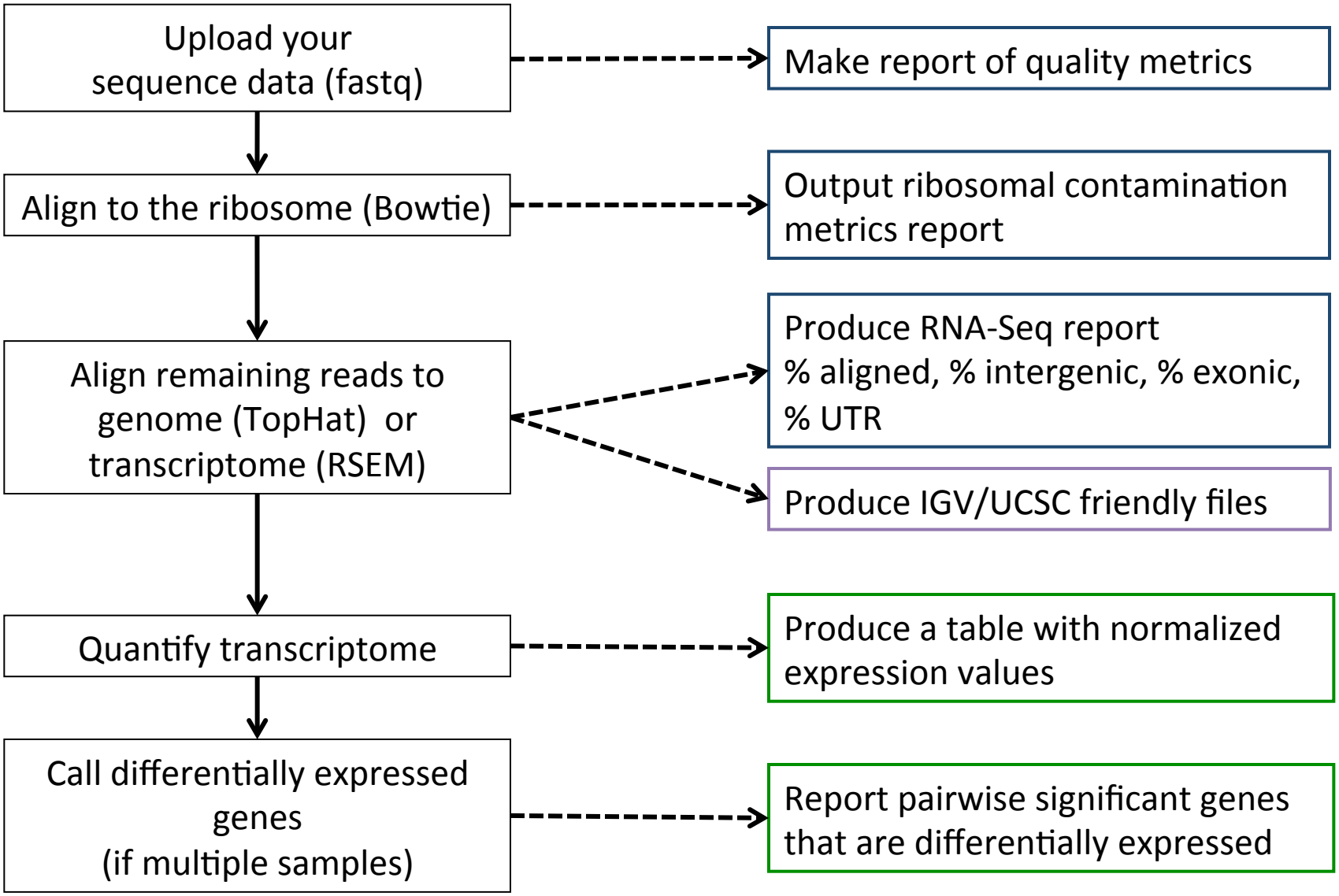
University of Massachusetts Medical School

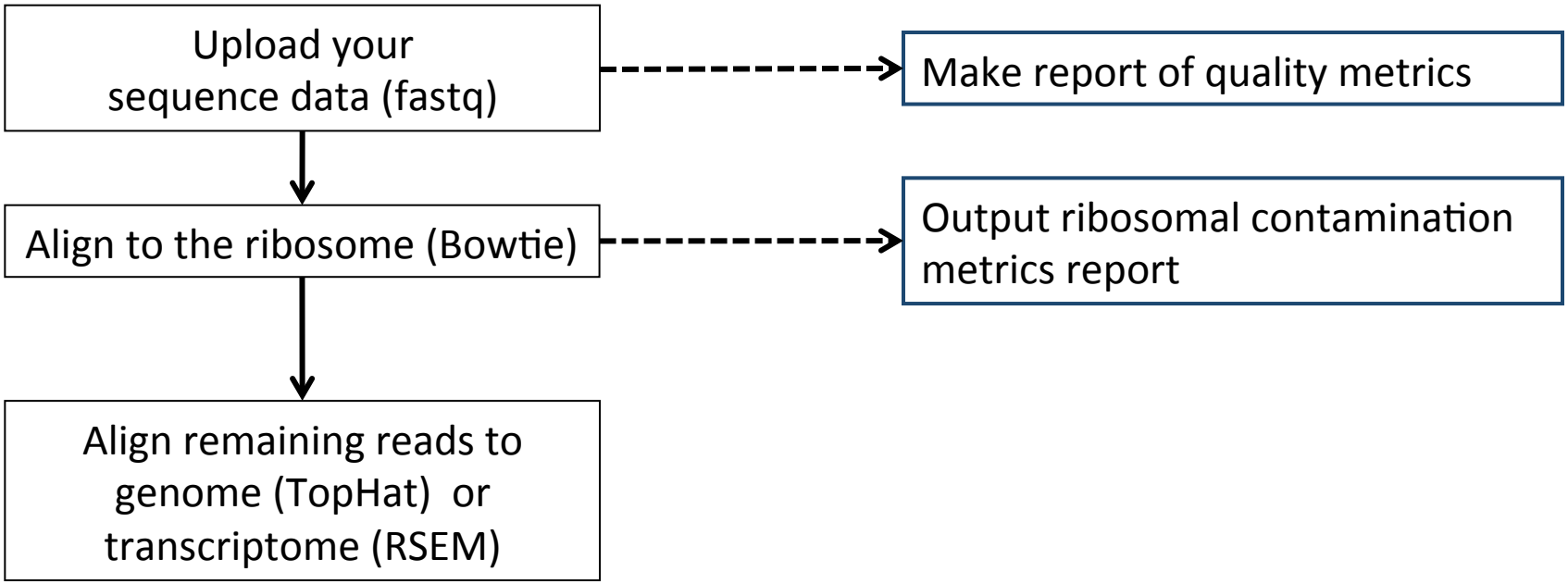
UMASS

RNA-Seq primer

Our typical RNA quantification pipeline

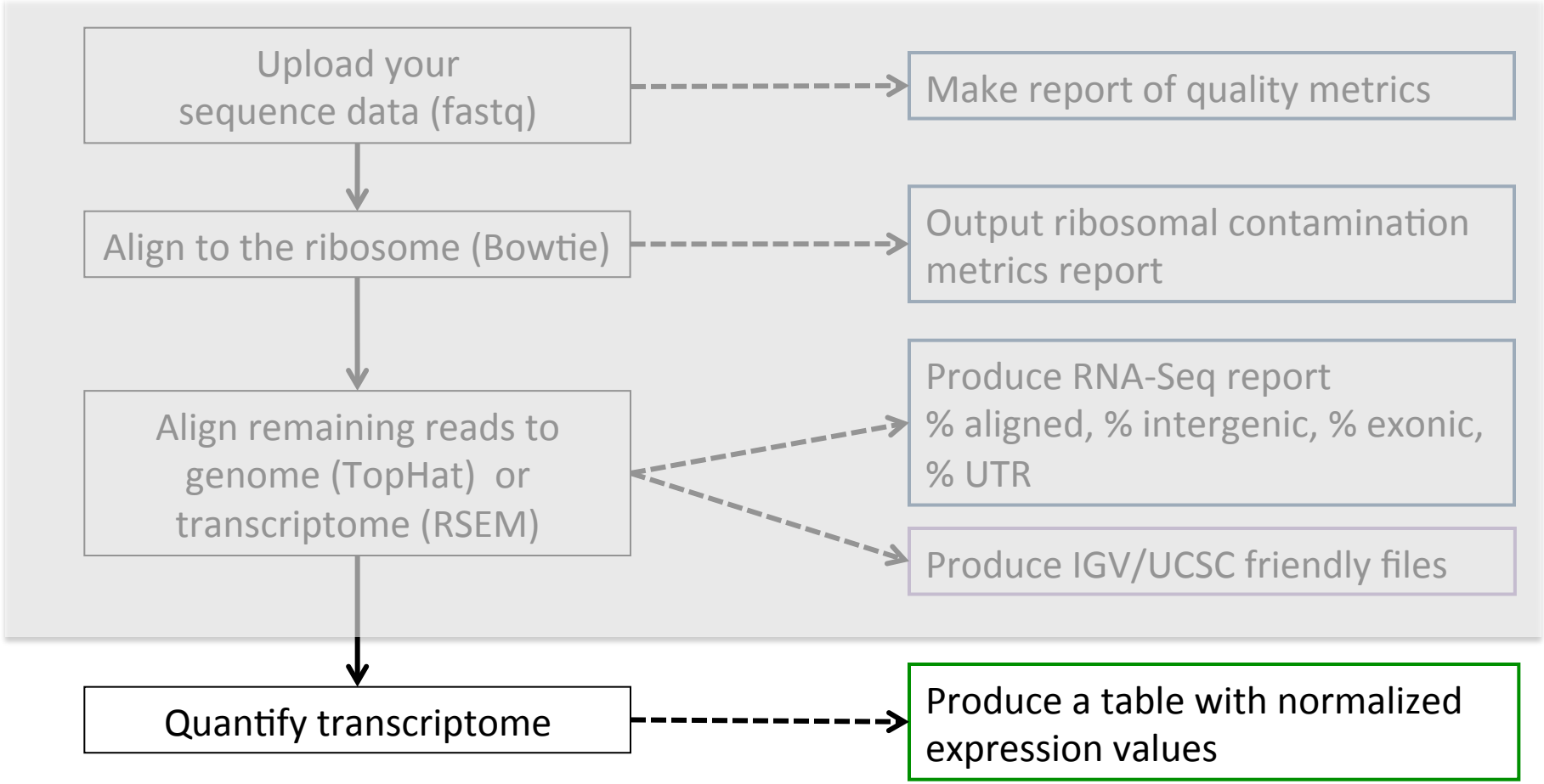


Alignment requires pre-processing



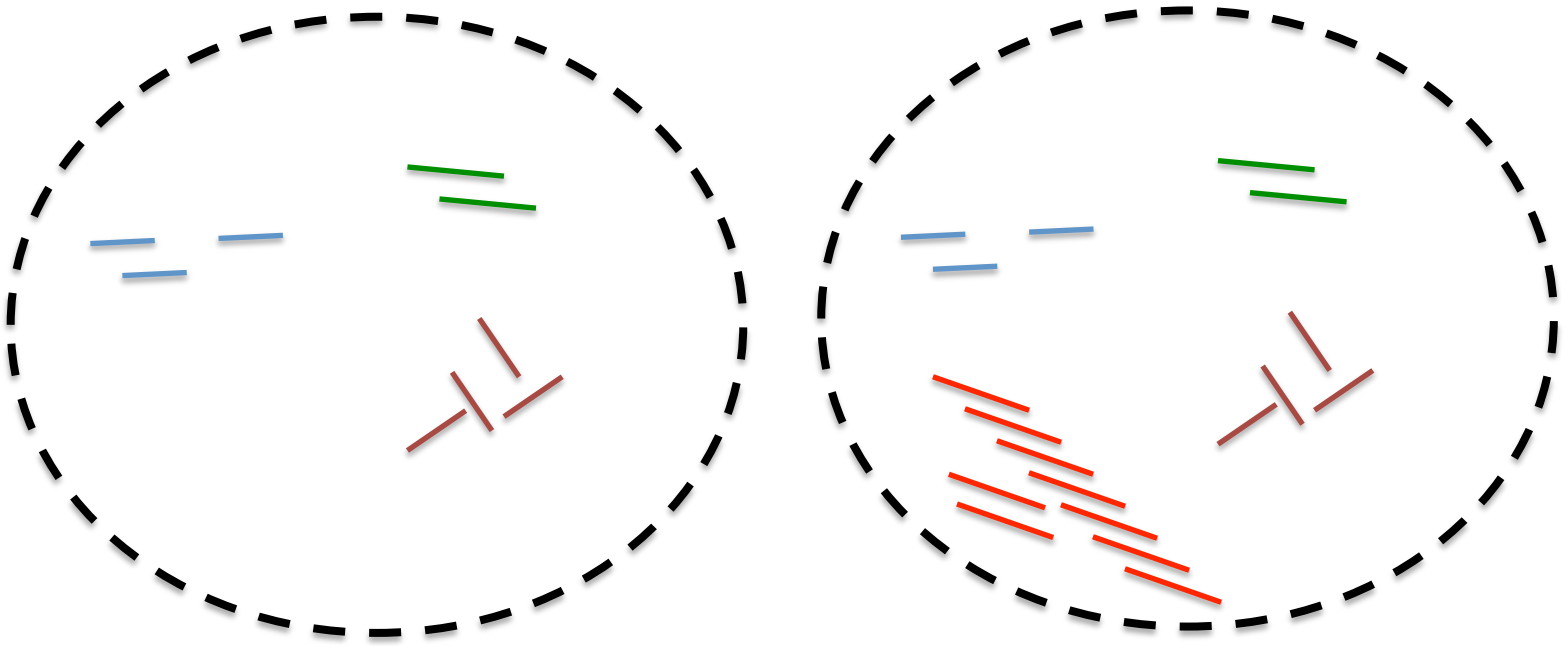

```
bowtie2-build -f mm10.fa mm10

rsem-prepare-reference \
--gtf ucsc.gtf --transcript-to-gene-map ucsc_into_genesymbol.rsem \
mm10.fa mm10.rsem
```

