RNA-SEQ VARIANT CALLING AND ALLELE-SPECIFIC EXPRESSION ANALYSIS

John Didion, PhD

RNA-seq Variant Calling and Allele-Specific Expression Analysis

OUTLINE

1. Calling variants from RNA-seq data
2. Allele-specific expression concepts
3. Correcting for mappability bias using WASP
4. Identifying allele-specific expression using ASARP
5. Interpreting results of allele-specific expression analysis

RNA-seq Variant Calling and Allele-Specific Expression Analysis

OUTLINE

1. Calling variants from RNA-seq data
2. Allele-specific expression concepts
3. Correcting for mappability bias using WASP
4. Identifying allele-specific expression using ASARP
5. Interpreting results of allele-specific expression analysis

Variant Calling

- The human reference genome is haploid – one nucleotide per position
- NGS is (essentially) the process of transforming molecules (e.g. DNA or RNA) in a biological sample into sequence alignments against the reference genome (a BAM file)
- Variant calling is (essentially) the process of identifying the positions where one or both parental alleles in the sample differ from the reference genome

Variant Calling is a Two-Step Process

1. Variant discovery
   - Identify positions at which there is variation within a population
   - This has been a primary focus of the genomics field; collectively, global human variant discovery efforts have identified ~150M SNPs
   - Each new individual that is sequenced harbors a few dozen variants that have never been seen before

Variant Calling is a Two-Step Process

2. Variant genotyping
   - Determine the two parental alleles at each potentially variable position
   - For a bi-allelic SNP, there are three possible genotypes:
     - Homozygous reference (aa)
     - Homozygous alternate (AA)
     - Heterozygous (aA)
   - Genotypes may be phased. If two alleles are in-phase, the both originated from the same molecule (haplotype).

Sources of Genotype Information

- Genotype array
  - Pros: easy, minimal input requirement
  - Cons: only types a fraction of sites in the genome
  - Cost: $
  - Untyped variants can be imputed using a probabilistic method and a panel of reference haplotypes
- Whole-exome sequencing (WES)
  - Pros: type all variants in coding sequences
  - Cons: multiple sources of bias
  - Cost: $$
- Whole-genome sequencing (WGS)
  - Pros: minimal bias
  - Cons: more complex analysis, large data storage requirements
  - Cost: $$$

Calling Variants from RNA-seq

- If you already have RNA-seq data, it's (essentially) free. Two-for-one!
- Provides similar genotype data to WES
- **Both false-negative and false-positive variant calls are inflated relative to DNA-derived data**

Interlude: Allelic Imbalance (AI)

- Unequal transcription rates from the two chromosomal alleles in a cell
- Biological sources of AI in a cell:
  - Chromatin differences between the two chromosomes
  - Differential transcription factor binding due to genetic variation in binding sites
  - Differential polymerase efficiency
  - Differential splicing (differential stability of transcripts)
  - Temporal (*e.g.* cell cycle) effects
- These effects can be due to genetic differences between alleles, environmental variables, or stochastic variation

Interlude: Allelic Imbalance (AI)

- Cell-to-cell variation is averaged out in bulk mRNA sequencing; unless you're using single-cell RNA-seq, only systematic AI is detectable
- However, technical artifacts can either give the false impression of AI or mask true AI. Transcripts from the two alleles can differ in terms of
  - cDNA conversion efficiency
  - PCR amplification efficiency
  - Mappability

Mitigating technical artifacts

- Sequencing
  - Use high-quality reagents, especially for cDNA conversion
  - Reduce/eliminate PCR amplification (requires large amount of input material)
  - Use long (100bp or more) paired-end reads
- Read mapping
  - Mark PCR duplicates
  - Filter alignments for mapping quality (MAPQ)
- Downstream analysis
  - Correct for mappability bias

Sources of Error in RNA-seq Genotypes

- Sequencing errors introduce low-frequency false-positive variants
- Variant calling algorithms try to reduce false-positives by ignoring low-frequency variants
- However, AI (whether true or false) results in unequal representation of alleles
- **In other words, variant callers make an assumption of equal allelic representation that is violated in RNA-seq**
- Thus, variant calling parameters must be adjusted to allow for AI when calling heterozygous genotypes
- Additional caveats:
  - RNA editing may introduce false-positive variants
  - Variant calls will be wrong for genes that are imprinted, *i.e.* expressed from only one allele

