Getting Genomics Done with R

Stephen A. Sefick

2017-03-28

Outline

Topic

Motivation

1. How to arrive at analysis ready variants?

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
    - False Positives

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
    - False Positives
3. How to separate True/False Positives

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
   - False Positives
3. How to separate True/False Positives
   - classification/machine learning

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
   - False Positives
3. How to separate True/False Positives
   - classification/machine learning
   - filtering

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
   - False Positives
3. How to separate True/False Positives
   - classification/machine learning
   - filtering
   - both

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
   - False Positives
3. How to separate True/False Positives
   - classification/machine learning
   - filtering
   - both
4. What software to use?

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
   - False Positives
3. How to separate True/False Positives
   - classification/machine learning
   - filtering
   - both
4. What software to use?
   - R

Motivation

1. How to arrive at analysis ready variants?
2. GATK HaplotypeCaller is "permissive"
   - False Positives
3. How to separate True/False Positives
   - classification/machine learning
   - filtering
   - both
4. What software to use?
   - R
   - Bioconductor Project

Bioconductor

1. CRAN

Bioconductor

1. CRAN
2. Bioconductor- additional software repository

Bioconductor

1. CRAN
2. Bioconductor- additional software repository
3. "Open Source Software For Bioinformatics"

Bioconductor

1. CRAN
2. Bioconductor- additional software repository
3. "Open Source Software For Bioinformatics"
    - DNA micro-array

Bioconductor

1. CRAN
2. Bioconductor- additional software repository
3. "Open Source Software For Bioinformatics"
   - DNA micro-array
   - sequence

Bioconductor

1. CRAN
2. Bioconductor- additional software repository
3. "Open Source Software For Bioinformatics"
   - DNA micro-array
   - sequence
   - SNP

Bioconductor

1. CRAN
2. Bioconductor- additional software repository
3. "Open Source Software For Bioinformatics"
   - DNA micro-array
   - sequence
   - SNP
   - etc.

Bioconductor

1. CRAN
2. Bioconductor- additional software repository
3. "Open Source Software For Bioinformatics"
   - DNA micro-array
   - sequence
   - SNP
   - etc.
4. started 2001

Bioconductor

1. CRAN
2. Bioconductor- additional software repository
3. "Open Source Software For Bioinformatics"
   - DNA micro-array
   - sequence
   - SNP
   - etc.
4. started 2001
5. 1294 user contributed packages

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")
2. Documentation familiar and excellent

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")
2. Documentation familiar and excellent
3. In addition, most packages have vignettes

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")
2. Documentation familiar and excellent
3. In addition, most packages have vignettes
   - Vignettes are short "how-tos"

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")
2. Documentation familiar and excellent
3. In addition, most packages have vignettes
   - Vignettes are short "how-tos"
4. Most object oriented providing consistent "feel" for bioconductor packages

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")
2. Documentation familiar and excellent
3. In addition, most packages have vignettes
   - Vignettes are short "how-tos"
4. Most object oriented providing consistent "feel" for bioconductor packages
   - CRAN mix of no OO, S3, S4, and newer R6

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")
2. Documentation familiar and excellent
3. In addition, most packages have vignettes
   - Vignettes are short "how-tos"
4. Most object oriented providing consistent "feel" for bioconductor packages
   - CRAN mix of no OO, S3, S4, and newer R6
   - Flexibility is R's strength and weakness

Bioconductor Packages

1. Installation similar to CRAN packages (automatic dependency resolution)
   - Install/update Bioconductor
     - source("https://bioconductor.org/biocLite.R")
     - biocLite()
   - Install packages
     - biocLite("VariantAnnotation")
2. Documentation familiar and excellent
3. In addition, most packages have vignettes
   - Vignettes are short "how-tos"
4. Most object oriented providing consistent "feel" for bioconductor packages
   - CRAN mix of no OO, S3, S4, and newer R6
   - Flexibility is R's strength and weakness
5. https://bioconductor.org/

Topic

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?
   - understanding the data

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?
   - understanding the data
   - filtering

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?
   - understanding the data
   - filtering
   - classification/machine learning

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?
   - understanding the data
   - filtering
   - classification/machine learning
4. Non-model systems analysis ready variants

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?
   - understanding the data
   - filtering
   - classification/machine learning
4. Non-model systems analysis ready variants
   - hard-filtered call set

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?
   - understanding the data
   - filtering
   - classification/machine learning
4. Non-model systems analysis ready variants
   - hard-filtered call set
   - Bootstrapped Variant Quality Score Re-calibration (VQSR) call set using HF as training/truth data

GATK Haplotype Caller Annotations

1. HaplotypeCaller includes/calculates annotations
   - VCFs have a number of annotations
     - e.g., genotype quality, depth, etc.
2. Annotations change
   - e.g., HaplotypeCaller's gVCFs with Reference Genotype Quality
   - Custom annotations
3. How do we use annotations?
   - understanding the data
   - filtering
   - classification/machine learning
4. Non-model systems analysis ready variants
   - hard-filtered call set
   - Bootstrapped Variant Quality Score Re-calibration (VQSR) call set using HF as training/truth data
5. Today concentrate on hard-filtering SNPs

Hard-filtering

1. Context specific

Hard-filtering

1. Context specific
    - SNPs

Hard-filtering

1. Context specific
   - SNPs
   - INDELs

Hard-filtering

1. Context specific
   - SNPs
   - INDELs
2. GATK hard-filtering recommendations

Hard-filtering

1. Context specific
   - SNPs
   - INDELs
2. GATK hard-filtering recommendations
3. These are recommendations, developed with human data, and might/likely will need to be modified based on the data

Variant	#	Annotation	Remove If
Both	1	DP	min=empirical; max=5 or 6 sigma
	2	GQ (or RGQ)	empirical
SNP	3	QD	< 2.0
	4	MQ	< 40
	5	FS	> 60
	6	SOR	> 3.0
	7	MQRankSum	< -12.5
	8	ReadPosRankSum	< -8.0

1 Depth (DP) - Depth of coverage

1. Min - no empirical guidance- look at plots

1 Depth (DP) - Depth of coverage

1. Min - no empirical guidance- look at plots
2. Max - 5 or 6x the standard deviation

1 Depth (DP) - Depth of coverage

1. Min - no empirical guidance- look at plots
2. Max - 5 or 6x the standard deviation
3. How many reads cover a position

1 Depth (DP) - Depth of coverage

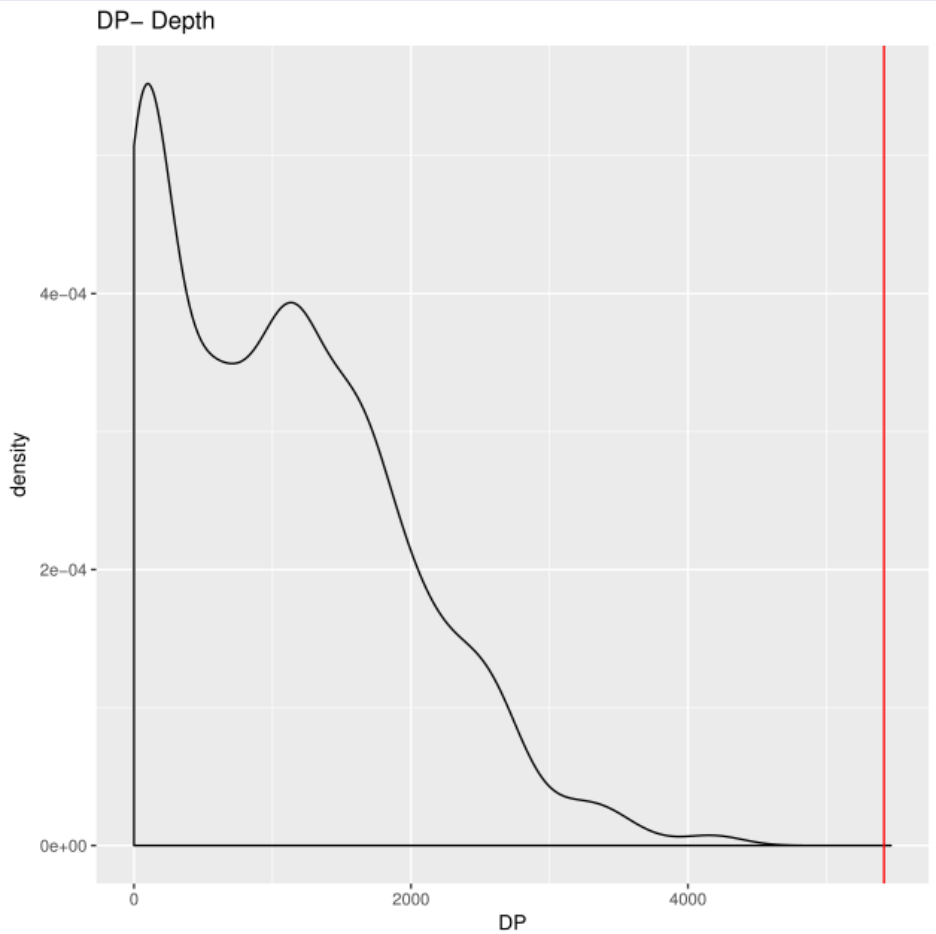
1. Min - no empirical guidance- look at plots
2. Max - 5 or 6x the standard deviation
3. How many reads cover a position
   - GATK Caveat- slightly different from raw depth of coverage