Computing gene expression




```
rsem-calculate-expression --paired-end --strand-specific -p 2 \
 --output-genome-bam fastq.quantification/control_rep1.1.fq \
fastq.quantification/control_rep1.2.fq genome.quantification/mm10.rsem \
rsem/ctrl1.rsem
```

Normalization for comparing a gene across samples

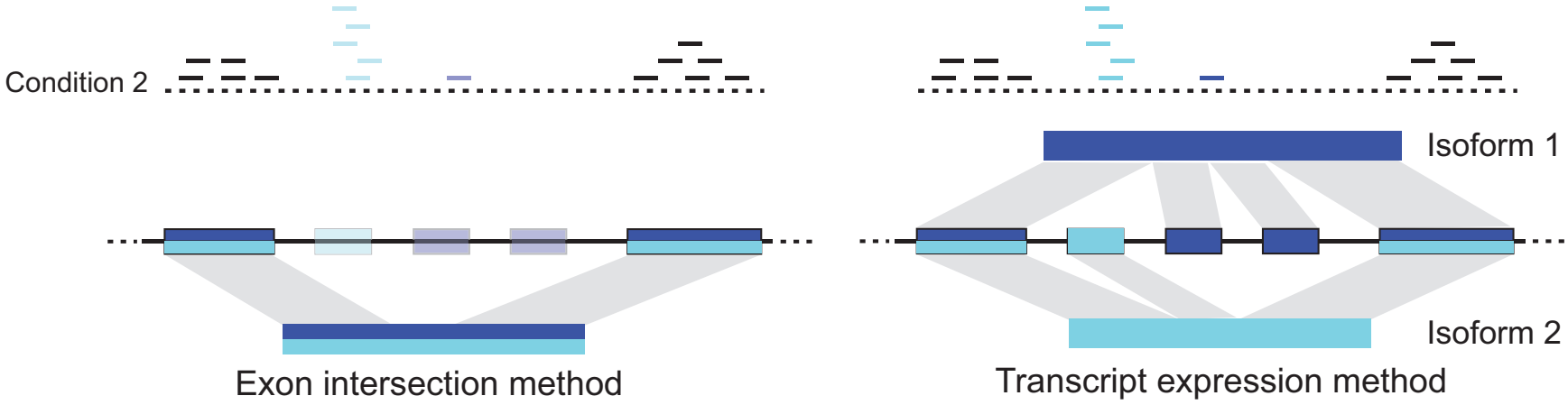


Cell type I

Cell type II

Normalizing by total reads does not work well for samples with very different RNA composition

But, how to compute counts for complex gene structures?



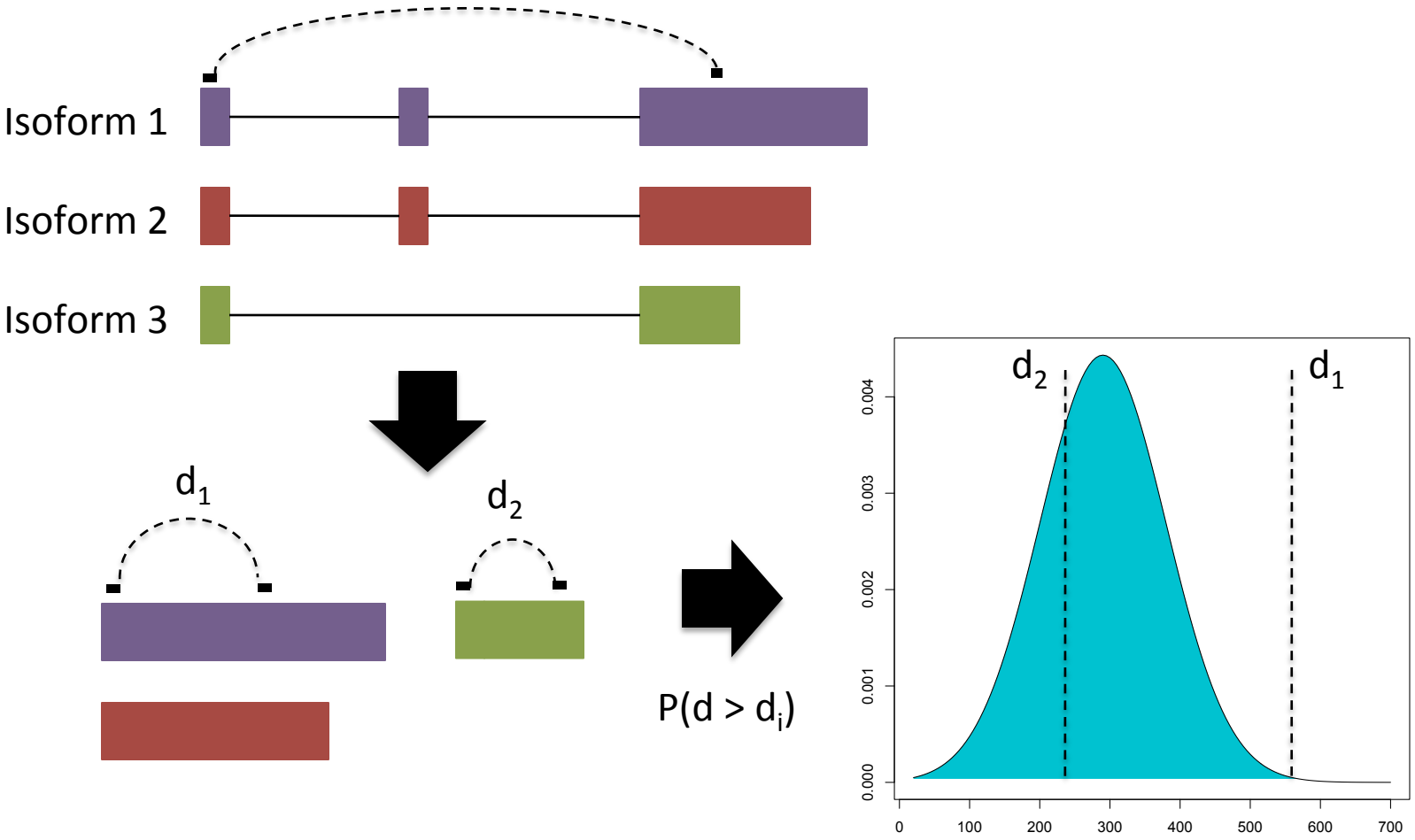
Three popular options:

Exon *intersection* model: Score constituent exons

Exon *union* model: Score the the "merged" transcript

Transcript expression model: Assign reads uniquely to different isoforms. *Not a trivial problem!*

… and use it for probabilistic read assignment



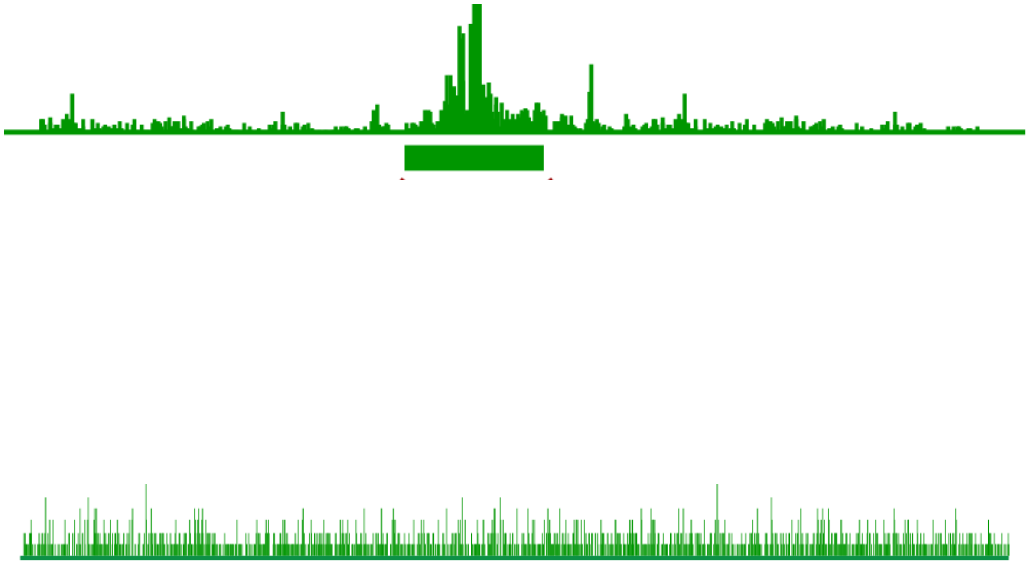
For methods such as MISO, Cufflinks and RSEM, it is critical to have paired-end data

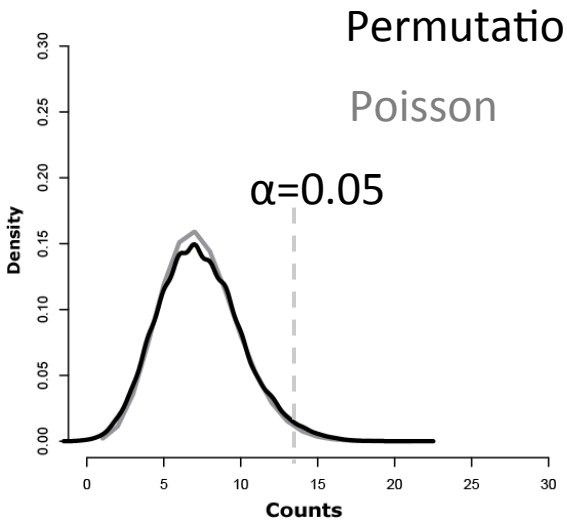
RNA-Seq quantification summary

- Counts must be estimated from ambiguous read/ transcript assignment.
  - Using simplified gene models (intersection)
  - Probabilistic read assignment
- Counts must be normalized
  - RPKM is sufficient for intra-library comparisons
  - More sophisticated normalizations to account for differences in library composition for inter-library comparisons.

IS A GENE EXPRESSED?