Mitigating RNA-seq Genotype Errors

- Variant calling
  - Ignore duplicate reads
  - Excluded known RNA-editing sites, variants near splice junctions, and variants at repeat regions
  - Additional filters can be applied, such as excluding variants with allele frequencies that differ substantially from expectation
- Downstream analysis
  - Exclude known imprinted genes

INTERLUDE: THE COMMAND LINE

Conventions

Comments preceded by hash

$ Command prompt starts with a dollar sign

VARIABLES_UPPERCASE_ITALICS

Single line command continued \
onto next line by a backslash

INTERLUDE: THE COMMAND LINE

Moving Around the Filesystem

```
# what directory am I in now?
$ pwd
# navigate to where the example files live within
# your home directory
$ cd ~/rnaseqvariant
# what files are in the current directory?
$ ls –la # -l (long mode), -a (show .files)
# make an output folder
$ mkdir output
# view contents of a file
$ less FILENAME # page through file
$ head -N FILENAME # show first N lines
$ tail –N FILENAME # show last N lines
```

INTERLUDE: THE COMMAND LINE

More Useful Commands

```
# <tab> to auto-complete current command
# <up arrow> and <down arrow> to see previous commands
# <Ctrl-r> to search previous commands

# cat concatenates two or more files
# | sends output of one command to another command
# wc -l counts the number of lines
# > sends command output to a file
$ cat FILE1 FILE2 | wc –l > FILE

# read the manual for a command
$ man COMMAND
```

Exercise: RNA-seq Variant Calling

Software

- bcftools is a variant calling toolkit
- We will `bcftools mpileup` to count the occurrence of each allele at each genomic position, and to compute the likelihood of each genotype given the observed allele counts
- We will use `bcftools call` to call genotypes from the likelihoods
- We will combine these tools into a "pipeline"

Exercise: RNA-seq Variant Calling

Data

- We are using the ASARP demo data, which contains alignments at 3 small regions
- A sample pipeline for aligning and filtering the reads can be found in the handout

Exercise: RNA-seq Variant Calling

```
# variant discovery and genotyping pipeline
$ bcftools mpileup -f ref/hg19.fasta -q 20 --ff DUP \
-a FORMAT/AD,FORMAT/DP input/input.bam | \
bcftools call -c -v -O z -o output/variants.vcf.gz

# here's what the mpileup options mean:
-f  # reference genome FASTA file
-q  # only use reads with MAPQ >= 20
--ff DUP  # ignore duplicate reads
-a  # add additional information fields to VCF output

# some additional mpileup parameters:
-C 50  # recommended for BWA alignments
-d 10000 # may be necessary for very deep sequencing
```

Exercise: RNA-seq Variant Calling

```
# variant discovery and genotyping pipeline
$ bcftools mpileup -f ref/hg19.fasta -q 20 --ff DUP \
-a FORMAT/AD,FORMAT/DP input/input.bam | \
bcftools call -c -v -O z -o variants.vcf.gz

# here's what the bcftools options mean:
-c  # use the consensus caller, which assumes
    # bi-allelic variants; don't use for standard
    # DNA-seq variant calling!
-v  # only output variant sites
-O z  # output gzip-compressed output
-o variants.vcf.gz  # the output file name
```

VARIANT CALLING

VCF Header

```
##fileformat=VCFv4.2
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="…">
…
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="…">
…
```

- FILTER: marker that the variant failed a condition
- INFO: information given for each variant
- FORMAT: information given in each genotype record