1 Depth (DP) - Depth of coverage

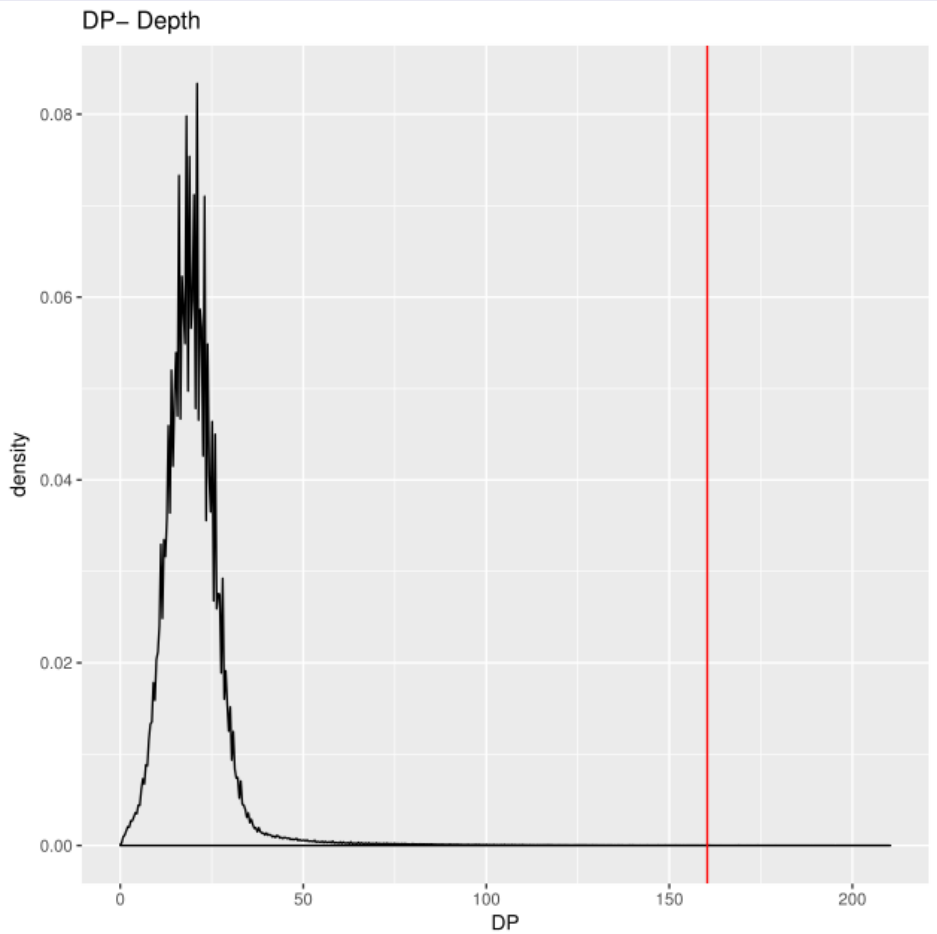
1. Min - no empirical guidance- look at plots
2. Max - 5 or 6x the standard deviation
3. How many reads cover a position
   - GATK Caveat- slightly different from raw depth of coverage
   - QC of the caller will remove reads (MAQ)

1 Depth (DP) - Depth of coverage

Mitochondria



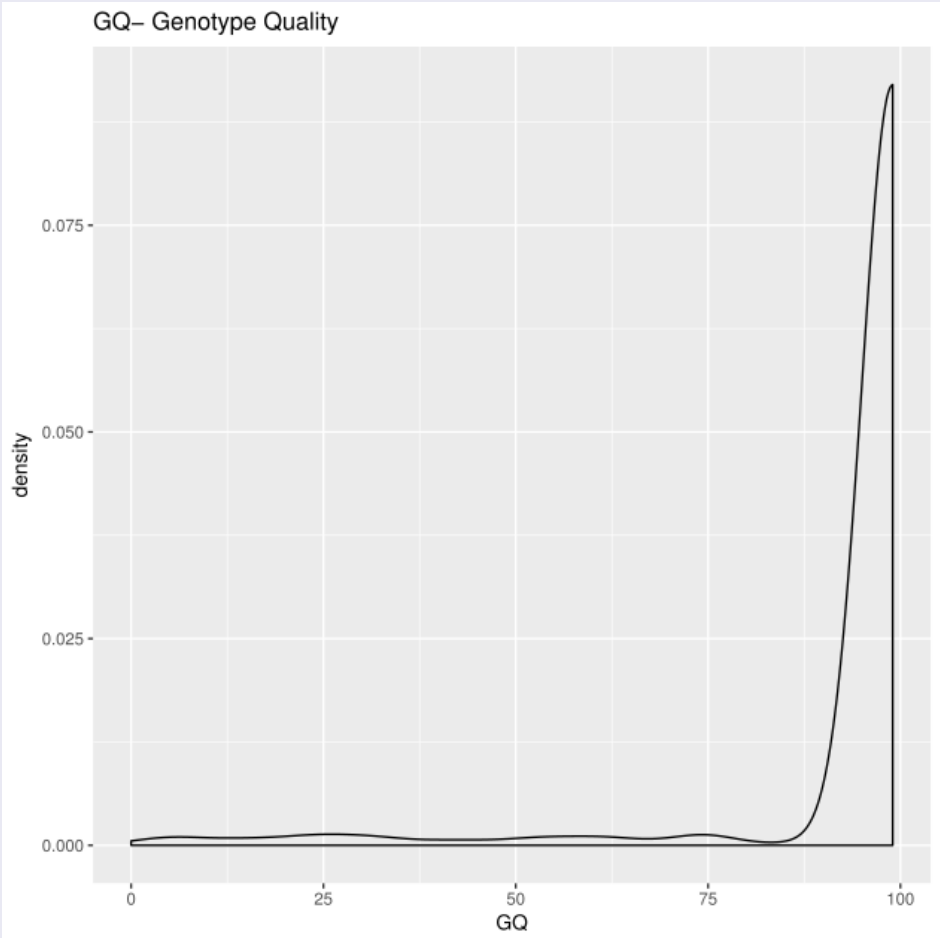
Chr1-20 and X



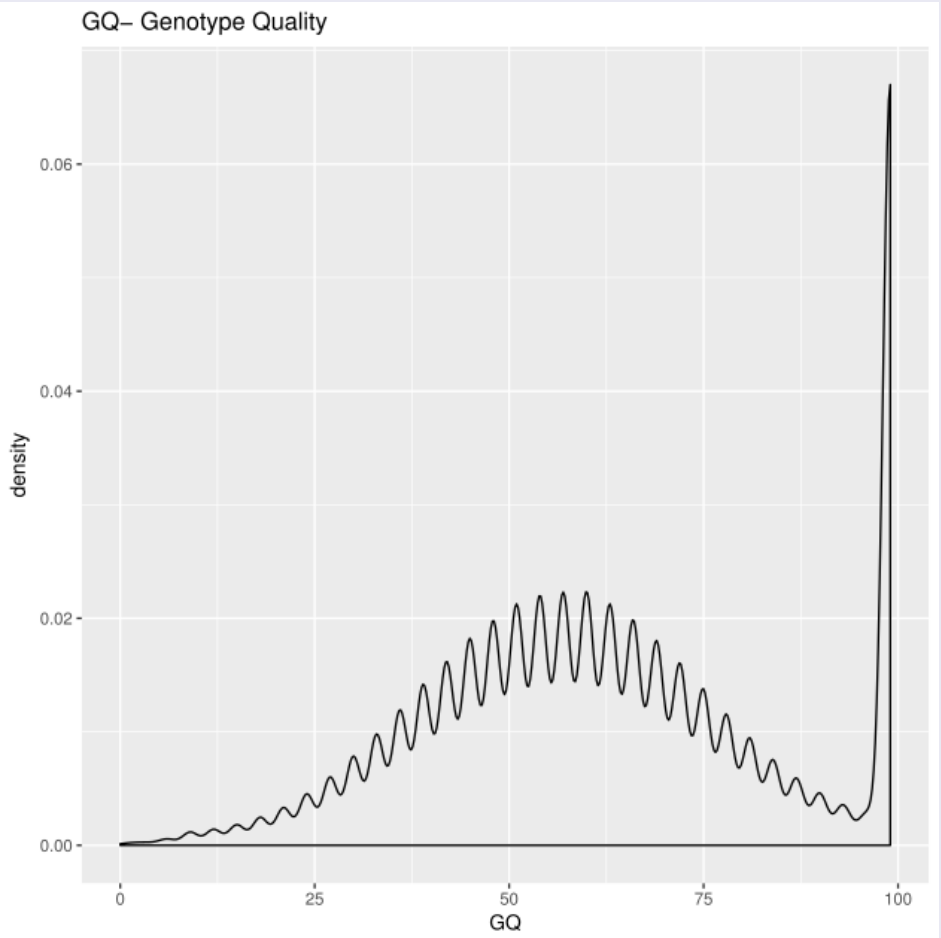
2 Genotype quality (GQ); Reference GQ (RGQ)

