Analysis of genome-scale count data in Bioconductor

Mark D. Robinson[1,2] and Davis J. McCarthy[1]

[1]Bioinformatics Division, The Walter and Eliza Hall Institute of Medical Research
[2]Epigenetics Laboratory, Garvan Institute of Medical Research

BioC 2010

Outline

1. Applications
2. Summarization
3. Statistical models for count data
4. “Normalization”

Preliminaries
(~40min)

Practical
(~20min)

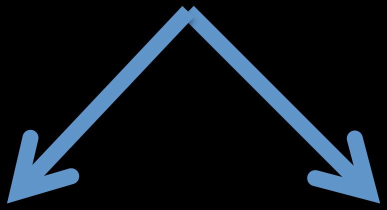
5. Sharing information over entire dataset
6. Statistical testing
7. Other considerations – error model and more complex designs

More advanced topics
(~30min)

Practical
(~30min)

(Current) Bioconductor tools:
baySeq, DEGseq, DESeq, **edgeR**

Sequencing experiments used for:



Sequence of (mapped) read

e.g. genome sequencing, SNP/mutation mapping, genomic rearrangements, etc.

Position of mapped read

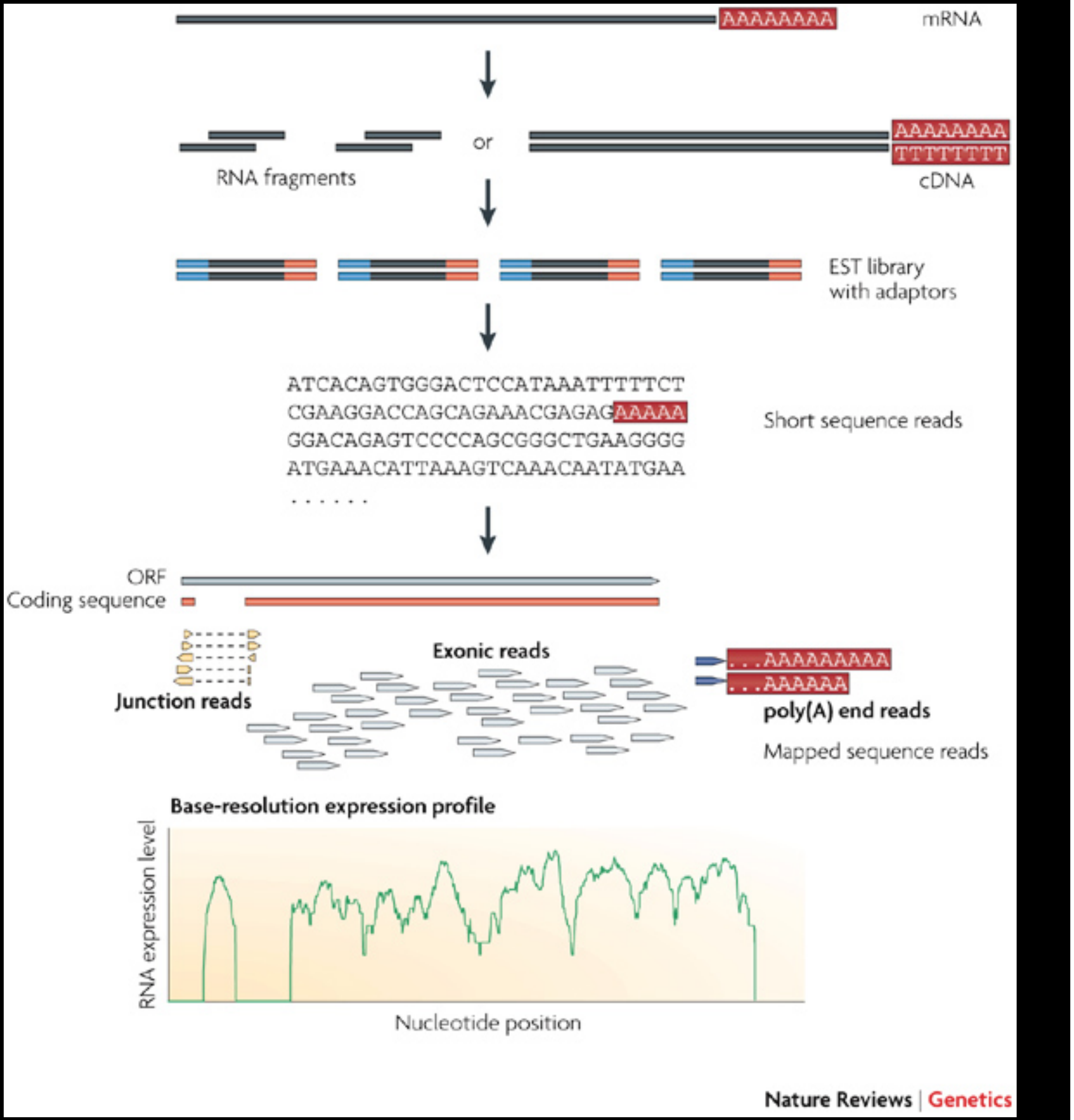
e.g. RNA-seq, tag-seq for expression, ChIP-seq for TF binding or histone modifications, MeDIP-seq for DNA methylation, etc.

Applications

- **Differential** gene expression: RNA-seq, "Tag"-seq, etc.
- **Differential** enrichment: histone modifications, other types of "enrichment"-based sequencing e.g. ChIP-seq, MeDIP-seq, etc.
- Analyses of **changes** in other tables of counts: e.g. peptide counts from MS/MS experiments, metagenomics experiments.

Example:

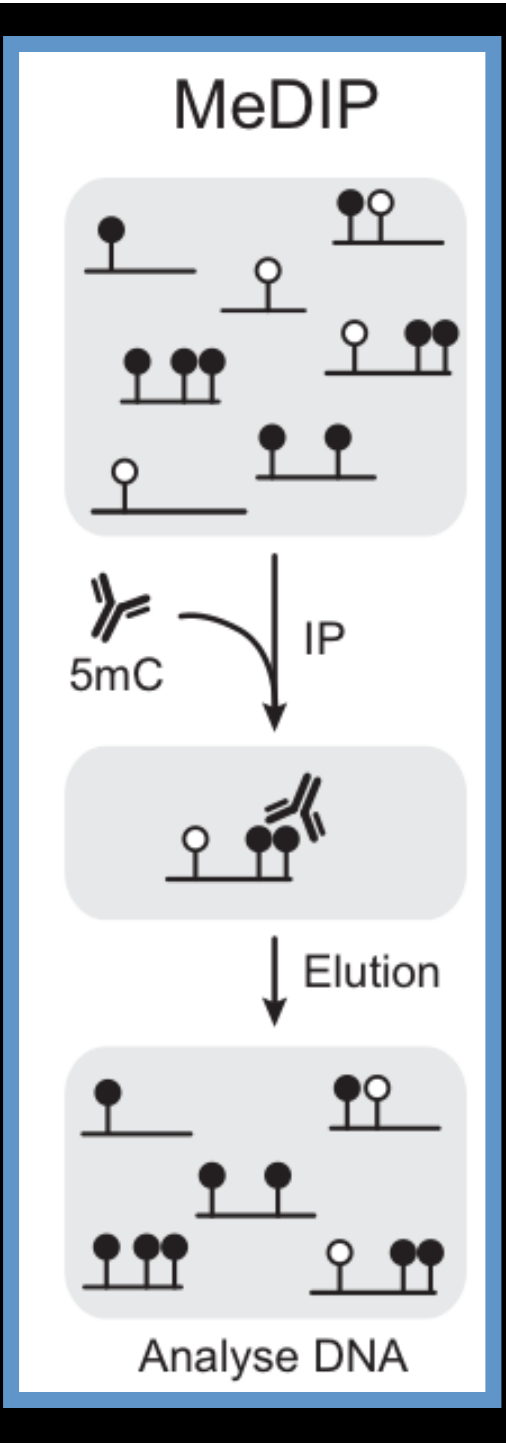
RNA-seq (or similar) for gene expression



Nature Reviews | Genetics

Example:

Enrichment of subset of the genome (e.g. ChIP for histone modifications or DNA methylation)



