Introduction to Next-Generation Sequencing Data and Analysis

Utah State University – Spring 2014

STAT 5570: Statistical Bioinformatics

Notes 6.3

References

- Auer & Doerge (2009), "Statistical Issues in Next-Generation Sequencing", Proceedings of Kansas State University Conference on Applied Statistics in Agriculture
- Anders & Huber (2010), "Differential Expression Analysis for Sequence Count Data", Genome Biology 11:R106
- Gohlmann & Talloen (2009) Gene Expression Studies Using Affymetrix Microarrays [Ch. 9 – "Future Perspectives"]
- Backman, Sun, and Girke (2011) HT Sequence Analysis with R and Bioconductor [accessed March 2014 at http://manuals.bioinformatics.ucr.edu/home/ht-seq]

General DNA sequencing

- Sanger
  - 1970's – today
  - most reliable, but expensive
- Next-generation [high-throughput] (NGS):
  - Genome Sequencer FLC (GS FLX, by 454 Sequencing)
  - Illumina's Solexa Genome Analyzer
  - Applied Biosystems SOLiD platform
  - others …
  - Key difference from microarrays: no probes on arrays, but sequence (and identify) all sequences present

Common features of NGS technologies (1)

- fragment prepared genomic material
  - biological system's RNA molecules → RNA-Seq
  - DNA or RNA interaction regions → ChIP-Seq, HITS-CLIP
  - others …
- sequence these fragments (at least partially)
  - produces HUGE data files (~10 million fragments sequenced)

Common features of NGS technologies (2)

- align sequenced fragments with reference sequence
  - usually, a known target genome (gigo…)
  - alignment tools: ELAND, MAQ, SOAP, Bowtie, others
  - often done with command-line tools
  - still a major computational challenge

- count number of fragments mapping to certain regions
  - usually, genes
  - these read counts linearly approximate target transcript abundance

Example – 3 treated vs. 4 untreated; read counts for 14,470 genes

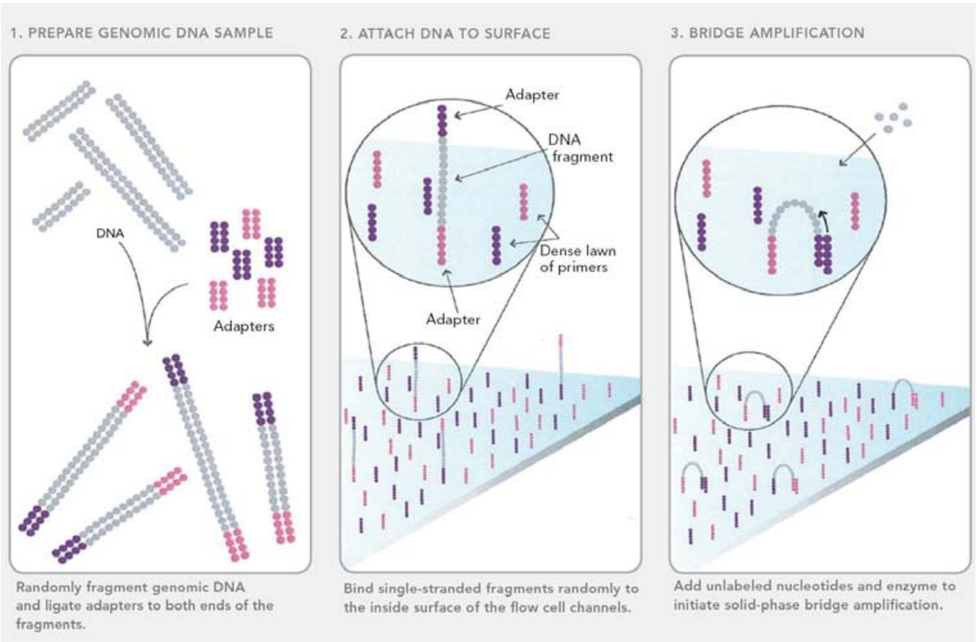
- Published 2010 (Brooks et al., Genome Research)
- Drosophila melanogaster
- 3 samples “treated” by knock-down of “pasilla” gene (thought to be involved in regulation of splicing)

```
              T1   T2   T3   U1   U2   U3   U4
FBgn0000003    0    0    1    0    0    0    0
FBgn0000008   78   46   43   47   89   53   27
FBgn0000014    2    0    0    0    0    1    0
FBgn0000015    1    0    1    0    1    1    2
FBgn0000017 3187 1672 1859 2445 4615 2063 1711
FBgn0000018  369  150  176  288  383  135  174
```

```
library(pasilla); data(pasillaGenes)
eset <- counts(pasillaGenes)
colnames(eset) <- c('T1','T2','T3','U1','U2','U3','U4')
head(eset)
```