VARIANT CALLING

VCF Variant Rows

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  SAMPLE1  SAMPLE2...
chr1    51479   rs123   T       A       3442.03 PASS    AC=5;AF=0.404;AN=20
GT:AD:DP:GQ     0/0:14,0:14:42  0/0:2,0:2:6
```

- ID: If variant is known, it's database ID (e.g. dbSNP)
- REF, ALT: the reference and alternate alleles
  - ALT may contain more than one allele, comma-separated
- QUAL: PHRED-scale probability that there is no variant at this site

VARIANT CALLING

VCF Variant Rows

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  SAMPLE1  SAMPLE2...
chr1    51479   rs123   T       A       3442.03 PASS    AC=5;AF=0.404;AN=20
GT:AD:DP:GQ     0/0:14,0:14:42  0/1:2,2:4:6
```

- For each sample:
  - GT: genotype call; 0 is reference, and 1, 2, … are alternate
    - 0/0 is homozygous reference
    - 0/1 is heterozygous
    - 1/1 is homozygous alternate
    - '.' means no call
    - If there is a '|' rather than '/', the genotype is phased
  - GQ: PHRED-scale probability that the call is incorrect

Exercise: RNA-seq Variant Calling

Filtering

- We will use `bcftools annotate` to annotate variants based on whether they are near splice sites or are known RNA editing sites
- We will use `bcftools filter` to remove annotated variants
- This annotation file is provided for you (annotations/filter_sites.bed)

Exercise: RNA-seq Variant Calling

```
# annotate problematic sites
# -c tells what columns to use for matching for BED file
# -m adds a “EXCLUDE” tag in the info field
$ bcftools annotate -a annotations/filter_sites.bed.gz \
-c CHROM,-,POS -m +EXCLUDE -O z \
-o outputs/variants_annotated.vcf.gz \
outputs/variants.vcf.gz

# check that variants have been tagged
$ zcat results/variants_annotated.vcf.gz| grep EXCLUDE

# filter out sites annotated with a “EXCLUDE” tag
# -e gives an expression specifying which sites to exclude
$ bcftools filter -e EXCLUDE -O z \
-o outputs/variants_filtered.vcf.gz \
outputs/variants_annotated.vcf.gz

# check that annotated variants are gone
$ zcat results/variants_filtered.vcf.gz| grep EXCLUDE
```

RNA-seq Variant Calling and Allele-Specific Expression Analysis

OUTLINE

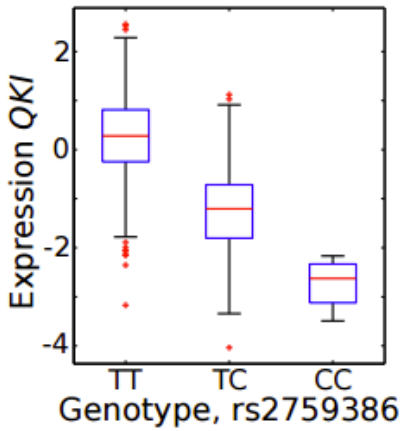
1. Calling variants from RNA-seq data
2. Allele-specific expression concepts
3. Correcting for mappability bias using WASP
4. Identifying allele-specific expression using ASARP
5. Interpreting results of allele-specific expression analysis

Allele-Specific Expression (ASE)

- General term for RNA-seq analyses that attempt to identify exons or genes with true allelic imbalance
- ASE is always relative to one or more genetic variant(s)
- The most common analysis is individual-specific ASE:
  - Identify coding SNPs with significant AI (ASE SNPs)
  - Power is increased by aggregating SNP-level information at the exon or gene level (ASE exons/genes)
  - With multiple samples, can identify ASE SNPs/exons/genes that are common or different among individuals, conditions, and/or tissues