Summarization

Summarization

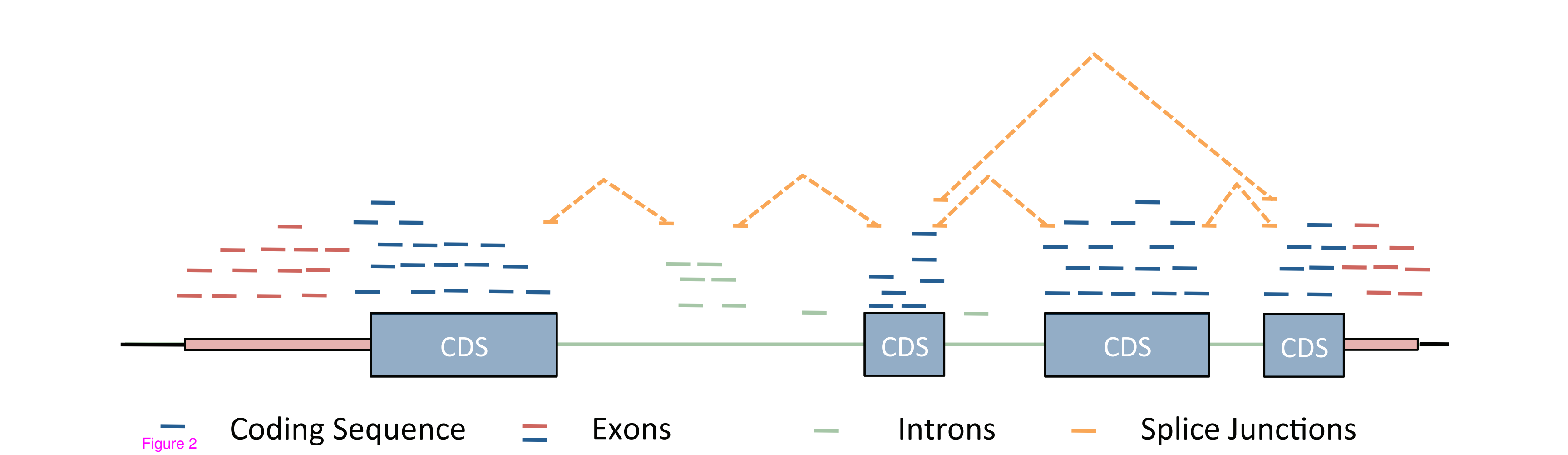
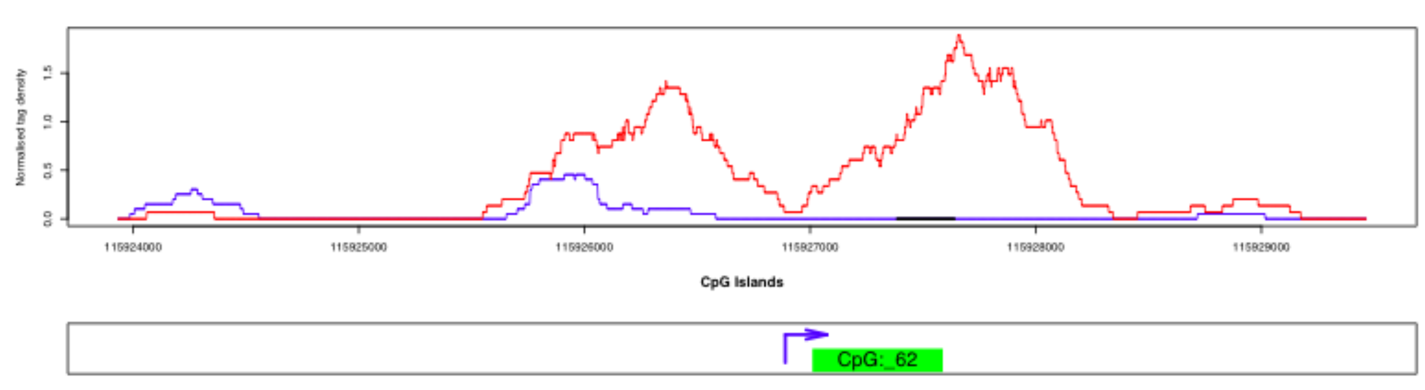


Figure 2



What does genome-scale count data look like?

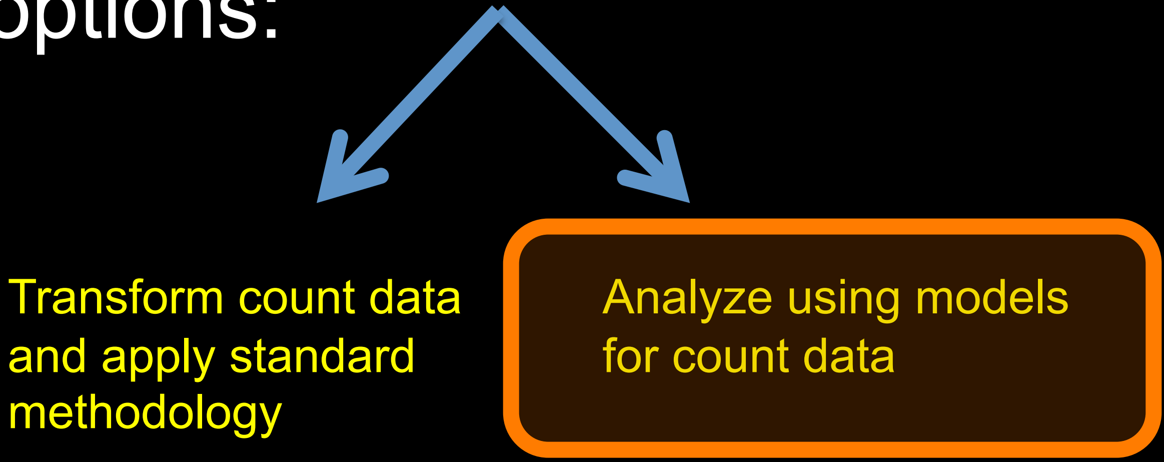
- e.g. RNA-seq

Tag ID	A1	A2	A3	A4	B1	B2	B3
ENSG00000124208	478	619	628	744	483	716	240
ENSG00000182463	27	20	27	26	48	55	24
ENSG00000125835	132	200	200	228	560	408	103
ENSG00000125834	42	60	72	86	131	99	30
ENSG00000197818	21	29	35	31	52	44	20
ENSG00000125831	0	0	2	0	0	0	0
ENSG00000215443	4	4	4	0	9	7	4
ENSG00000222008	30	23	29	19	0	0	0
ENSG00000101444	46	63	58	71	54	53	17
ENSG00000101333	2256	2793	3456	3362	2702	2976	1320
…	… tens of thousands more tags …						

Statistical models for count data

Count data

- Count data (e.g. RNA-seq) is discrete, not continuous
- Statistical methods designed for microarrays are not directly applicable
- Two options:



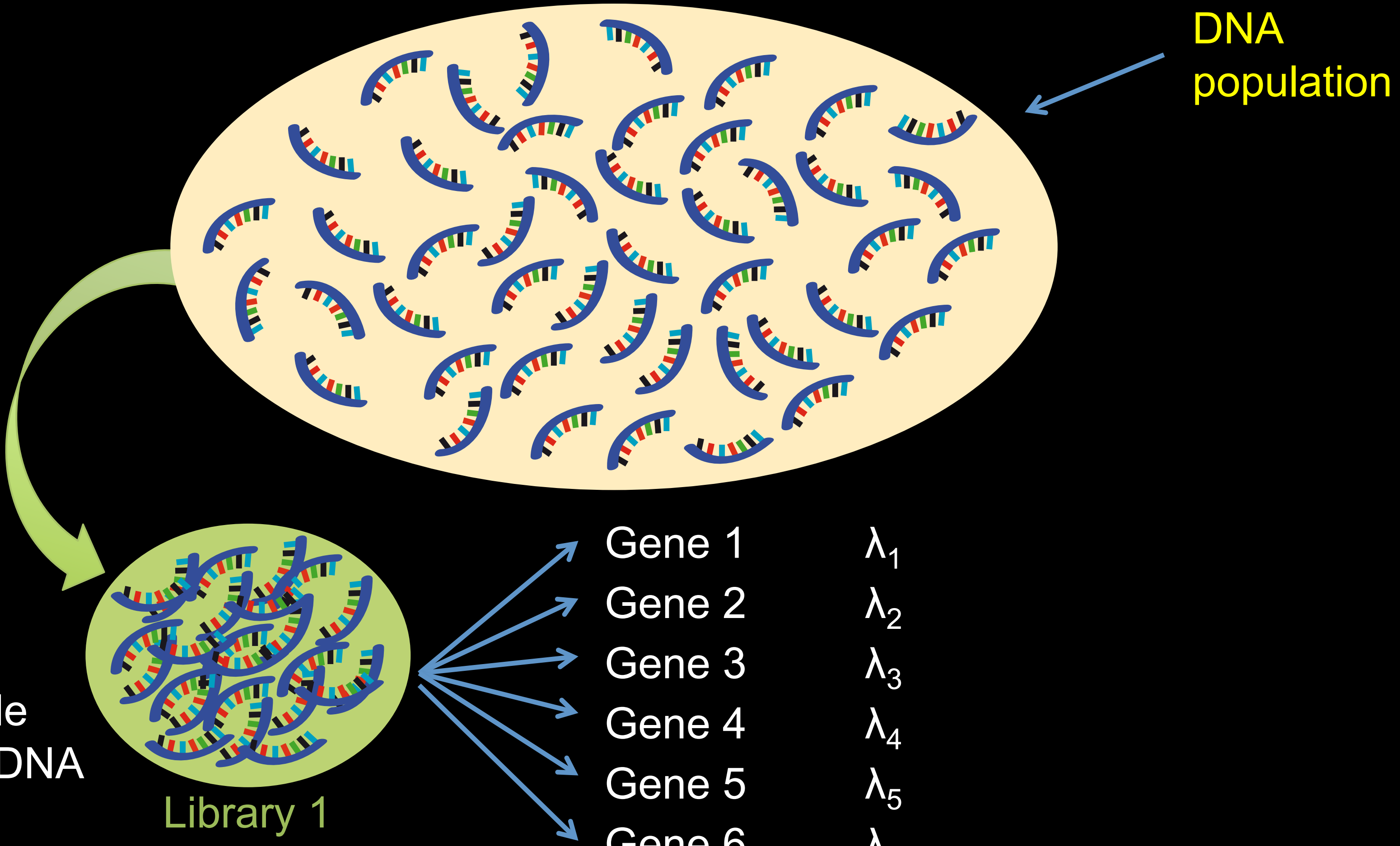
Transform count data and apply standard methodology

Analyze using models for count data

Count data

- BUT we have learned much from the analysis of microarray data
- Methods that share information over the whole dataset generally:
  - stabilize parameter estimation
  - improve performance of making inferences

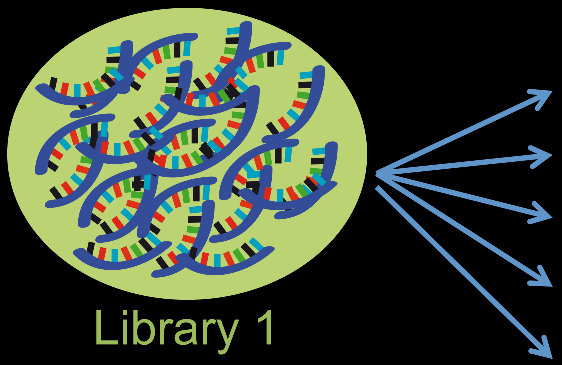
Poisson arises naturally from multinomial sampling



- Take sample
- Sequence DNA

Gene 1	λ_1
Gene 2	λ_2
Gene 3	λ_3
Gene 4	λ_4
Gene 5	λ_5
Gene 6	λ_6
…	

Reads for a single gene (single library) are binomial distributed



Library 1

…

Gene i λ_i

…

Y_i ~ Binomial(M, λ_i)

