Specific tools for genomics in UNIX: bedtools, bedops, vcftools,…

Course: Work with genomic data in the UNIX

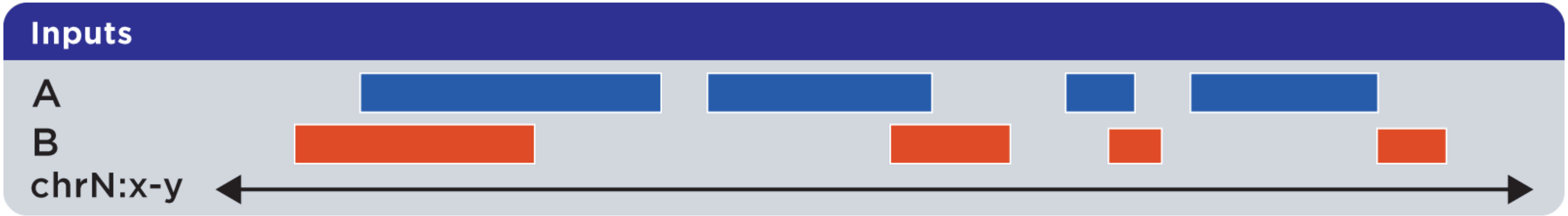
April 2015

Genome arithmetics

- Operations with genomic data based on their physical position in genome
- Variables:
  - chromosome
  - feature start, feature end
  - id
  - strand
- Basic data format: BED

Genome arithmetics: Examples

- Two sets of features (BED files):



http://bedops.readthedocs.org

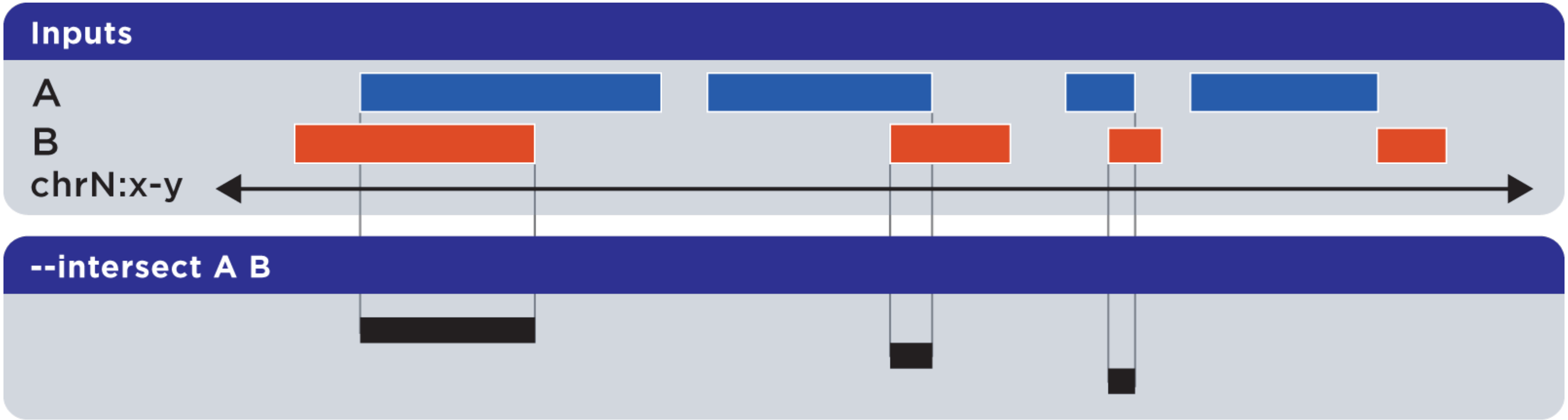
```
chr1    1000    1200
chr1    1700    2100
chr2    1100    1500
```

```
chr1    700     1100
chr1    1400    1500
chr1    1600    1900
```

New set of features based on combination of the previous sets using a specific rule

Genome arithmetics: Examples

- The rule: Get parts of features that overlap



http://bedops.readthedocs.org

```
1   9000    21000   gene1
1   30000   35000   gene2
1   65000   80000   gene3
2   32000   45000   gene4
2   55000   70000   gene5
```

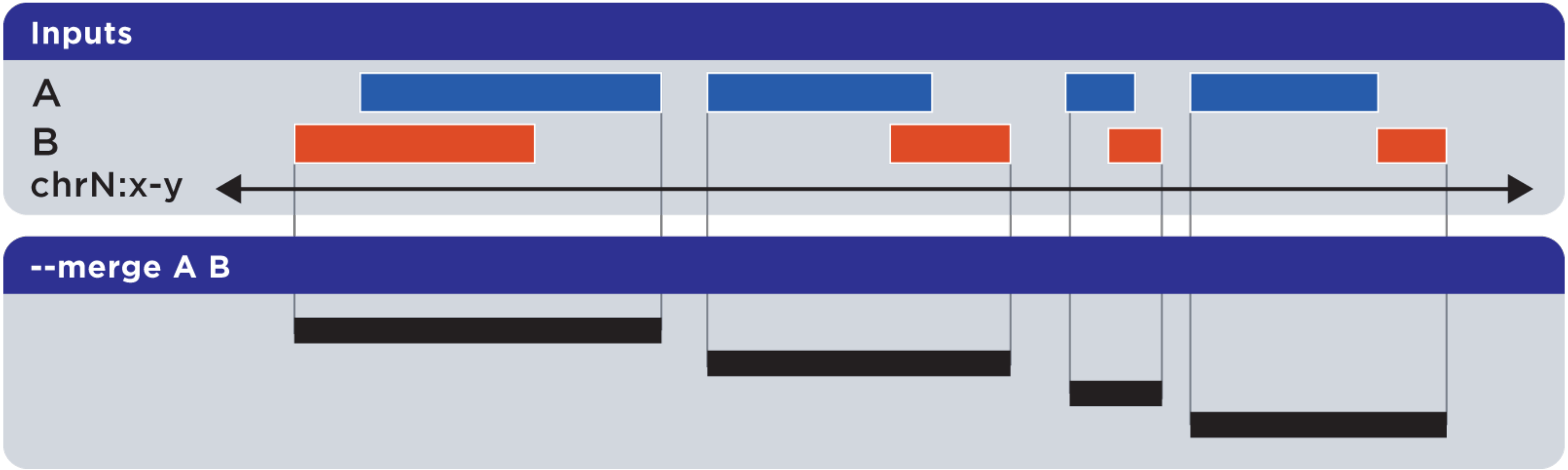
```
1   8000    10000   feature1
1   16000   18000   feature2
1   24000   26000   feature3
1   38000   45000   feature4
1   60000   70000   feature5
2   10000   13000   feature6
2   40000   44000   feature7
```

```
bedops --intersect genes.bed features.bed
bedtools intersect -a genes.bed -b features.bed
```

```
1   9000    10000
1   16000   18000
1   65000   70000
2   40000   44000
```