1. Phred scaled probability of incorrect genotype
   - 20 - 0.01; 30 - 0.001; 40 - 0.0001

Mitochondria

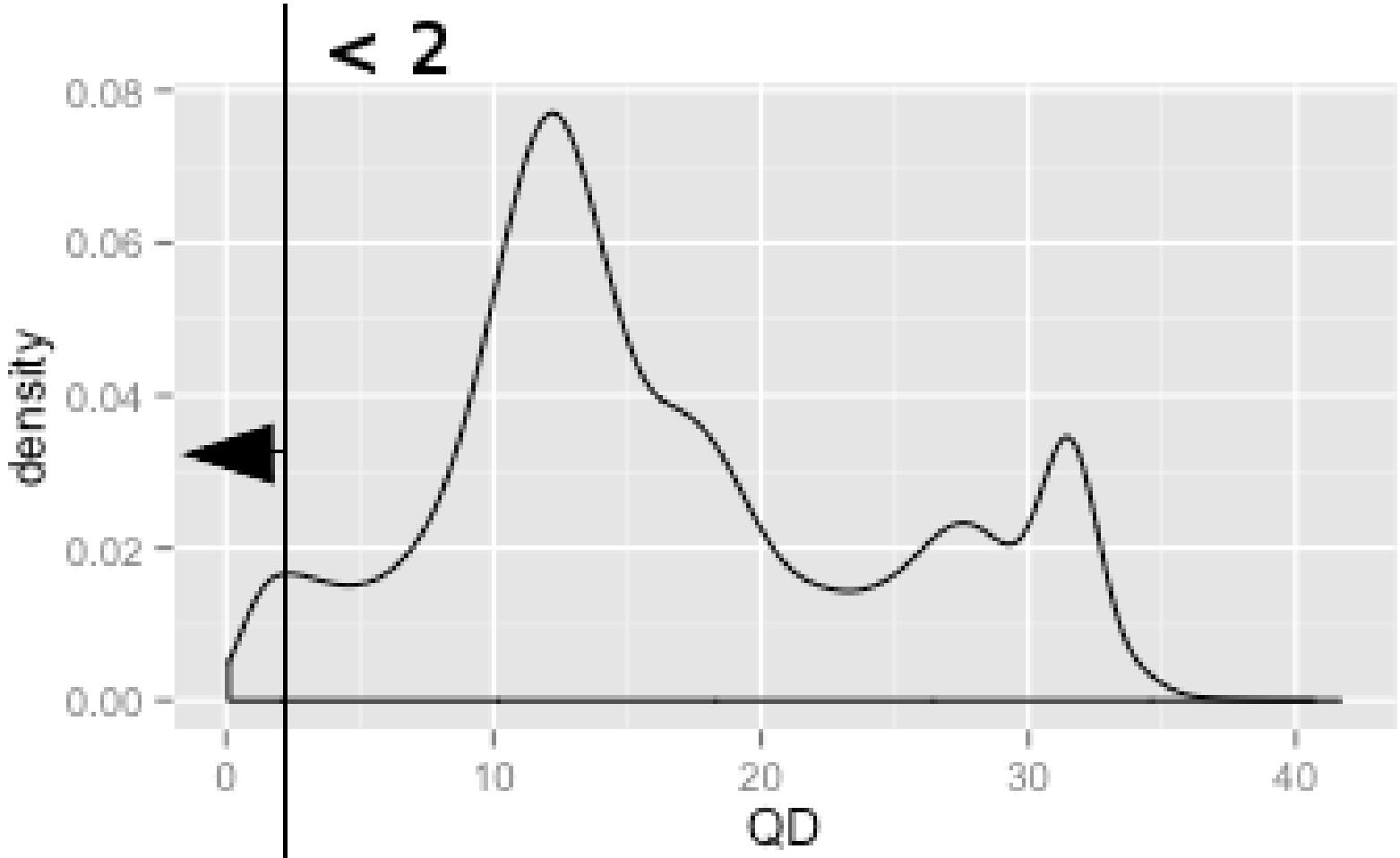


Chr1-20 and X



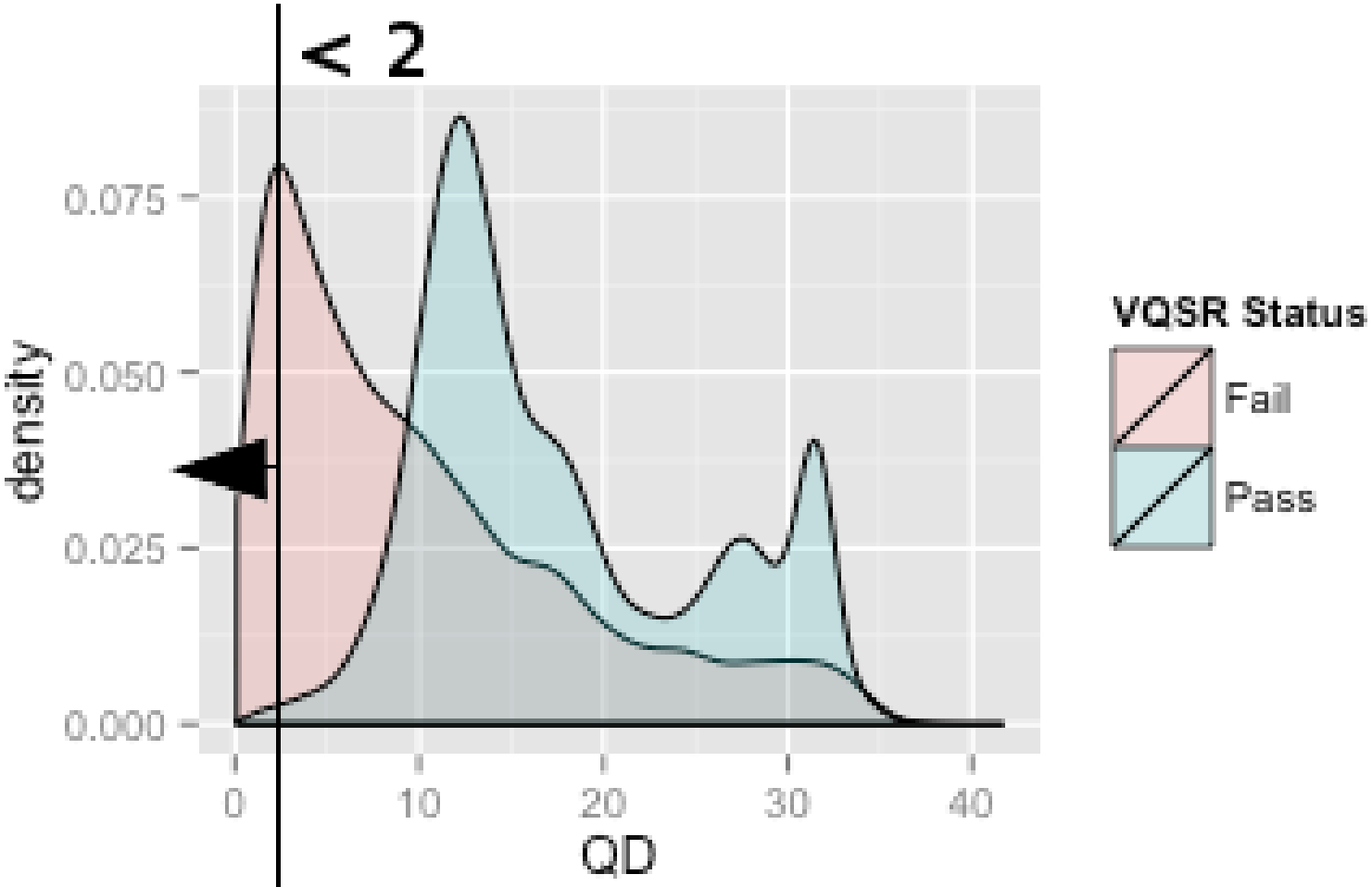
3 Variant quality/allele depth (QD)

1. Variant Quality (QUAL) is the phred scaled probability that the variant is wrong.
2. allele depth is actual depth of each observed allele (How many actual reads; in contrast to DP).