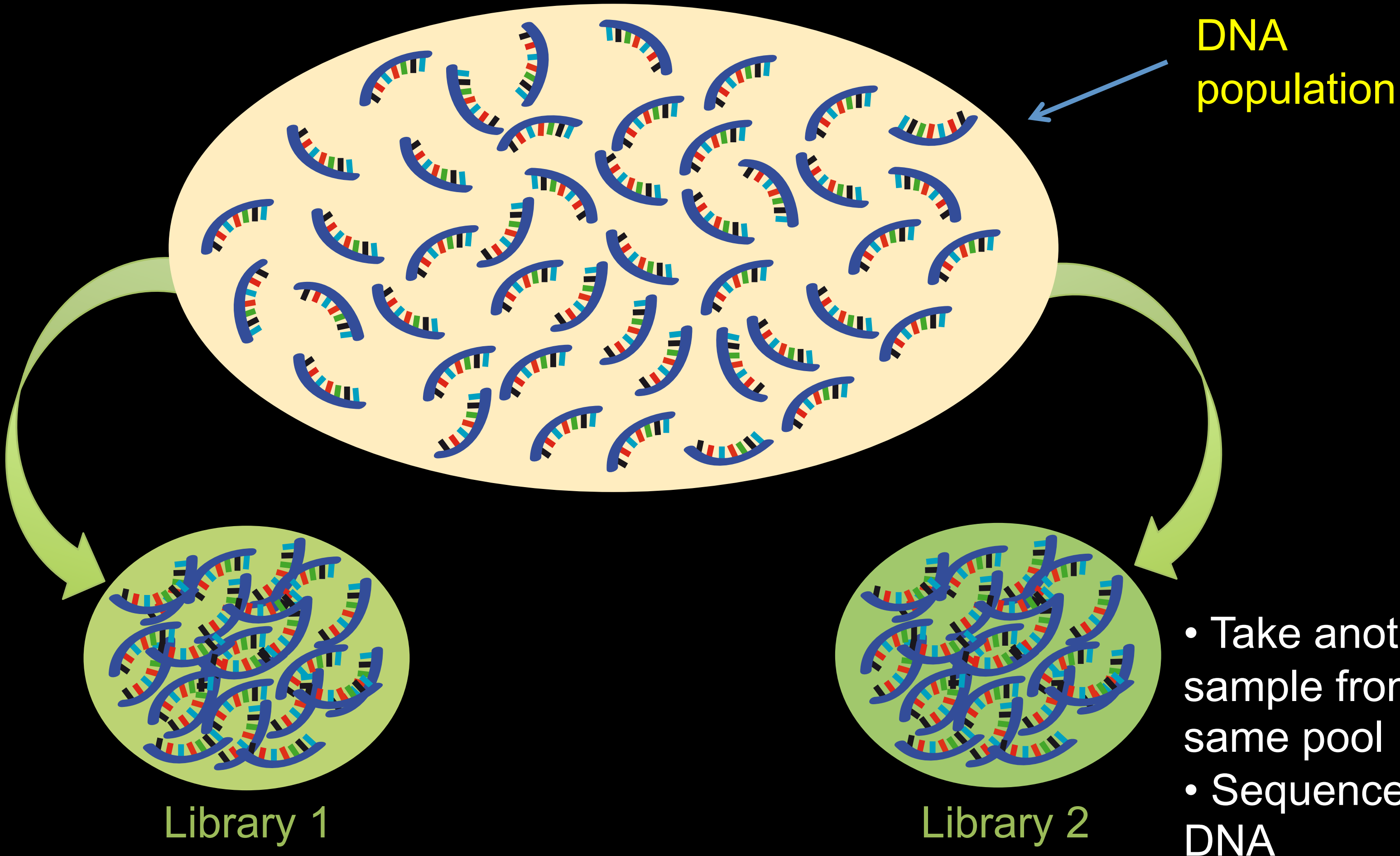
Y_i - observed number of reads for gene i

M - total number of sequences

λ_i - proportion

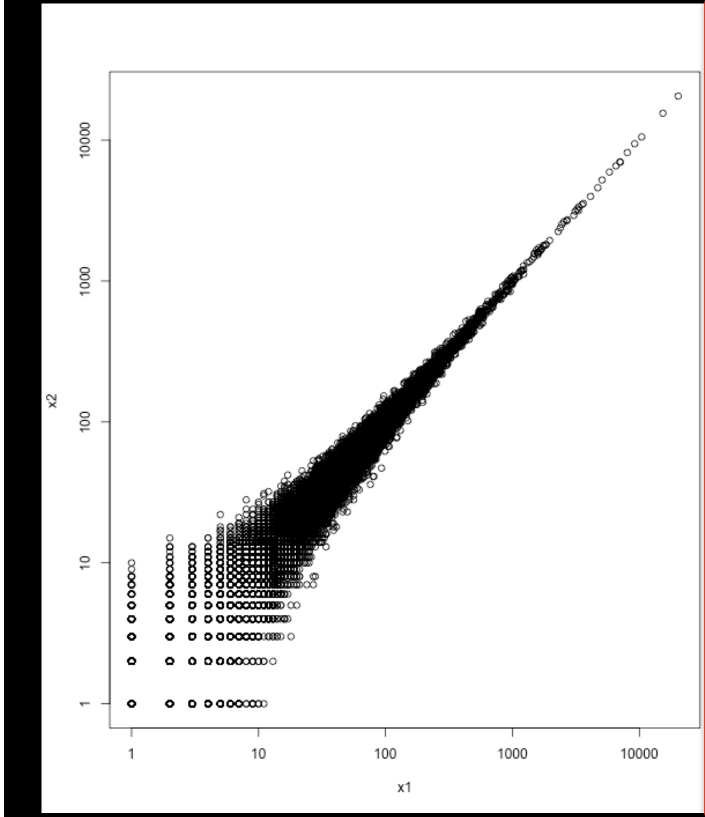
Large M, small λ_i → approximated well by Poisson($\mu_i = M \cdot \lambda_i$)

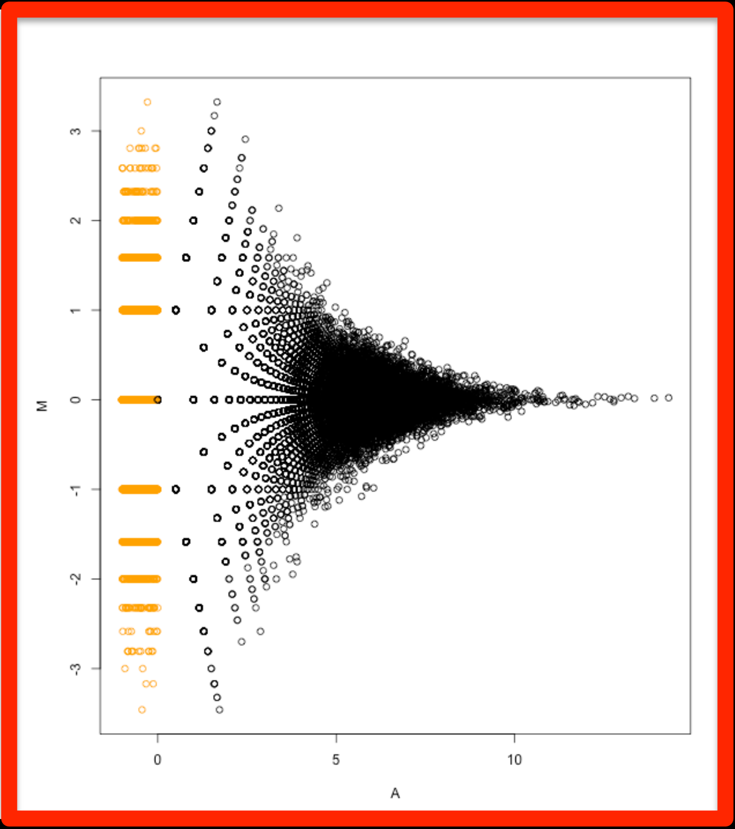
Technical replication



- Take another sample from same pool
- Sequence DNA

Poisson replication induces a vuvuzela-shaped "MA"-plot





And the theory validates that this behaviour should exist: M is essentially a **log-relative-risk**

Power (to detect changes) is higher at higher counts Implications for downstream analysis.

$$M_g = \log_2 \frac{Y_{gk}/N_k}{Y_{gk'}/N_{k'}}$$

$$A_g = \frac{1}{2}\log_2\left(Y_{gk}/N_k \bullet Y_{gk'}/N_{k'}\right) \text{ for } Y_{g\bullet} \neq 0$$

Statistical models

- For count data, variance increases with mean
- Starting point: Poisson model
- Poisson has simplest mean-variance relationship

Poisson

- Variance is equal to the mean
- One-parameter model: mean for each gene

$$Y_i \sim \mathrm{Pois}(\ \mu_i\ )$$

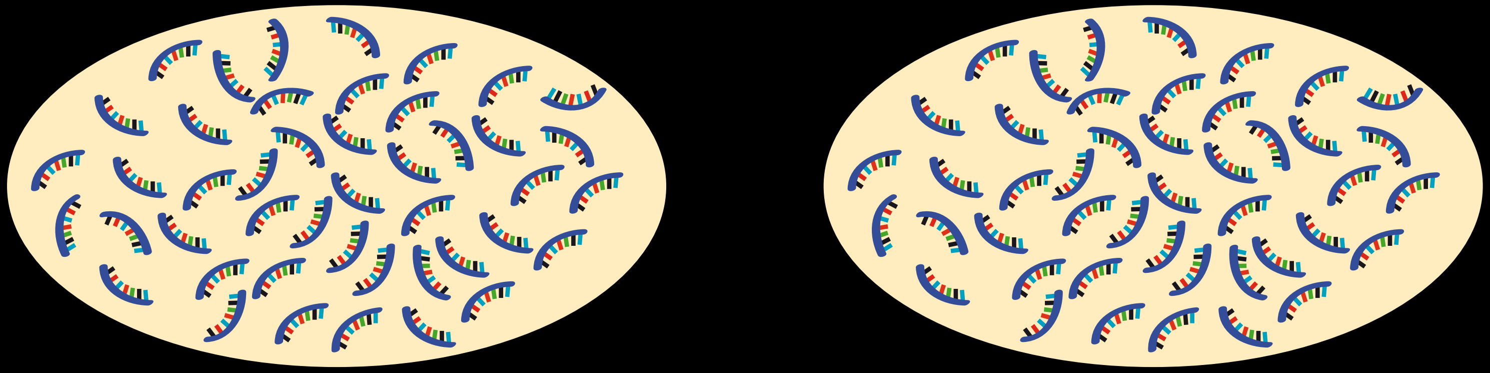
$$\mu_i = M * \lambda_i$$

- M = library size
- λ_i = relative contribution of gene i

Poisson describes technical variance

- Marioni et al (2008) show that there is little technical variance in RNA-seq
- Poisson model is (probably) adequate for assessing DE when there are only technical reps
- But this is not the end of the story …