Expression QTL

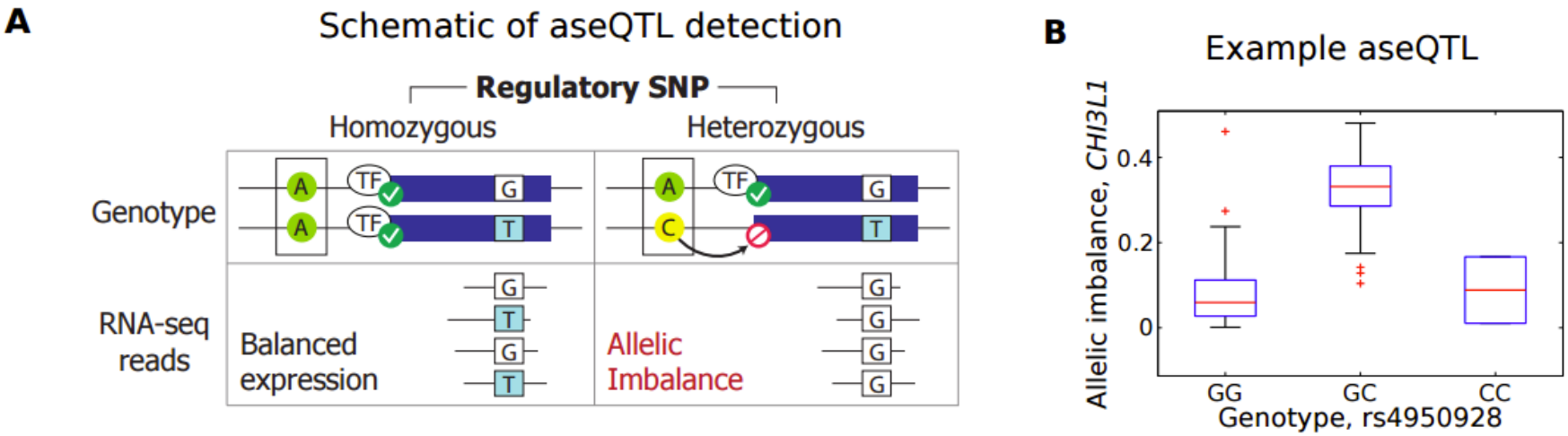
- An expression quantitative trait locus (eQTL) is a variant that is significantly associated with the expression level of a gene
  - When the associated SNP and gene are nearby (e.g. within 100kb), it is considered a *cis*-eQTL
  - When they are far away (e.g. >1MB), it is considered a *trans*-eQTL



Battle et al. 2013, Fig 3C

Allele-Specific eQTL

- Allele-specific eQTL (aseQTL) are SNPs at which the homzogyous and heterozygous genotypes are associated with significantly different AI at a nearby gene
- Many (hundreds) of individuals are required



Battle et al. 2013, Fig 1

Workflow for Individual ASE

- Generate RNA-seq data
- Map and filter reads to reference genome
- For the same sample, either
  - Genotype DNA (microarray),
  - Sequence DNA (WGS or WES), or
  - Call variants from RNA-seq
- Filter variants
- Correct RNA-seq alignments for mappability bias
- Use corrected RNA-seq alignments to test for ASE at variant sets
- Optionally, aggregate SNP-level information to identify ASE exons and/or genes

RNA-seq Variant Calling and Allele-Specific Expression Analysis

OUTLINE

1. Calling variants from RNA-seq data
2. Allele-specific expression concepts
3. Correcting for mappability bias using WASP
4. Identifying allele-specific expression using ASARP
5. Interpreting results of allele-specific expression analysis

Mappability Bias

- NGS reads that map equally well to multiple locations ("multi-mapping") likely originate from repetitive sequence (e.g. centromeres, LINEs, tandem repeats). These reads will either fail to align, or will have low mapping quality scores (MAPQ) and filtered out.
  - RNA-seq reads are less likely to be multi-mapping than WGS reads.
- Heterozygous variants increase the probability that reads from different alleles will map to different genomic locations.
- The read with more reference alleles has a higher probability of mapping to the correct location (mappability bias), and thus reference allele counts are prone to inflation in ASE analysis.