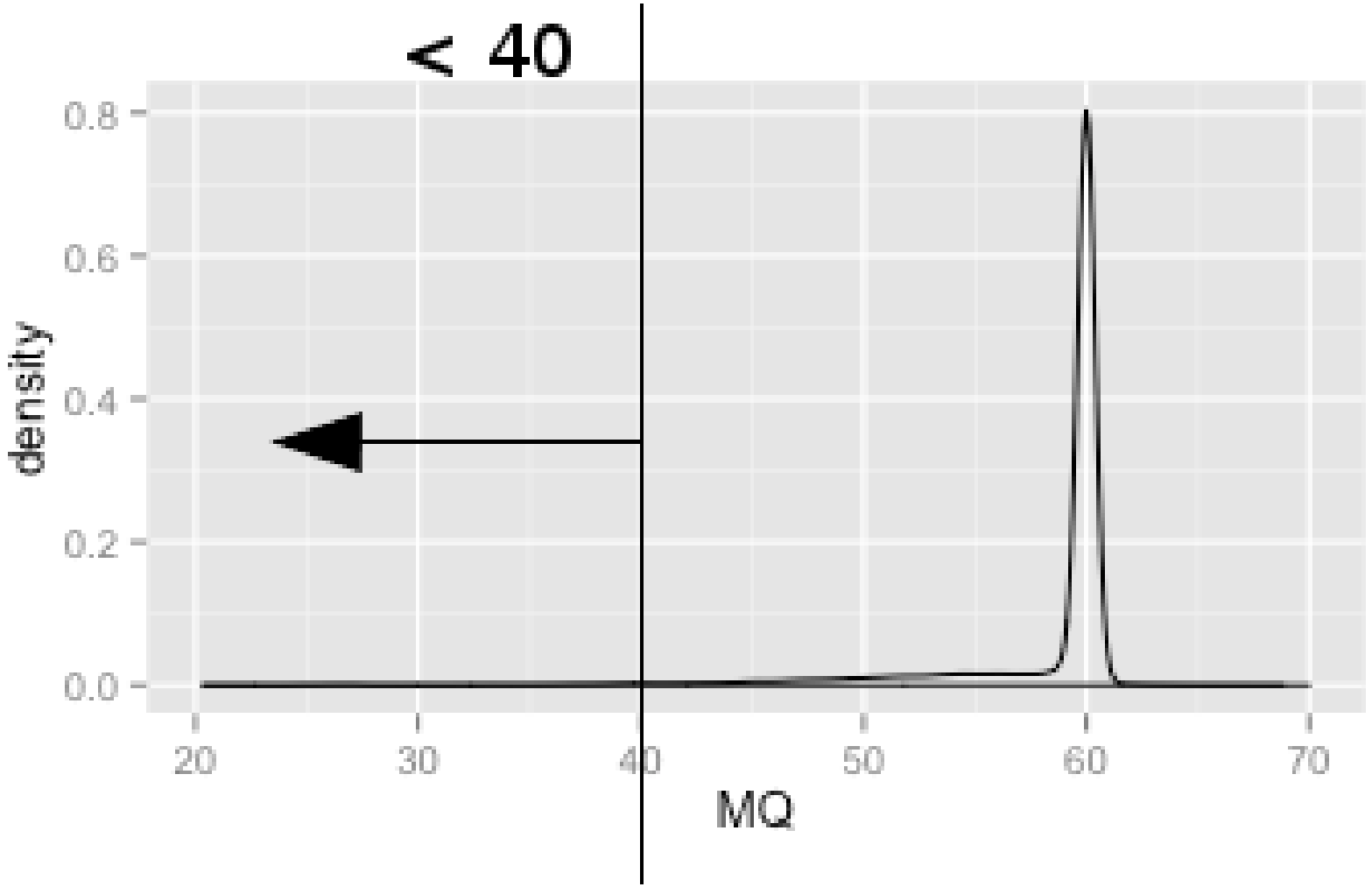
3 Variant quality/allele depth (QD)

1. Variant Quality (QUAL) is the phred scaled probability that the variant is wrong.
2. allele depth is actual depth of each observed allele (How many actual reads; in contrast to DP).



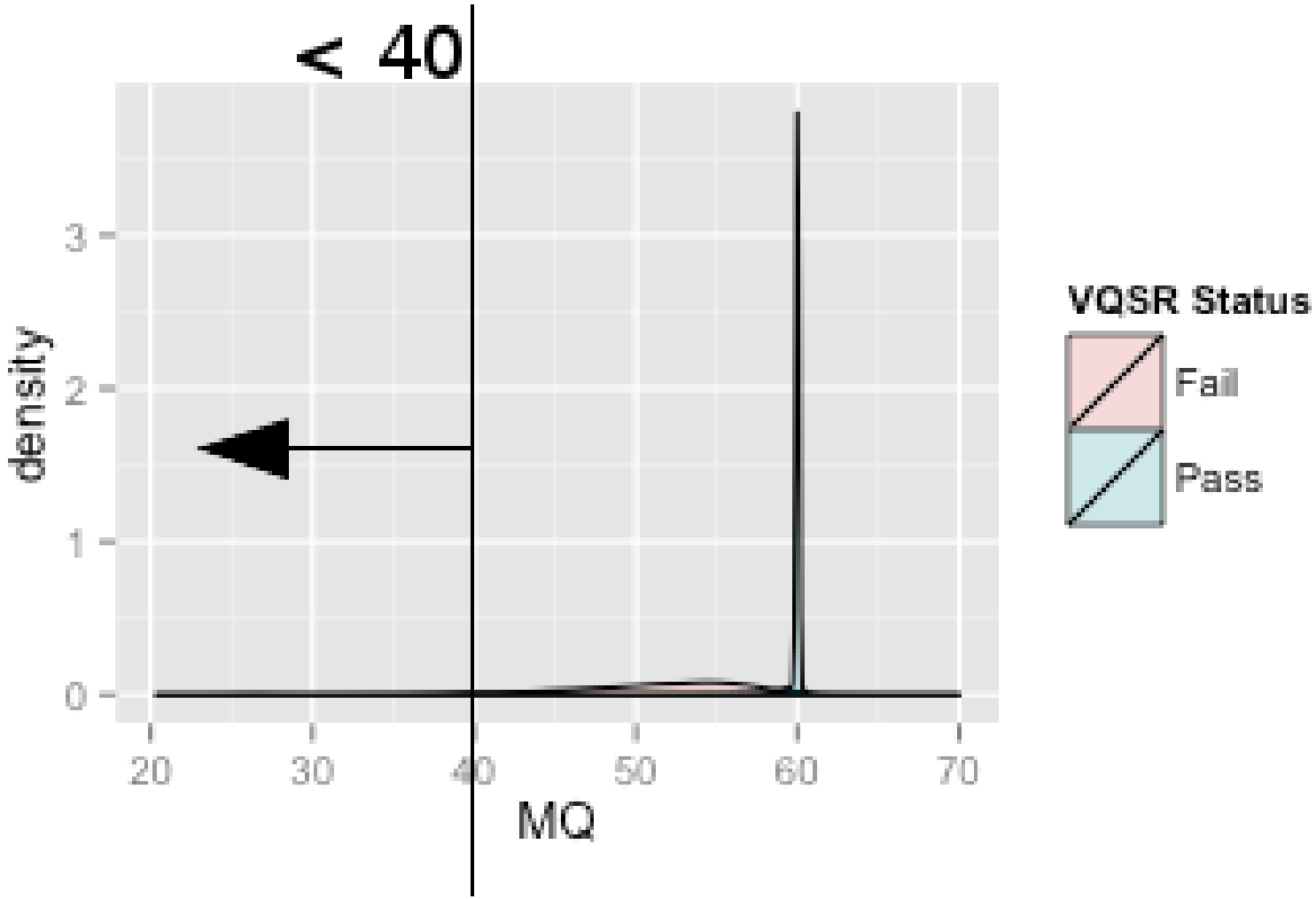
4 Root mean square mapping quality (MQ)