Biological replication



2 or more independent DNA populations from the same experimental condition

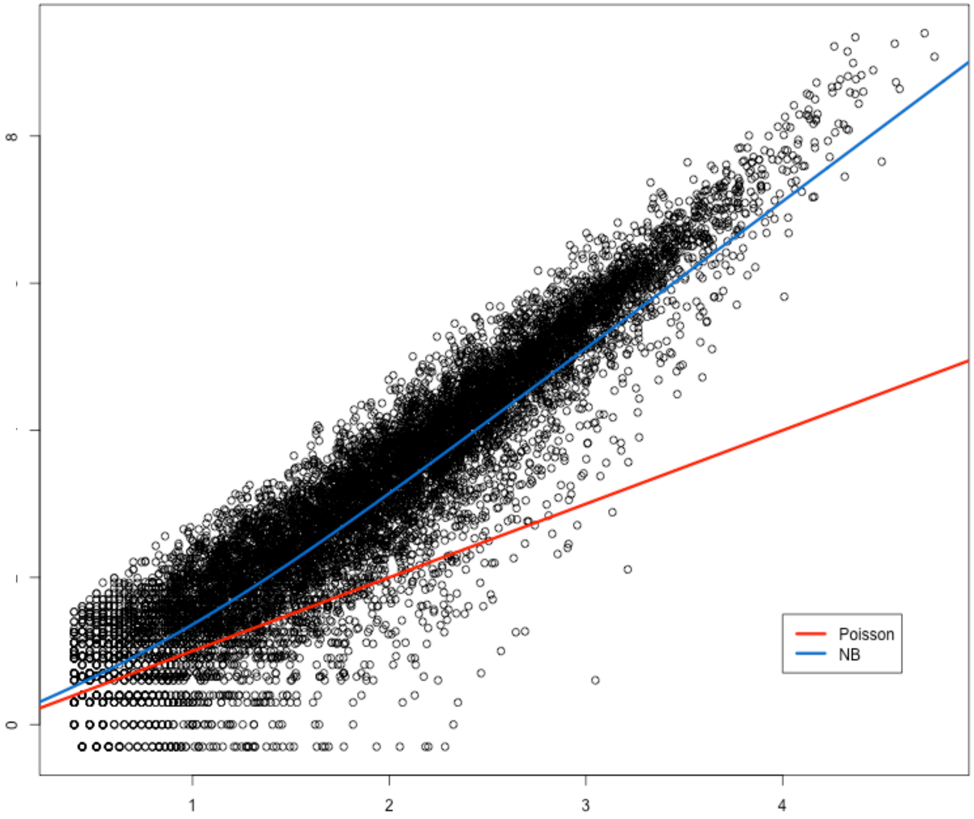
Generally, experimenters will want biological replication for generalizable results

Overdispersion: extra-Poisson variation

- If there are ANY further sources of variation, there is more variation in data than Poisson model can account for
- Poisson model underestimates variation -> false positives
- Need a model that can account for this extra variation

Overdispersion is present in real data

Mean-variance plot for slime-mould dataset hr00 and hr24 (2 vs 2)



Comparing expression levels from Dictyostelium discoideum at hr00 and hr24 – two biological replicates at each time point.
RNA-seq data from Parikh et al. *Genome Biology* 2010, 11:R35 http://genomebiology.com/2010/11/3/R35

Sources of variation: technical and biological

- Technical: same pool of RNA sequenced separately (e.g. different lanes)
- Biological: RNA from different biological sources (e.g. individuals) under the same experimental conditions
- Other: extra-Poisson variation also introduced by other processes, e.g. different library preparations, protocols etc.

Natural extension to Poisson: negative binomial model

- Introduce the **dispersion** parameter

$$Y_i \sim NB(\mu_i , \varphi_i)$$

- Still have mean expression level

$$\mu_i = M * \lambda_i$$

- M = library size, λ_i = “conc” of gene DNA
- Variance is a quadratic function of mean:

$$Var(Y_i) = \mu_i (1 + \mu_i \varphi_i)$$

Coefficient of variation

- Dispersion is squared coefficient of variation
- Measure of similarity/variability btw samples
- E.g. dispersion = 0.2 -> coef of var = 0.45
- Interpretation: true expression levels of genes vary by 45% btw replicates
- Separate biological and technical variation

Problem: small sample size

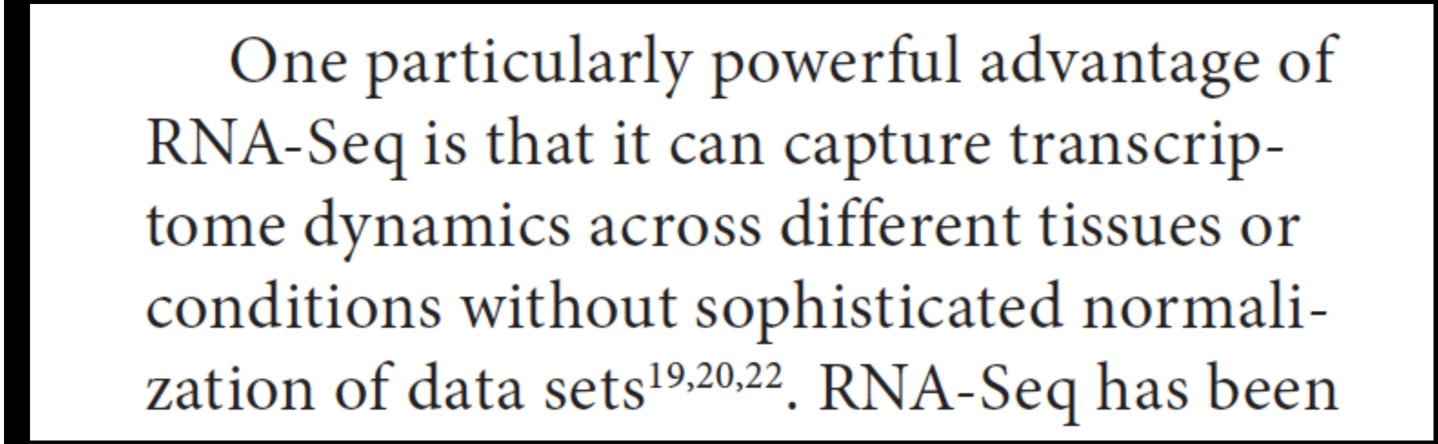
- RNA-seq experiments will typically have small sample sizes (e.g. n=7)
- Standard methods for estimating the dispersion for each gene produce very unreliable estimates
- Lesson from microarrays: share information between genes (variance structure) to improve inference

Common dispersion model

- One approach: use same value for the dispersion for all genes
- Estimate using all genes in dataset (conditional max likelihood)
- Produces a reliable estimate
- Nice biological interpretation, but can be heavy handed

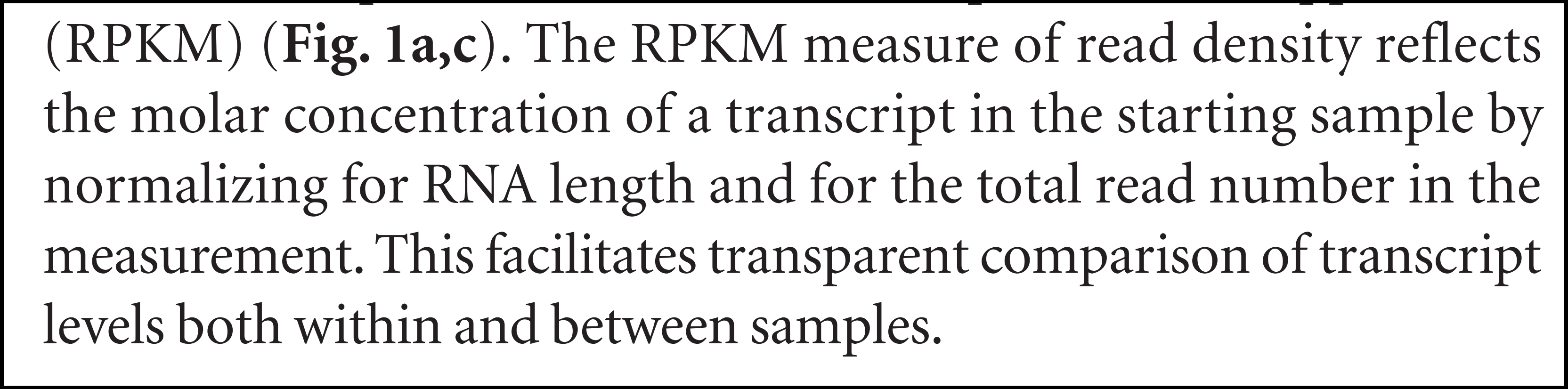
Normalization

Wang et al. 2008 Nature Reviews Genetics



One particularly powerful advantage of RNA-Seq is that it can capture transcriptome dynamics across different tissues or conditions without sophisticated normalization of data sets[19,20,22]. RNA-Seq has been