Genome arithmetics: Examples

- The rule: Merge entire features



http://bedops.readthedocs.org

```
1   9000    21000   gene1
1   30000   35000   gene2
1   65000   80000   gene3
2   32000   45000   gene4
2   55000   70000   gene5
```

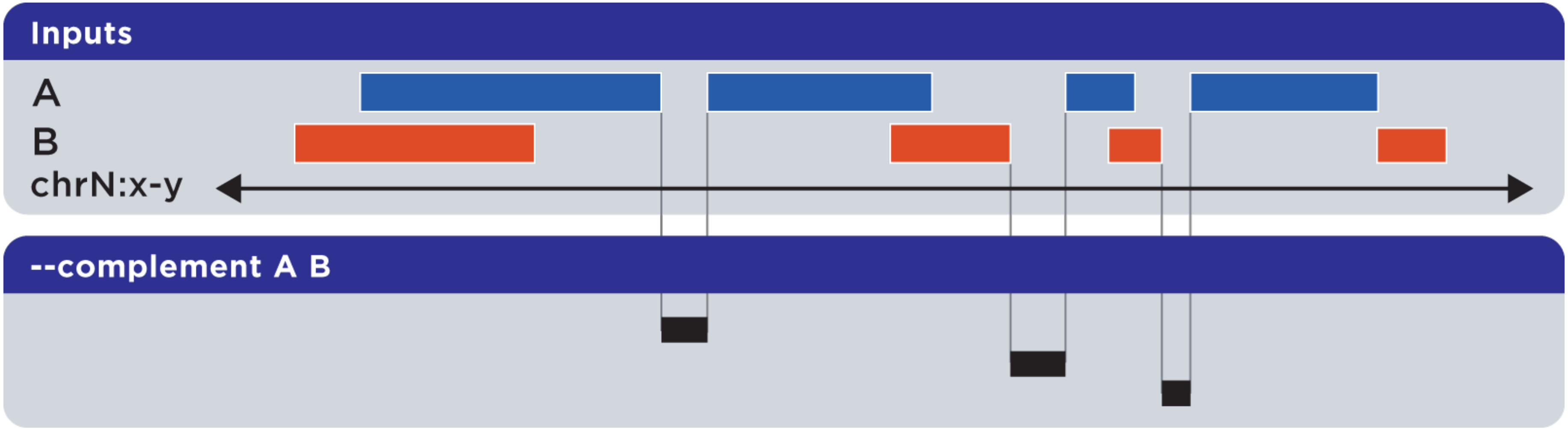
```
1   8000    10000   feature1
1   16000   18000   feature2
1   24000   26000   feature3
1   38000   45000   feature4
1   60000   70000   feature5
2   10000   13000   feature6
2   40000   44000   feature7
```

```
bedops --merge genes.bed features.bed

cat *.bed | sortBed > features2.bed
bedtools merge -i features2.bed
```

```
1   8000 21000
1   24000   26000
1   30000   35000
1   38000   45000
1   60000   80000
2   10000   13000
2   32000   45000
2   55000   70000
```

Genome arithmetics: Examples

- The rule: Get complement features



http://bedops.readthedocs.org

```
1   9000    21000   gene1
1   30000   35000   gene2
1   65000   80000   gene3
2   32000   45000   gene4
2   55000   70000   gene5
```

```
1   8000    10000   feature1
1   16000   18000   feature2
1   24000   26000   feature3
1   38000   45000   feature4
1   60000   70000   feature5
2   10000   13000   feature6
2   40000   44000   feature7
```