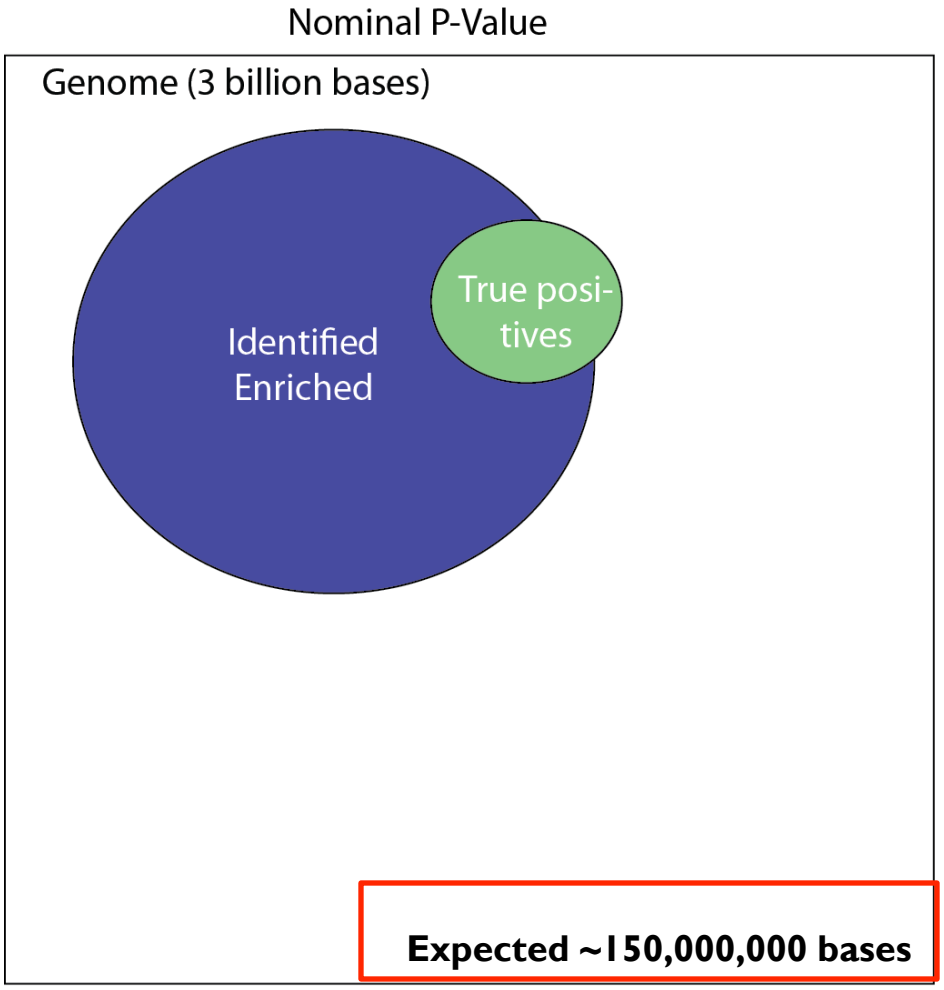
Our approach





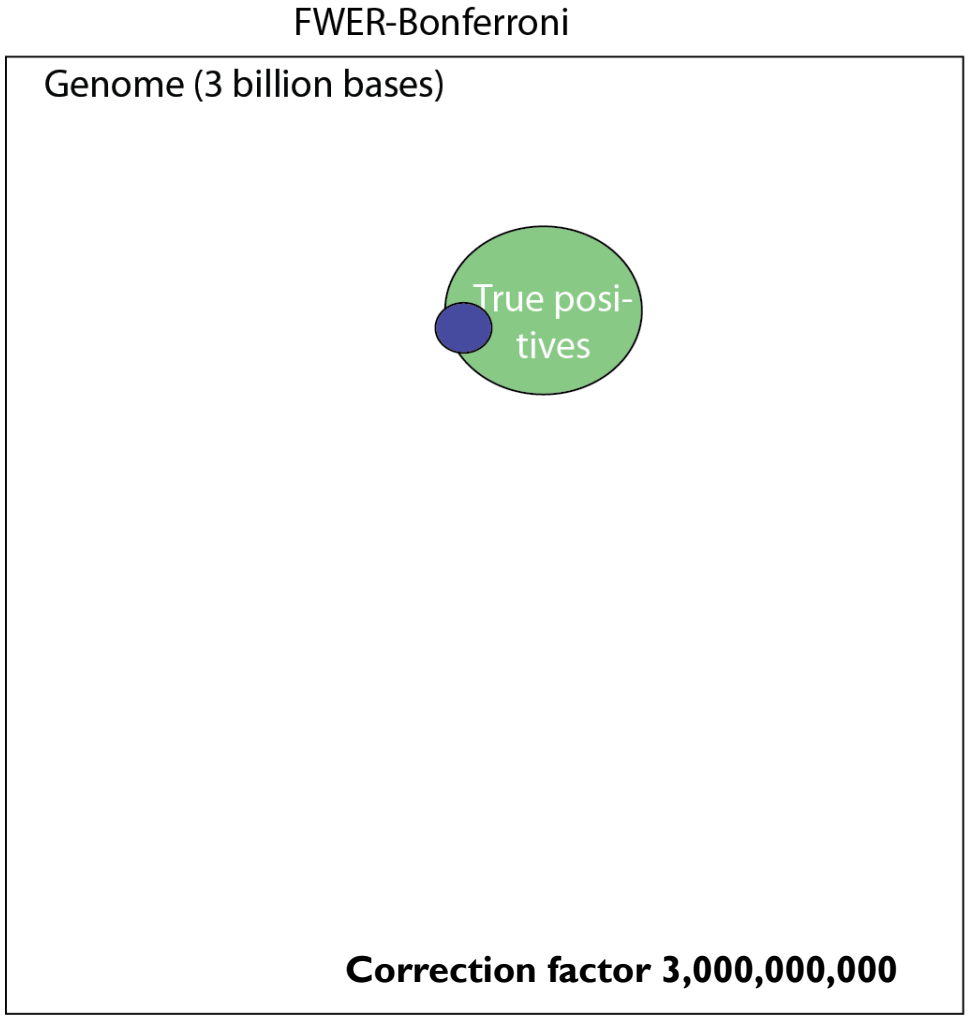
We have an efficient way to compute read count p-values …

The genome is large, many things happen by chance



We need to correct for multiple hypothesis testing

Bonferroni correction is way to conservative

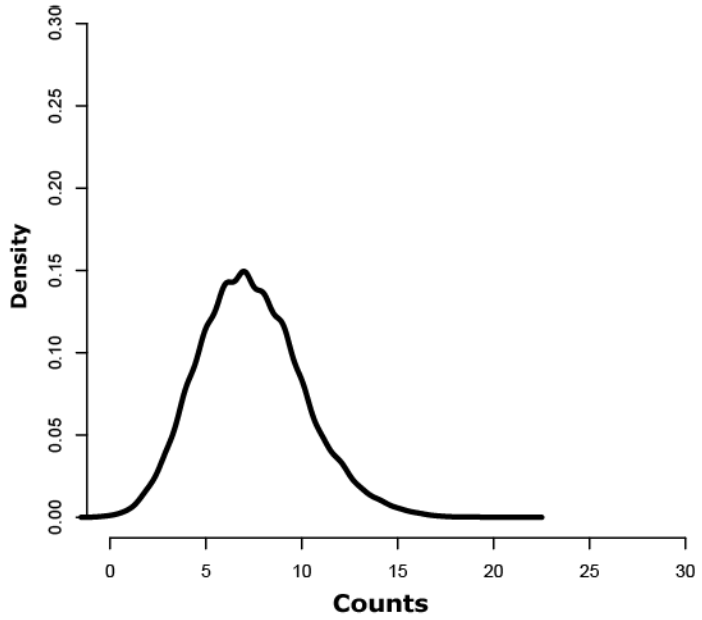


Bonferroni corrects the number of hits but misses many true hits because its too conservative – How do we get more power?

Controlling FWER

Max Count distribution

α=0.05 α_{FWER}=0.05



Count distribution (Poisson)

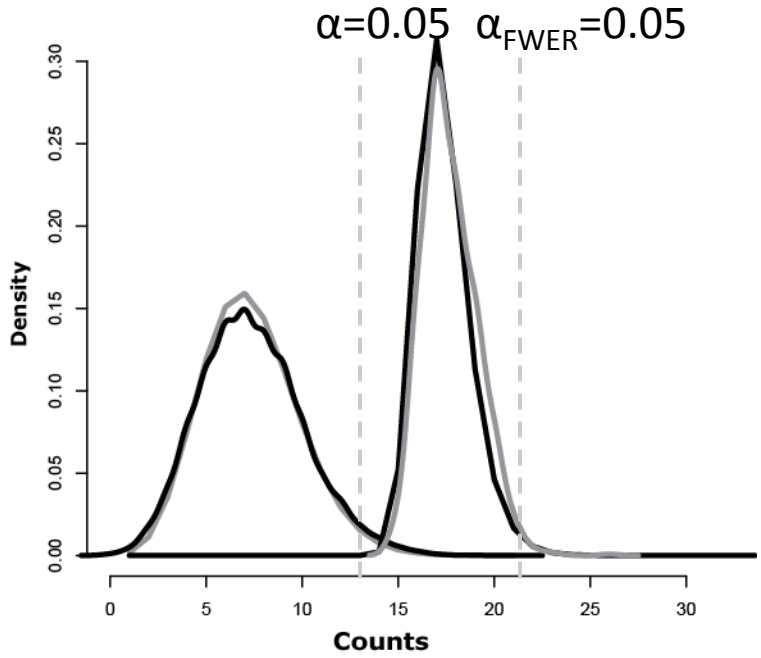
Given a region of size w and an observed read count n. What is the probability that one or more of the $3x10^9$ regions of size w has read count >= n under the null distribution?

We could go back to our permutations and compute an FWER: **max of the genome-wide distributions of same sized region**)→
but really really really slow!!!

Scan distribution, an old problem

- Is the observed number of read counts over our region of interest high?
- Given a set of Geiger counts across a region find clusters of high radioactivity
- Are there time intervals where assembly line errors are high?

Scan distribution



Poisson distribution

Thankfully, the ***Scan Distribution*** computes a closed form for this distribution.

ACCOUNTS for dependency of overlapping windows thus more powerful!

Scan distribution for a Poisson process

The probability of observing k reads on a window of size w in a genome of size L given a total of N reads can be approximated by (Alm 1983):

$$P(k|\lambda w, N, L) \approx 1 - F_p(k-1|\lambda w)e^{-\frac{k-w\lambda}{k}\lambda(T-w)P(k-1|\lambda w)}$$

where

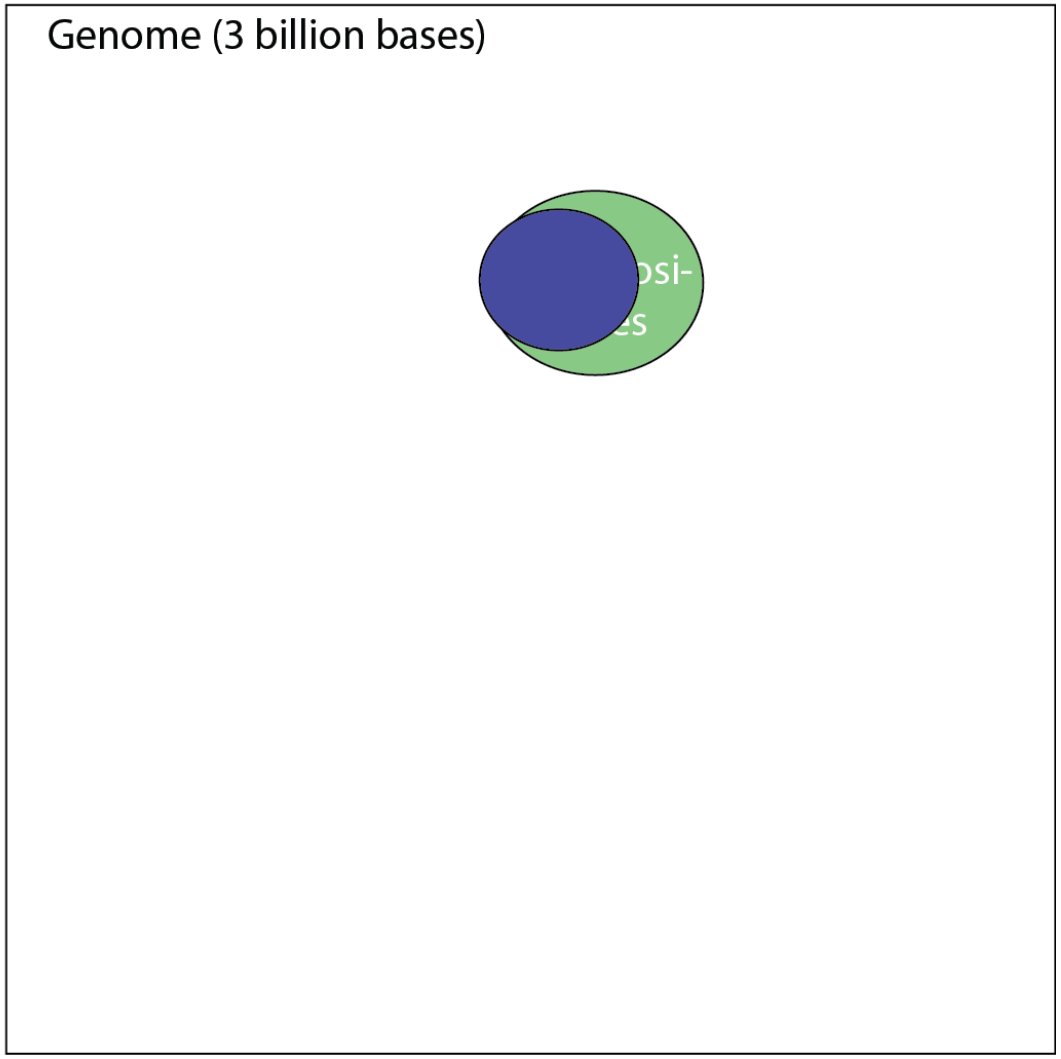
$P(k-1|\lambda w)$ is the Poisson probability of observing $k-1$ counts given an expected count of λw

and

$F_p(k-1|\lambda w)$ is the Poisson probability of observing $k-1$ or fewer counts given an expectation of λw reads

