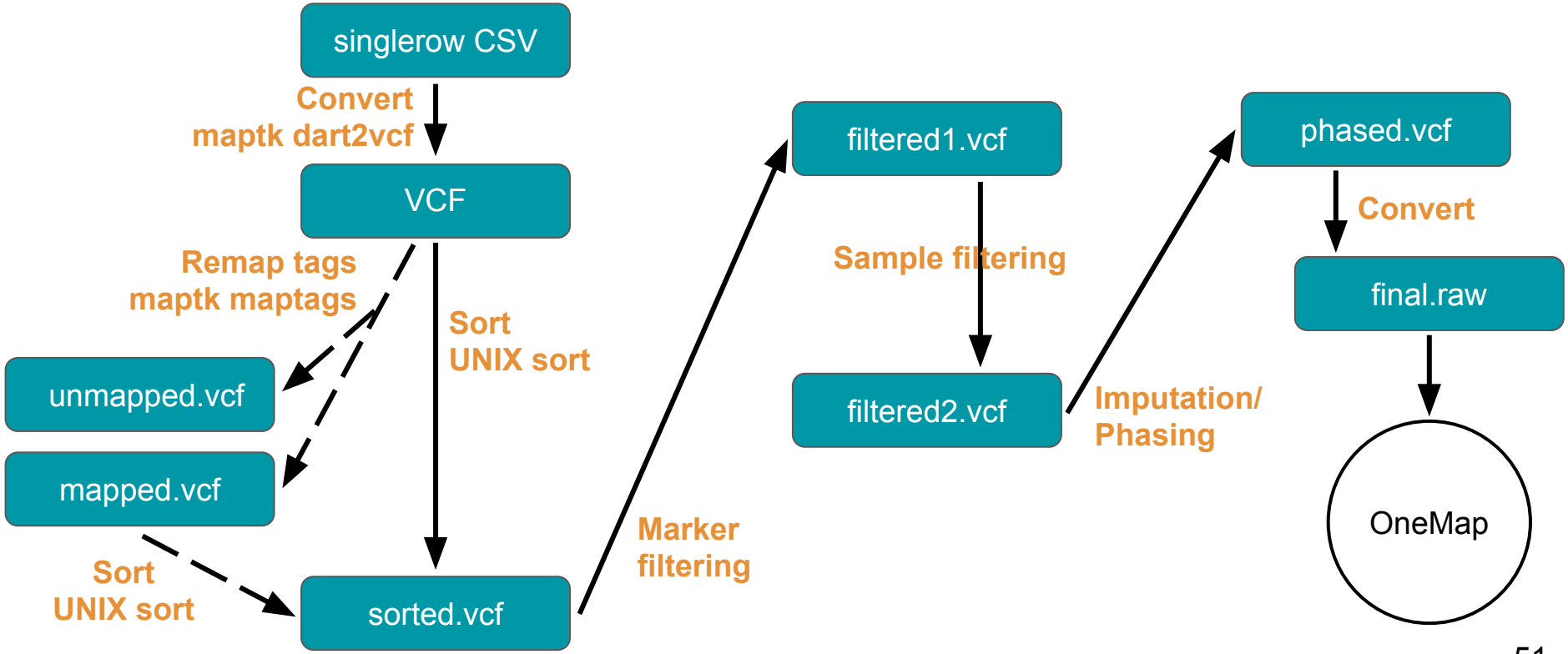
Preparing the (mapped) VCF file for filtering




```
##contig=<ID=Chr1,length=23221533>
##contig=<ID=Chr2,length=20822479>
##contig=<ID=Chr3,length=20178995>
##fileDate=20180624
##reference=file:///home/lyons/reference.fasta
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  F1-0A01 F1-01A02        F1-01A03        F1-01A04        F1-01A05
Chr2    13345100        000020000|F|0-37        C       T       .       .       AC=64;AF=0.2254;AN=284;AlleleID=000020000|F|0-37:C>T-37:C>
Chr2    13350112        000020001|F|0-37        G       T       .       .       AC=74;AF=0.2450;AN=302;AlleleID=000020001|F|0-37:G>T-37:G>
Chr1    20793838        000020002|F|0-37        G       A       .       .       AC=137;AF=0.4507;AN=304;AlleleID=000020002|F|0-37:G>A-37:G
Chr1    20845949        000020003|F|0-37        C       T       .       .       AC=308;AF=1.0000;AN=308;AlleleID=000020003|F|0-37:C>T-37:C
Chr1    20849268        000020004|F|0-37        T       A       .       .       AC=198;AF=0.6828;AN=290;AlleleID=000020004|F|0-37:T>A-37:T
Chr1    20854234        000020005|F|0-37        A       C       .       .       AC=162;AF=0.5260;AN=308;AlleleID=000020005|F|0-37:A>C-37:A
Chr1    20857859        000020006|F|0-37        G       C       .       .       AC=136;AF=0.4444;AN=306;AlleleID=000020006|F|0-37:G>C-37:G
Chr3    13428126        000020007|F|0-37        C       T       .       .       AC=5;AF=0.0175;AN=286;AlleleID=000020007|F|0-37:G>A-37:G>A
Chr3    13264041        000020008|F|0-37        C       A       .       .       AC=151;AF=0.4935;AN=306;AlleleID=000020008|F|0-37:G>T-37:G
Chr3    13198719        000020009|F|0-37        G       T       .       .       AC=78;AF=0.2549;AN=306;AlleleID=000020009|F|0-37:C>A-37:C>
Chr3    13198270        000020010|F|0-37        C       T       .       .       AC=83;AF=0.2767;AN=300;AlleleID=000020010|F|0-37:G>A-37:G>
Chr3    12859148        000020011|F|0-37        A       G       .       .       AC=290;AF=1.0000;AN=290;AlleleID=000020011|F|0-37:T>C-37:T
Chr3    12859121        000020012|F|0-37        T       C       .       .       AC=288;AF=1.0000;AN=288;AlleleID=000020012|F|0-37:A>G-37:A
Chr3    12858768        000020013|F|0-37        T       G       .       .       AC=78;AF=0.2566;AN=304;AlleleID=000020013|F|0-37:A>C-37:A>
Chr3    12849493        000020014|F|0-37        A       C       .       .       AC=154;AF=0.5000;AN=308;AlleleID=000020014|F|0-37:T>G-37:T
Chr3    12837629        000020015|F|0-37        C       T       .       .       AC=235;AF=0.7630;AN=308;AlleleID=000020015|F|0-37:G>A-37:G
Chr3    7887414 000020016|F|0-37        T       C       .       .       AC=282;AF=1.0000;AN=282;AlleleID=000020016|F|0-37:A>G-37:A>G;Allel
Chr3    7871580 000020017|F|0-37        A       G       .       .       AC=81;AF=0.2736;AN=296;AlleleID=000020017|F|0-37:T>C-37:T>C;Allele
Chr3    7819650 000020018|F|0-37        C       T       .       .       AC=77;AF=0.2674;AN=288;AlleleID=000020018|F|0-37:G>A-37:G>A;Allele
```

Why sort?

DArTseq to mapping: analysis flow chart



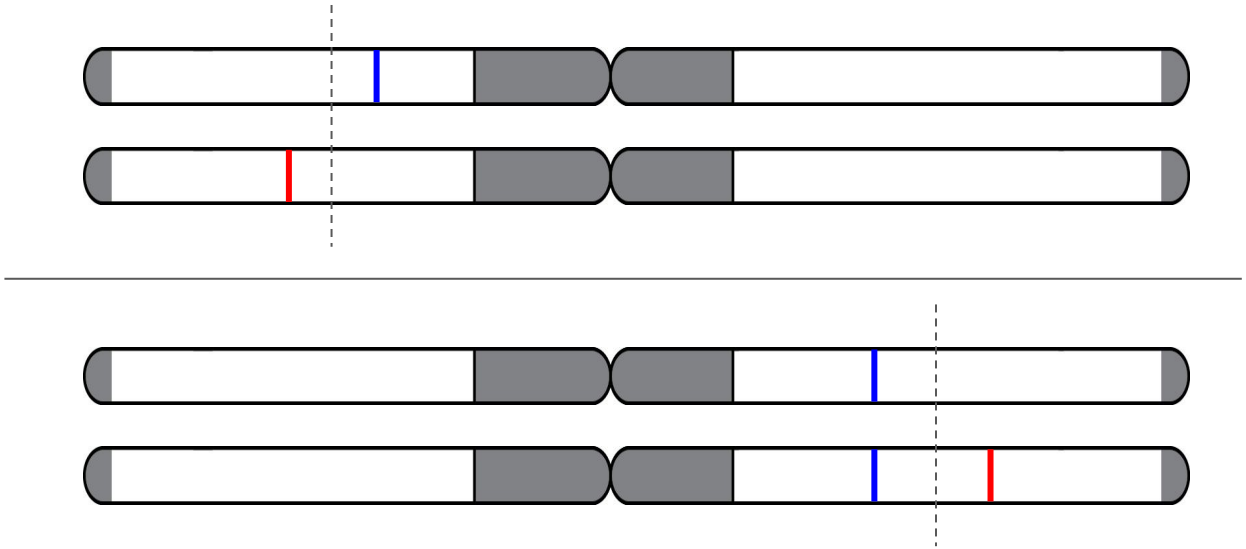
Filtering loci for segregating markers

What markers are useful for F1 linkage mapping?