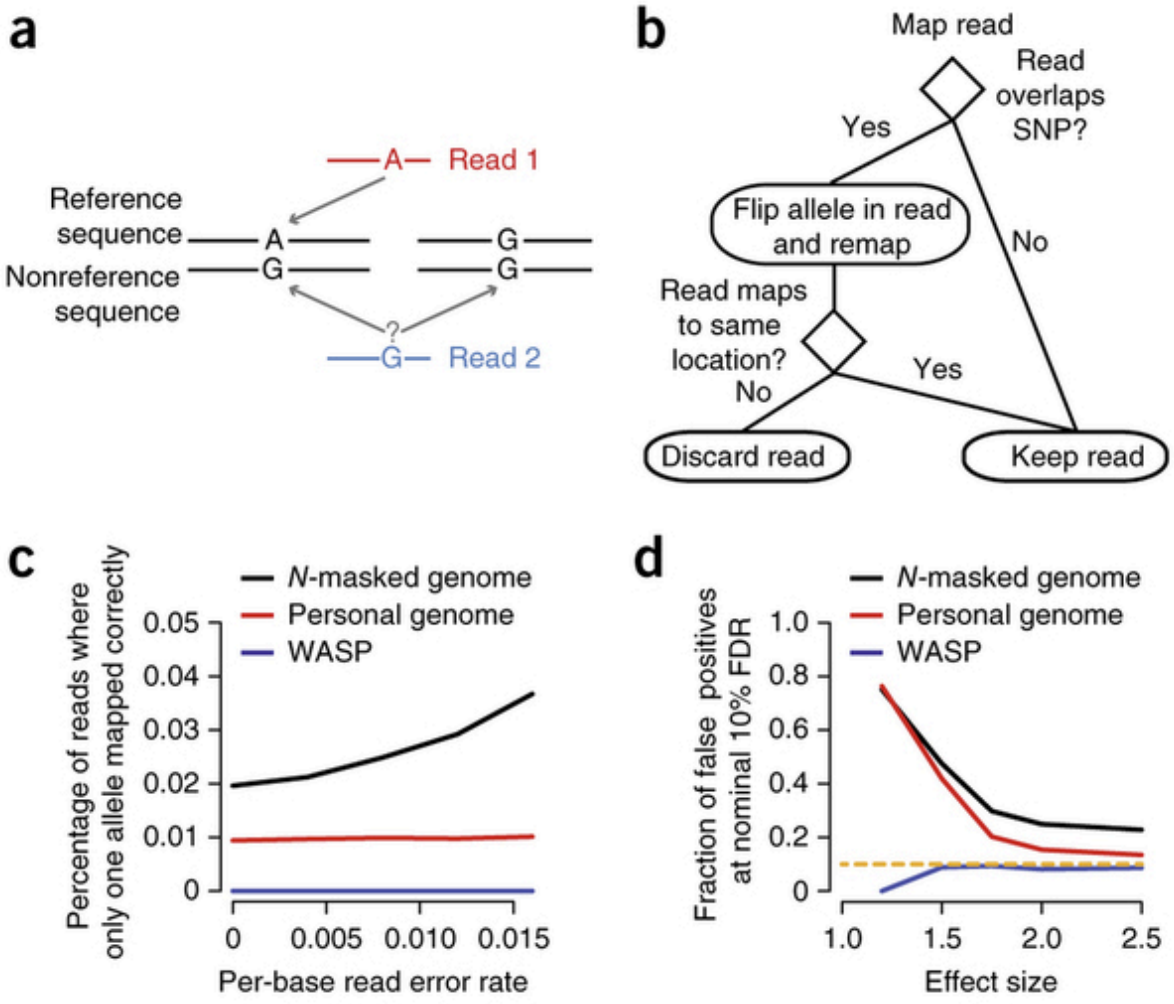
The Problem with RNA-seq Variant Calls

- In extreme cases of mappability bias, all reads from one allele will map to a different location than the reads from the other allele.
- Thus, the variant caller will see only one allele, and will call a homozygous genotype.
- Since mappability bias can only be discovered at heterozygous sites, the bias will be missed and affect downstream ASE analysis.
- If ASE is an important part of your experimental design, it is **strongly** recommended to independently genotype your subjects from DNA samples.

Correcting for Mappability Bias

- Software: WASP
- Consists of two tools:
  - Bias correction pipeline
  - Combined Haplotype Test pipeline for aseQTL analysis
- WASP pipelines are run via Snakemake – a general-purpose tool you can use to create your own pipelines. Highly recommended!