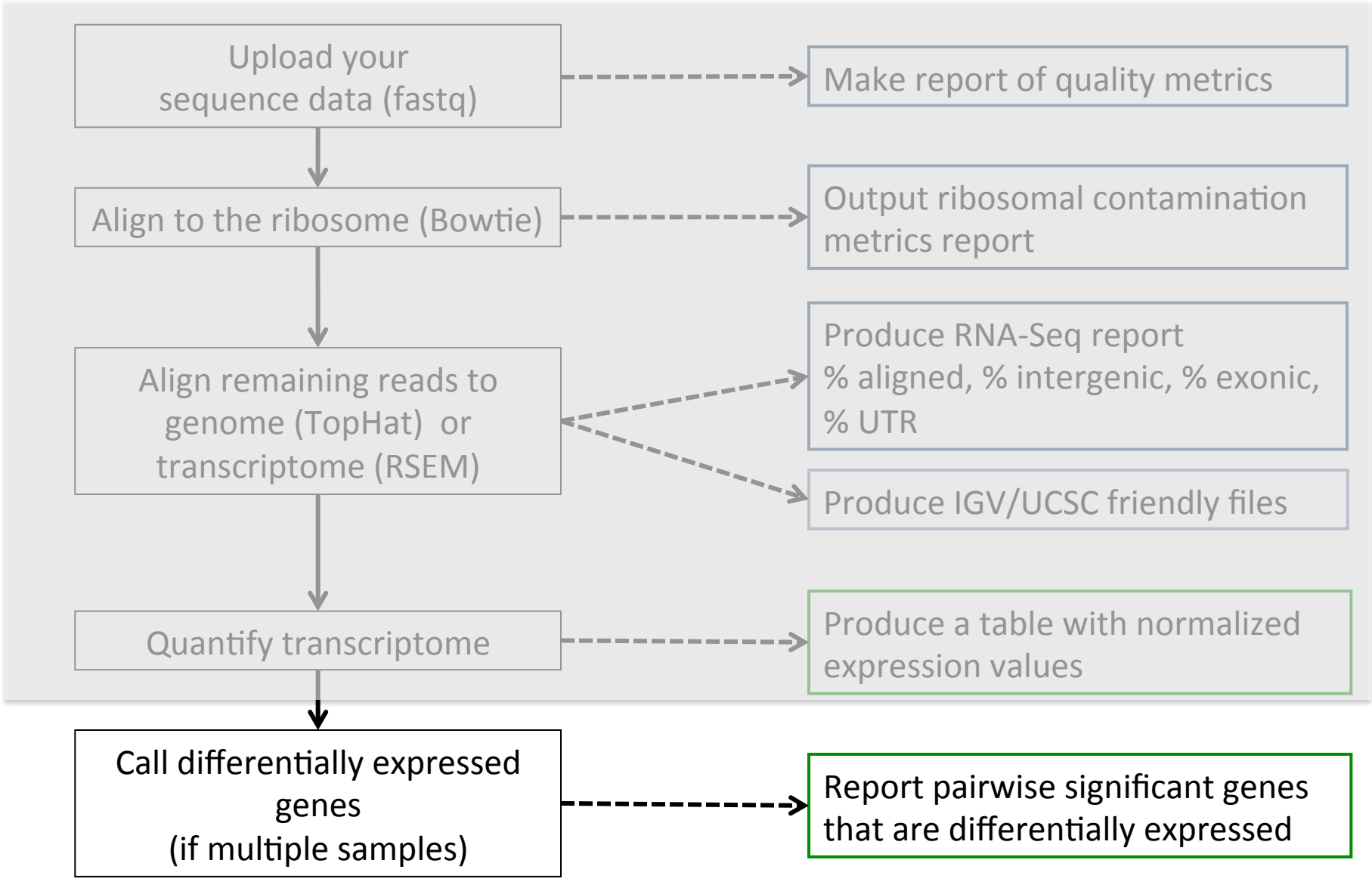
The scan distribution gives a computationally very efficient way to estimate the FWER

FWER-Scan Statistics

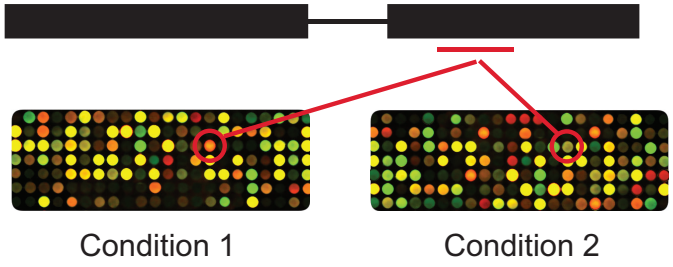


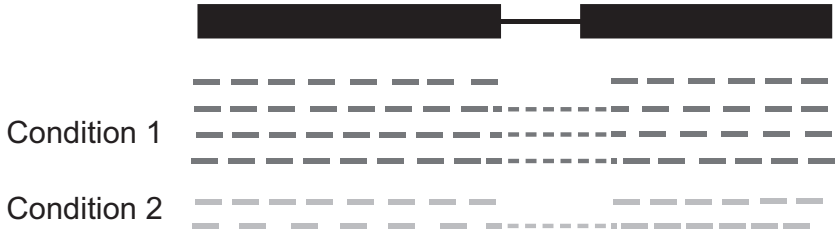
By utilizing the dependency of overlapping windows we have greater power, while still controlling the same genome-wide false positive rate.

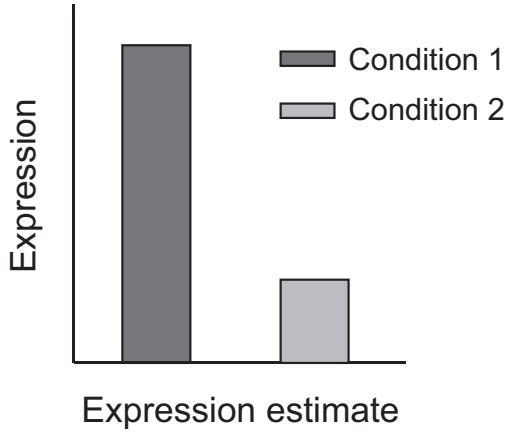
Differential gene expression

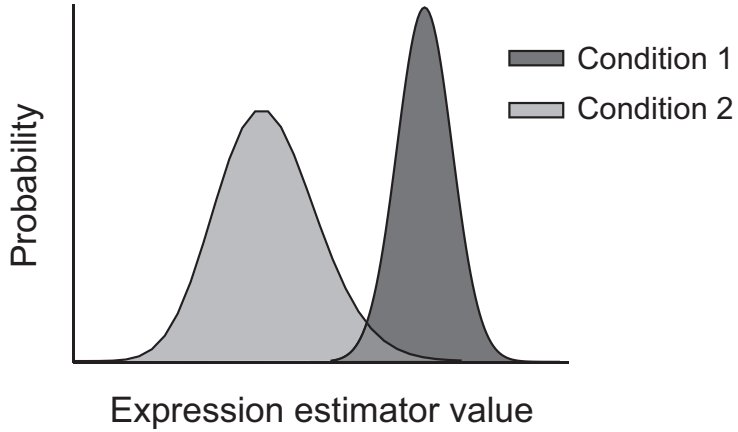


Differential gene expression using RNA-Seq









•(Normalized) read counts ←→ Hybridization intensity

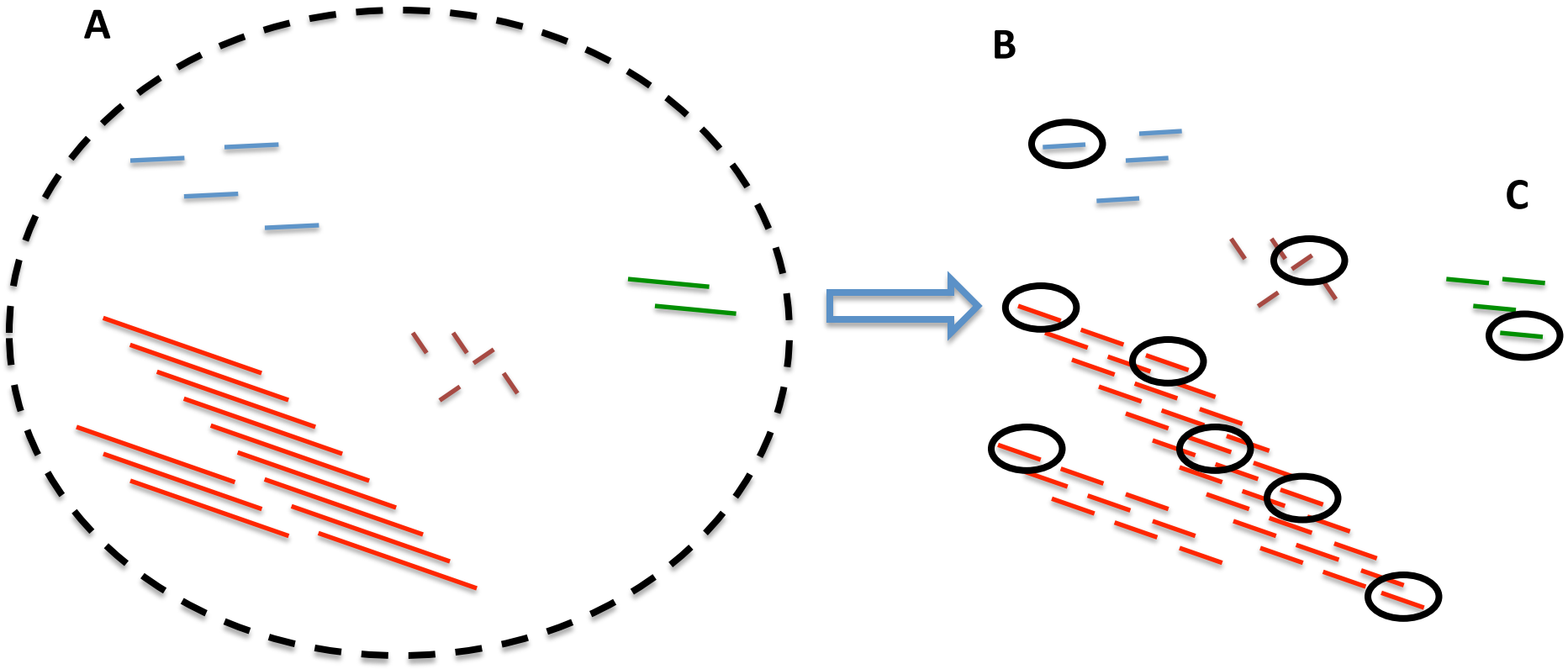
Differential analysis strategies

- Use read counts
  - Standard Fisher exact test

	Condition A	Condition B
Gene A reads	n_a	n_b
Rest of reads	N_a	N_b

  - Model read counts (Poisson, negative binomial) and test whether models are distinct
  - Use empirical approaches that do not rely on parametric assumptions (more on this later)

Differential analysis: Lets revisit quantification



RNA-Seq quantification: Infer # molecules in A from observed fragments in C

Quantification assumptions for differential expression

$$\mathcal{G} \propto f_g \times N$$

$\mathcal{G}$ = Read counts for gene g
f_g = the fraction of mRNA molecules for gene g
N = The total number of **aligned** reads

Note: We can at best estimate f_g

Modeling the RNA-Seq process

$$\mathcal{G} \propto f_g \times N$$

$$P(\mathcal{G}|N) = \binom{N}{\mathcal{G}} (f_g)^{\mathcal{G}} (1 - f_g)^{N-\mathcal{G}}$$

RNA-Seq counts should distribute "binomially"

Binomial? Why then we talk about Poisson

$$P(\mathcal{G}|N) = \binom{n}{\mathcal{G}} (f_g)^{\mathcal{G}} (1 - f_g)^{1-\mathcal{G}}$$

$$f_g \ll 1 \quad \text{and } \mathcal{G} \ll N \quad \text{and say } g = f_g \times M$$

M: # of mRNAs

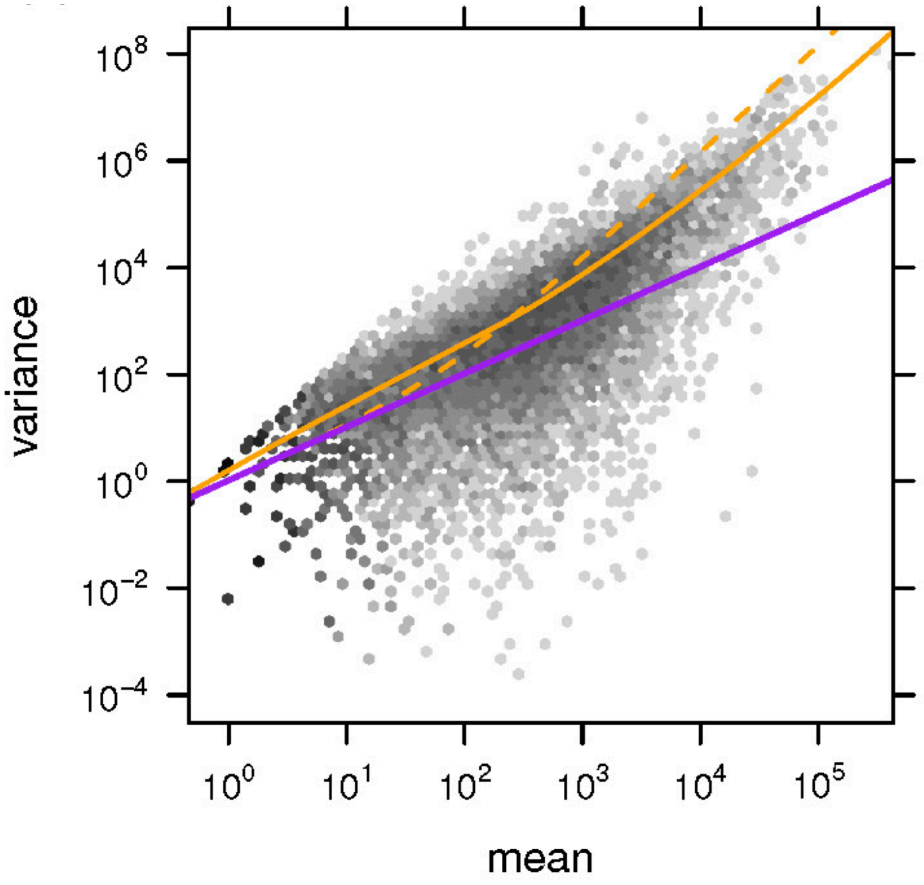
g: # mRNAs for the gene

Binomial? Why then we talk about Poisson

$$P(\mathcal{G}|N) = \frac{N!}{\mathcal{G}!(N-\mathcal{G})!}(\frac{g}{M})^{\mathcal{G}}(1-\frac{g}{M})^{N-\mathcal{G}}$$