- Useful markers have two or more alleles segregating in a Mendelian manner
- At least one of the parents must be heterozygous

	A	A
a	Aa	Aa
a	Aa	Aa

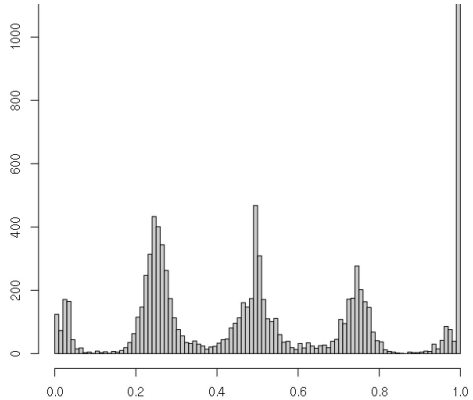
1 Aa



X^2 test for Mendelian segregation

Goodness-of-fit test:

- Expected counts vs. observed counts
- “Test statistics that follow a chi-squared distribution arise from an assumption of independent normally distributed data” – Wikipedia



	A	a
A	AA	Aa
a	Aa	aa

1 AA : 2 Aa : 1 aa

	A	a
A	AA	Aa
A	AA	Aa

1 AA : 1 Aa

	a	a
A	Aa	Aa
a	aa	aa

1 Aa : 1 aa

MapTK chif1 command

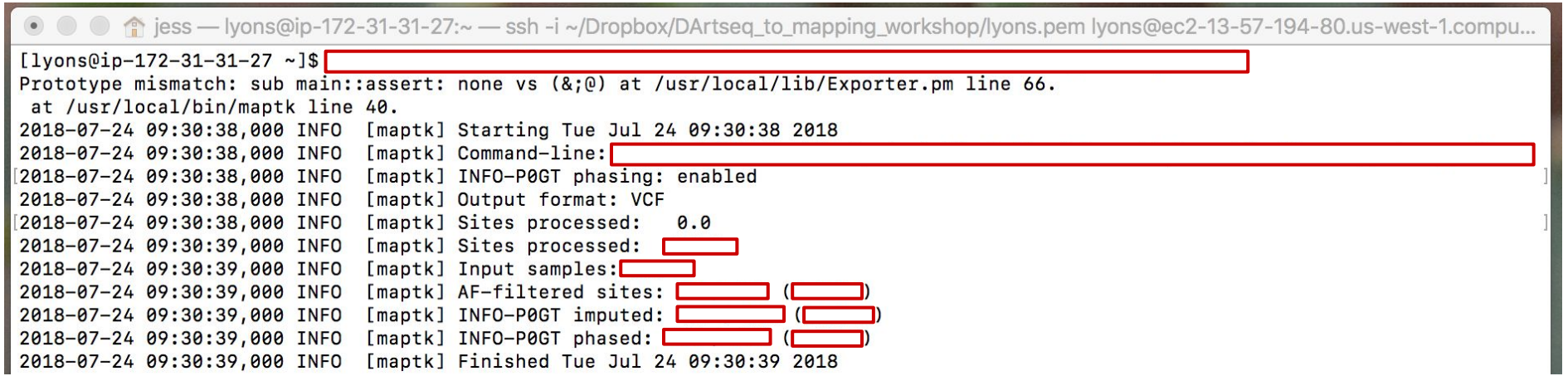
```
jess — lyons@ip-172-31-31-27:~ — ssh -i ~/Dropbox/DArtseq_to_mapping_workshop/lyons.per
[lyons@ip-172-31-31-27 ~]$ maptk chif1
Prototype mismatch: sub main::assert: none vs (&;@) at /usr/local/lib/Exporter.pm line 66.
 at /usr/local/bin/maptk line 40.

Usage:   maptk chif1 [options] <in.vcf>

Options: -o <file>      Write output to FILE [stdout]
         -A <ufloat>    Max. phred allele freq. best-fit P-val [50.0]
         -G <ufloat>    Max. phred genotype freq. best-fit P-val [9999.0]
         -P <P1>[,<P2>] Parental IDs for the input population (recommended)
         -e <uint>      Phred-scaled genotype-call error (see Notes 3) [30]
         -a             Perform chi-sqr on allele depths (AD) field [GT]
         -F             Exclude sites for which scores cannot be applied
         -p             Force pseudo-testcross markers only
         -S             Silence/disable verbose reporting
         -t             Write output as a table (see Note 4) [VCF]
         -h             This help document
```

```
Notes:

  1. This script applies a chi-squared goodness-of-fit test for Mendelian
     genotype frequencies, for a determined allele frequency class
     (3:1, 1:1, etc). The allele frequency class is selected within a
     goodness-of-fit tolerance threshold set by the '-A' option. The chi-
     squared values calculated here have been compared to those calculated
     by JoinMap4 and there is (almost) complete agreement.

  2. Parental genotypes are inferred and stored by the 'P0GT' key in the
     INFO field. If the IDs of the parents are passed via the '-P' option,
     and the samples are included in the input VCF, an attempt to orient
     the inferred genotype calls with respect to parent (determined by the
     relative order of the parental samples listed in the VCf header) is
     made. If successful, a 'P0PHASED' key is applied to the INFO field.

  3. Requires at least five diploid F1 samples (ten or more recommended).

  4. Yates' correction for continuity is applied to sites with less than
     ten observed chromosomes, and no calculations are performed on sites
     with less than five observed chromosomes.

  5. Value passed to the '-e' argument must be a positive integer. It is
     recommended to set '-e' to the minimum GQ value use for filtering.

  6. When enabling the '-a' flag, and the input data are at low-coverage,
     it is necessary to include the parental IDs via '-P' to calculate the
     P-value statistics accurately.

  7. The default output format is VCF, the '-t' option outputs a tab-
     delimited table (list of sub-field values) of relevant statistics
     a-la `vcftools --get-INFO`.
```

Note: this command requires all sites to have less than 50% missing data.
VCFtools can help you remove sites with >50% missing data.

MapTK chif1

How do you know if chif1 ran correctly?