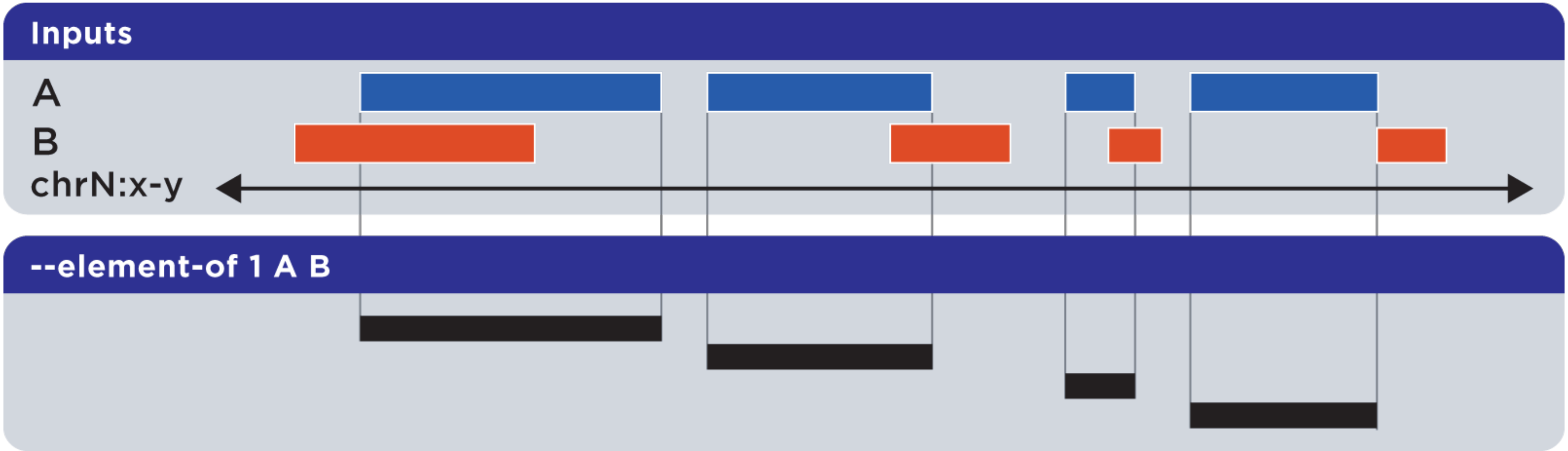
```
bedops --complement genes.bed features.bed
bedtools complement -i <(cat *.bed | sortBed) -g my.genome
```

```
1   21000   24000
1   26000   30000
1   35000   38000
1   45000   60000
2   13000   32000
2   45000   55000
```

```
1   0   8000
1   21000   24000
1   26000   30000
1   35000   38000
1   45000   60000
1   80000   100000
2   0   10000
2   13000   32000
2   45000   55000
2   70000   120000
```

Genome arithmetics: Examples

- The rule: Report A which overlaps B



http://bedops.readthedocs.org

```
1   9000   21000  gene1
1   30000  35000  gene2
1   65000  80000  gene3
2   32000  45000  gene4
2   55000  70000  gene5
```

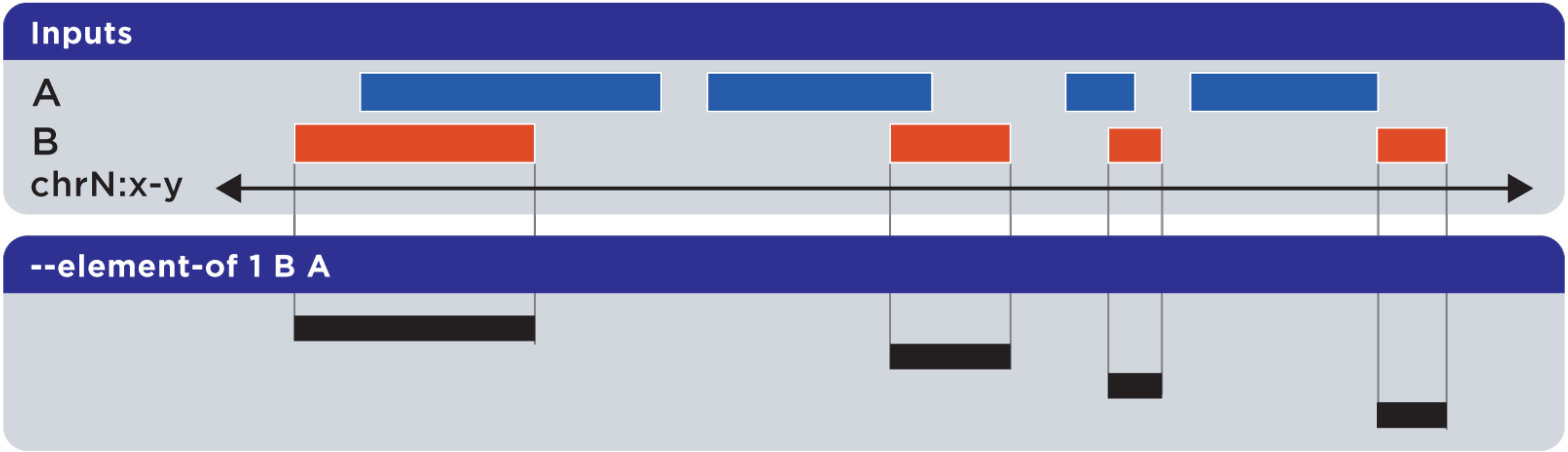
```
1   8000   10000  feature1
1   16000  18000  feature2
1   24000  26000  feature3
1   38000  45000  feature4
1   60000  70000  feature5
2   10000  13000  feature6
2   40000  44000  feature7
```

```
bedops --element-of 1 genes.bed features.bed
bedtools intersect -u -a genes.bed -b features.bed
```

```
1   9000   21000  gene1
1   65000  80000  gene3
2   32000  45000  gene4
```

Genome arithmetics: Examples

- The rule: Report B which overlaps A



http://bedops.readthedocs.org

```
1   9000    21000   gene1
1   30000   35000   gene2
1   65000   80000   gene3
2   32000   45000   gene4
2   55000   70000   gene5
```

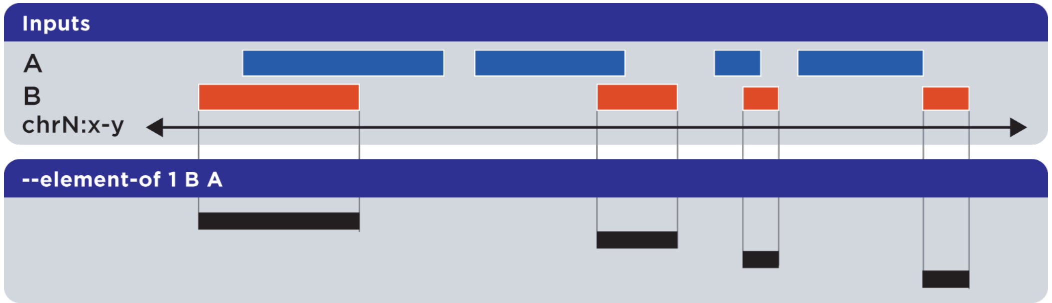
```
1   8000    10000   feature1
1   16000   18000   feature2
1   24000   26000   feature3
1   38000   45000   feature4
1   60000   70000   feature5
2   10000   13000   feature6
2   40000   44000   feature7
```

```
bedops --element-of 1 features.bed genes.bed
bedtools intersect -u -a features.bed -b genes.bed
```