Correcting for Mappability Bias



Van de Geijn et al. 2015, Fig. 1

Correcting for Mappability Bias

Two ways to run mappability bias correction

1. Using phased genotypes
   - Requires either:
     - Known parental genotypes, *e.g.* in a trio experimental design
     - A panel of reference haplotypes, *e.g.* 1000 Genomes Project
   - Use software such as SHAPEIT or fastPhase to phase your samples based on the reference panel
   - This is more complicated but leads to more accurate results
2. Using unphased genotypes: we will use this approach for simplicity, but phasing is recommended for real analysis!

Exercise: Correcting Mappability Bias with WASP

```
# Execute the pipeline
# If you have phased genotypes, use
# “snakefile.phased” instead.
$ cd ~/rnaseq/variant/WASP
$ snakemake -s snakefile.unphased
```

Correcting for Mappability Bias

WASP Output

```
# There will now be several subdirectories in your
# output directory. Most are intermediate files that can
# be deleted.

# remove unsorted files from rmdup dir:
$ ls rmdup/ | grep -v sort | xargs rm

# remove intermediate files and directories:
$ rm -rf map1 map1_sort find_intersecting_snps \
Map2 map2_sort filter_remapped_reads merge
```

Correcting for Mappability Bias

WASP Output

```
# The starting input file was
# input/map1_sort/18501.bam
# and the final output file is:
# output/rmdup/18501.keep.merge.rmdup.sort.bam
# Let's see how many reads were removed.
$ samtools view input1/map1_sort/18501.bam | wc –l
$ samtools view \
output/rmdup/18501.keep.merge.rmdup.sort.bam \
| wc -l
```

RNA-seq Variant Calling and Allele-Specific Expression Analysis

OUTLINE

1. Calling variants from RNA-seq data
2. Allele-specific expression concepts
3. Correcting for mappability bias using WASP
4. Identifying allele-specific expression using ASARP
5. Interpreting results of allele-specific expression analysis