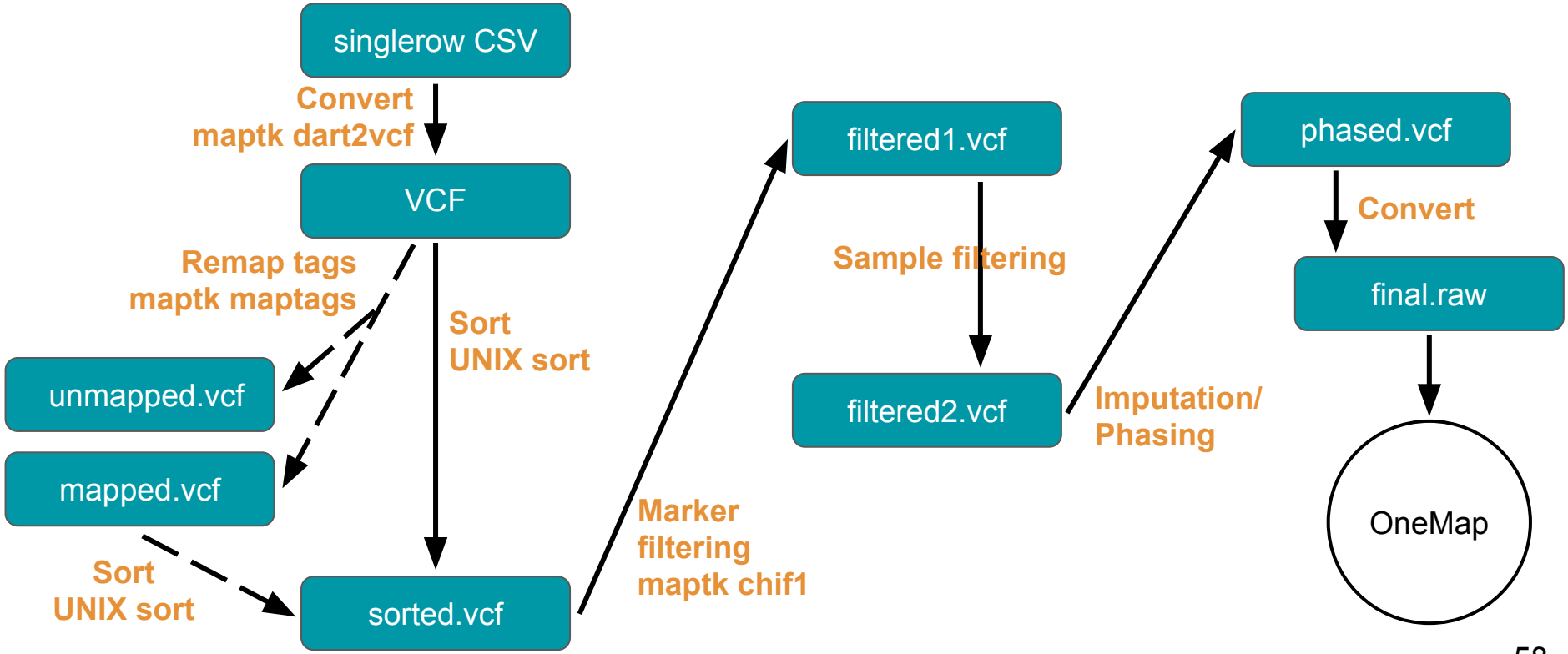


MapTK chif1: test statistics

- **F1AFP**: Phred-scaled P-value for X^2 goodness-of-fit test on allele frequencies
- **F1GTP**: Phred-scaled P-value for X^2 goodness-of-fit test on genotype frequencies
- **F1X2**: X^2 value for goodness-of-fit test on locus allele frequencies
- **P0GT**: The Inferred parental genotypes, ordered alphabetically by Sample ID
- **P0PHASED**: Boolean tag indicating the locus passed the specified X^2 test (and, if the option was enabled, the parental genotypes were able to be inferred)

$$\text{Phred} = -10 \log_{10}(\text{Prob})$$

DArTseq to mapping: analysis flow chart



Filtering samples for full-sib progeny

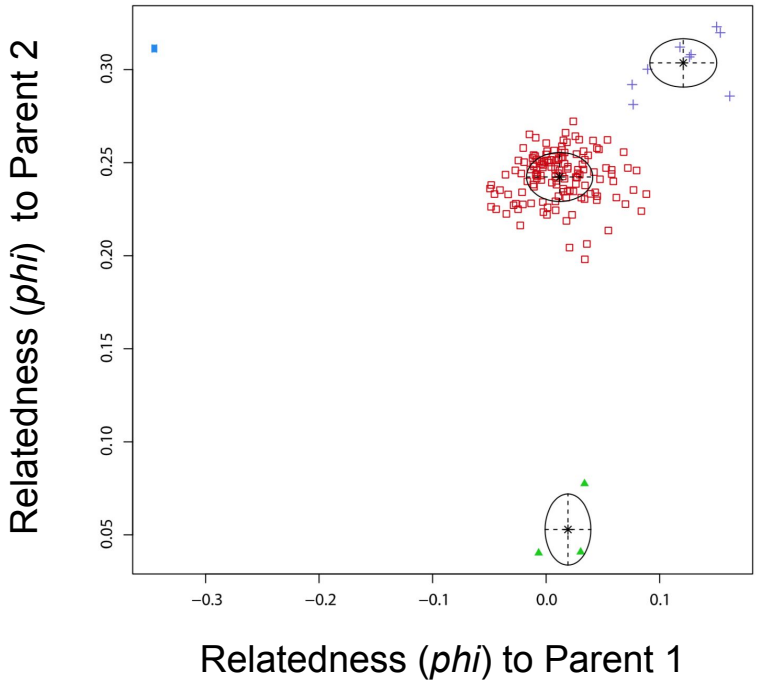
Sources of error in crossing experiments

1. Half-sibs
2. Selfs
3. Volunteer seedlings
4. Human error

Filtering samples for full-sib progeny

- Calculate relatedness* of putative offspring w.r.t. the parents.
- Relatedness *phi* about half of relatedness expected:

relationship	**Expected**	**Reported *phi***
clonal	1.0	0.5
parent–child	0.5	0.25
off-type	0.0	0.0



*Manichaikul. 2010. Bioinfo. doi: 10.1093/bioinformatics/btq559
ICGMC. 2015. G3 Journal. doi: 10.1534/g3.114.015008
https://bitbucket.org/rokhsar-lab/gbs-analysis/src/master/

Filtering samples for full-sib progeny

vcftools relatedness2

➔ Remember to check where it wants a file, and where a prefix

maptk mclust

➔ Check for files ending in .dat and .dat.pdf

➔ **Note**: mclust runs R to produce the plots, if the Mclust R package is not installed, R will not print the plots

Filtering samples for full-sib progeny

Download the plots to your laptop:

Mac/Linux:

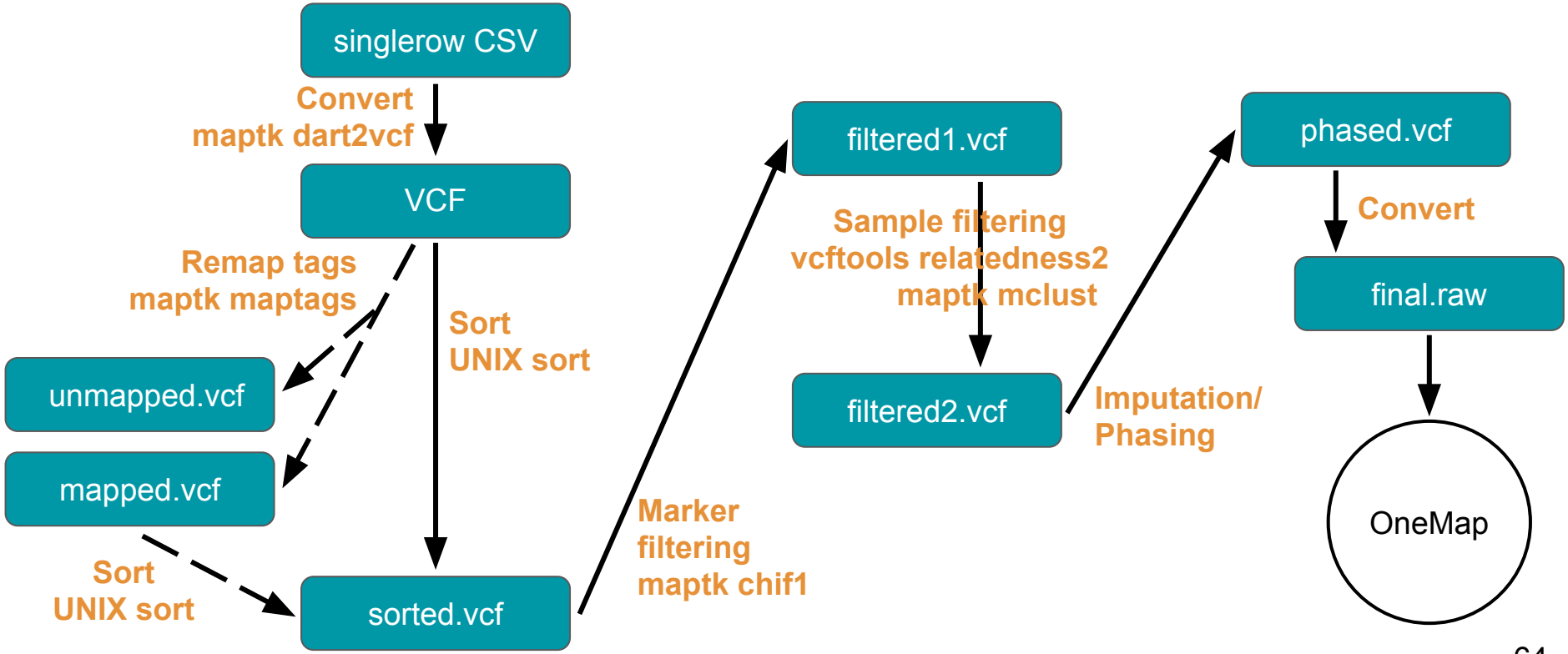
```
scp -i ~/fn.pem \
fn@ec2-13-57-194-80.us-west-1.compute.amazonaws.com:/home/fn/mclust.dat.pdf .
```