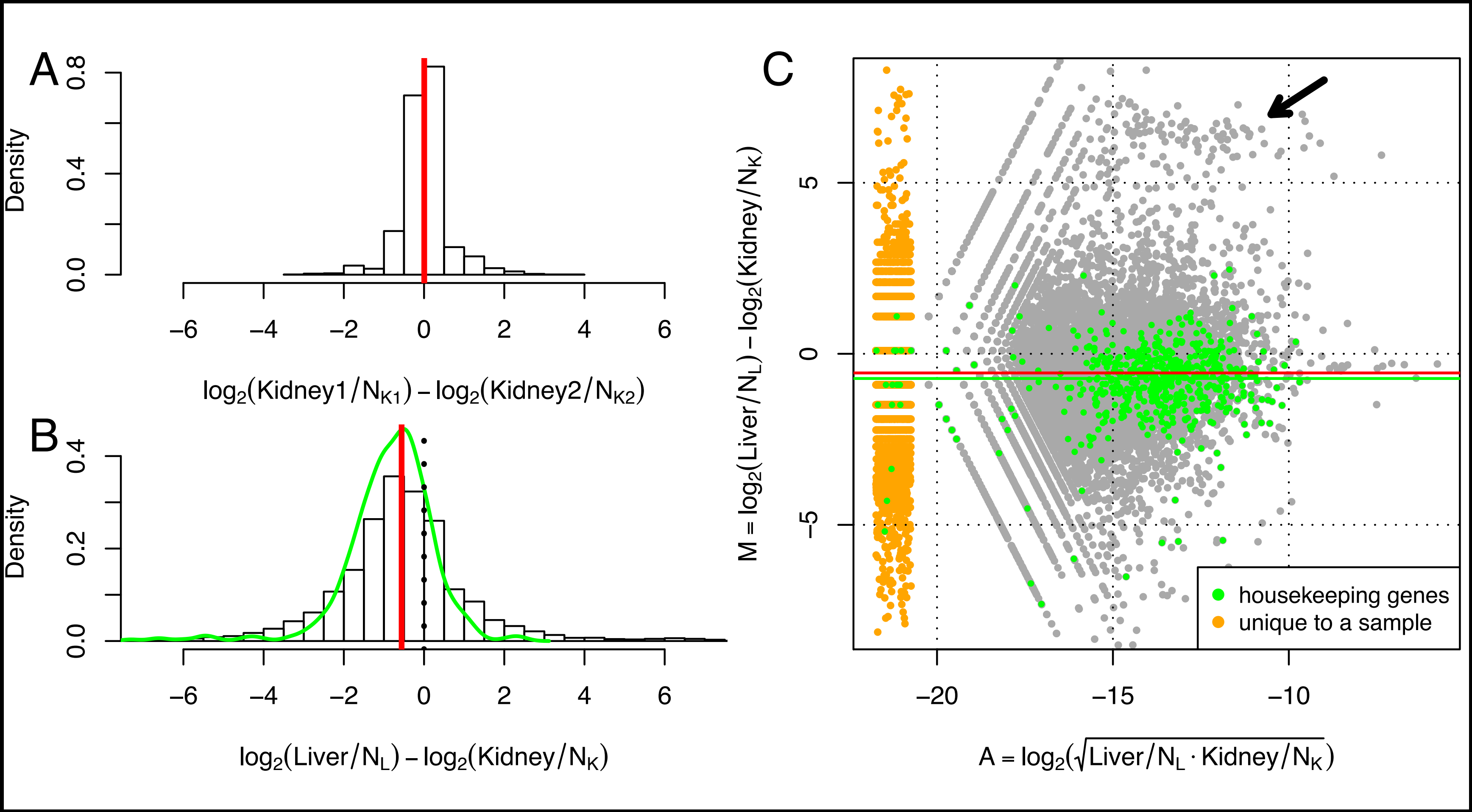
Mortazavi et al. 2008 Nature Methods



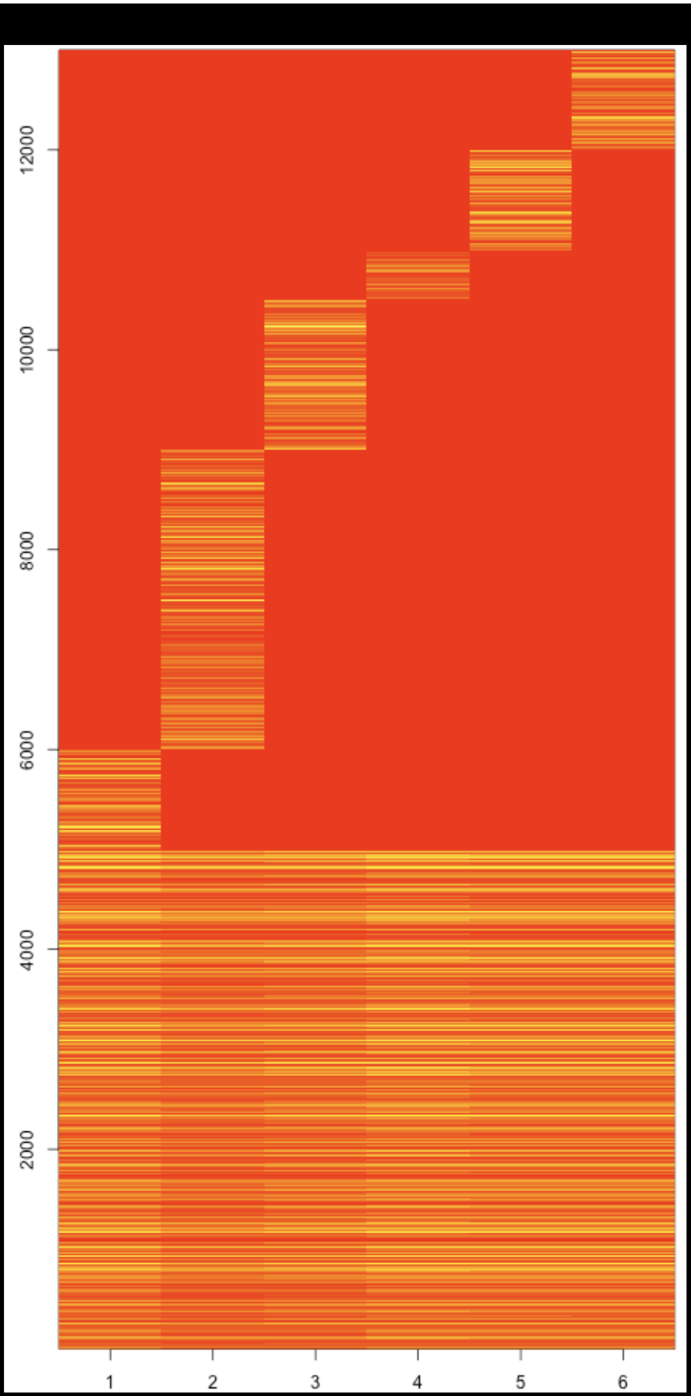
(RPKM) (**Fig. 1a,c**). The RPKM measure of read density reflects the molar concentration of a transcript in the starting sample by normalizing for RNA length and for the total read number in the measurement. This facilitates transparent comparison of transcript levels both within and between samples.

But, this is not the full story.

Kidney and Liver RNA have very different composition



Robinson and Oshlack (2010) Genome Biology



“Composition” of sampled DNA can be an important consideration

- Hypothetical example: Sequence 6 libraries to the **same** depth, with varying levels of *unique-to-sample* counts
- Composition can induce (sometimes significant) differences in counts

Red=low, goldenyellow=high

The adjustment to data analysis is straightforward

- Assumption: core set of genes that do not change in expression.
- Pick a reference sample, compute trimmed mean of M-values (TMM) to reference
- LTM($[Y_{gk}/M_k] / [Y_{gk'}/M_{k'}]$) estimates $S_{k'}/S_k$
- Adjustment to statistical analysis:
  - Use “effective” library size (edgeR)
  - Use additional offset (GLM)

Outline

1. Applications
2. Summarization
3. Statistical models for count data
4. “Normalization”

Preliminaries (~40min)

Practical (~20min)

5. Sharing information over entire dataset
6. Statistical testing
7. Other considerations – error model and more complex designs

More advanced topics (~30min)

Practical (~30min)

(Current) Bioconductor tools:
baySeq, DEGseq, DESeq, **edgeR**

Sharing information over entire dataset

Extending the common dispersion model

- Common dispersion offers sig. stabilization vs. naïve tagwise estimation, esp. in small samples.
- Have found common dispersion model to give good results
- Downside: not generally true that each tag has the same dispersion.
- Would like stabilized individual tagwise dispersions

Moderated tagwise dispersions

- Moderate individual dispersions towards common value
- Stabilize dispersion ests. by sharing variance structure over all genes
- IDEA: ‘Squeeze’ individual dispersion ests. towards common value---larger ests. shrink, smaller ests. get larger

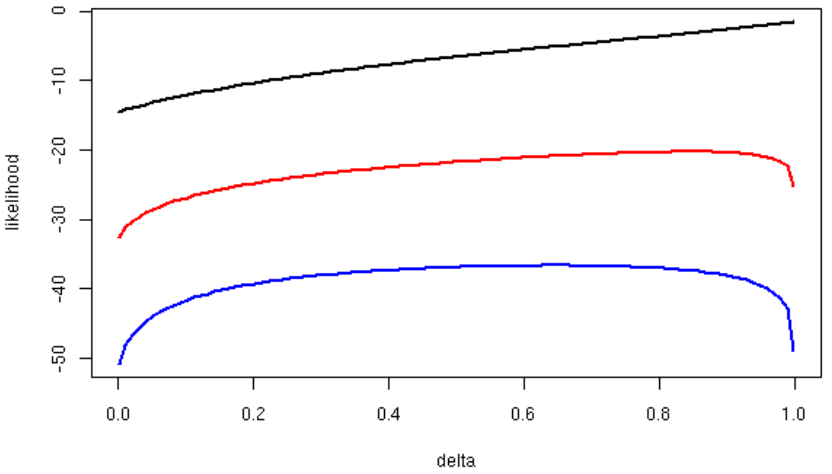
Weighted Likelihood

- WL is the individual log-likelihood plus a weighted version of the common **log-likelihood**:

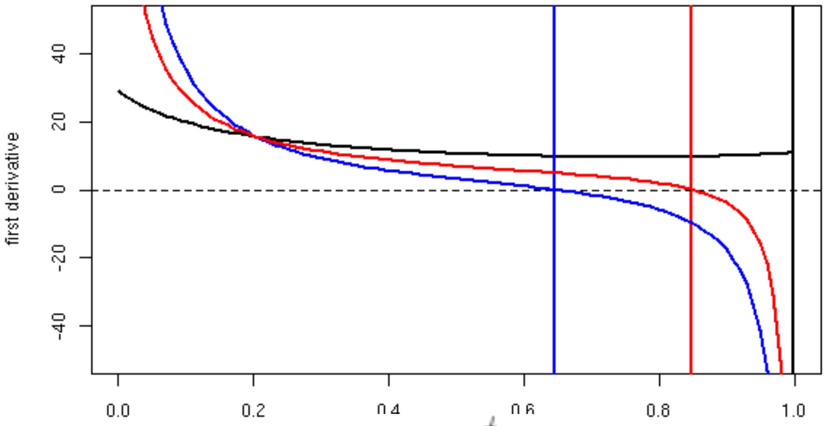
$$\mathrm{WL}(\phi_g) = \underset{\overset{\uparrow}{(1-\alpha)}}{\vphantom{l}} l_g(\phi_g) + \alpha\, l_C(\phi_g)$$

- l_g here is the the quantile-adjusted conditional likelihood
- Plot shows:
  - – Black: Likelihood for single tag
  - – Blue: Likelihood averaged over all tags (common dispersion)
  - – Red: Linear combination of the two

Log-Likelihood



Score (1st derivative of LL)



$$\delta = \frac{\phi}{\phi+1}$$

New alternatives

- DESeq: fit an empirical mean-variance relationship using all data [Anders and Huber 2010]
- baySeq: use all data to form an empirical distribution [Tom Hardcastle]

Statistical testing for count data

Assessing DE: a statistical problem

- Two group setting*: for each gene, estimate λ_1 and λ_2 (mean level for each group) and the dispersion

Tag ID	A1	A2	A3	A4	B1	B2	B3
ENSG00000215443	14	12	5	13	6	16	14
ENSG00000222008	97	113	90	101	10	13	10
ENSG00000101444	46	63	58	71	54	53	1001
ENSG00000101333	256	793	4156	5463	1705	976	1320
…	… tens of thousands more tags …						

- Conduct a hypothesis test for λ_1 and λ_2
- Obtain a p-value for the significance of DE for each gene

*Generalises to n groups

Significance testing

- Simple hypothesis test

$$H_0: \lambda_1 = \lambda_2$$

vs

$$H_A: \lambda_1 != \lambda_2$$

- Easy to state, but requires some sophisticated statistics to test appropriately

Multiple testing

- We fit the same model to each gene
- Fit the same model thousands of times
- Expect some (many) genes to appear significantly DE just by chance
- Need to adjust p-values for multiple testing (control the false discovery rate)
- Need accurate p-values to start with

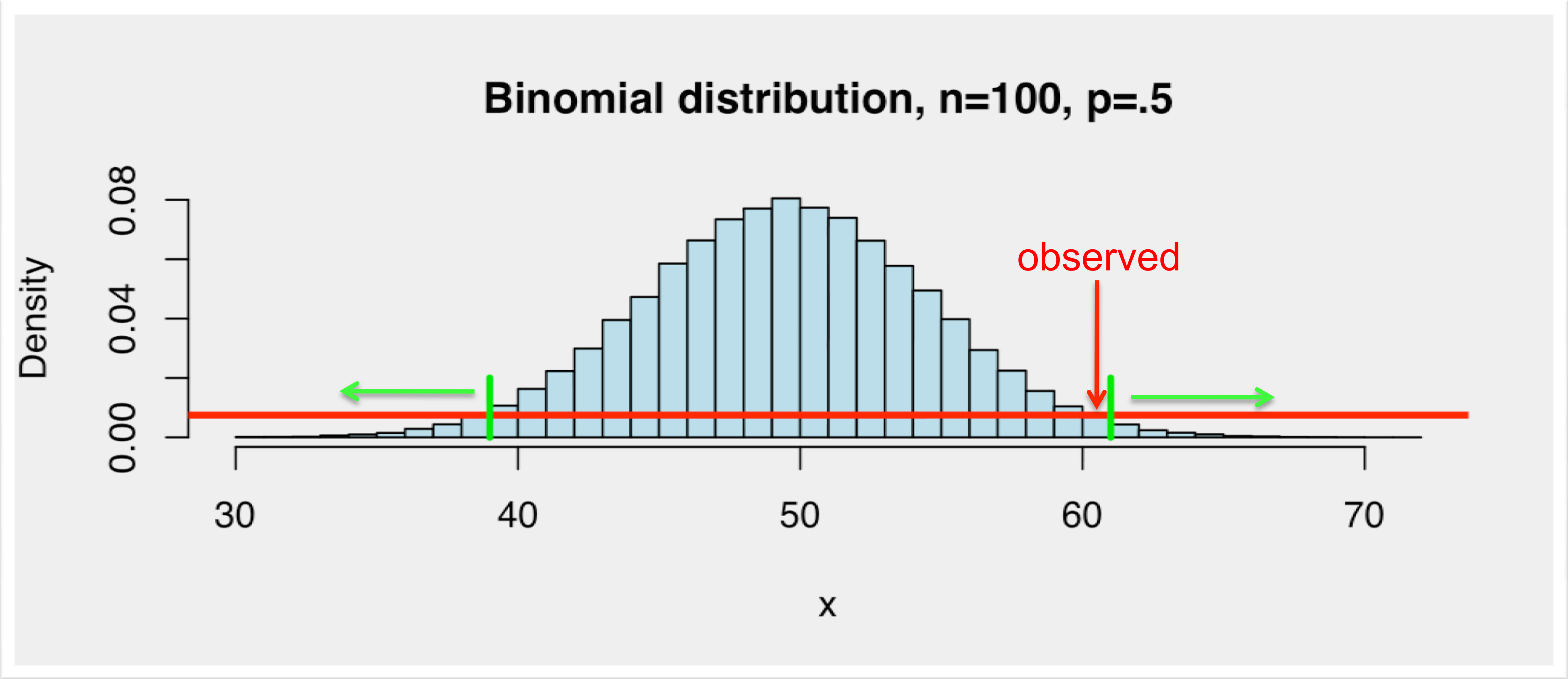
Further considerations

- RNA-seq experiments: very small sample-sizes but need accurate p-values
- Asymptotic tests (Score, Likelihood Ratio, Wald) not ideal
- Instead: exact tests for the Poisson and NB models
- Exact tests give accurate p-values in small sample experiments

Exact testing

- By conditioning on the total sum of counts for each gene we obtain conditional distributions
- Can compute exact p-values from conditional distributions

Binomial exact testing



- Poisson model: sum of Poisson RVs is a Poisson RV
- Conditional distribution (on total sum for a gene) is multinomial
- Two groups: can compute exact p-value for DE from binomial distribution

Exact test for NB distribution

- Sum of NB RVs is a NB RV, if library sizes (means) are equal, under the null hypothesis of no difference
- Conditioning gives ‘overdispersed multinomial’ from which we can compute exact p-values as per binomial test
- Statistical sophistication: quantile-adjustment to equalise library sizes and enable exact test for NB model
- Size of dispersion has big effect on significance of DE

Effect of dispersion

```
> d.tuch$counts[hicom.lotgw,order(d.tuch$samples$group)]
        N8 N33   N51   T8  T33   T51
FABP4   62  62   387    0   37  2022
MMP1    68  74 11190 1883 1998 24955
TTTY15 241   1     0   46    0     0
> de.tuch.com$table[hicom.lotgw,]
       logConc  logFC  p.value
FABP4   -15.59  2.016 0.005006
MMP1    -11.59  1.865 0.008713
TTTY15  -17.90 -2.281 0.002998
> de.tuch.tgw$table[hicom.lotgw,]
       logConc  logFC p.value
FABP4   -15.60  2.018 0.05040
MMP1    -11.59  1.866 0.05771
TTTY15  -17.87 -2.238 0.07857
> d.tuch$common.dispersion
[1] 0.3325
> d.tuch$tagwise.dispersion[hicom.lotgw]
[1] 0.6694 0.6207 0.9417
```

Limitations of exact tests

- Exact tests only implemented for *pairwise* comparisons between groups
- Can only be used for single-factor (one-dimensional) experimental design
- Cannot include any other factors or covariates in our model for DE
- qCML approach to estimating dispersion also only for single-factor design

Limitations of exact testing

- E.g. cannot account for **paired** samples in Tuch et al (2010) data
- Matched tumour/normal oral tissue from 3 patients (6 RNA samples)

	Normal	**Tumour**
Patient 8	N8	T8
Patient 33	N33	T33
Patient 51	N51	T51

Paired oral squamous cell carcinoma and healthy oral tissue samples from three patients. RNA-seq data from Tuch et al. Tumor transcriptome sequencing reveals allelic expression imbalances associated with copy number alterations. *PLoS ONE* (2010) vol. 5 (2) pp. e9317. doi:10.1371/journal.pone.0009317

Further considerations

More complicated experiments

- We would like to be able to analyse more complicated experimental designs
- Paired samples, time-series, covariates, batch/day effects etc.
- Need to go beyond the qCML and exact tests (sadly)

GLM methods for complicated designs

- Propose to use GLM (generalized linear model) methods for more complicated designs
- Currently implementing likelihood ratio tests
- Cox-Reid approximate conditional inference for estimating dispersion
- Cutting edge…hopefully ready to go soon!