RNA-Seq counts can be approximated by a Poisson distribution

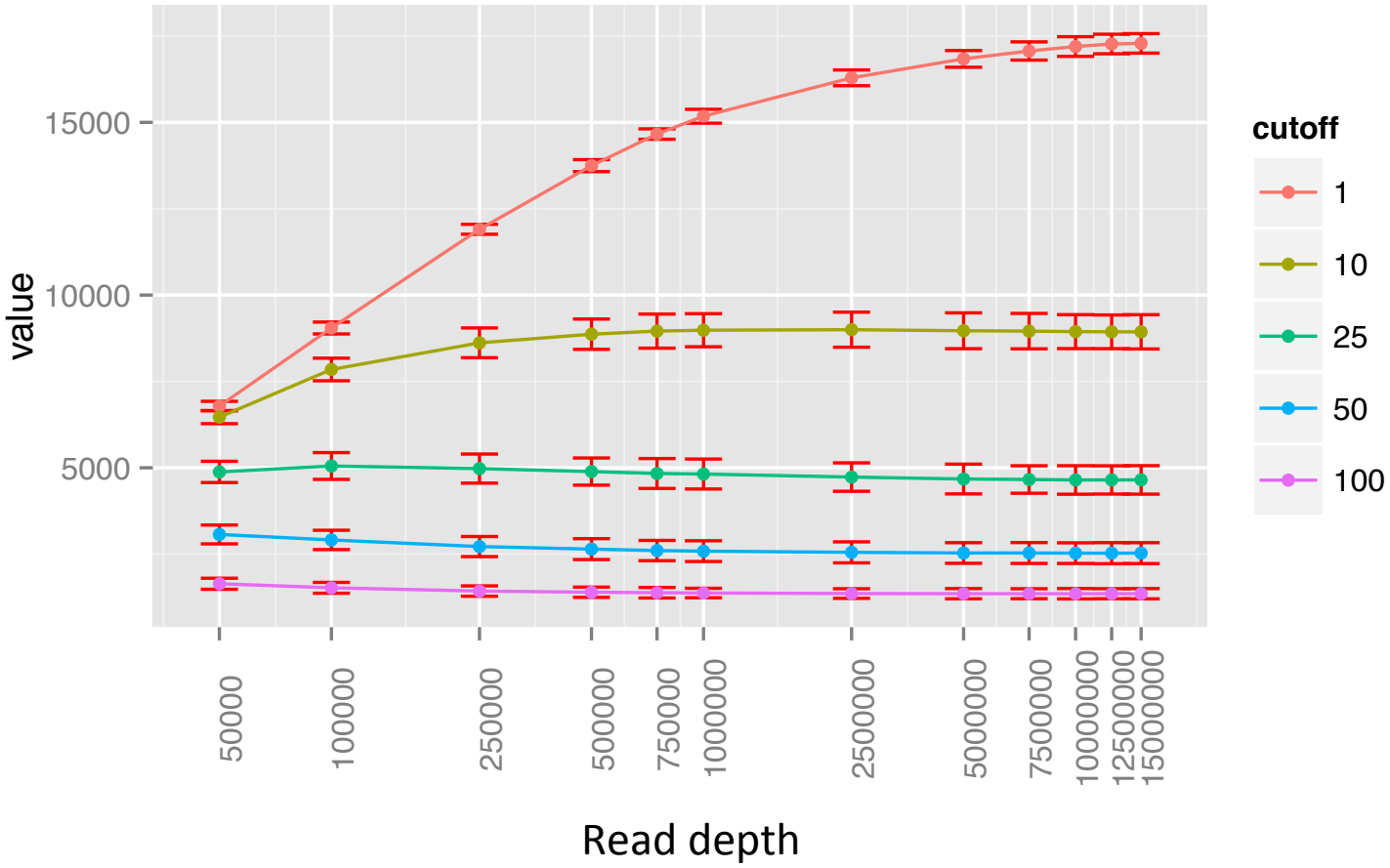
Poisson model does not work



Adapted from Anders, 2010

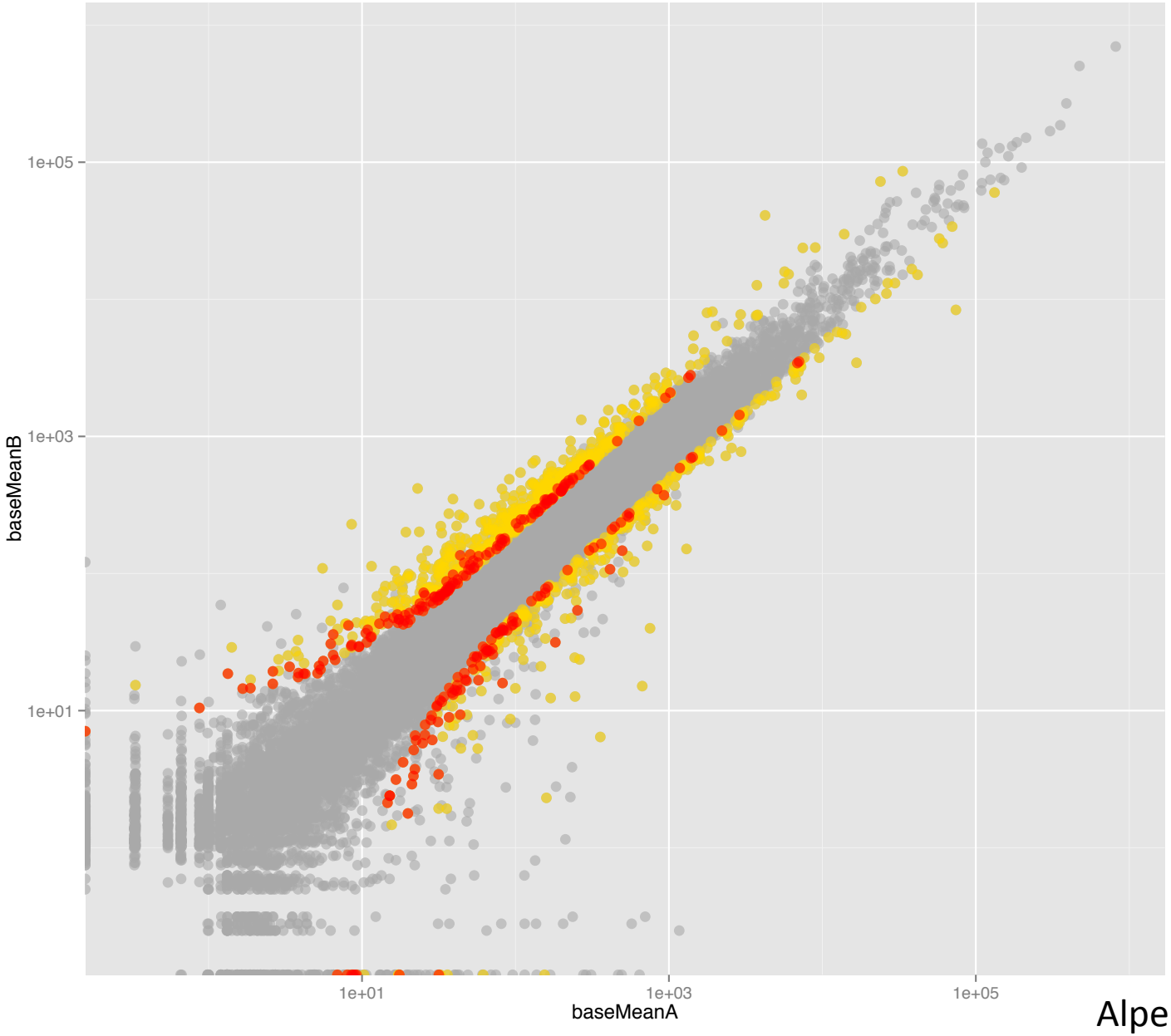
Biological variance does not follow a Poisson model