Where “fn” is your family name

Windows:

Download and install WinSCP

DArTseq to mapping: analysis flow chart



Imputing genotypes

Imputation and phasing

What is imputation:

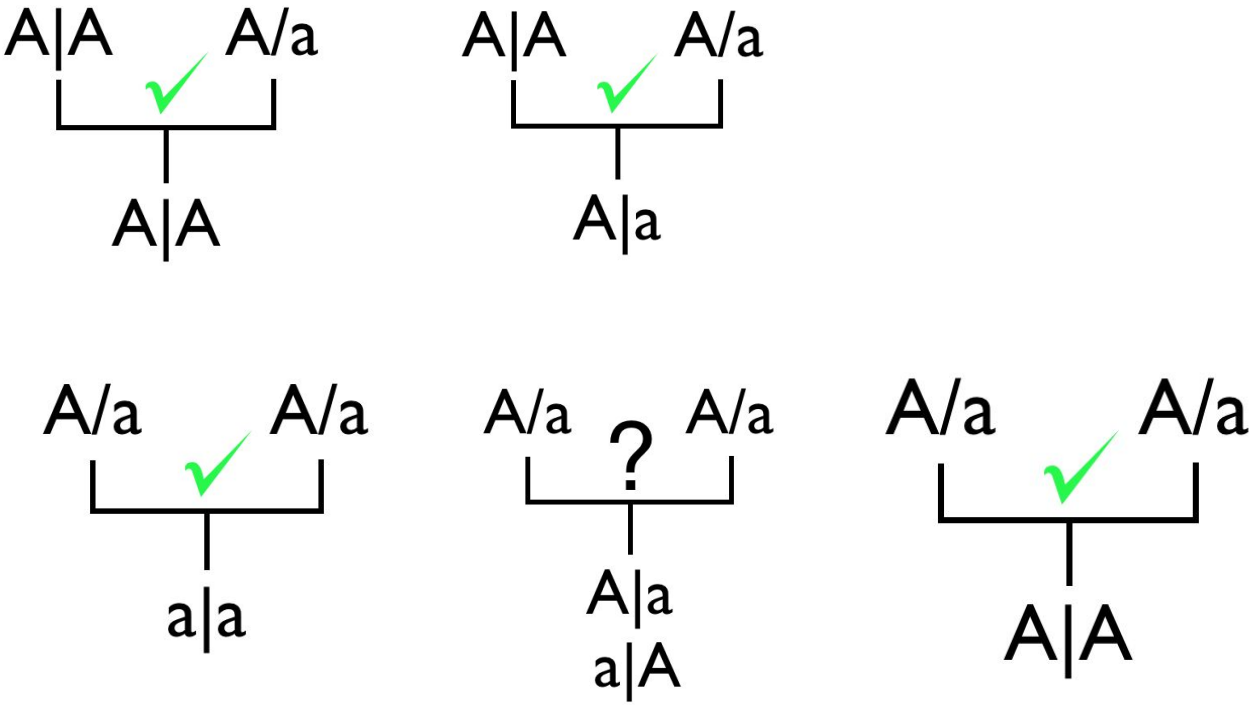
➔ Filling in missing values (*i.e.*, genotypes) with data inferred from patterns/correlations in the dataset.

What is phasing?

➔ Assigning alleles to their maternal or paternal chromosome.

phaseF1 script

Imputation and phasing



Imputation and phasing

Mapping algorithms are sensitive to missing data, but *more* sensitive to incorrect data.

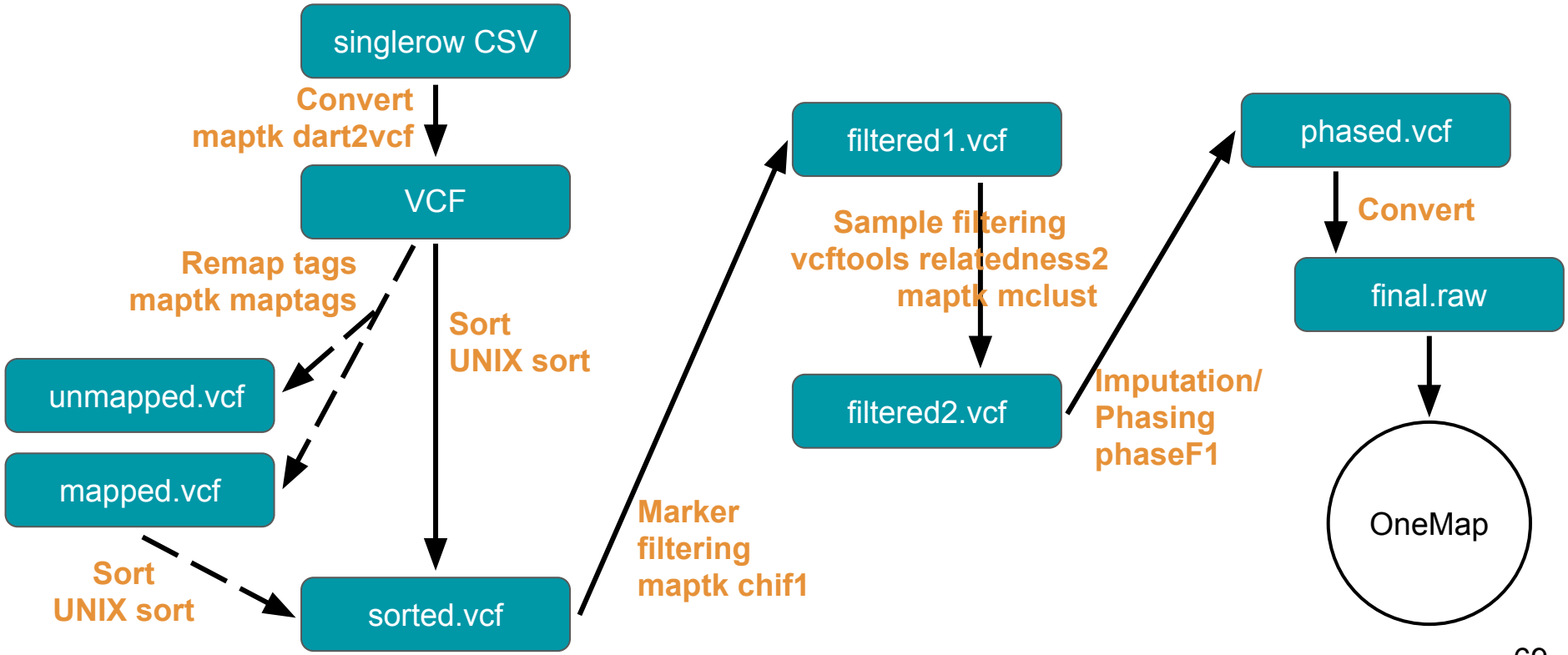
When to impute? When not?

- With OneMap we'll be able to judge correctness of imputation.
- Iteration/parameter sweeps.

Imputation/phasing software:

1. Beagle
2. MACH
3. IMPUTE2

DArTseq to mapping: analysis flow chart



Thursday, 26 July

- Convert (remapped), sorted, filtered, (phased) .vcf to OneMap input file (.raw format)
- Using R
- Linkage mapping with OneMap