Example: Cancer dataset

- RNA-seq data from Tuch et al (2010)
- Comparing oral squamous cell carcinoma tissue to matched healthy oral tissue
- 6 samples, paired design

Sample	Description
N8	healthy oral tissue from patient 8
T8	oral tumour tissue from patient 8
N33	healthy oral tissue from patient 33
T33	oral tumour tissue from patient 33
N51	healthy oral tissue from patient 51
T51	oral tumour tissue from patient 51

*Ignore paired design for now and treat as simple comparison of healthy and tumour groups

Exact test in edgeR: tagwise disp

```
> de.tuch.tgw <- exactTest(d.tuch,common.disp=FALSE)
Comparison of groups: tumour - normal
> topTags(de.tuch.tgw, n=5)
Comparison of groups: tumour - normal
        logConc         logFC           PValue          FDR
TNNC2   -16.63025       -6.439491       6.237545e-12    1.146710e-07
KRT36   -19.02052       -8.087423       1.723154e-11    1.583923e-07
ADIPOQ  -19.88465       -7.30664        1.133512e-10    6.946160e-07
SPP1    -14.90146       6.057058        3.448317e-10    1.288116e-06
CA3     -15.43170       -6.462589       3.782377e-10    1.288116e-06
> top.tgw <- rownames(topTags(de.tuch.tgw, n=5)$table)
> d.tuch$counts[top.tgw,c(1,3,5,2,4,6)]
        N8   N33    N51     T8  T33  T51
TNNC2   590  1627  1239     1   8    39
KRT36   711  104    70      2   1    1
ADIPOQ  111  12     575     1   1    1
SPP1    19   29     158     378 8517 1681
CA3     1859 4259   557     1   35   73
```

GLM results

```
> glm.res.com[o1[1:10],]
          LRT p-val    N8   N33   N51  T8  T33  T51
TMPRSS11B 9.508e-15  2601  7874  3399   3  322    9
TNNC2     2.388e-13   590  1627  1239   1    8   39
CKM       2.609e-13  4120  5203 24175   5   24 1225
MAL       4.009e-13  2742  3977  1772   3  264    8
CRNN      6.646e-13 24178 22055 12533  49 2353   26
PI16      6.781e-13   231   216  1950   0    2   35
KRT36     2.229e-12   711   104    70   2    1    1
IL1F6     3.513e-12   367  1825   809  10   45    1
MYBPC1    3.641e-12  4791  4145 15766  10   14 1319
MUC21     1.376e-11  4161  3432  1722   7  517    5
```

Dispersion estimation

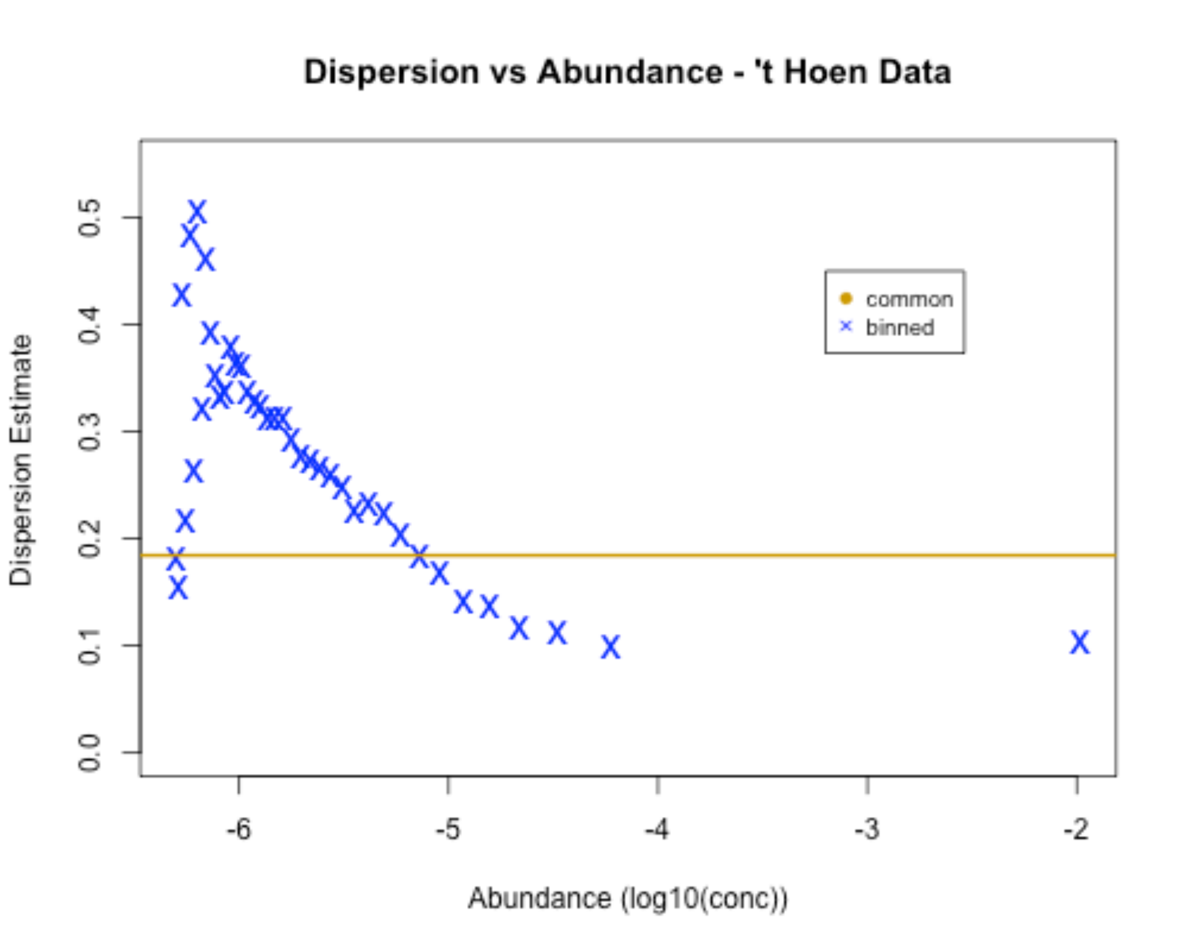
- Estimating the dispersion appropriately for GLMs

→ Cox-Reid approximate conditional inference

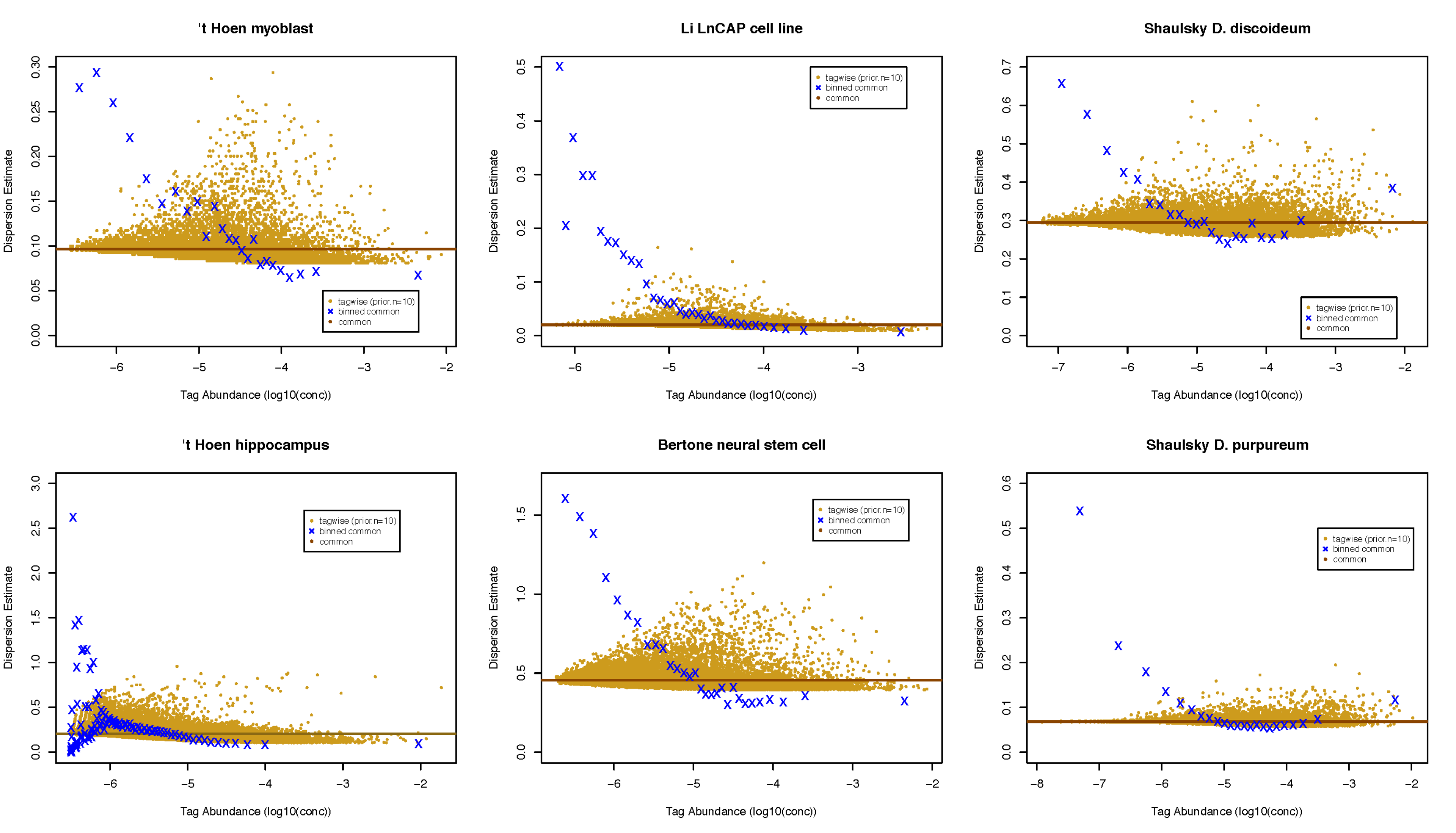
Mean-dispersion relationship

- There is evidence of that the value of the dispersion parameter varies with the expression level of the tag
- Noted by Anders and Huber (2010)
- Generally, dispersion is larger for low abundance tags and decreases as abundance increases

Mean-dispersion rel.: ‘t Hoen



Also seems true for more datasets



Consequences

- Looks like dispersion is much larger for lower abundance tags
- Including this in the model would decrease ability to call low abundance tags DE (but further increase power for high abundance tags; is perhaps more correct)
- DESeq has been designed to deal with this
- edgeR will soon also include an option for allowing dispersion to vary with abundance

Concluding remarks

- Must understand and account for biological variability (overdispersion) in RNA-seq data
- Negative binomial model, sharing information between genes
- Exact and multiple testing for accurate p-values