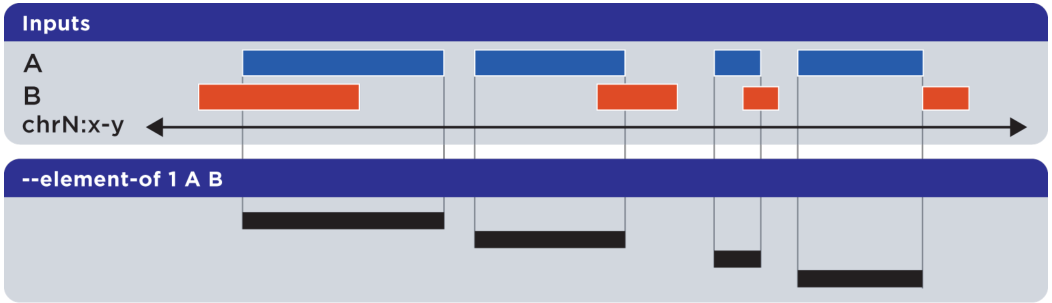
```
1   8000    10000   feature1
1   16000   18000   feature2
1   60000   70000   feature5
2   40000   44000   feature7
```

Genome arithmetics: Examples

- The rule: Report A,B which overlap each other



+



http://bedops.readthedocs.org

```
1   9000   21000  gene1
1   30000  35000  gene2
1   65000  80000  gene3
2   32000  45000  gene4
2   55000  70000  gene5
```

```
1   8000   10000  feature1
1   16000  18000  feature2
1   24000  26000  feature3
1   38000  45000  feature4
1   60000  70000  feature5
2   10000  13000  feature6
2   40000  44000  feature7
```

```
bedtools intersect -wa -wb -a genes.bed -b features.bed
```

```
1   9000   21000  gene1  1   8000   10000  feature1
1   9000   21000  gene1  1   16000  18000  feature2
1   65000  80000  gene3  1   60000  70000  feature5
2   32000  45000  gene4  2   40000  44000  feature7
```

Genome feature summary

- Statistics, summary
- bedmap, bedtools (coverageBed, groupBy)
- e.g. depth coverage, base pair coverage, etc.

Genome feature summary: Example

- What is the base coverage of features within genes?

```
1   9000    21000   gene1
1   30000   35000   gene2
1   65000   80000   gene3
2   32000   45000   gene4
2   55000   70000   gene5
```

```
1   8000    10000   feature1
1   16000   18000   feature2
1   24000   26000   feature3
1   38000   45000   feature4
1   60000   70000   feature5
2   10000   13000   feature6
2   40000   44000   feature7
```

```
1    9000 21000    gene1
1    30000    35000    gene2
1    65000    80000    gene3
2    32000    45000    gene4
2    55000    70000    gene5
```

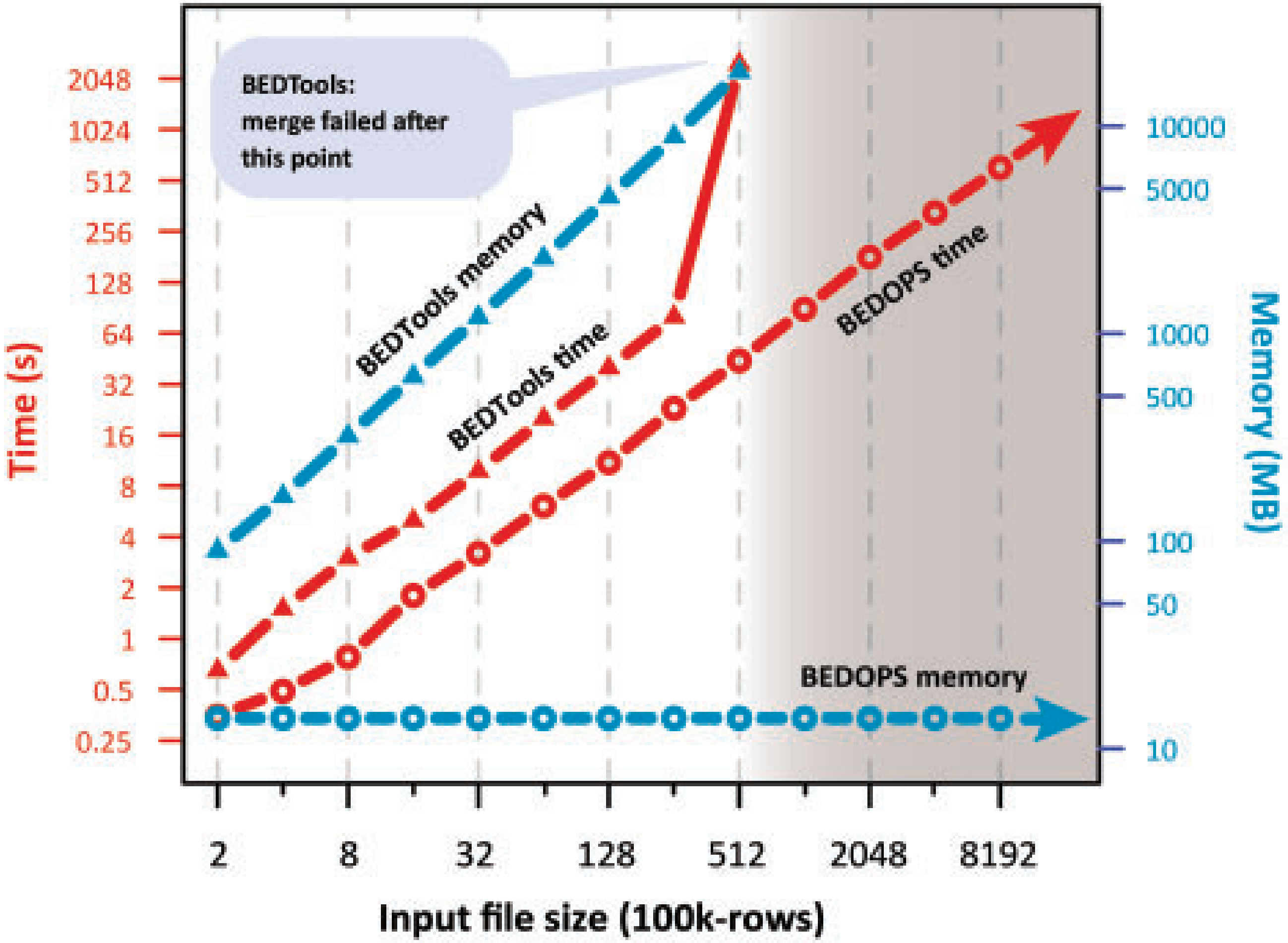
```
1    8000 10000    feature1
1    16000    18000    feature2
1    24000    26000    feature3
1    38000    45000    feature4
1    60000    70000    feature5
2    10000    13000    feature6
2    40000    44000    feature7
```

```
bedmap --echo --count --bases-uniq genes.bed features.bed
coverageBed –b genes.bed –a features.bed
```

```
1    9000 21000    gene1|2|3000
1    30000    35000    gene2|0|0
1    65000    80000    gene3|1|5000
2    32000    45000    gene4|1|4000
2    55000    70000    gene5|0|0
```