1. phred scaled probability that the mapping position is wrong



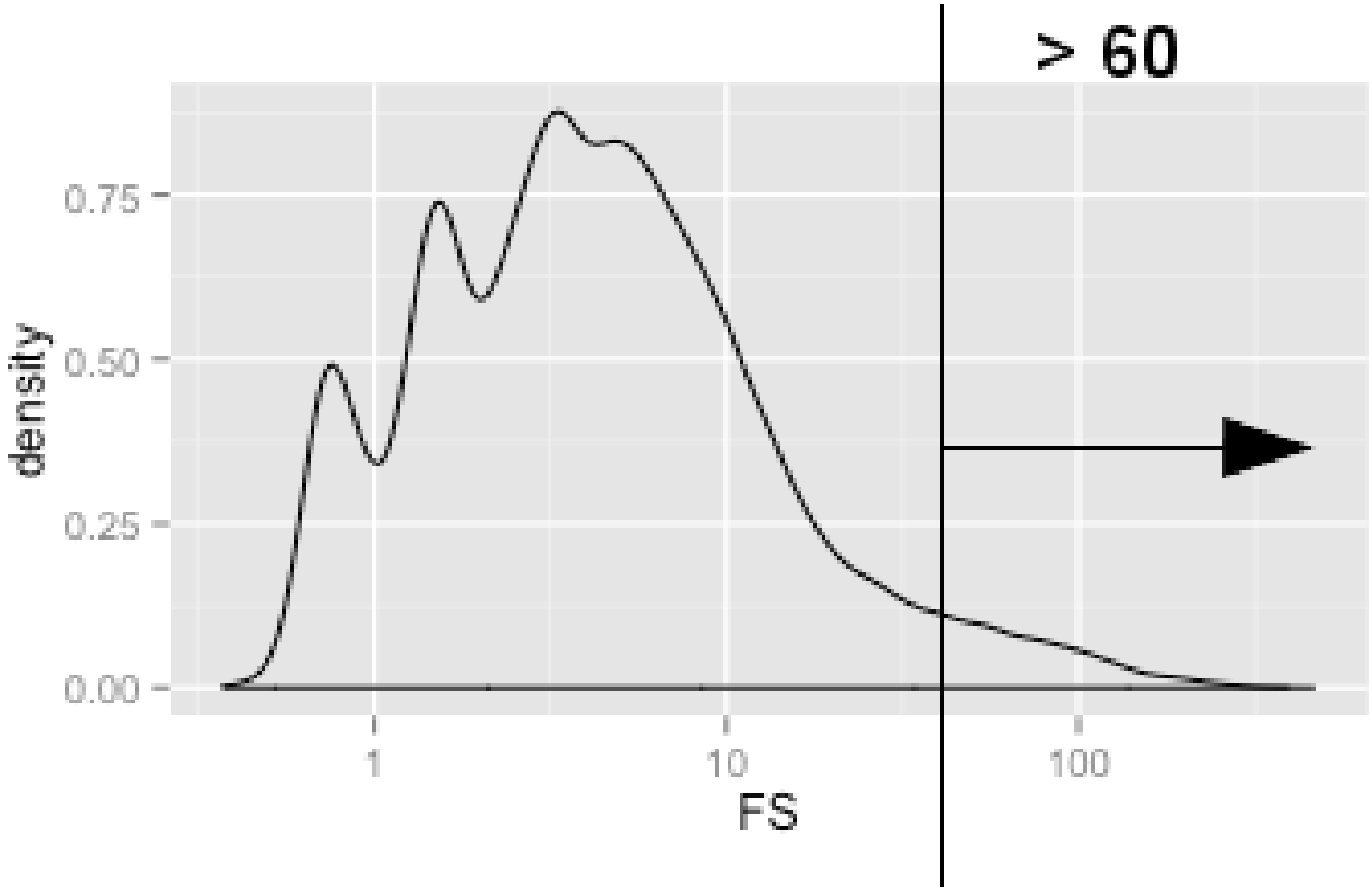
4 Root mean square mapping quality (MQ)

1. phred scaled probability that the mapping position is wrong



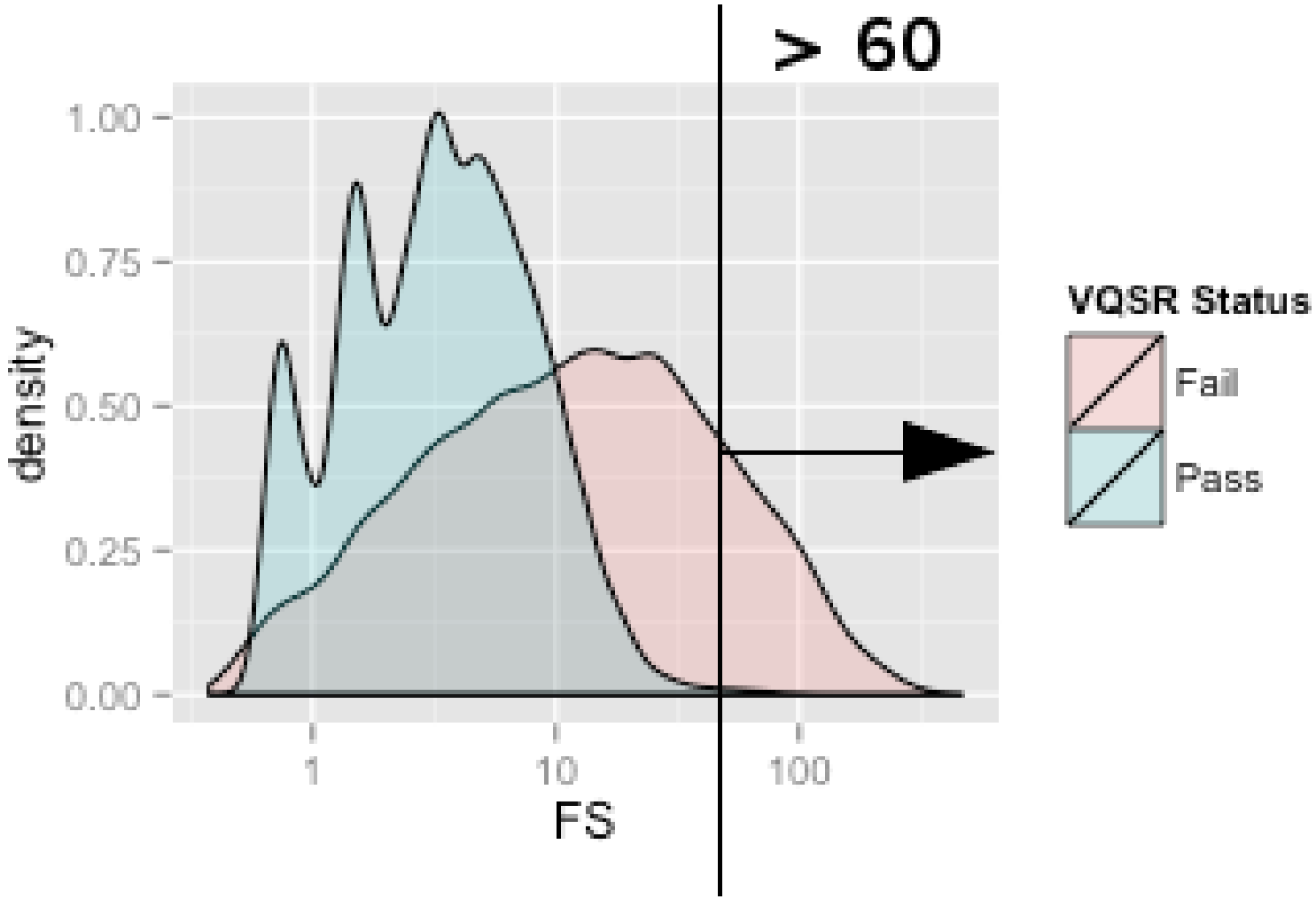
5 Fisher strand bias (FS)

1. phred scaled probability ALT on forward or reverse strand more or less than REF