Robustness to low depth: Transcripts detected



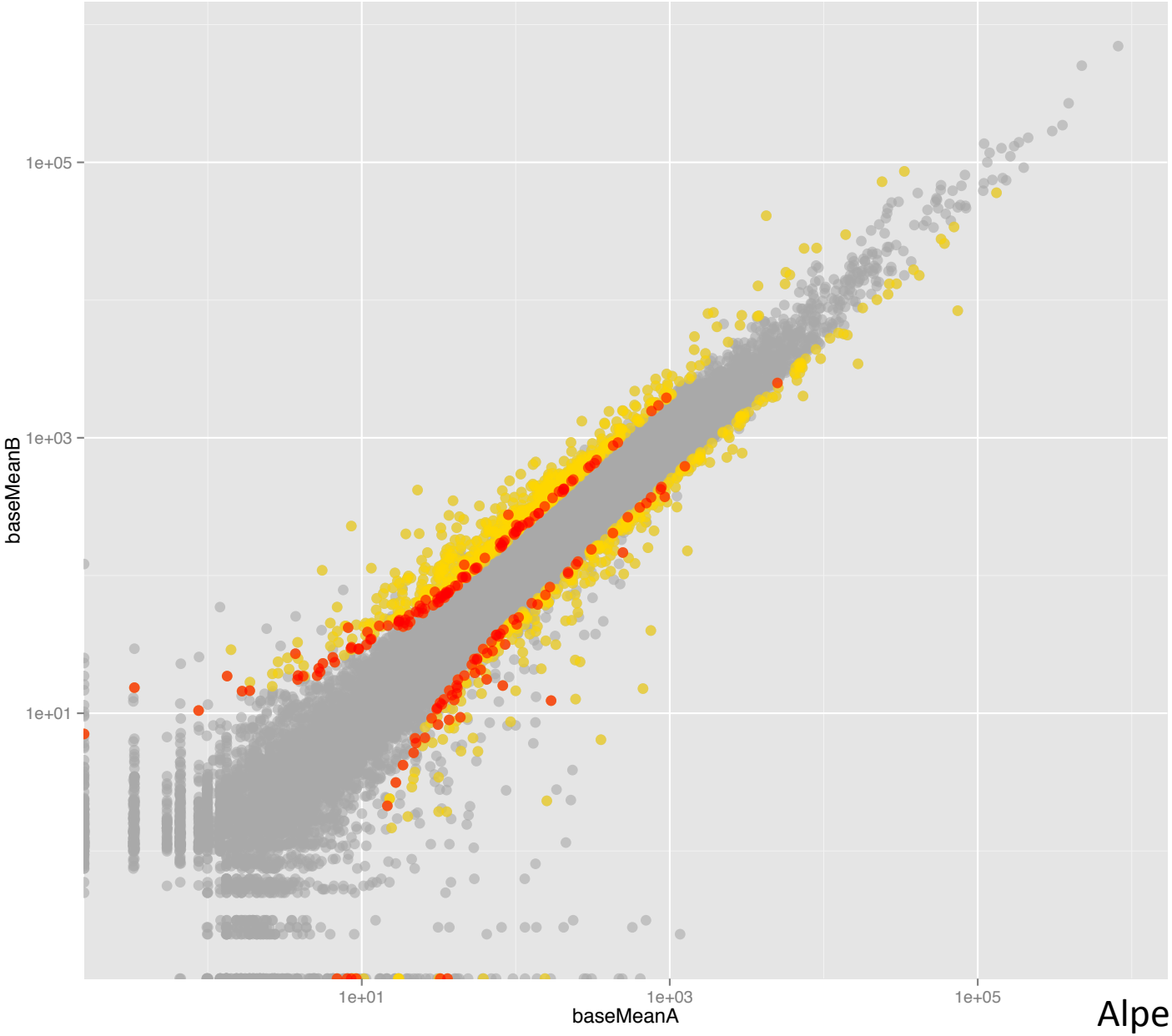
Alper Kucukural

RSEM/DESeq: 15 mill reads in mouse



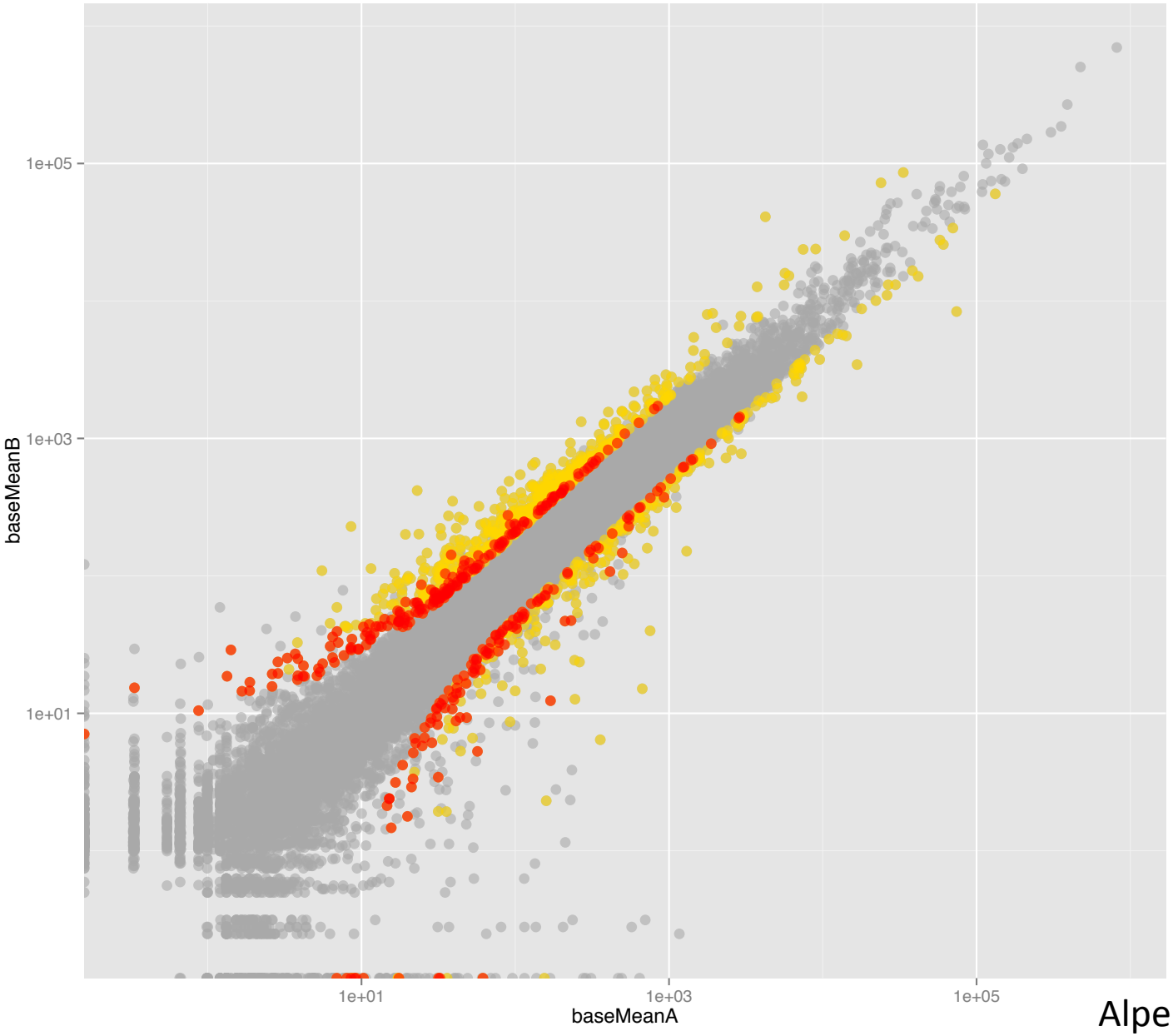
Alper Kucukural

RSEM/DESeq: 10 mill reads in worm



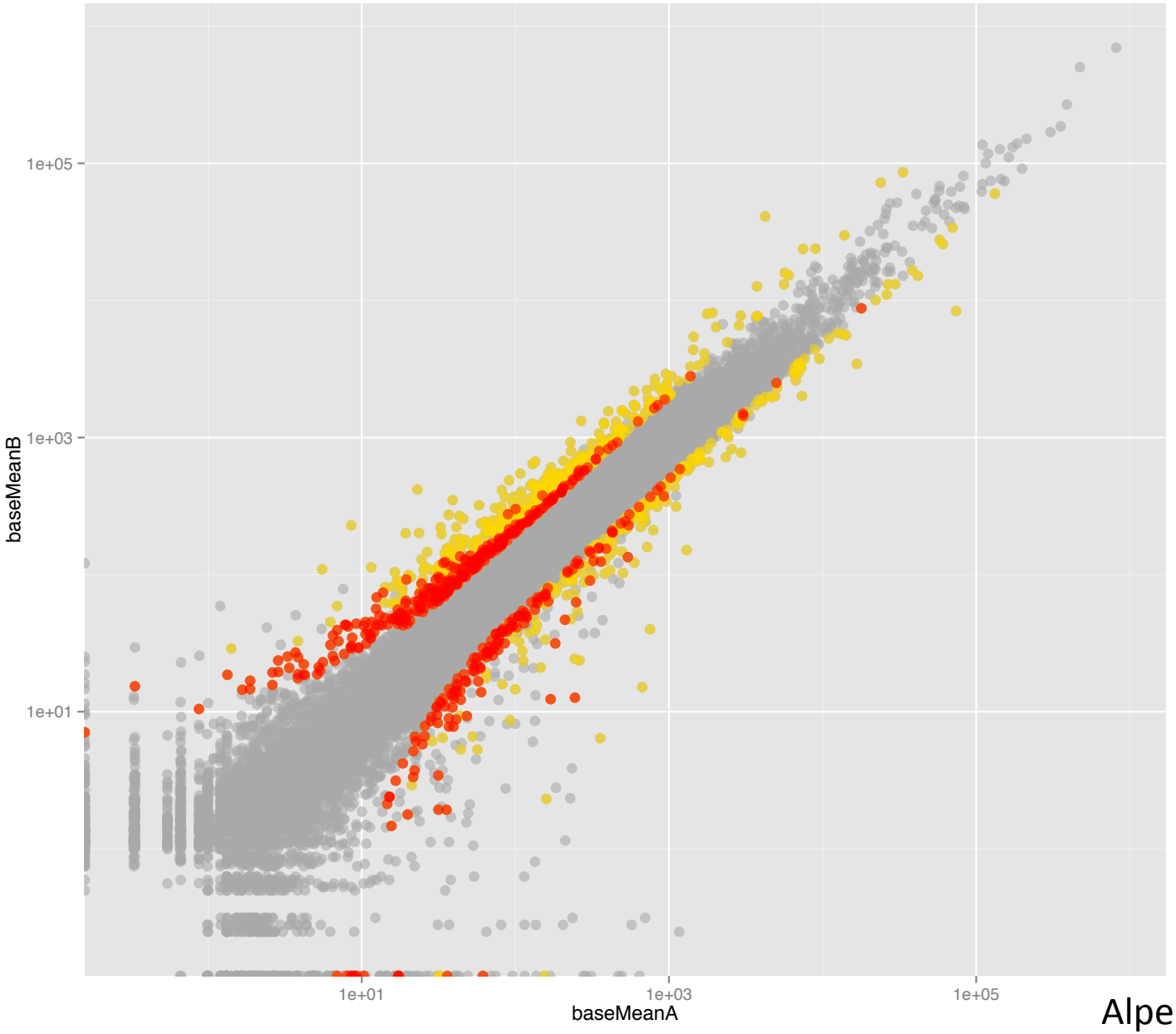
Alper Kucukural

RSEM/DESeq: 7.5 mill reads in worm



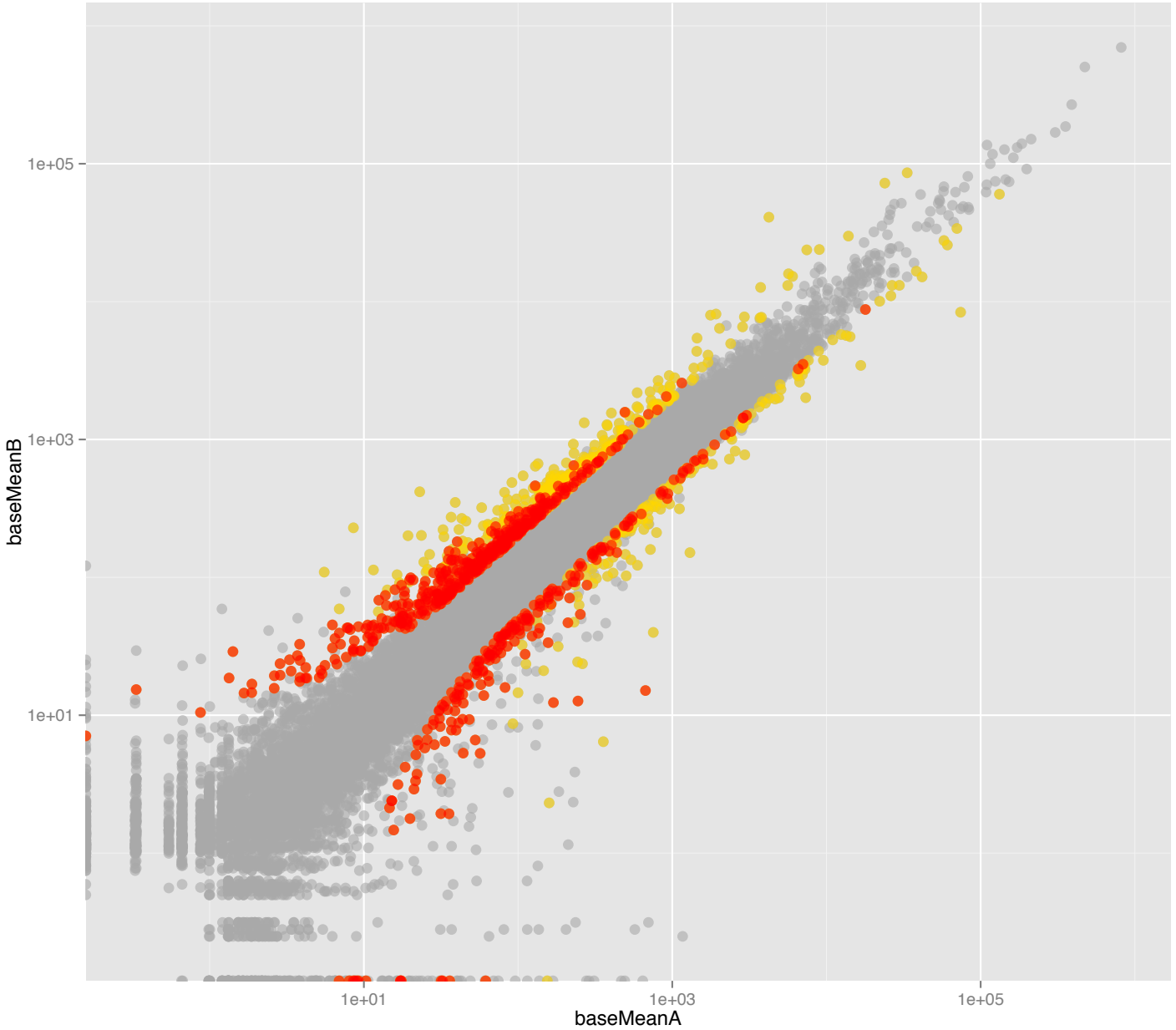
Alper Kucukural

RSEM/DESeq: 5 mill reads in worm



Alper Kucukural

RSEM/DESeq: 2.5 mill reads in worm



University of Massachusetts Medical School

Initial analysis

Summary I – Data types, file formats and utilities

- Annotation: Genomic regions
  - Genes
  - Peaks
  - *bedtools*
- Alignment: Map reads
  - BAM/SAM
  - *Samtools*
- Aggregation: Summary files
  - Wig (UCSC)
  - TDF (IGV)

Summary II – Data process

- Short read alignment (Bowtie, BWA)
  - Making the genome searchable: Hashing/BW
  - Seed an extend (hashing) vs suffix searches (BW)
  - New aligners are mix
- Spliced aligners (TopHat, STAR, GSNAP)
  - Map read fragments then strung them
  - Choosing the fragment size
  - Avoiding biases using information (junctions)
- Quantifying (RSEM/Cufflinks)
  - Read/Isoform assignment
  - Normalization procedures
- Differential expression (DESeq/EdgeR/Cufflinks)

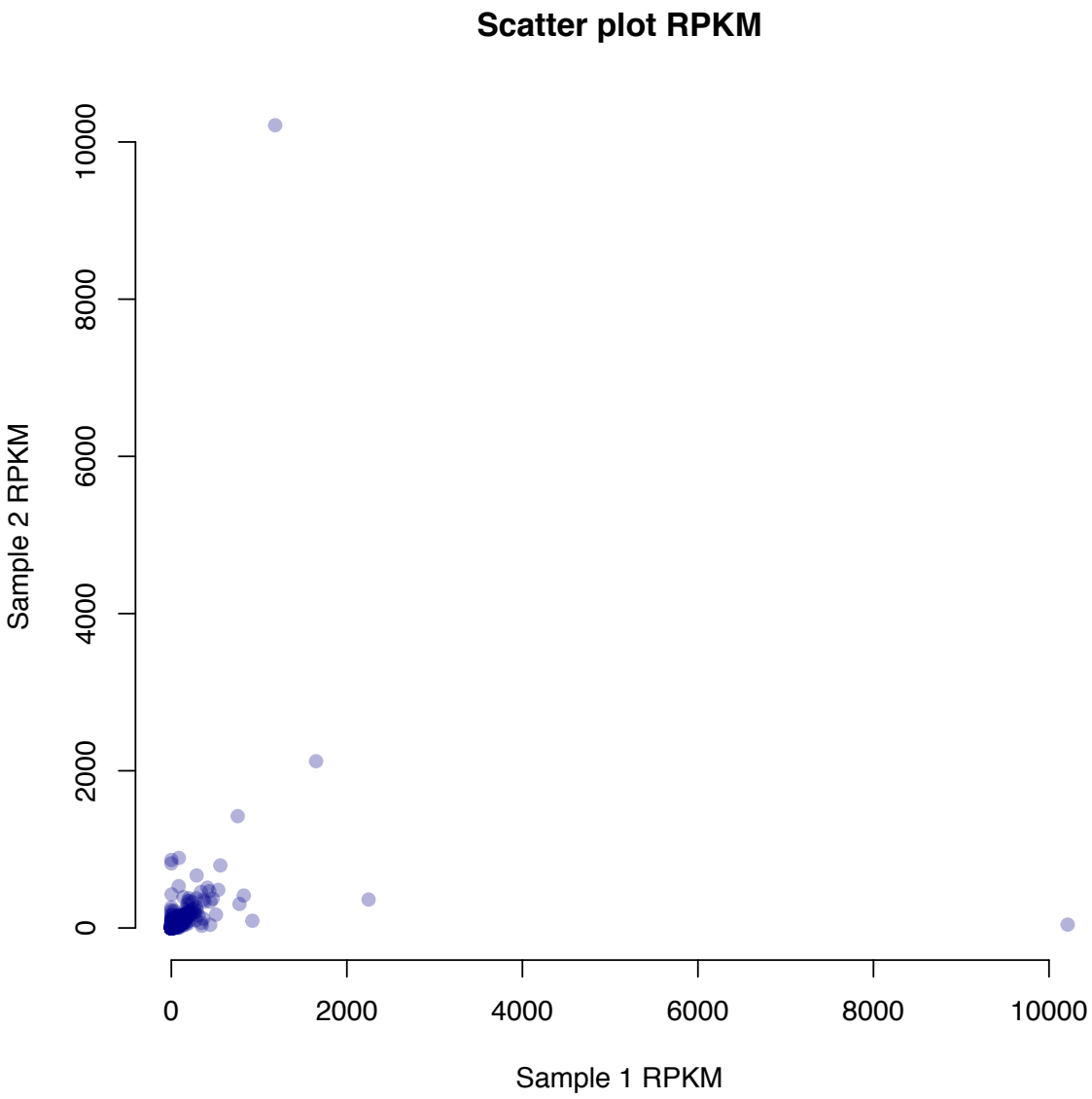
Summary III – Using a graphical user interface