Cygwin: a LINUX-like environment for Windows

https://www.cygwin.com/

https://cygwin.com/faq.html

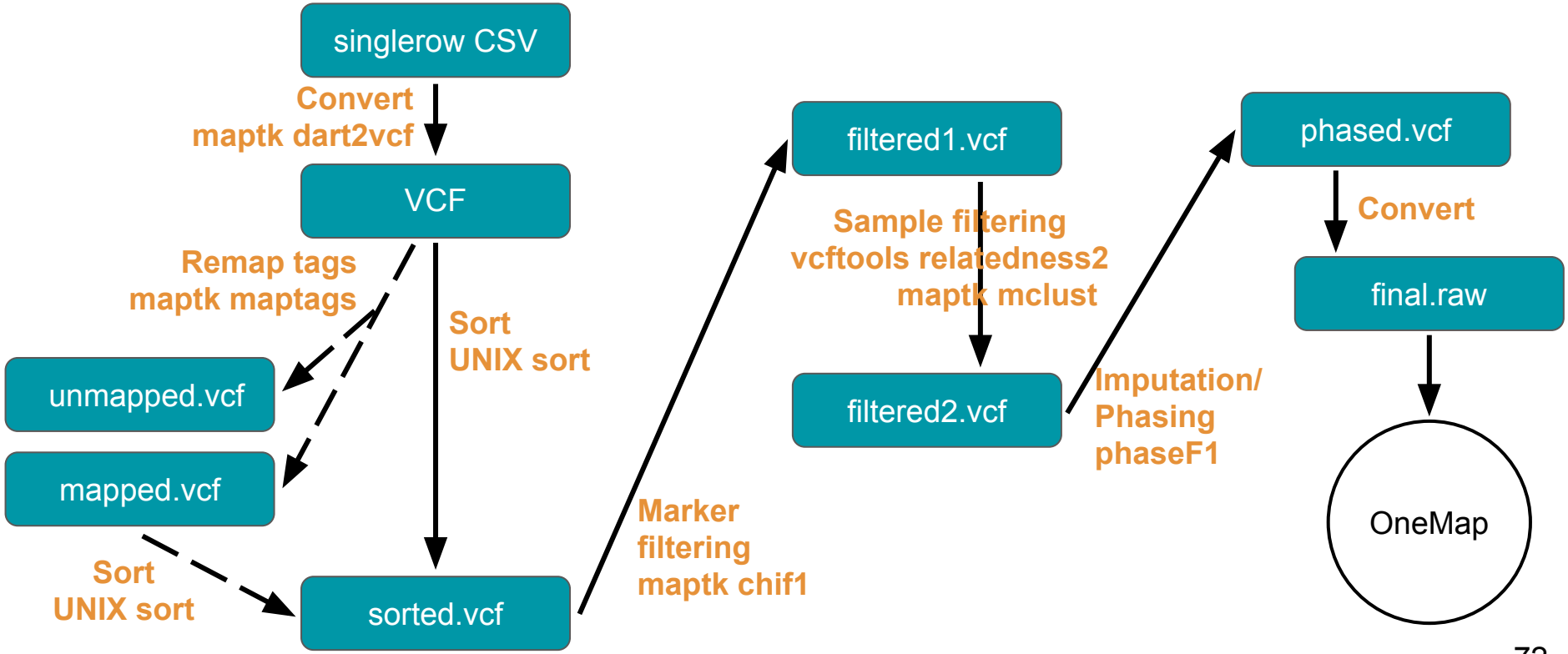
https://cygwin.com/cygwin-ug-net/cygwin-ug-net.html

for e.g. vcftools, you will have to compile it on your computer

any arguments that reference the filesystem must be in Windows (or DOS) format or translated

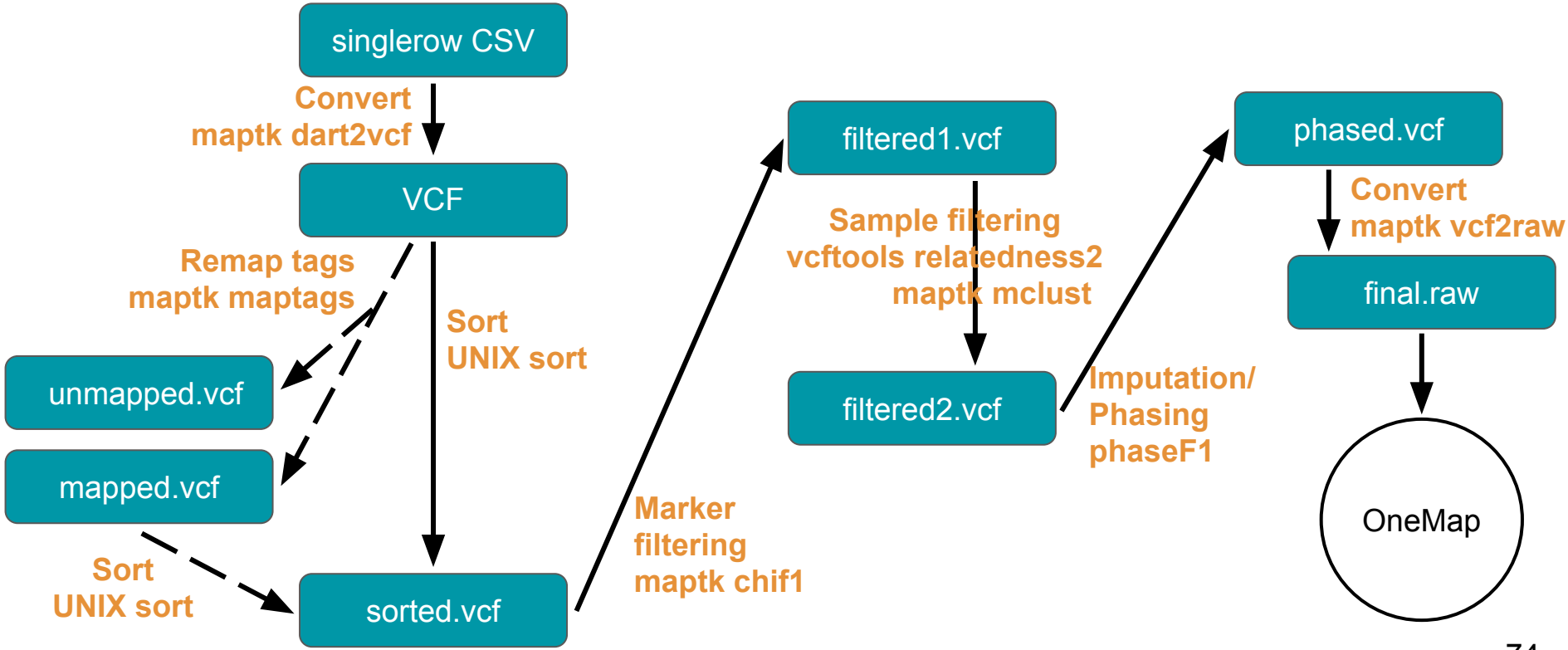
➔ `cygpath` utility (https://cygwin.com/cygwin-ug-net/using-effectively.html)

DArTseq to mapping: analysis flow chart



maptk vcf2raw

DArTseq to mapping: analysis flow chart



RRRRR matey

R statistical programming

Very good linear algebraic support (vectors, matrices)

Publication-quality figures

Interactive interpreter and Rscript

Portable on all operating systems

Goals: be able to use R/OneMap and understand what you are doing

Teaching yourself R

Comments:

```
# Write yourself lots of notes
```

Learn more about a function/package:

```
?thing
```

Rstudio, a GUI for R:
https://www.r-project.org
https://www.rstudio.com

R: Variables, Math, Logic

Valid variable names may contain A–Z, a–z, 0–9 characters, as well as "_" and "."
(Cannot begin with a number)

Assignment:
`x = 5`
`x <- 5`

Logic:
And: `x & y`
Or: `x | y`
Negate: `!x`

Math operations:
Addition: `x + y`
Subtraction: `x - y`
Multiplication: `x * y`
Division: `x / y`
Power: `x^y or x**y`
Modulo: `x %% y`
Log: `log(x)`
Square root: `sqrt(x)`

Comparison:
Equivalent to: `x == y`
Not equivalent: `x != y`
Less-than: `x < y`
Less-than-or-equal-to: `x <= y`
Greater-than: `x > y`
Greater-than-or-equal-to: `x >= y`

R: Objects

Variables: are “objects” (containers for data):

```
object = data
```

Belong to a class/have “type” (character, numeric, vector, factor, etc.):

```
class(object)
```

May contain attributes:

```
names(object) # what attributes are there?
object$attribute  # use the attribute as a variable
```

variable types and coercion

NULL: `NULL`

numeric: `-1, 0, 1, -1.0, 0.05, 1e-6`

character: `“string”, “1”, “FALSE”`

vector: `1D sequence`

```
V = c(1, 5, 10);
value = V[index];
V[index] = value
```

matrix: 2D table (typically numeric)

```
M = matrix(1:9, nrow=3, ncol=3)
value = M[i,j]
M[i,j] = value
```

data.frame: Excel-like table

```
D = data.frame(field1=1:9, field2=11:19)
value = D[i,j]; D[i,j] = value
value = D$field1; D$field1 = value
```

logical: `TRUE, FALSE, NA`

list: Multi-D table

```
L = list(field1=1:9, field2=11:19)
value = L[i,j]; L[i,j] = value
value = L$field1; L$field1 = value
L = list(1:9, 11:19)
value = L[[i]][j]; L[[i]][j] = value
```

Testing for type:

`is.`***`type`***`(object)`

Coerce to type:

`new.type = as.`***`type`***`(old.type)`

Conditional statements

Template:

```
if (test1) {
    execute this line

} else if (test2) {
    execute this line

} else { # defaults to this
    execute this line
}
```