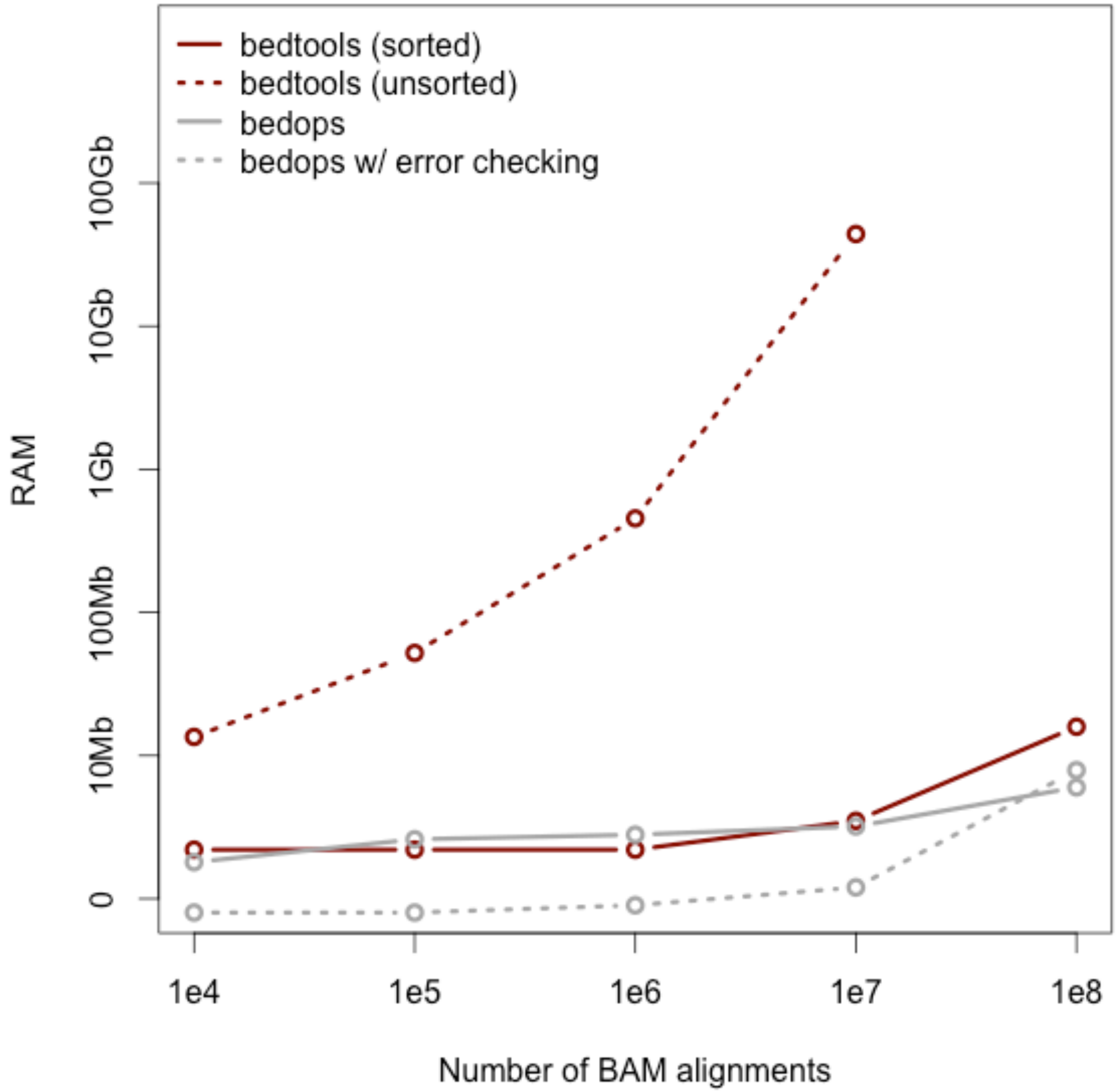
```
1    9000 21000    gene1    2    3000 12000    0.2500000
1    30000    35000    gene2    0    0    5000 0.0000000
1    65000    80000    gene3    1    5000 15000    0.3333333
2    32000    45000    gene4    1    4000 13000    0.3076923
2    55000    70000    gene5    0    0    15000    0.0000000
```