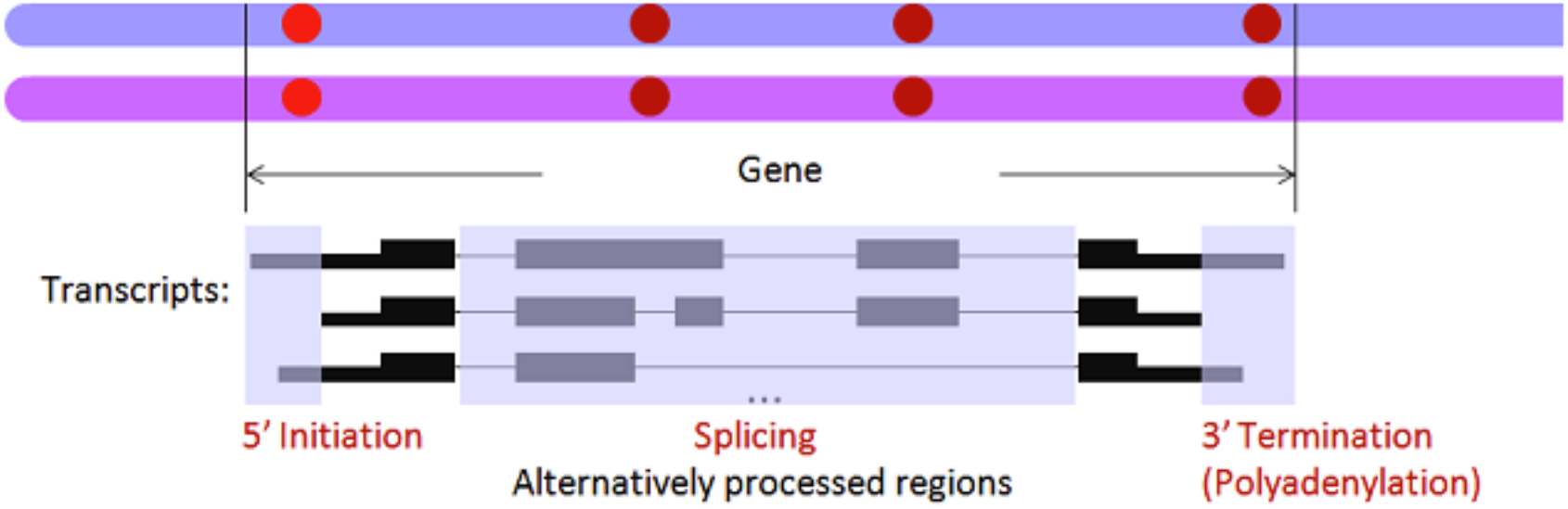
Identifying ASE with ASARP

- ASARP tests all heterozygous SNPs for ASE
- P-values are controlled for FDR (0.05 by default)
- If all heterozygous SNPs in a gene that are above a certain read-count threshold (20 by default) exhibit significant ASE, it is considered an ASE gene
- Otherwise, ASARP tests each SNP in the gene for other allele-specific events