Here, RNA-Seq:

- similar biological objective to microarrays
  - recall central dogma: DNA → mRNA → protein → action
  - quantify [mRNA] transcript abundance
- Isolate RNA from cells, fragment at random positions, and copy into cDNA
- Attach adapters to ends of cDNA fragments, and bind to flow cell (Illumina has glass slide with 8 such lanes – so can process 8 samples on one slide)
- Amplify cDNA fragments in certain size range (e.g., 200-300 bases) – using PCR → clusters of same fragment
- Sequence – base-by-base for all clusters in parallel



1. PREPARE GENOMIC DNA SAMPLE

Randomly fragment genomic DNA and ligate adapters to both ends of the fragments.

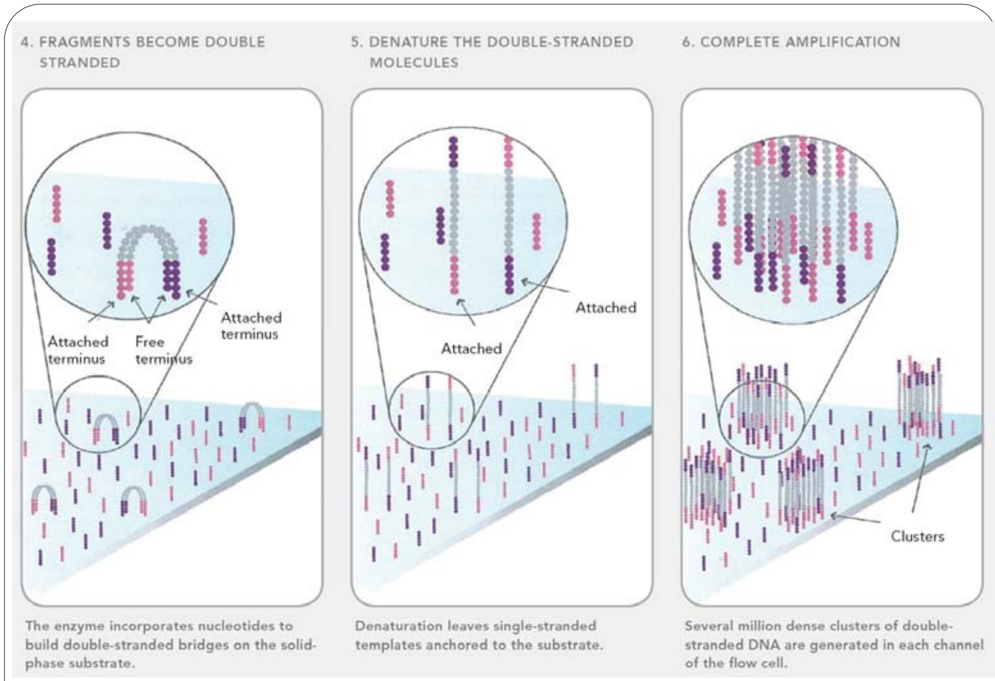
2. ATTACH DNA TO SURFACE

Bind single-stranded fragments randomly to the inside surface of the flow cell channels.

3. BRIDGE AMPLIFICATION

Add unlabeled nucleotides and enzyme to initiate solid-phase bridge amplification.

9 (originally illumina.com download; see http://res.illumina.com/documents/products/techspotlights/techspotlight_sequencing.pdf)



4. FRAGMENTS BECOME DOUBLE STRANDED

The enzyme incorporates nucleotides to build double-stranded bridges on the solid-phase substrate.