bedtools vs. bedops



Neph et al. (2012) *Bioinformatics*

bedtools vs. bedops



Always use sorted data: sort-bed (bedops), sortBed (bedtools)

http://bedtools.readthedocs.org

Other tools in bedtools

- makewindows
- cluster
- shuffle
- random
- jaccard
- reldist
- …

Variation data: vcftools

- Efficient manipulation with VCF data
- Control quality
- Molecular evolution & population genetics measures/statistics
  - transition/transversion
  - heterozygosity, relatedness
  - Hardy-Weinberg
  - Weir & Cockerham's Fst
  - Nucleotide diversity
  - Linkage Disequilibrium

vcftools: starting

- Opening and viewing a vcf file:

```
vcftools --gzvcf popdata_mda.vcf.gz --recode --stdout | less -S
```

- Creating a new vcf file:

```
vcftools --gzvcf popdata_mda.vcf.gz --recode --out new_vcf
```

vcftools: data filtering

- Sample/Variant retrieval by name:
  - Individual/Variant names to keep/remove have to be specified in a separate file

```
--keep ind.txt # Keep these individuals
--remove ind.txt # Remove these individuals
--snps snps.txt # Keep these SNPs
--snps snps.txt --exclude # Remove these SNPs
```

```
vcftools --gzvcf popdata_mda.vcf.gz --keep euro_samples.txt --
recode --stdout | less -S
```

vcftools: data filtering

- Variant filtering based on physical location

```
--chr 11 # Keep just this chromosome
--not-chr 11 # Remove this chromosome
--not-chr 11 -not-chr 2 # Remove these two chromosomes
--from-bp 20000000 # Keep SNPs from this position
--to-bp 22000000 # Keep SNPs to this position
--bed keep.bed # Keep only SNPs overlapping with locations
listed in a file
--exclude-bed remove.bed # The opposite of the previous
```

```
vcftools --gzvcf popdata_mda.vcf.gz --keep euro_samples.txt --
chr 11 --from-bp 22000000 --to-bp 23000000 --recode --stdout |
less -S
```

vcftools: data filtering

- Variant filtering based on other features

```
--maf 0.2 # Keep just variants with Minor Allele Freq higher
than 0.2
--hwe 0.05 # Keep just variants which do not deviate from HW
equilibrium (p-value = 0.05)
--max-missing (0-1) # Remove SNPs with given proportion of
missing data (0 = allowed completely missing, 1 = no missing
data allowed)
--minQ 20 # Minimal quality allowed (Phred score)
```

```
vcftools --gzvcf popdata_mda.vcf.gz --keep euro_samples.txt --
recode --stdout | vcftools --vcf - --max-missing 1 -maf 0.2 --
recode --stdout | less -S
```

stdin

vcftools: data filtering

- Variant filtering based on other features

```
--maf 0.2 # Keep just variants with Minor Allele Freq higher
than 0.2
--hwe 0.05 # Keep just variants which do not deviate from HW
equilibrium (p-value = 0.05)
--max-missing (0-1) # Remove SNPs with given proportion of
missing data (0 = allowed completely missing, 1 = no missing
data allowed)
--minQ 20 # Minimal quality allowed (Phred score)
```

```
vcftools --gzvcf popdata_mda.vcf.gz --keep euro_samples.txt --
recode --stdout | vcftools --vcf - --max-missing 1 -maf 0.2 --
recode --stdout > popdata_mda_euro.vcf
```

vcftools: summary/statistics

- molecular evolution/population genetic

```
--site-pi # Calculates per-site nucleotide diversity (π)
--window-pi 1000000 --window-pi-step 250000 # Calculates per-site nucleotide diversity for windows of 1Mb with 250Kb step
--weir-fst-pop pop1.txt --weir-fst-pop pop2.txt # Calculates Weir & Cockerham's Fst
--fst-window-size 1000000 --fst-window-step 250000 # Calculates Fst for windows of 1Mb with 250Kb step
```

```
vcftools --vcf popdata_mda_euro.vcf
--weir-fst-pop musculus_samps.txt
--weir-fst-pop domesticus_samps.txt --stdout | less -S
```

Exercise: Population differentiation

- Get a population differentiation calculated as Fst between *M. m. musculus* and *M. m. domesticus* within a given sliding window and find candidate genes within highly differentiated regions
  - use vcftools to filter data and calculate Fst for individual SNPs
  - use bedtools makewindows to create sliding windows of three sizes
    - 100 kb + 10 kb step
    - 500 kb + 50 kb step
    - 1 Mb + 100 kb step
  - use bedmap (bedops) to calculate average Fst for each window
  - use Rstudio and ggplot2 to plot Fst values across the genome
  - use R to obtain 99th percentile and use it to obtain a set of candidate genomic regions
  - use bedtools intersect to get a list of candidate genes

Exercise: Population differentiation

1. use vcftools to filter data and calculate Fst for individual SNPs

```
## Prepare files

cd
mkdir data/diff

cp /data/mus_mda/00-popdata/*.txt data/diff/.
mv /data/mus_mda/00-popdata/popdata_mda.vcf.gz data/diff/.

cd data/diff/
```

```
vcftools --gzvcf popdata_mda.vcf.gz --keep euro_samples.txt --
recode --stdout | vcftools --vcf - --max-missing 1 –maf 0.2 --
recode --stdout > popdata_mda_euro.vcf
```