- Galaxy – for knowledgable users who are not comfortable with UNIX
- All tools available
- Not great for many samples

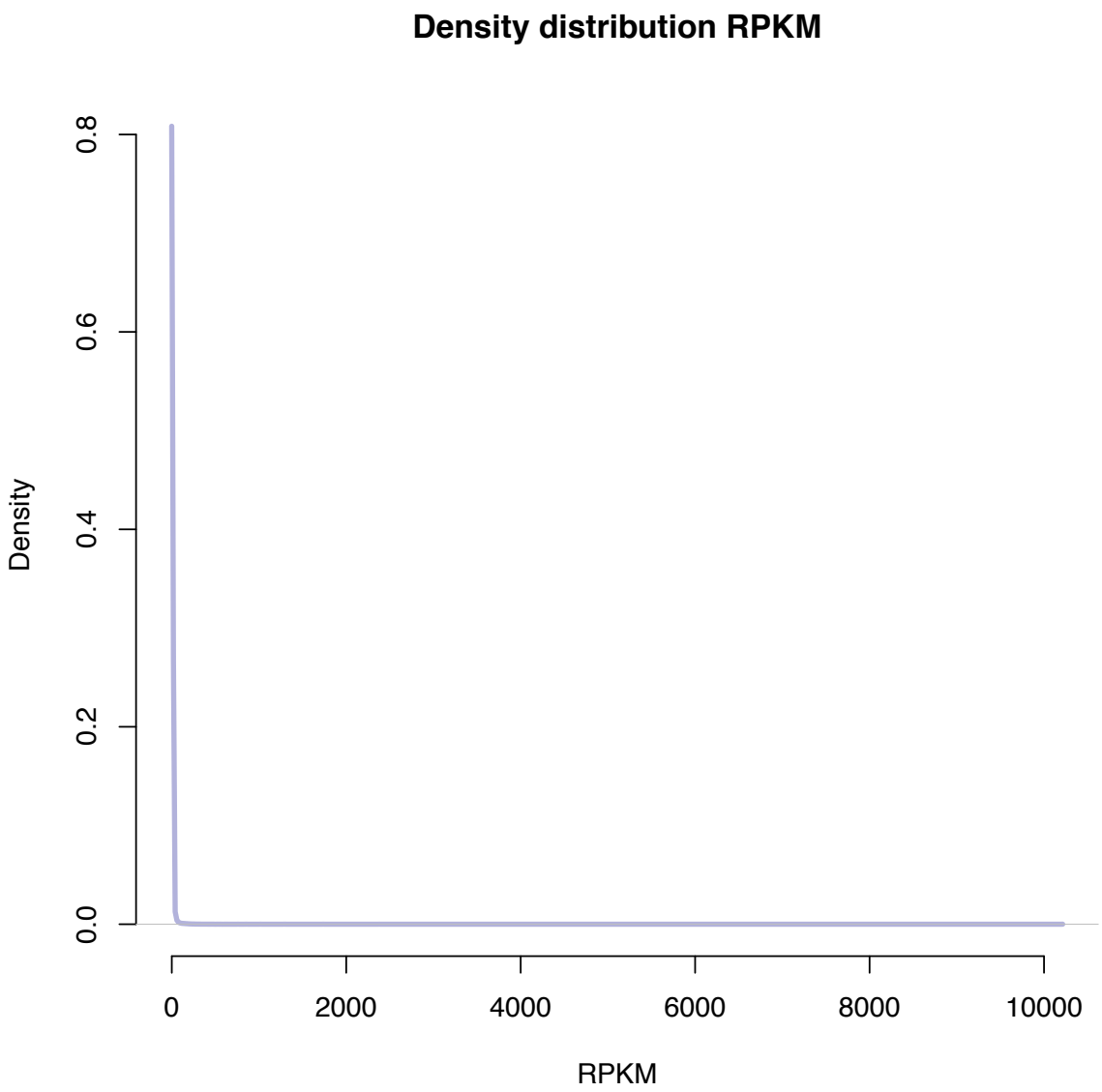
Todays topics

- Looking at ALL of your data

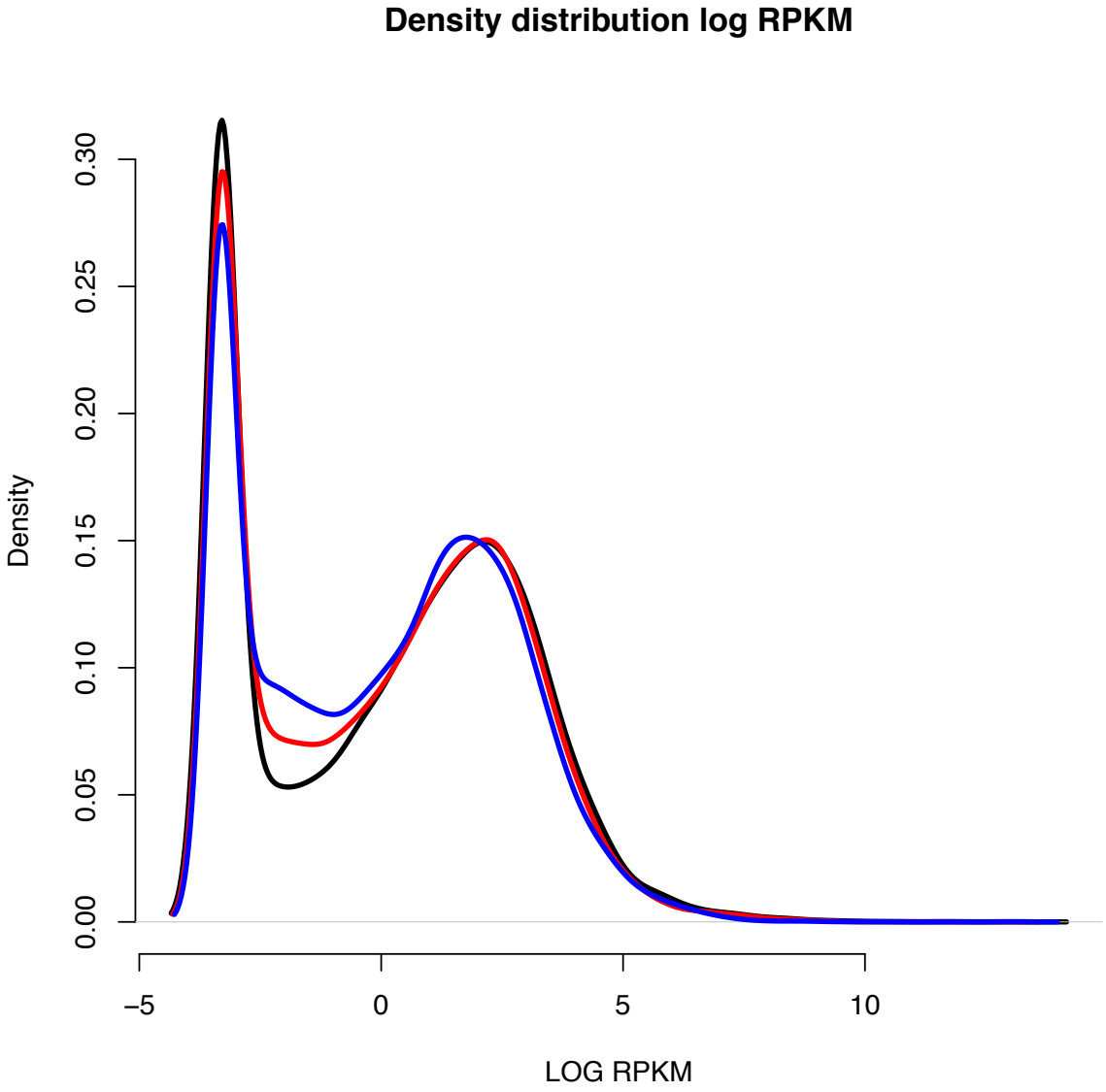
Comparing samples: Scatter plots



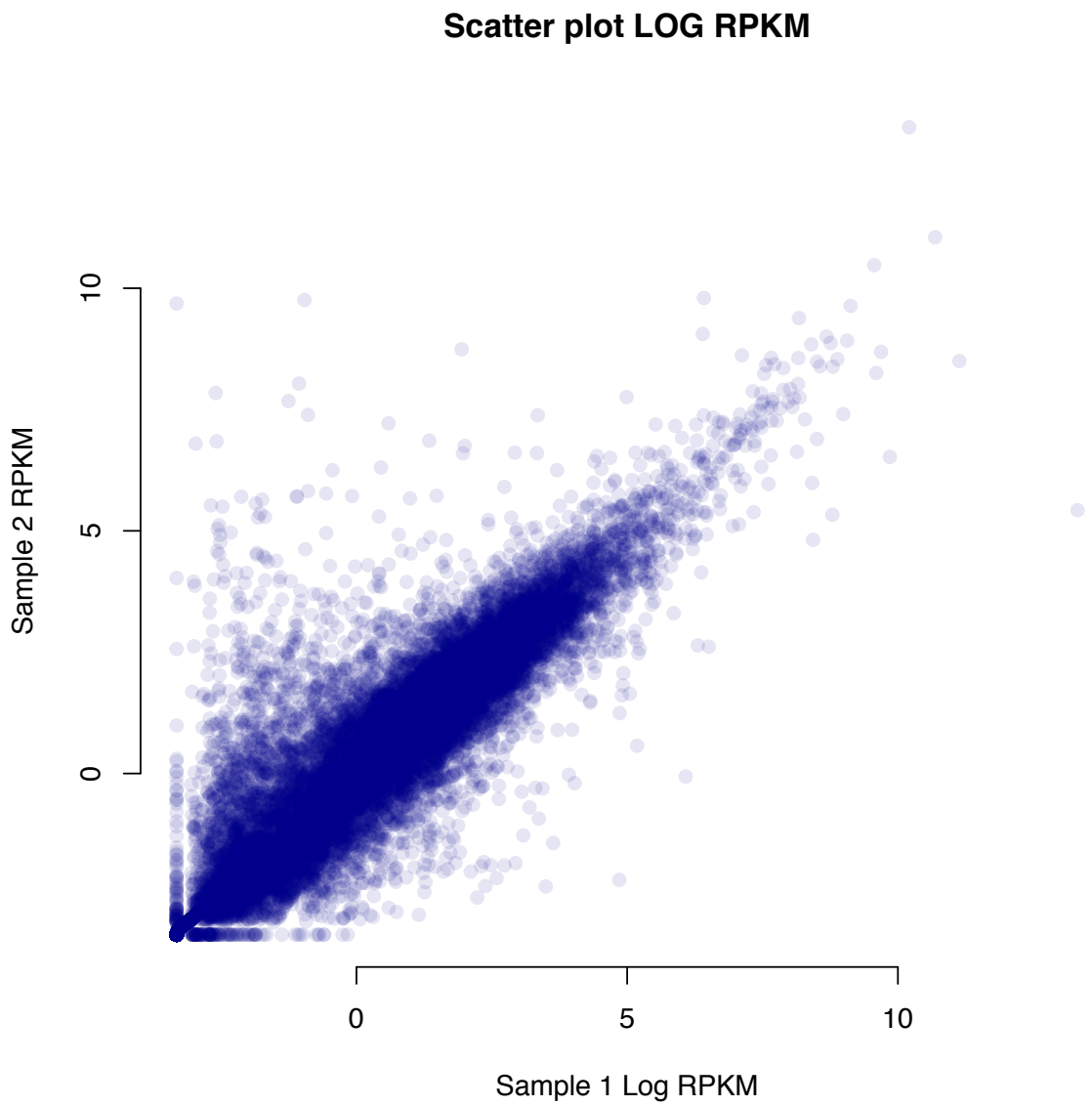
Raw counts/RPKMs are NOT Gaussian



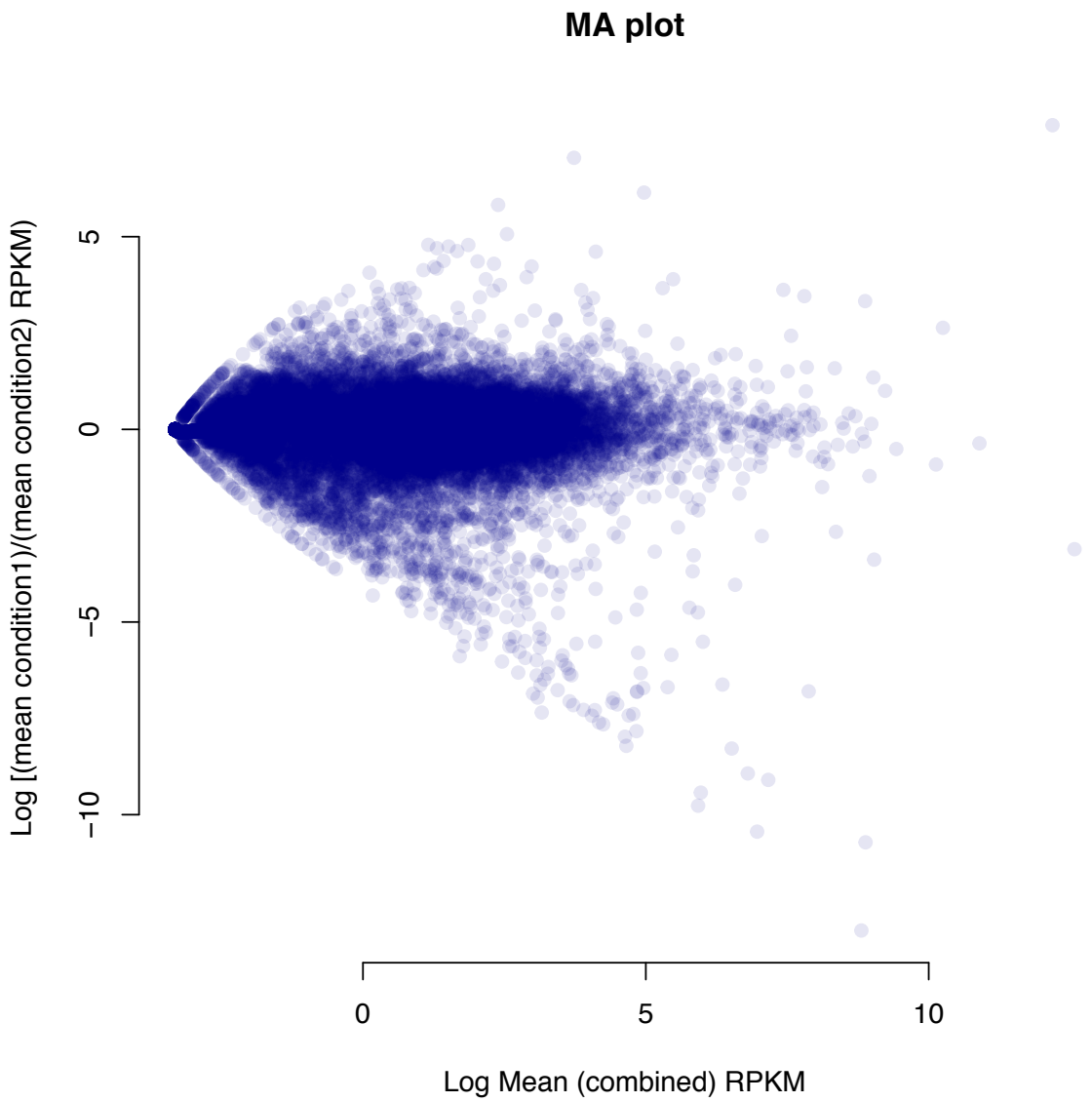
... they are more like Log-Gaussian



And log counts/RPKM can be scatter-plotted



Which can also be looked at as an “MA-Plot”



Hierarchical clustering – when are vector similar?

Gene	Cond1	Cond2	Cond3	Cond4
g_1	2.5	5	7.5	10
g_2	0.2	0.5	0.8	1.1
g_3	0.2	0.3	0.4	11
g_4	2.5	8	8	9

Clustering is about similarity:

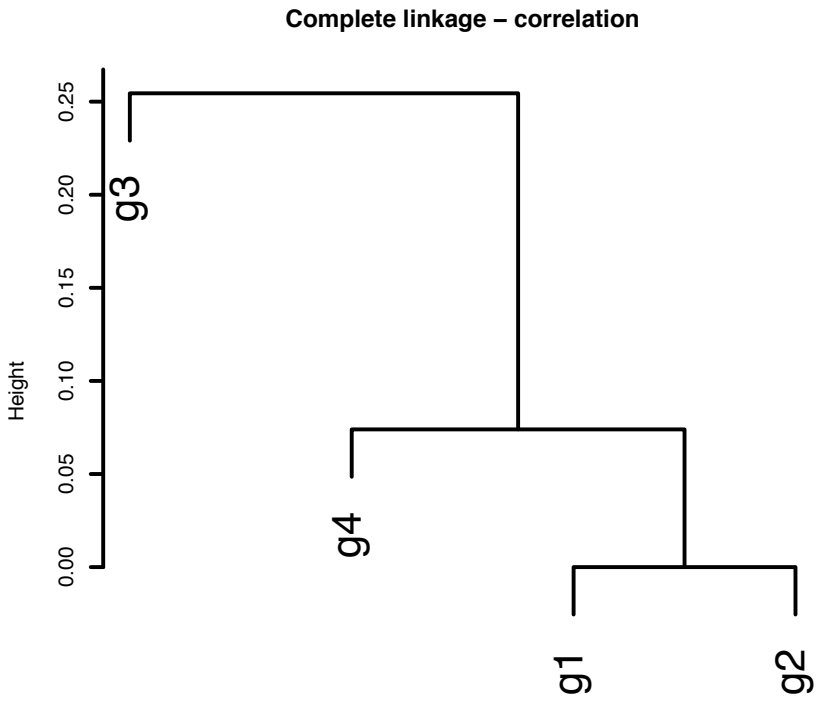
- Between two rows (specified by a distance function)
- Between two sets of rows (specified by the linkage method)

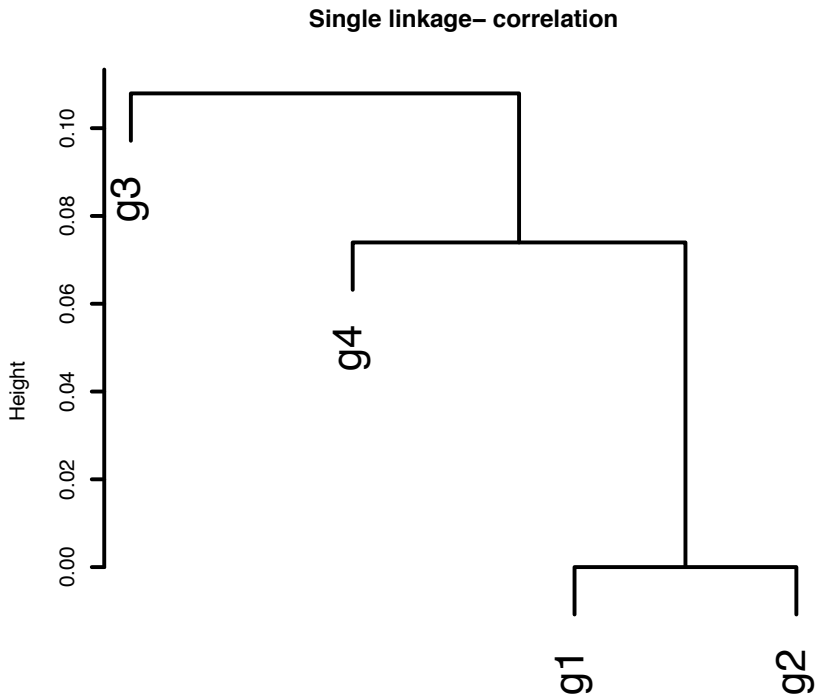
Common similarity approaches

- Distance between rows (or columns)
  - Correlation: `d(r,s) = (1- cor(r,s))/2`
  - Euclidean: $d(r,s) = \sqrt{\Sigma_i (r_i - s_i)^2}$
- Linkage: Distance betwee two sets (`d(R,S)`)
  - Complete: $max\{d(r,s), s \in S, r \in R\}$
  - Average: $mean\{d(r,s), s \in S, r \in R\}$
  - Single: $min\{d(r,s), s \in S, r \in R\}$

Gene	Cond1	Cond2	Cond3	Cond4
g_1	2.5	5	7.5	10
g_2	0.2	0.5	0.8	1.1
g_3	0.2	0.3	0.4	11
g_4	2.5	8	8	9

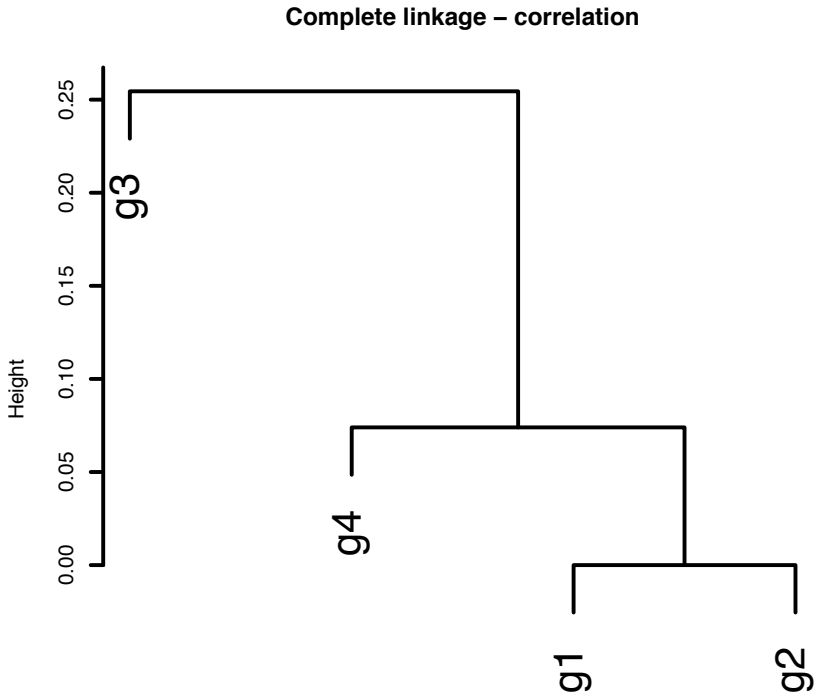
The effect of the linkage method

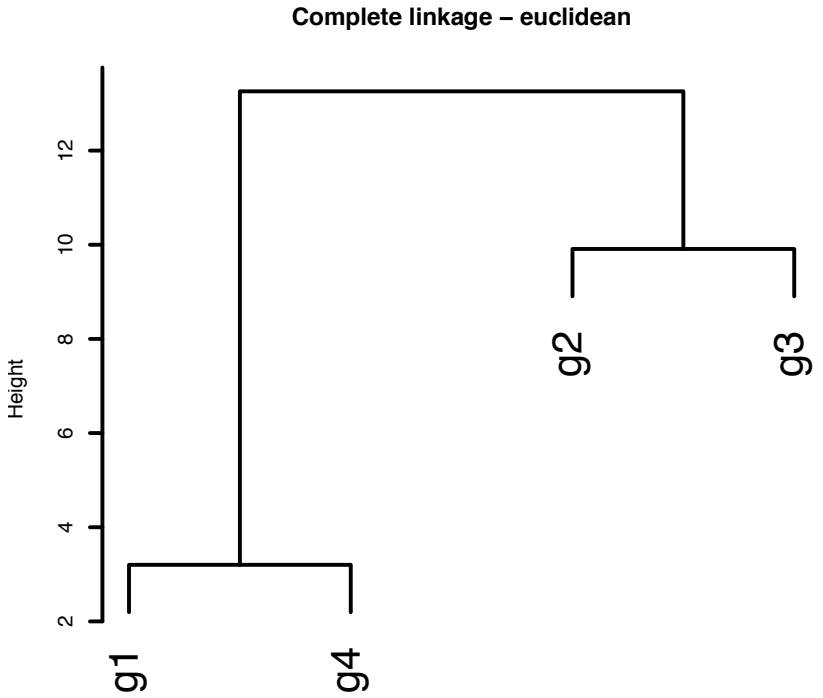




Gene	Cond1	Cond2	Cond3	Cond4
g_1	2.5	5	7.5	10
g_2	0.2	0.5	0.8	1.1