Template:

```
if (test1) {
    if (test2) {
        execute this line
    }
    execute this line

} else { # defaults to this
    execute this line
}
```

Conditional statements

Example:

```
i = 0
if (i < 10) {
    print(“< 10”)

} else { # defaults to this
    print(“>= 10”)
}
```

Example:

```
i = 5
if ((0 <= i) & (i <= 5)) {
    print(“in range [0, 5]”)

} else if (i < 0){
    print(“negative”)

} else {
    print(“greater than 5”)
}
```

for and while

Template:

```
for (variable in sequence) {
    execute this code
}


while (condition-is-true) {
    execute this code
}
```

Example:

```
for (i in 1:10) {
    print(i)
}

j = 0
while (j < 15) {
    print(j)
    j = j + 1
}
```

functions

Template:

```
f = function(x=default) {
    compute something
    return(result)
}

y = f(x)
```

Example:

```
f = function(x, add=FALSE) {
    if (add) {
        return(x + 1)
    } else {
        return(x - 1)
    }
}

y = f(x)
```

Other useful functions

`plot`	Plot X, Y data	`read.csv`	Read CSV file
`hist`	Plot a histogram	`write.csv`	Write CSV file
`seq`	Create a sequence	`length`	Calc. length of object
`pdf`	Open a PDF device for writing	`names`	Get attribute names
`dev.off`	Close device to save	`append`	Append item to vector
`paste`	Paste together data to make a string	`setwd`	Set working dir
`sort`	Sort a vector, list, etc.	`getwd`	Get working dir
`read.table`	Read in a space-/tab-delimited file	`ls`	Show defined variables
`write.table`	Write a table to file	`summary`	Summarize object data
`install.packages`	Install R packages from CRAN	`lapply`	Iterate and create list

R exercises

1. Use R to calculate the `sum` of 5 and 8 and assign the output to a variable `x`.
2. Use R to determine if `x` is divisible by 2 (hint: see modulo operator).
3. Use `is.integer` type test to determine whether `x` is an integer. If not, what type is it?
4. Use a conditional statement to test whether x is between 12 AND 13
5. Use a conditional statement to test whether x is equivalent to 12 OR 13
6. Use the `seq` function to output even numbers between 0 and 50
7. Combine your solution to question 6 with a `for` loop to print odd numbers.
8. Create a function to calculate the difference between two numbers.
9. Use `data(cars)` to load some example data. The `cars` variable becomes defined. What columns does it contain? Plot a histogram of the first column. What is the `mean`? Use `sd` to calculate the standard deviation.

R exercises

10. To save a plot to PDF, you must first call `pdf("name_of.pdf")`, then the plotting function of interest, then `dev.off()` to close the file. Save a PDF of your histogram from question 9.

R exercises Solutions

1. Use R to calculate the `sum` of 5 and 8 and assign the output to a variable `x`.

   ```
   x = 5 + 8 OR x <- 5 + 8 OR x = sum(5, 8) OR x <- sum(5, 8)
   ```

2. Use R to determine if `x` is divisible by 2 (hint: see modulo operator).

   ```
   (x %% 2) == 0  # returns 'FALSE'
   ```

3. Use `is.integer` type test to determine whether `x` is an integer. If not, what type is it?

   ```
   is.integer(x)  # returns 'FALSE'
   class(x)  # returns 'numeric'
   ```

R exercises Solutions

4. Use a conditional statement to test whether x is between 12 AND 13

It was not specified whether the comparisons should be inclusive or not, it is up to the programmer to decide what is best for his/her code. Correct answers will have used > or >= for the first condition, and < or <= for the second condition below. In order to require *both* conditions to be `TRUE`, we use the & operator. If `"True!"` is not printed to the terminal, then *one or both* of the conditional statements must not be TRUE:

```
if ((x >= 12) & (x <= 13)) {
   print("True!")
}
```

R exercises Solutions

5. Use a conditional statement to test whether x is equivalent to 12 OR 13

   In order to determine if one *OR* both conditions are `TRUE`, we use the | operator. If `"True!"` is not printed to the terminal, then *none* of the conditions are TRUE.:

```
if ((x == 12) | (x == 13)) {
   print("True!")
}
```

   NOTE: It is incorrect to use a single = sign for *equivalence*, you must use two (==). A single = sign means *assignment* not equivalence!

6. Use the `seq` function to output even numbers between 0 and 50

   `seq(from=0, to=50, by=2)` OR `seq(from=2, to=50, by=2)` are both correct.

R exercises Solutions

7. Combine your solution to question 6 with a `for` loop to print odd numbers.

   It would be correct to start with either solution to question 6, but the best is the first, *i.e.* `seq(from=0, to=50, by=2)`. Put it together with the `for` loop to get:

```
for (i in seq(from=0, to=50, by=2)) {
   print(i + 1)
}
```

8. Create a function to calculate the difference between two numbers.

   Use the `function` declarator to assign a variable name (one that is suggestive of its function is best practice) to compute the difference. We must first define the variables to be used (in parentheses) and the code it executes (in squiggly brackets):

```
diff = function (x, y) { return(x - y) }
diff(5, 2)  # returns 3
```

R exercises Solutions

9. Use `data(cars)` to load some example data. The `cars` variable becomes defined. What columns does it contain? Plot a histogram of the first column. What is the `mean`? Use `sd` to calculate the standard deviation.

```
data(cars)  # loads the cars data
names(cars)  # returns column/attribute names in the object: “speed” and “dist”
hist(cars[,1])  # plots a histogram in a new window
mean(cars[,1])  # 15.4
sd(cars[,1])  # 5.287644
```

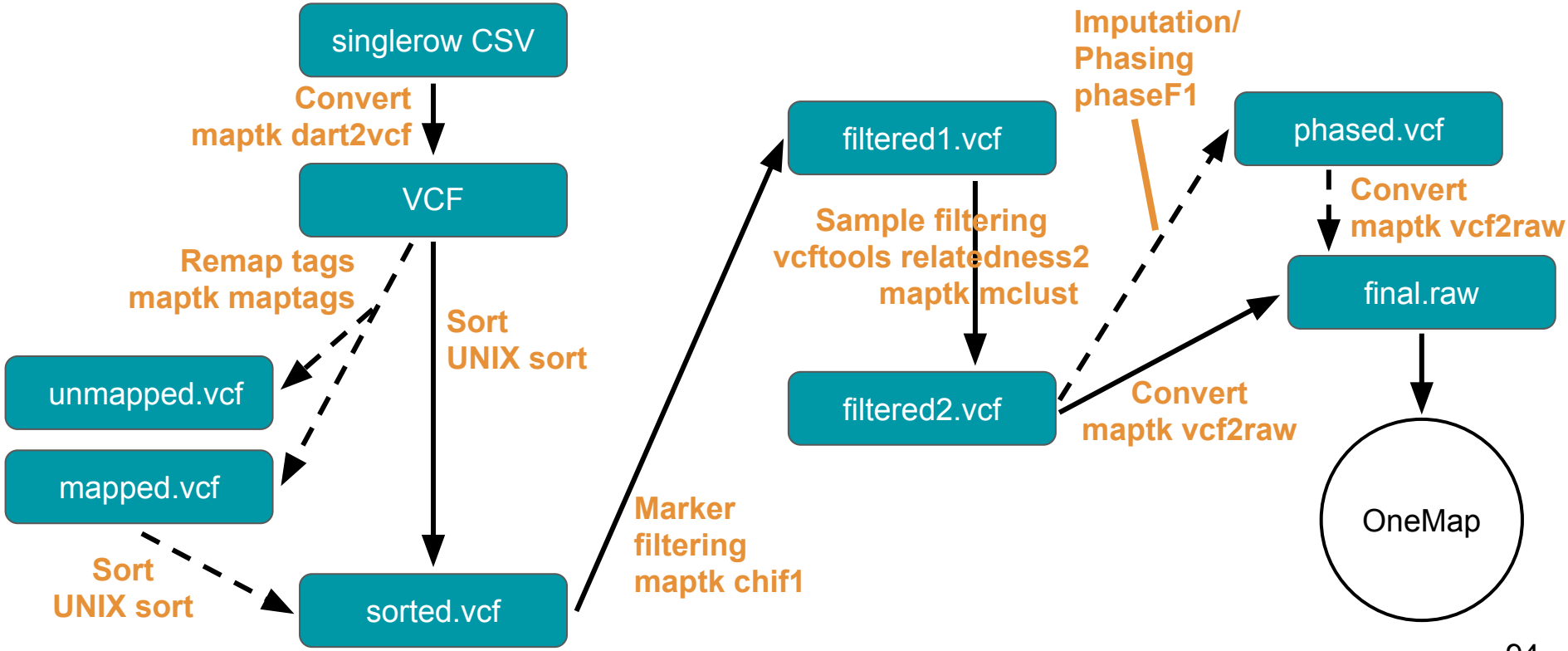
R exercises Solutions

10. To save a plot to PDF, you must first call `pdf("name_of.pdf")`, then the plotting function of interest, then `dev.off()` to close the file. Save a PDF of your histogram from question 9.

```
pdf("cars_speed.pdf")  # opens a blank PDF file called "cars_speed.pdf"
hist(cars$speed)  # sends the histogram to the PDF
dev.off()  # writes all data to the PDF and closes it
```

Genetic linkage map estimation

DArTseq to mapping: analysis flow chart



Linkage mapping with OneMap

Why OneMap?