Exercise: Population differentiation

1. use vcftools to filter data and calculate Fst for individual SNPs

```
vcftools --gzvcf popdata_mda.vcf.gz--keep euro_samples.txt --
recode --stdout | vcftools --vcf - --max-missing 1 –maf 0.2 --
recode --stdout > popdata_mda_euro.vcf
```

```
vcftools --vcf popdata_mda_euro.vcf
--weir-fst-pop musculus_samps.txt
--weir-fst-pop domesticus_samps.txt --stdout |
tail -n +2 |
awk -F $'\t' 'BEGIN{OFS=FS}{ print $1,$2-1,$2,$1":"$2,$3}' >
popdata_mda_euro_fst.bed
```

Exercise: Population differentiation

2. use bedtools makewindows to create sliding windows of three sizes
   - 100 kb + 10 kb step
   - 500 kb + 50 kb step
   - 1 Mb + 100 kb step

Inputting from subshell
<(command producing input)

```
cp /data/mus_mda/02-windows/genome.fa.fai .

bedtools makewindows -g <(grep '^2\|^11' genome.fa.fai) -w
1000000 -s 100000 -i winnum | awk '{ print $0":1000kb" }' >
windows_1000kb.bed
```

```
cat windows_*.bed > windows.bed
```

Exercise: Population differentiation

3. use bedmap (bedops) to calculate average Fst for each window

```
sort-bed windows.bed > windows_sorted.bed
sort-bed popdata_mda_euro_fst.bed >
popdata_mda_euro_fst_sorted.bed

bedmap --echo --mean --count windows_sorted.bed
popdata_mda_euro_fst_sorted.bed | grep -v NA |
tr "|:" "\t" > windows2snps_fst.bed
```

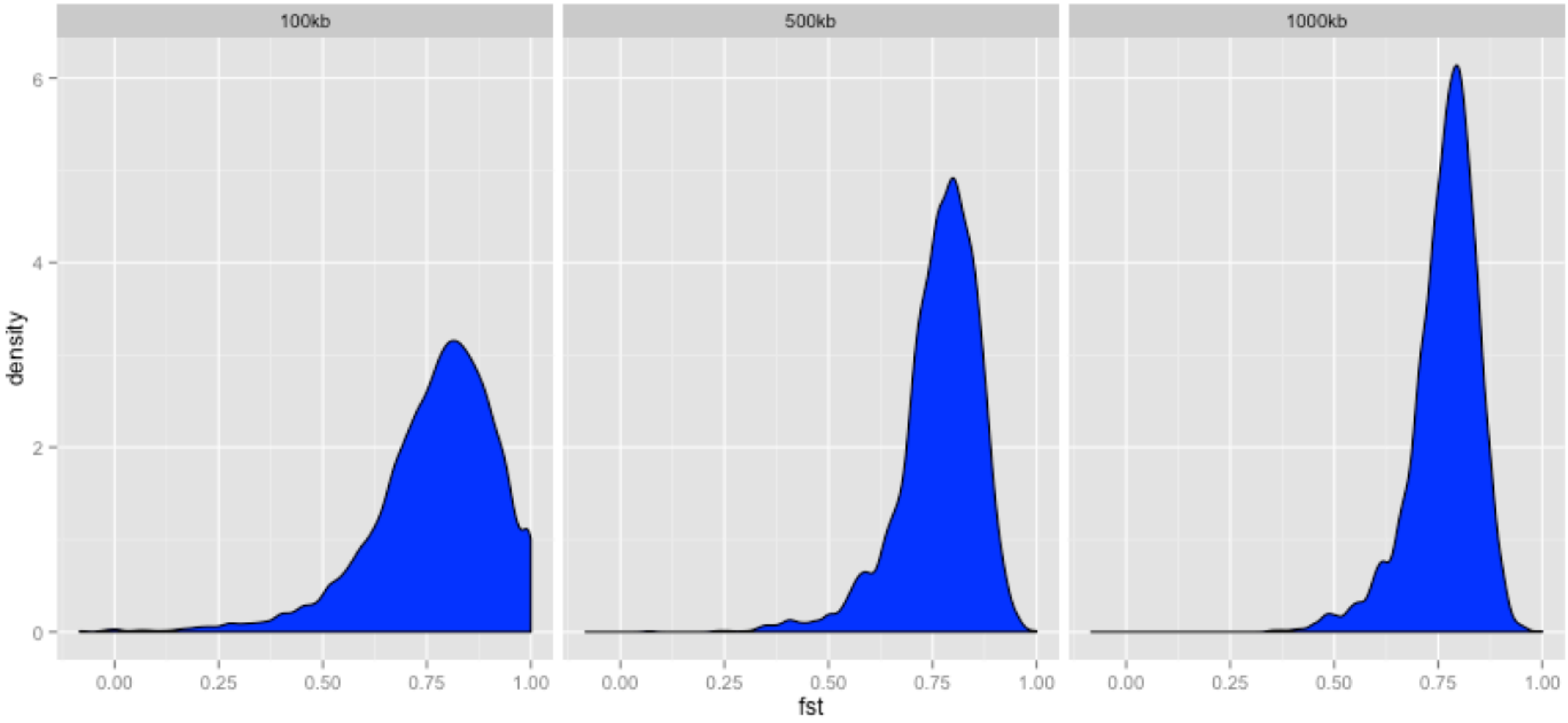
Exercise: Population differentiation

4. use Rstudio and ggplot2 to plot Fst values across the genome

```
library(ggplot2)

setwd("~/data/diff")

fst <- read.table("windows2snps_fst.bed", header=F,sep="\t")

names(fst) <- c("chrom", "start", "end", "win_id", "win_size",
"fst", "cnt_snps")

fst$win_size <- factor(fst$win_size, levels=c("100kb",
"500kb", "1000kb"))

qplot(fst, data=fst, geom="density",fill=I("blue")) +
facet_wrap(~win_size)
```