g_3	0.2	0.3	0.4	11
g_4	2.5	8	8	9

Effect of the distance!





Gene	Cond1	Cond2	Cond3	Cond4
g_1	2.5	5	7.5	10
g_2	0.2	0.5	0.8	1.1
g_3	0.2	0.3	0.4	11
g_4	2.5	8	8	9

Playing with clustering

```
#Define the toy matrix#
#######################
m =  rbind (c(2.5,5,7.5,10), c(0.2,0.5,0.8,1.1), c(0.2,0.3,0.4,11), c(2.5,8,8,9))

#Give column and row names#
###########################
rownames(m) = c("g1","g2","g3","g4");
colnames(m) = c("c1","c2","c3","c4");


#Compute the correlation distance matrix#
########################################
submat.dist = as.dist( (1 - cor(t(m)) )  /2 );


#Plot clustering with the three main methods#
#############################################
plot( hclust(submat.dist, method="complete",members=NULL), main="Complete linkeage -
correlation", sub="", xlab="", lwd=3);
plot( hclust(submat.dist, method="average",members=NULL),  main = "Average Linkeage -
correlation", sub="", xlab="", lwd=3);
plot( hclust(submat.dist, method="single",members=NULL),  main = "Single Linkeage- correlation",
sub="", xlab="", lwd=3);

#Plot clustering with the three main methods, using the euclidean distance#
############################################################################
plot( hclust(dist(m), method="complete",members=NULL), main="Complete linkeage - euclidean",
sub="", xlab="", lwd=3);
plot( hclust(dist(m), method="average",members=NULL),  main = "Average Linkeage -  euclidean",
sub="", xlab="", lwd=3);
plot( hclust(dist(m), method="single",members=NULL),  main = "Single Linkeage - euclidean",
sub="", xlab="", lwd=3);
```