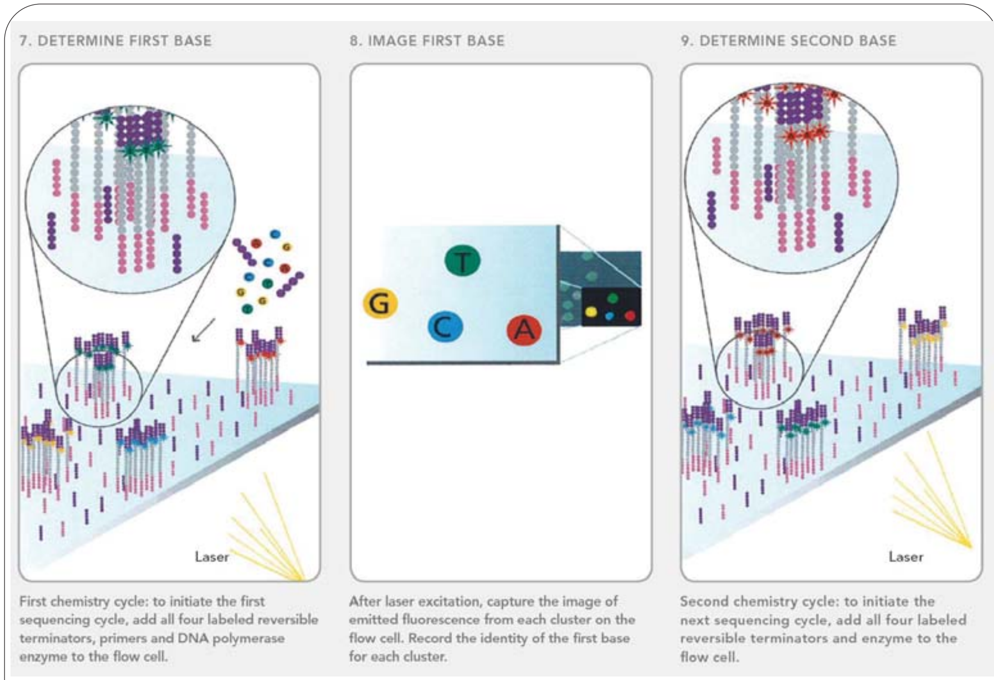
5. DENATURE THE DOUBLE-STRANDED MOLECULES

Denaturation leaves single-stranded templates anchored to the substrate.

6. COMPLETE AMPLIFICATION

Several million dense clusters of double-stranded DNA are generated in each channel of the flow cell.

10 (originally illumina.com download)



7. DETERMINE FIRST BASE

First chemistry cycle: to initiate the first sequencing cycle, add all four labeled reversible terminators, primers and DNA polymerase enzyme to the flow cell.

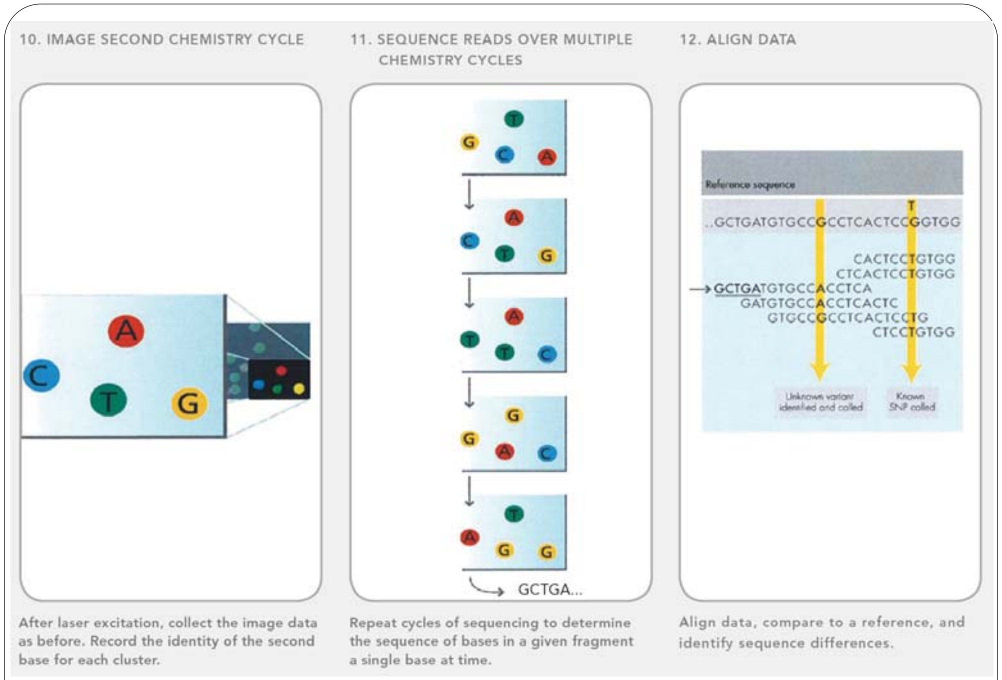
8. IMAGE FIRST BASE

After laser excitation, capture the image of emitted fluorescence from each cluster on the flow cell. Record the identity of the first base for each cluster.