Exercise: Population differentiation

4. use Rstudio and ggplot2 to plot Fst values across the genome



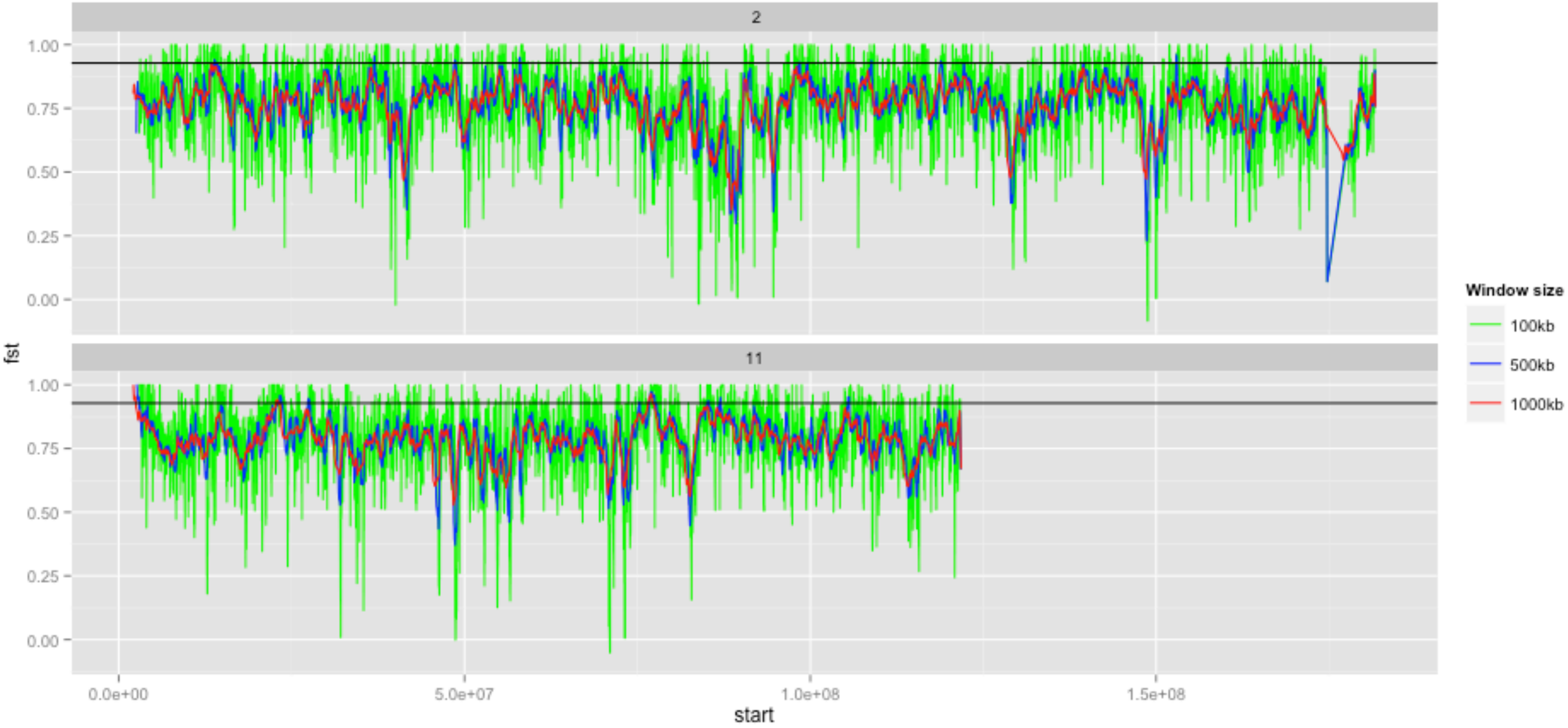
Exercise: Population differentiation

4. use Rstudio and ggplot2 to plot Fst values across the genome

```
ggplot(fst, aes(y=fst, x=start, colour=win_size)) +
    geom_line() +
    facet_wrap(~chrom, nrow=2) +
    scale_colour_manual(name="Window size", values=c("green",
"blue","red"))

q <- quantile(subset(fst,win_size=="500kb",select="fst")[,
1],prob=0.99)[[1]]

ggplot(fst, aes(y=fst, x=start, colour=win_size)) +
    geom_line() +
    facet_wrap(~chrom, nrow=2) +
    geom_hline(yintercept=q,colout="black") +
    scale_colour_manual(name="Window size", values=c("green",
"blue","red"))
```

Exercise: Population differentiation

4. use Rstudio and ggplot2 to plot Fst values across the genome



Exercise: Population differentiation

5. use R to obtain 99th percentile and use it to obtain a set of candidate genomic regions

Use of Shell variables

var=value

var=`command`

echo $var

```
q500=`grep 500kb windows2snps_fst.bed | cut -f 6 | Rscript -e
'quantile(as.numeric(readLines("stdin")),p=c(0.99))[[1]]' |
cut -d " " -f 2`
```

```
echo $q500
```

```
grep 500kb windows2snps_fst.bed | awk -v a=$q500 -F $'\t'
'BEGIN{OFS=FS}{ if($6 >= a){print $1,$2,$3} }' |
bedtools merge -i stdin > signif_500kb.bed
```

Exercise: Population differentiation

6. use bedtools intersect to get a list of candidate genes

```
bedtools intersect –a signif.bed –b
Mus_musculus.NCBIM37.67.gtf -wa -wb | grep protein_coding |
cut -f 1,2,3,4,13 | cut -d ' ' -f 1,3,9 | tr -d '"";' | sort |
uniq > fst2genes.tab
```