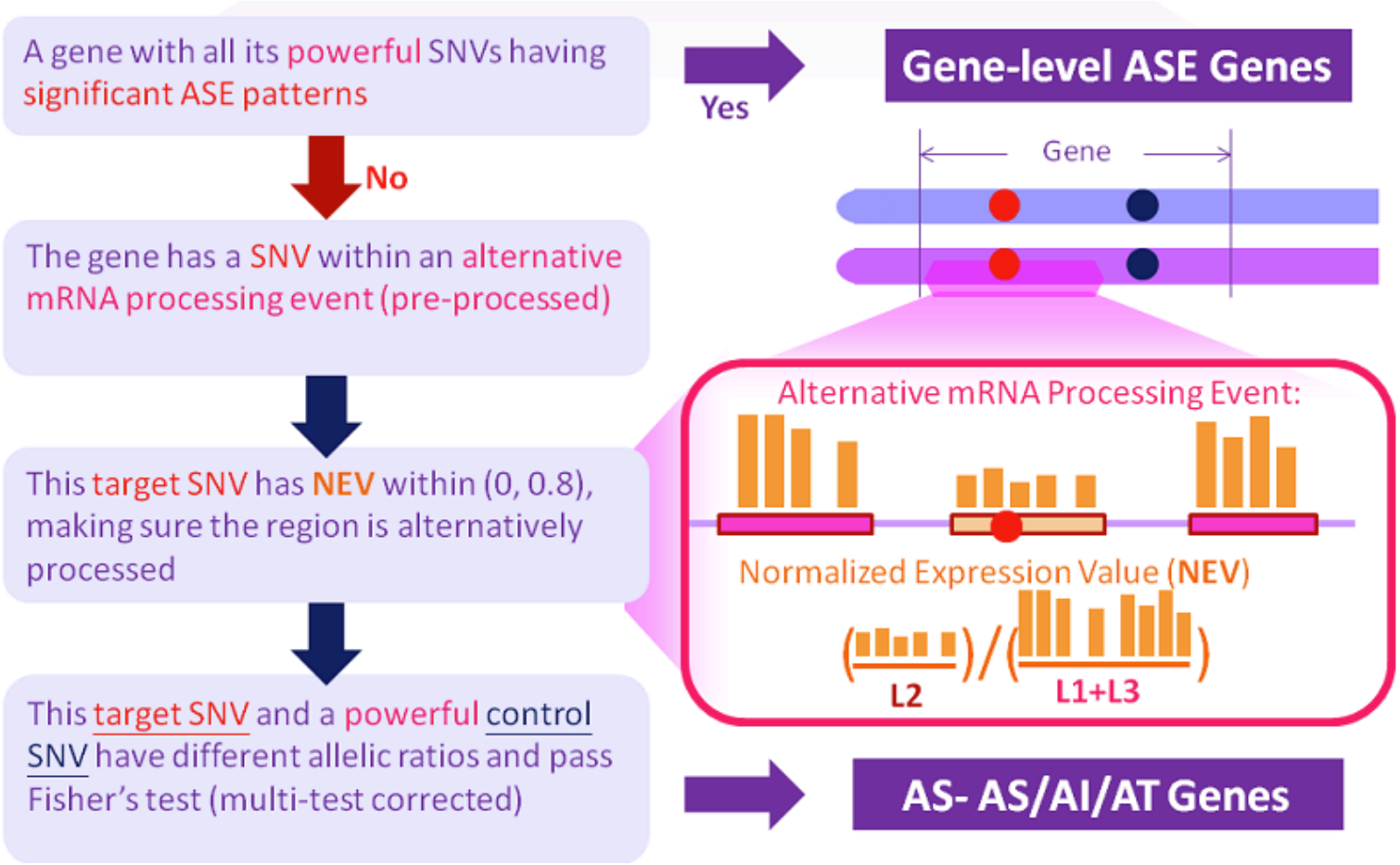
Identifying ASE with ASARP

In addition to ASE, ASARP detects Allele-Specific…

- Alternative Splicing (ASAS): SNP in an exon whose splice-in rate differs between alleles
- Transcription Initiation (ASTI): SNP in 5' UTR whose TSS differs between alleles
- Alternative Polyadenylation (ASAP): SNP in 3' UTR whose termination site differs between alleles



Identifying ASE with ASARP



Powerful = high-coverage (>= 20)

AS = Alternative Splicing; TI = Transcription Initiation; AP = Alternative Polyadenylation

Exercise: Idenify ASE with ASARP

```
# Step 1: ASARP processes one chromosome at a time and
# reads must be in SAM format and name-sorted
$ for chr in 1 5 10 ; do
samtools view -b input/input.bam \
chr$chr > input/chr$chr.bam
samtools sort -n -O SAM -o input/chr$chr.namesort.sam \
--reference ref/hg19.fasta input/chr$chr.bam
done
```

Exercise: Idenify ASE with ASARP

```
# Step 2: Preprocess reads. The last two options tell
# the program that our data is paired-end and
# strand-specific (make sure to set this correctly for
# your library conditions!)
$ for chr in 1 5 10 ; do
perl -I /usr/local/bioinf/ASARP \
/usr/local/bioinf/ASARP/procReads.pl chr$chr \
input/chr$chr.namesort.sam input/dna.snv.list \
output/chr$chr.candidate_snvs \
output/chr$chr.expression.bedgraph 1 2
done
```

Exercise: Idenify ASE with ASARP

```
# Step 3: merge SNVs for analysis
$ perl -I /usr/local/bioinf/ASARP \
/usr/local/bioinf/ASARP/mergeSnvs.pl \
output/ .candidate_snvs mono=0 output/rna.snv.lst 1

# Step 4: run ASARP
$ perl -I /usr/local/bioinf/ASARP \
/usr/local/bioinf/ASARP/asarp.pl \
output/asarp_output input/asarp.config \
input/asarp.params
```

ASARP Results

- ASARP generates four output files:
  1. output_file.ase.prediction: the detailed results of (whole-gene-level) ASE patterns (exclusive to other ASARP patterns: AI, AS or AT)
  2. output_file.gene.prediction: the detailed results of ASARP results (ASE patterns excluded) arranged by genes
  3. output_file.snv.prediction: the detailed results of ASARP results (ASE patterns excluded) of each individual SNV
  4. output_file.controlSNV.prediction: the control SNV information of each individual ASARP SNV

RNA-seq Variant Calling and Allele-Specific Expression Analysis

OUTLINE

1. Calling variants from RNA-seq data
2. Allele-specific expression concepts
3. Correcting for mappability bias using WASP
4. Identifying allele-specific expression using ASARP
5. Interpreting results of allele-specific expression analysis

Interpreting ASE results

- ASE SNP analysis
  - Annotate SNPs with public datasets using annovar
  - Annotate SNPs with predicted functional impact using variant effect predictor (VEP)
  - Intersect SNPs with GWAS catalog to identify possible disease associations
  - Intersect SNPs with ClinVar to identify known disease-causing variants

Interpreting ASE results

- ASE gene analysis
  - Identify biological processes or pathways enriched in the gene list using Gene Ontology (GO) or KEGG
  - Intersect genes with known disease genes from OMIM
  - Intersect genes with results of eQTL analysis (e.g. GTEx); associated eQTL SNPs can independently be analyzed for disease association and potential mechanism (e.g. disrupting TF binding site)

Interpreting ASE results

Remember: ASE predictions are just that: predictions. **Any important result should be experimentally validated (e.g. qPCR or ddPCR)!**