References

- Robinson and Smyth, Biostatistics, 2008, 9(2):321-32.
- Robinson and Smyth, Bioinformatics, 2007, 23(21):2881-7.
- Robinson, McCarthy and Smyth, Bioinformatics, 2010, 26(1): 139-40.
- Bullard et al. BMC Bioinformatics, 2010, 11:94.
- Robinson and Oshlack, Genome Biology, 2010, 11(3):R25.
- Anders and Huber, 2010, Nature Precedings (http://dx.doi.org/10.1038/npre.2010.4282.1)
- Wang et al. Bioinformatics, 2010, 26(1):136-8.
- Hardcastle, baySeq - (http://www.bioconductor.org/packages/release/bioc/html/baySeq.html)
- Oshlack and Wakefield, Biol Direct. 2009, 4:14.
- Young et al. Genome Biology 2010, 11(2): R14

R Practical

Analysis in R

- R/Bioconductor: open-source statistical software
- Four packages currently available for DE analysis of count data in R
- DEGSeq (Poisson), **edgeR**, baySeq and DESeq (NB)
- For NB, variations in the implementation of information sharing and statistical testing
- We work on **edgeR**, so this is our favourite

Reading in data

- Read the data into R session using a ‘targets’ file
- The function readDGE() creates a ‘DGEList’ object which stores our data in R

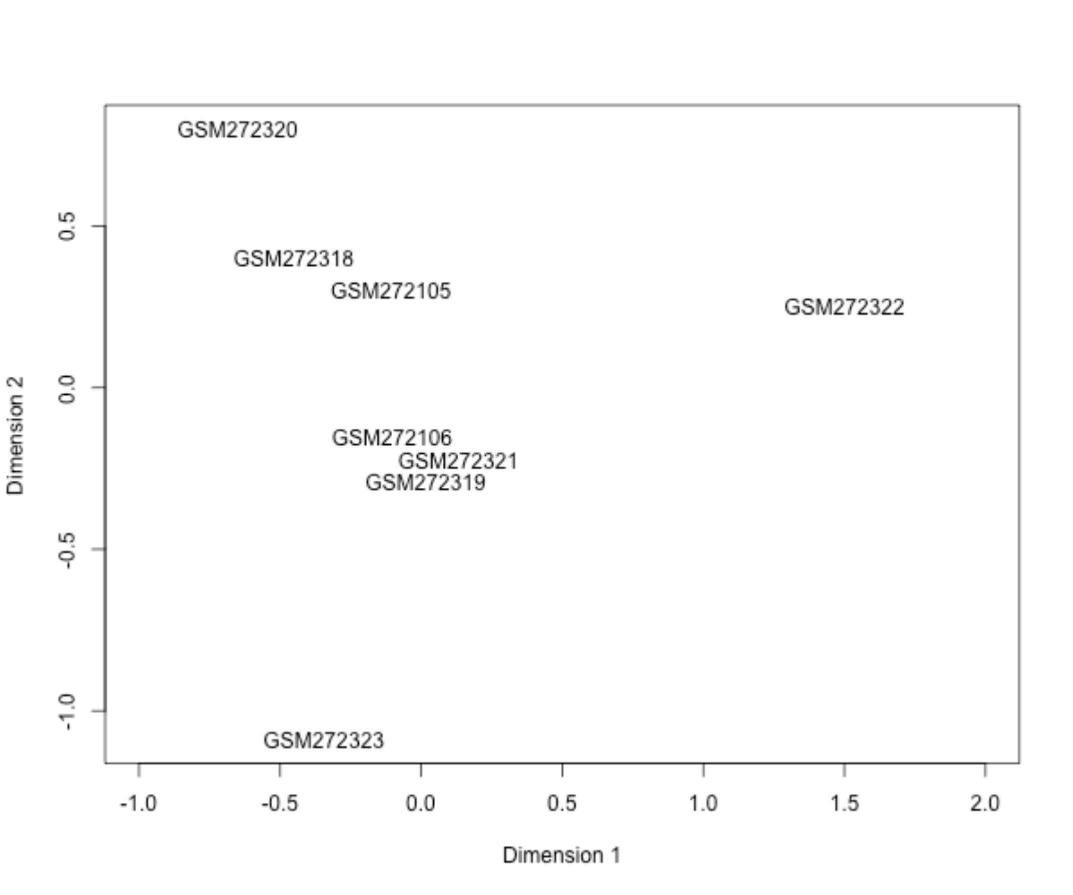
```
> library(edgeR)
> targets <- read.delim
  (file=‘Targets.txt',stringsAsFactors=
  FALSE)
> d <- readDGE
  (targets,skip=5,comment.char='#’)
```

DGEList object

```
> d
An object of class "DGEList"
$samples
                  files group                           description lib.size
GSM272105 GSM272105.txt  DCLK transgenic (Dclk1) mouse hippocampus  2582749
GSM272106 GSM272106.txt    WT          wild-type mouse hippocampus  3342705
GSM272318 GSM272318.txt  DCLK transgenic (Dclk1) mouse hippocampus  3207895
GSM272319 GSM272319.txt    WT          wild-type mouse hippocampus  3273243
GSM272320 GSM272320.txt  DCLK transgenic (Dclk1) mouse hippocampus  2428553
GSM272321 GSM272321.txt    WT          wild-type mouse hippocampus   358649
GSM272322 GSM272322.txt  DCLK transgenic (Dclk1) mouse hippocampus   714498
GSM272323 GSM272323.txt    WT          wild-type mouse hippocampus  2833329
$counts
                  GSM272105 GSM272106 GSM272318 GSM272319 GSM272320 GSM272321
TTTTTCTTCTTTCTTTT         3         1         2         6         3         0
CAGGGACCATCTGTAGA         5        19         2        16         2         0
GTGCGTGCAGCTGAGGG         7         4         6         5         7         1
ATACACACTGTAAAGAG         2         0         6         4         6         0
AATTATAGTGCAATTGA         5         3         3         3         2         0
                  GSM272322 GSM272323
TTTTTCTTCTTTCTTTT         1         2
CAGGGACCATCTGTAGA         2        13
GTGCGTGCAGCTGAGGG         2         3
ATACACACTGTAAAGAG         2         8
AATTATAGTGCAATTGA         0         4
76546 more rows ...
```

Multi-dimensional scaling plot

- Used to assess similarity btw libraries - identify outliers and problematic samples
- Common dispersion used as the ‘distance metric’
- Libraries quite similar here, apart from GSM272322



```
> plotMDS.dge(d)
```

Estimating the common dispersion

- We now compute common dispersion
- Estimate of the coefficient of variation is 0.44, quite large
- Genuine biological variation so reasonable that there is large inter-library variation

```
> d <- estimateCommonDisp(d)
> d$common.dispersion
[1] 0.1964033
> sqrt(d$common.dispersion)
[1] 0.4431741
```

Exact test in edgeR: common disp

```
> de.common <- exactTest(d)
Comparison of groups:  WT - DCLK
> topTags(de.common, n=5)
Comparison of groups:  WT - DCLK
                  logConc  logFC    PValue       FDR
AATTTCTTCCTCTTCCT  -17.25 11.671 2.803e-38 2.146e-33
TCTGTACGCAGTCAGGC  -18.42 -9.633 1.116e-23 4.270e-19
CCGTCTTCTGCTTGTCG  -10.70  5.290 3.524e-22 8.992e-18
AAGACTCAGGACTCATC  -32.22 35.600 1.516e-20 2.901e-16
CCGTCTTCTGCTTGTAA  -14.57  5.176 2.716e-20 4.158e-16
top.com <- rownames(topTags(de.common,n=5)$table)
> d$counts[top.com,order(d$samples$group)]
                  GSM272105 GSM272318 GSM272320 GSM272322 GSM272106 GSM272319 GSM272321 GSM272323
AATTTCTTCCTCTTCCT         1         0         0         0        44         1        76      3487
TCTGTACGCAGTCAGGC       160       101       440        33         0         1         0         0
CCGTCTTCTGCTTGTCG       106       268       601         5      1485       420      5156       242
AAGACTCAGGACTCATC         0         0         0         0         6         2         4       461
CCGTCTTCTGCTTGTAA        12        21        31         1        87        28       352        14

> sum(topTags(de.common,n=Inf)$table$FDR < 0.01)
[1] 399
```

Estimating the tagwise dispersions

- One function call required to estimate moderated tagwise dispersions
- The argument ‘prior.n’ determines amount of moderation or ‘squeezing’ towards common disp
- Larger prior.n → more squeezing

```
> d <- estimateTagwiseDisp(d, prior.n=10)
Using grid search to estimate tagwise
  dispersion.
> summary(d$tagwise.dispersion)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.119   0.185   0.193   0.197   0.207   0.809
```

Exact test in edgeR: tagwise disp

```
> de.tagwise <- exactTest(d, common.disp=FALSE)
Comparison of groups:  WT - DCLK
> topTags(de.tagwise, n=5)
Comparison of groups:  WT - DCLK
                  logConc   logFC    PValue       FDR
TCTGTACGCAGTCAGGC  -18.42  -9.633 3.244e-19 2.483e-14
CATAAGTCACAGAGTCG  -32.76 -34.508 1.995e-14 7.636e-10
AATTTCTTCCTCTTCCT  -17.26  11.668 1.223e-13 3.122e-09
AAAAGAAATCACAGTTG  -32.97 -34.089 6.105e-12 1.168e-07
ATACTGACATTTCGTAT  -16.74   4.213 9.744e-12 1.492e-07
> top.tgw <- rownames(topTags(de.tagwise, n=5)$table)
> d$counts[top.tgw,order(d$samples$group)]
                  GSM272105 GSM272318 GSM272320 GSM272322 GSM272106 GSM272319
TCTGTACGCAGTCAGGC       160       101       440        33         0         1
CATAAGTCACAGAGTCG        67        77        58         7         0         0
AATTTCTTCCTCTTCCT         1         0         0         0        44         1
AAAAGAAATCACAGTTG        31        90        42         3         0         0
ATACTGACATTTCGTAT         5         5         8         1       113       228
                  GSM272321 GSM272323
TCTGTACGCAGTCAGGC         0         0
CATAAGTCACAGAGTCG         0         0
AATTTCTTCCTCTTCCT        76      3487
AAAAGAAATCACAGTTG         0         0
ATACTGACATTTCGTAT         4       104
> > sum(topTags(de.tagwise,n=Inf)$table$FDR < 0.01)
[1] 237
```