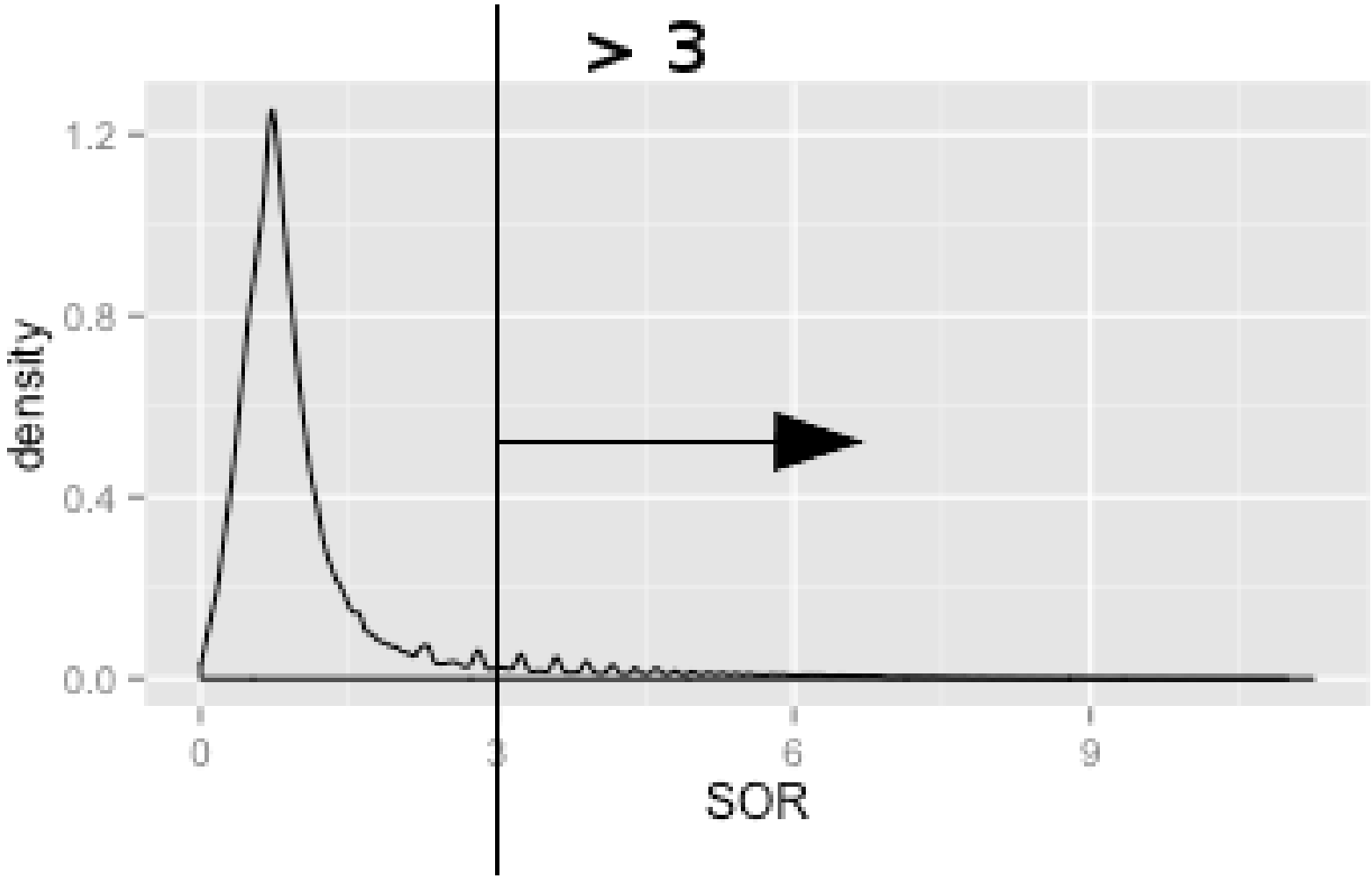
5 Fisher strand bias (FS)

1. phred scaled probability ALT on forward or reverse strand more or less than REF



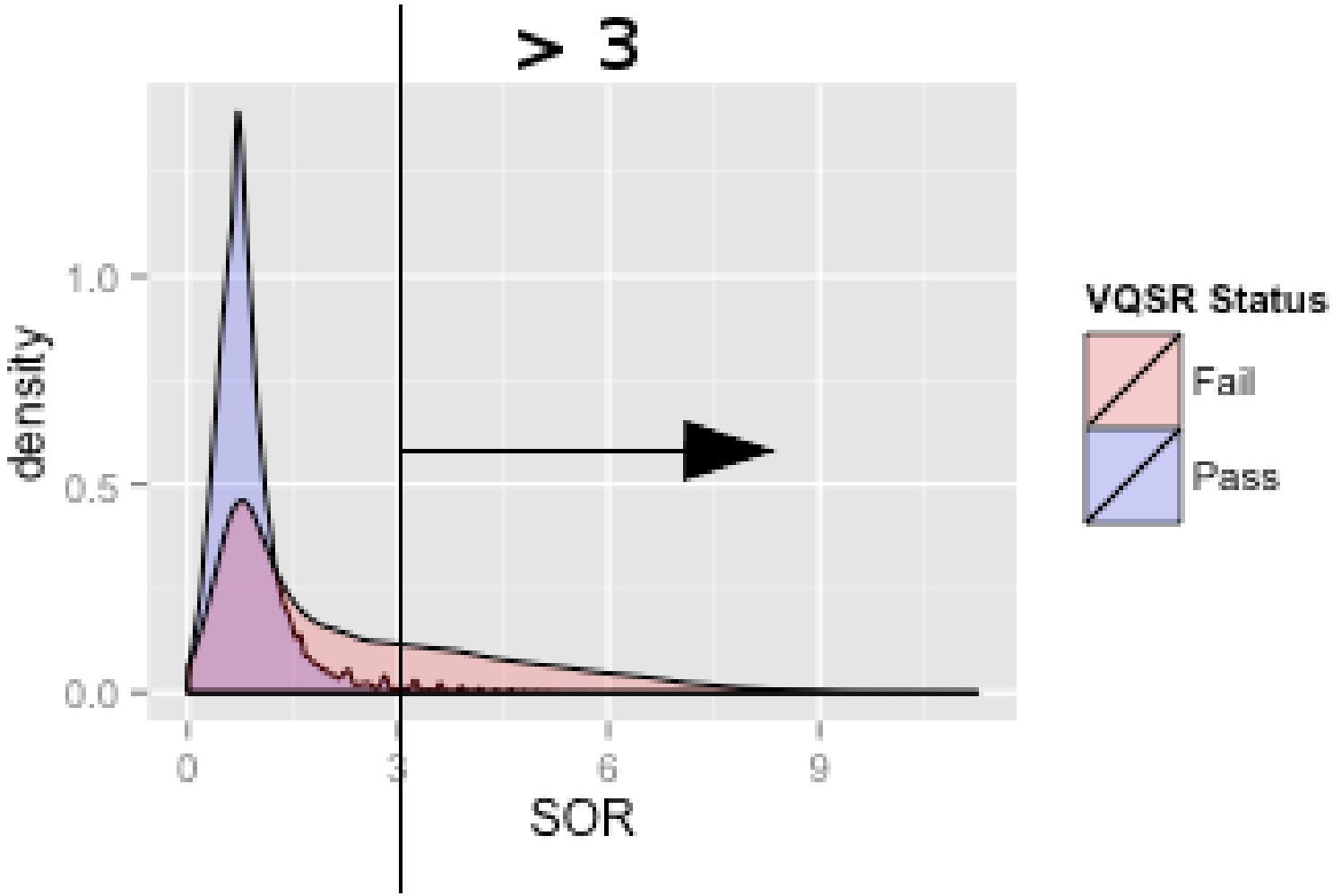
6 Strand odds ratio (SOR)

1. similar to FS, but updated for high coverage (NGS)
   - Ratio of reads that cover both alleles



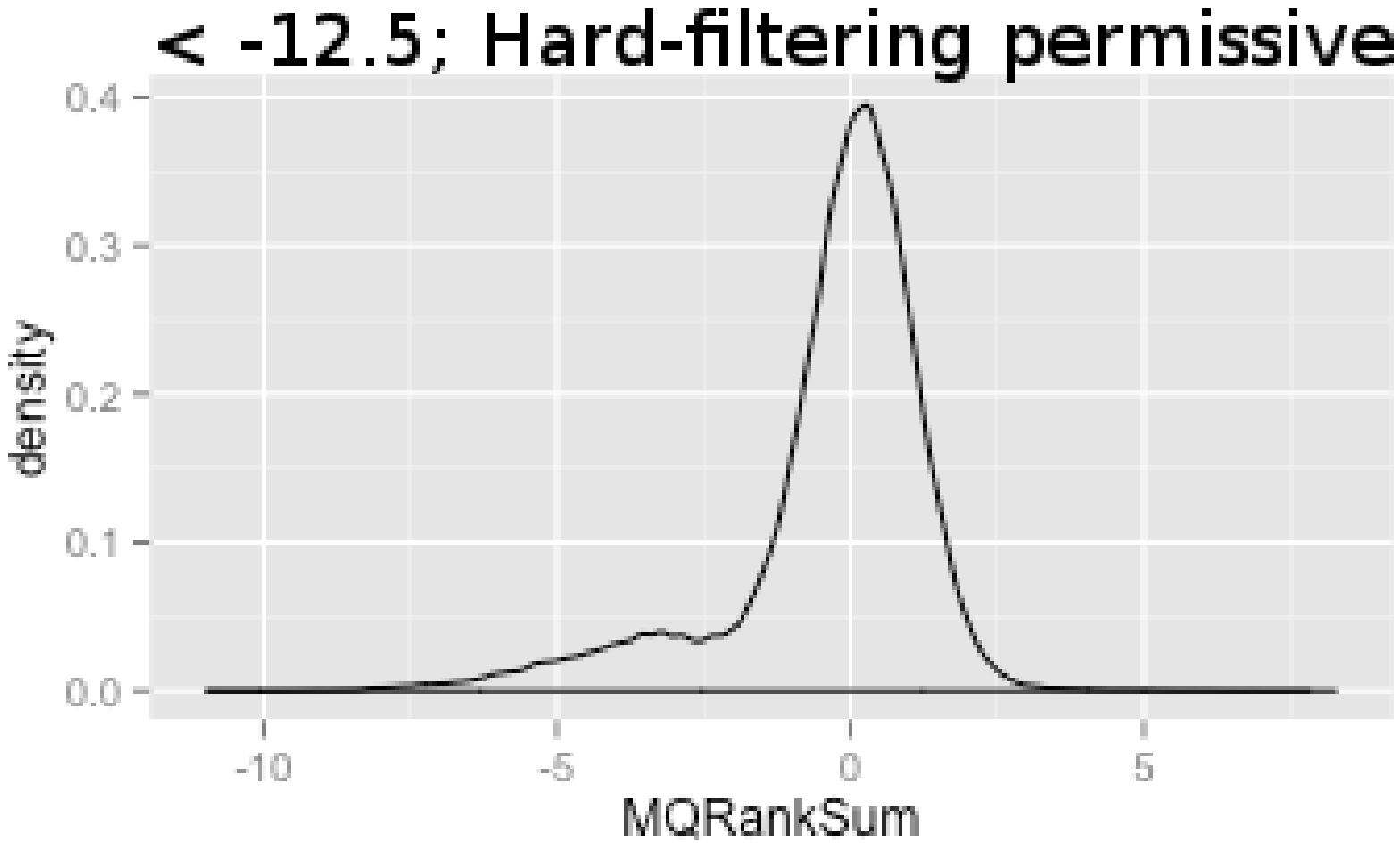
6 Strand odds ratio (SOR)