- Free!
- Supports many different cross types (F1, F2, RIL, BC)
- Parallelizable
- Very flexible (not only one analysis)
- Portable to any system

Loading the OneMap package and reading data

To load the OneMap package, use `require()`

```
require(onemap)
```

When you load mapping data, cross type is read from the file

```
cross = read_onemap("/path/to/dir", "F1.raw")
```

Checking your data quality

Plots color genotype matrix

```
plot(cross)
```

Was our chi-squared test stringent enough?

```
plot(test_segregation(cross))
```

OneMap can give you a recommended LOD score

```
lod.suggested = suggest_lod(cross)
```

$$LOD = Z = \log_{10} \frac{\text{probability of birth sequence with a given linkage value}}{\text{probability of birth sequence with no linkage}} = \log_{10} \frac{(1-\theta)^{NR} \times \theta^{R}}{0.5^{(NR+R)}}$$

https://en.wikipedia.org/wiki/Genetic_linkage

Friday, 27 July

- *Server disk-space issue fixed!*
- Linkage mapping with OneMap
- Class photograph ~10:45

See `HandsOnDArT/lecture_notes_and_exercises/onemap_notes.txt` for a detailed walk-through

Getting set up for R and OneMap

- To run R and OneMap on your computer, you should have:
  - R
  - OneMap
    - Ideal: https://github.com/augusto-garcia/onemap dev version
    - Also OK: `install.packages("onemap", dependencies=TRUE)`
  - **Mac users**:
    - Xcode (download from App Store)

- If all else fails, run on the server, and download plots to look at them

Loading the OneMap package and reading data

To load the OneMap package, use `require()`

```
require(onemap)
```

When you load mapping data, cross type is read from the file

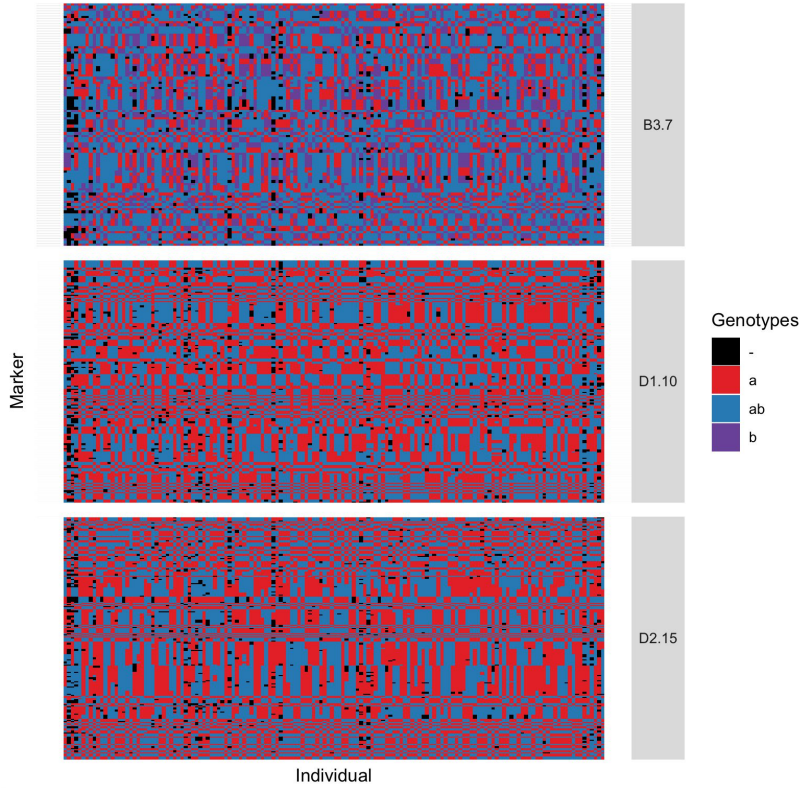
```
cross = read_onemap("/path/to/dir", "F1.raw")
```

Checking your data quality

Plot colorized genotype matrix:
`plot(cross, all=FALSE)`

Assess consistency of genotype patterns within colored blocks.

A pixel of a color in the block of another color is indicative of a genotyping error. How pervasive is it?

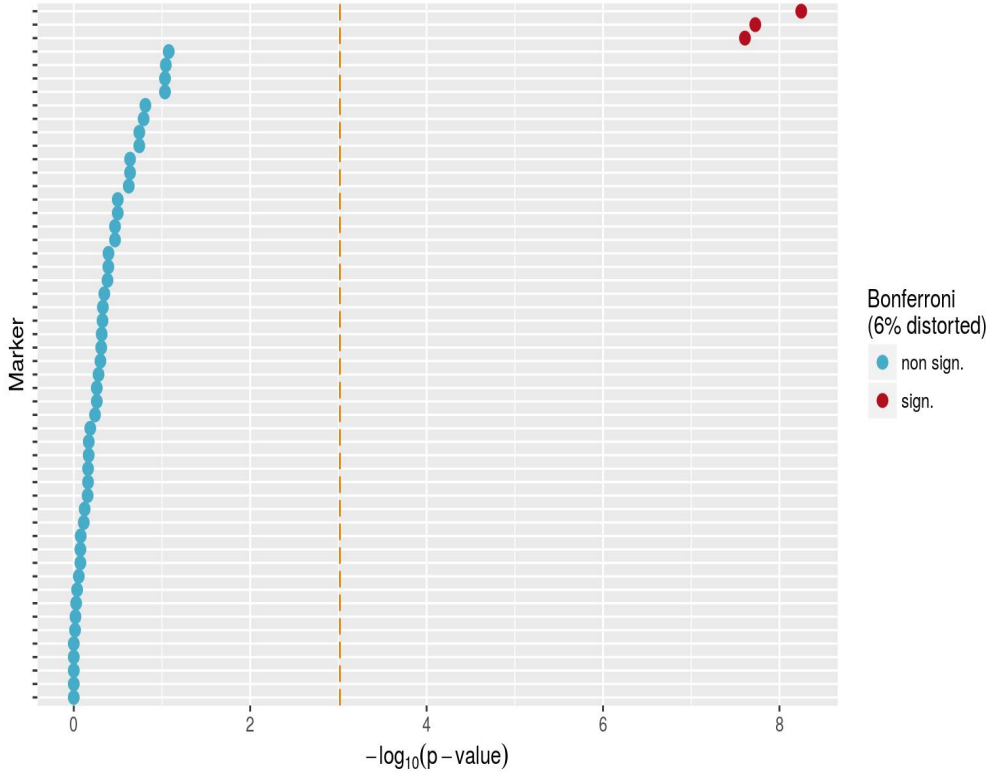


Checking your data quality

Was our chi-squared test stringent enough?

```
plot(test_segregation(cross))
```

Best markers are on the left of the orange vertical line; the presence of markers on the right suggests your X^2 test threshold should be more stringent.