1. similar to FS, but updated for high coverage (NGS)
   - Ratio of reads that cover both alleles



7 MQ rank sum test (MQRankSum)

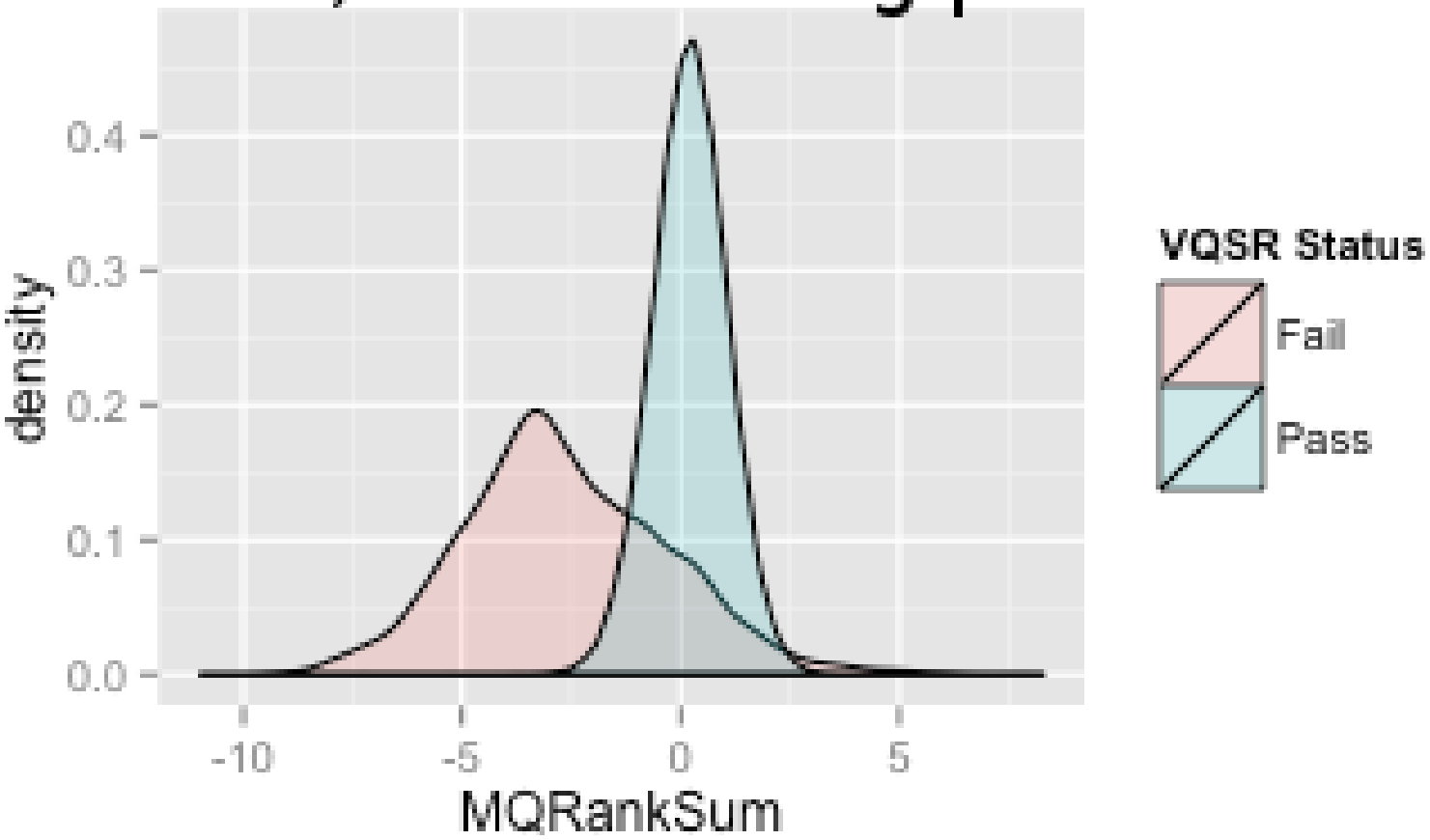
1. test compares MAQ ALT to REF
   - (-) Alt lower MAQ
   - (+) Ref lower MAQ



7 MQ rank sum test (MQRankSum)

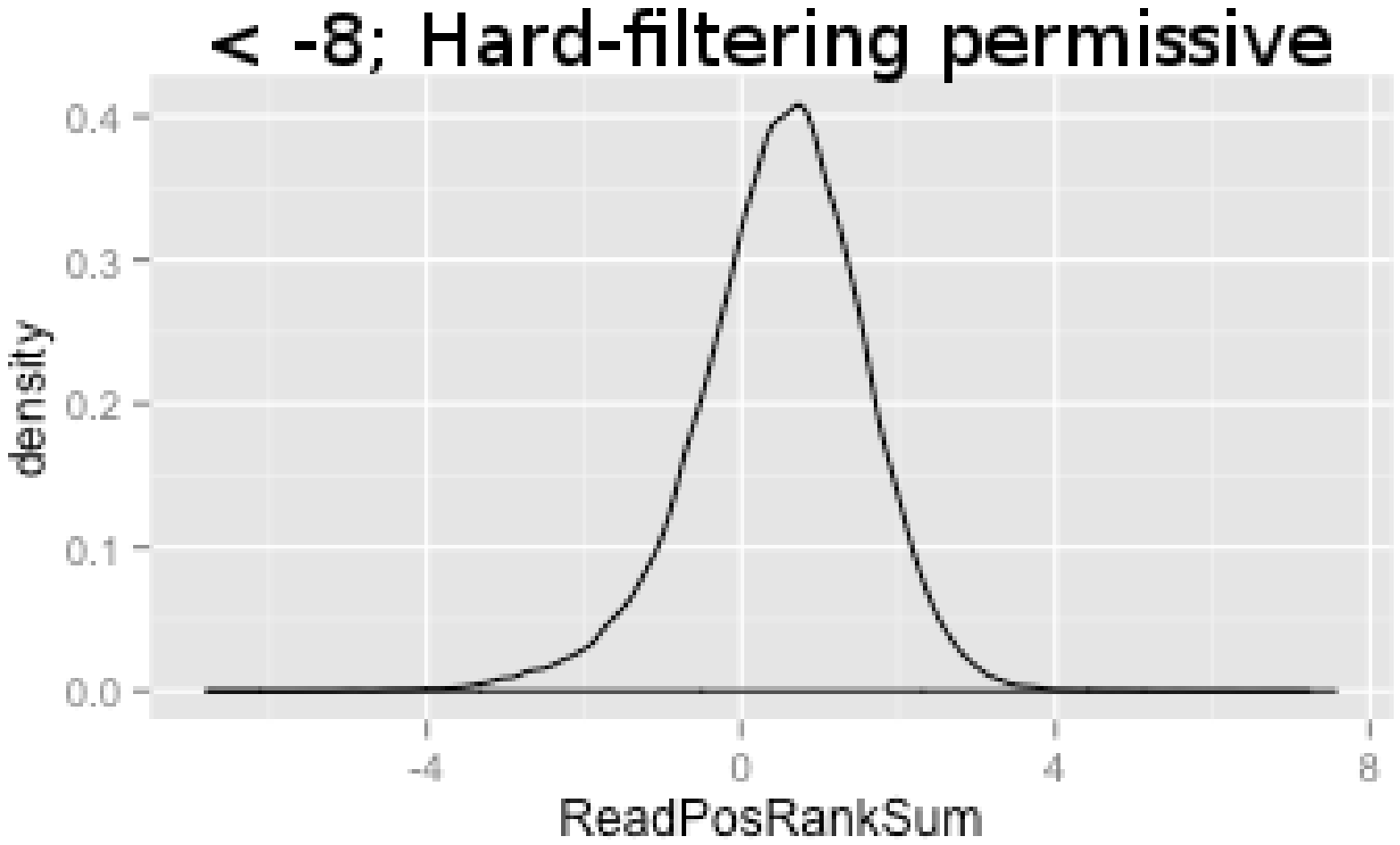
1. test compares MAQ ALT to REF
   - (-) Alt lower MAQ
   - (+) Ref lower MAQ