Linkage mapping

Three critical parameters:

Calculating linkage groups

1. Minimum LOD threshold (`LOD`)
2. Maximum recombination fraction (`max.rf`)

Estimating marker order

3. Number of initial markers (`n.init`)

Calculating linkage groups

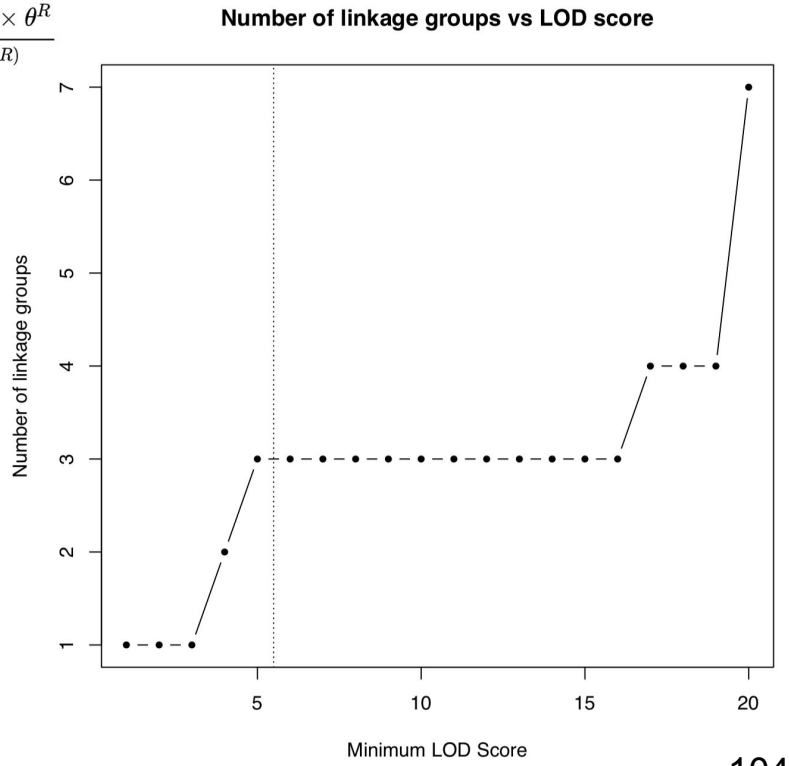
$$LOD = Z = \log_{10} \frac{\text{probability of birth sequence with a given linkage value}}{\text{probability of birth sequence with no linkage}} = \log_{10} \frac{(1-\theta)^{NR} \times \theta^{R}}{0.5^{(NR+R)}}$$

OneMap can recommend a LOD score:

```
lod.suggested = suggest_lod(cross)
```

It is always good practice to check that it is a valid suggestion.

➔ Calculate the number of linkage groups over a range of minimum LOD value and look for a plateau of stability in the plot, indicating strong/stable linkage groupings.



https://en.wikipedia.org/wiki/Genetic_linkage

Calculating linkage groups

We must calculate the genetic distance between every pair-wise combination of markers to estimate the strength of linkage:

```
recombination.fractions = rf_2pts(cross, LOD=0, max.rf=0.5)
```

Select which markers we want to group (all):

```
markers = make_seq(recombination.fractions, "all")
```

Partition markers with strong linkage into linkage groups:

```
linkage.groups = group(markers, LOD=min.lod, max.rf=0.5)
```

Estimating marker order

Given a group, produce a linear order of markers, representing the chromosome (Maximum Likelihood)

Extract LG *i* markers from OneMap “group” object and order them:

```
marker.ord.i = order_seq(make_seq(linkage.groups, i), n.init=7)
```

Extract marker order from from OneMap “order” object

```
marker.map.i = make_seq(marker.ord.i, "force")
```

“force” = all markers

“safe” = only those that pass the specified LOD threshold

Estimating marker order

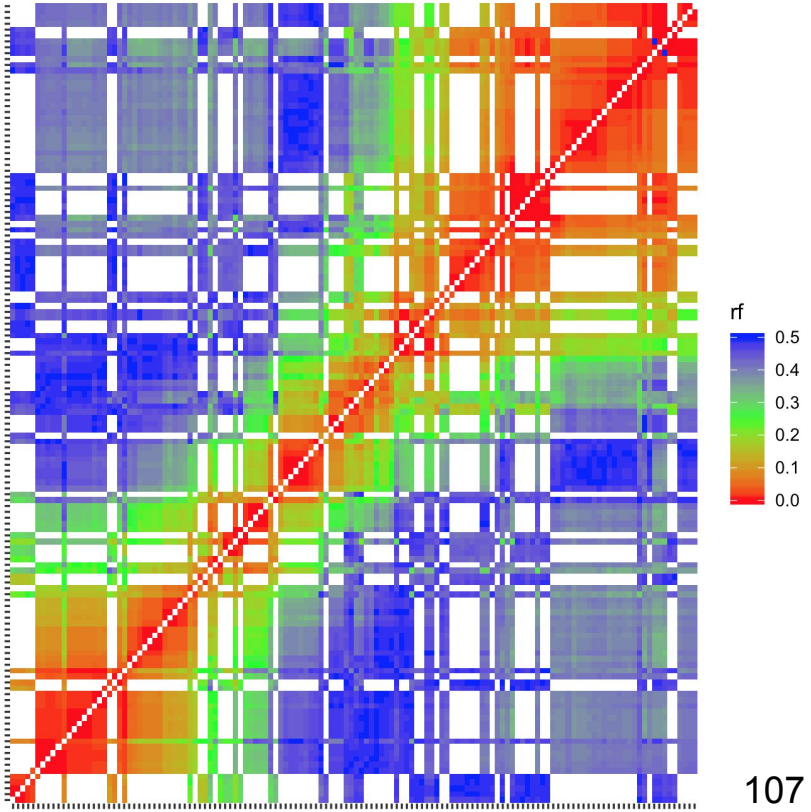
Refine the marker order by “rippling” markers across the linkage group:

```
ripple_seq(marker.map.i, ws=5)
```

ws = “window size”, number of markers to ripple each iteration

Plot a heatmap of recombination fractions between pairs of markers:

```
plot(rf_graph_table(marker.map.i,
  inter=F, graph.LOD=F, n.colors=3))
```



Visualizing the map

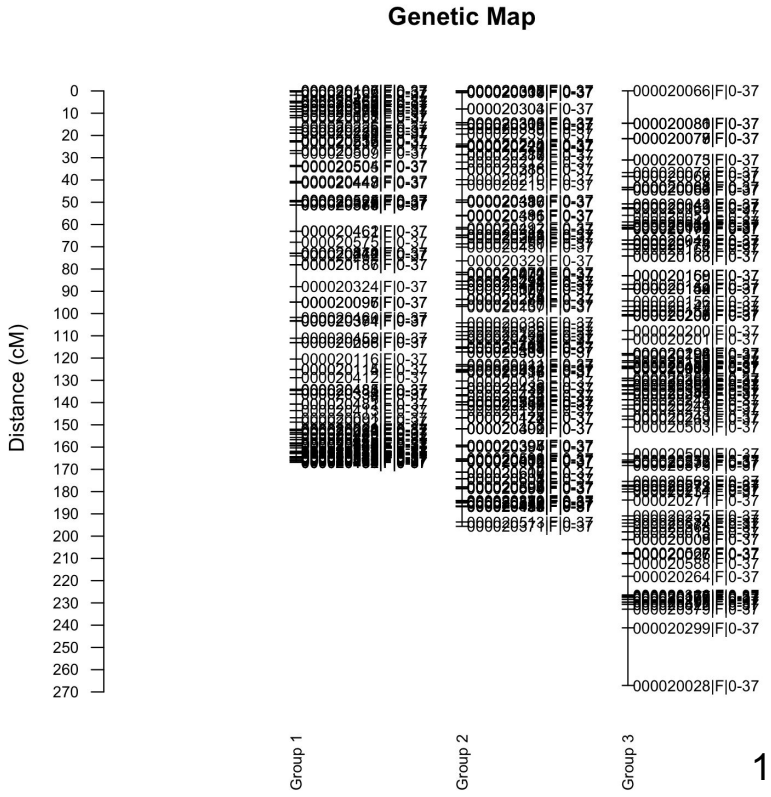
Draw a nice graphic of the linkage groups:

```
draw_map(group.table, names = TRUE)
```

Linkage groups with lengths much larger than 100 cM indicates the presence of many genotyping errors in the data

100 cM = 1 Morgan

= 1 recombination per chromosome



Saving your maps

Use the `write_map` function to write a space-delimited file of marker orders:

```
write_map(marker.map.i, file=paste(”file.prefix”,i,”txt”, sep=’.’))
```

```
1 000020105|F|0-37 0
1 000020106|F|0-37 9.99999999981846e-05
1 000020107|F|0-37 0.0001999999999996369
1 000020102|F|0-37 0.676102444861517
1 000020555|F|0-37 2.02796546722068
1 000020465|F|0-37 4.67331089992381
1 000020469|F|0-37 4.67341089992381
1 000020468|F|0-37 4.67351089992381
1 000020467|F|0-37 4.6736108999238
```

Feedback

1. Having completed the course, do you think the course is useful for your research?
2. How would you rate the course website and other materials (bad, just right, good)?
3. How would you rate the pace of the course (slow, just right, fast)?
4. In what way(s) can we improve this course?
5. Any additional comments are welcome.