< -12.5; Hard-filtering permissive



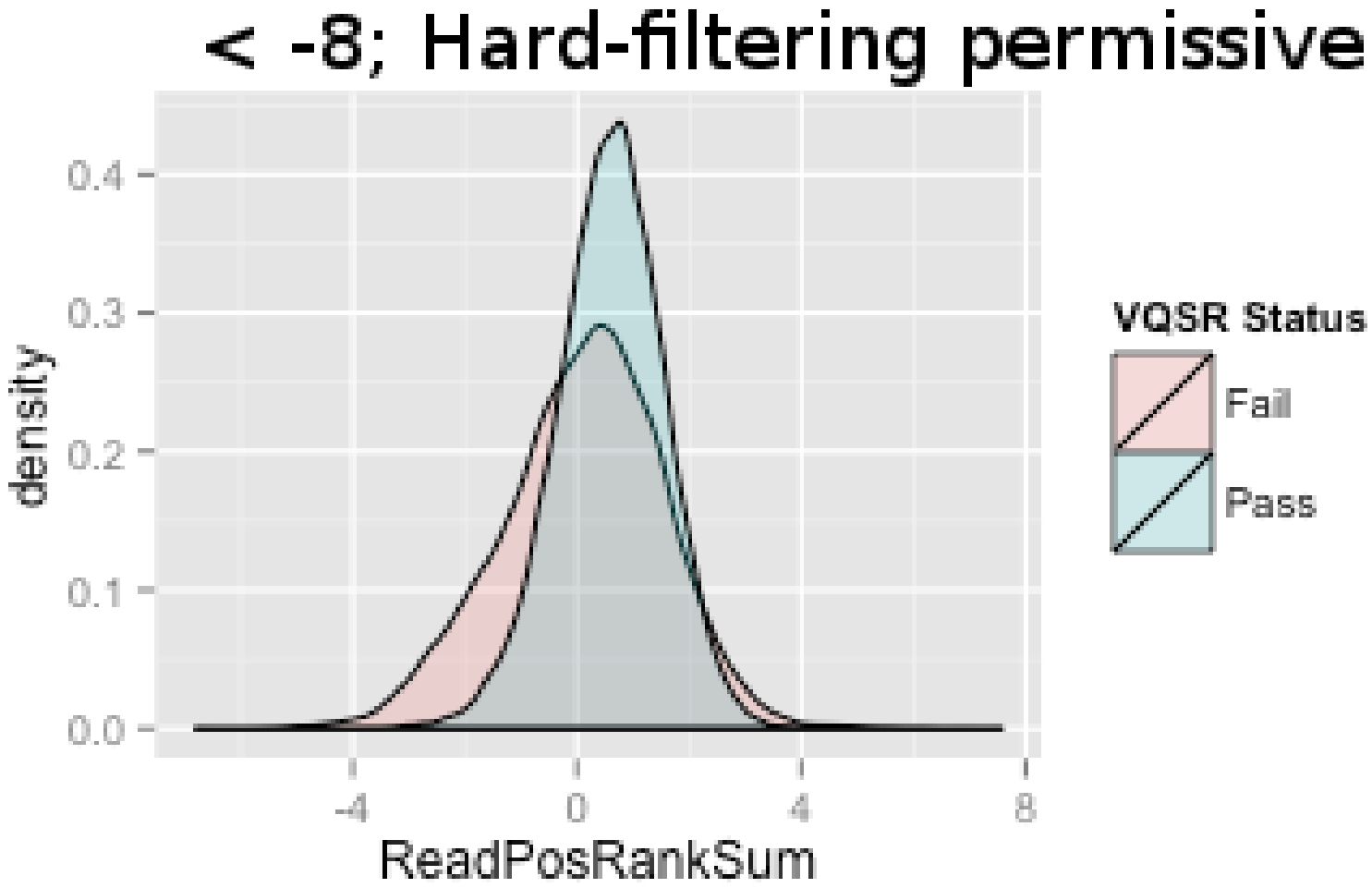
8 Read position rank sum test (ReadPosRankSum)

1. test for positional effects
   - (-) Alt close to end of read
   - (+) Ref close to end of read



8 Read position rank sum test (ReadPosRankSum)

1. test for positional effects
   - (-) Alt close to end of read
   - (+) Ref close to end of read



Hard-filtering Summary (SNPs and INDELS)

Variant	Annotation	Remove If
Both	DP	min=empirical; max=5 or 6 sigma
	GQ (or RGQ)	empirical
SNP	QD	< 2.0
	MQ	< 40
	FS	> 60
	SOR	> 3.0
	MQRankSum	< -12.5
	ReadPosRankSum	< -8.0
INDELs	QD	< 2.0
	ReadPosRankSum	< -20.0
	InbreedingCoeff (> 10 samples)	< -8.0
	FS	< 200.0
	SOR	> 10.0

Topic

Variant Annotation

1. Could write a script in favorite language.

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor
   - VariantAnnotation

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor
   - VariantAnnotation
     - general parsing and filtering

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor
   - VariantAnnotation
     - general parsing and filtering
3. Consistent interface

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor
   - VariantAnnotation
     - general parsing and filtering
3. Consistent interface
   - Learn 1 piece of software and reuse

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor
   - VariantAnnotation
     - general parsing and filtering
3. Consistent interface
   - Learn 1 piece of software and reuse
4. Custom filters

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor
   - VariantAnnotation
     - general parsing and filtering
3. Consistent interface
   - Learn 1 piece of software and reuse
4. Custom filters
   - flexible annotations (e.g., RGQ)

Variant Annotation

1. Could write a script in favorite language.
   - Know exactly what you did (+)
   - Time spent engineering software (-)
2. Hard Work already done
   - Bioconductor
   - VariantAnnotation
     - general parsing and filtering
3. Consistent interface
   - Learn 1 piece of software and reuse
4. Custom filters
   - flexible annotations (e.g., RGQ)
   - New annotations just "show up"

Topic

Extract, Filter, and Plot

1. Exercise folder on asc
   - Scripts: 1_initial_annotation_plot.sh; 2_filter_and_plot.sh
   - Data: D_PseudoFS14_16
   - UsefulBioinformaticScripts
2. Edit "Variables" in 1_initial_annotation_plot.sh

```
script_dir=${HOME}/Exercise/UsefulBioinformaticScripts
data_dir=${HOME}/Exercise/D_PseudoFS14_16
out_dir=${HOME}/Exercise
```

3. save script and run
4. Inspect graphs and decide upon filtering thresholds
5. add variable definitions in 2 to 2_filter_and_plot.sh
6. Edit

```
#################################################################################
##Filtering Parameters
##this is
${script_dir}/filter_SNPs_GATK_hard_filter.CHUNKS.R -I ${out_dir}/${vcf1}.gz -T
${out_dir}/${vcf1}.gz.tbi -O ${out_dir}/${vcf1}.filtered.vcf -C 10000 --QD=2
--FS=60 --SOR=3 --MQRankSum=-8 --min_Depth=4 --max_Depth=32 --Genotype_Quality=20

${script_dir}/filter_SNPs_GATK_hard_filter.CHUNKS.R -I ${out_dir}/${vcf2}.gz -T
${out_dir}/${vcf2}.gz.tbi -O ${out_dir}/${vcf2}.filtered.vcf -C 10000 --QD=2
--FS=60 --SOR=3 --MQRankSum=-8 --min_Depth=4 --max_Depth=32 --Genotype_Quality=20
```

7. save script and run
8. inspect graphs and write up.