9. DETERMINE SECOND BASE

Second chemistry cycle: to initiate the next sequencing cycle, add all four labeled reversible terminators and enzyme to the flow cell.

11 (orginally illumina.com download)



10. IMAGE SECOND CHEMISTRY CYCLE

After laser excitation, collect the image data as before. Record the identity of the second base for each cluster.

11. SEQUENCE READS OVER MULTIPLE CHEMISTRY CYCLES

Repeat cycles of sequencing to determine the sequence of bases in a given fragment a single base at time.

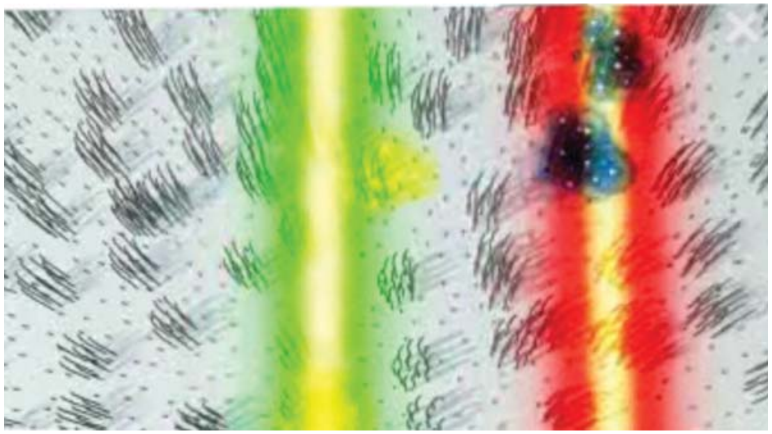
12. ALIGN DATA

Align data, compare to a reference, and identify sequence differences.

12 (orginally illumina.com download)

Cartoons

- Imaging the sequence ("cutting edge imaging technology")
  (1:40-2:20 of http://www.youtube.com/watch?v=d2AxXv_6UTQ)

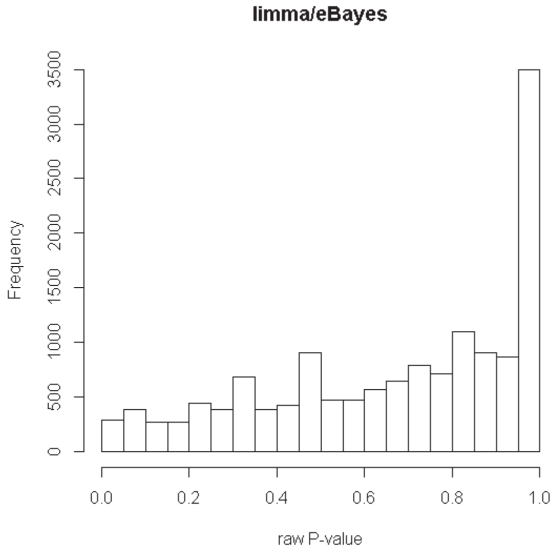


- See also "Illumina sequencing"
  http://www.youtube.com/watch?v=l99aKKHcxC4

Then align and map …

- For sequence at each cluster, compare to [align with] reference genome; file format:
  - millions of clusters per lane
  - approx. 1 GB file size per lane
- For regions of interest in reference genome (genes, here), count number of clusters mapping there
  - requires well-studied and well-documented genome

What would limma/eBayes results look like?



```
# (Defined eset object on slide 7; now define conditions)
conds <- c("T","T","T","U", "U", "U", "U")
# 3 treated, 4 untreated

# try analyzing as in limma/eBayes
#  (slides 8 and 13 of Notes 3.4)
library(limma)
trt <- as.factor(conds)
design <- model.matrix(~0+trt)
colnames(design) <- c('T','U')
fit <- lmFit(eset, design)
contrast.trt <- makeContrasts(T-U, levels=design)
fit.trt <- contrasts.fit(fit, contrast.trt)
final.fit.trt <- eBayes(fit.trt)
   # Warning message:
   # Zero sample variances detected, have been offset
top.trt <- topTableF(final.fit.trt, n=nrow(eset))

sum(top.trt$adj.P.Val<.05) # 0 sig. genes

hist(top.trt$P.Value, main='limma/eBayes', xlab='raw P-value')
```

(logged counts yield similar result)

But wait …

- limma/eBayes implicitly assumes continuous data for each gene k:
  - Recall matrix representation (slide 5 of Notes 3.4)

$$\mathbf{Y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}\,, \qquad \varepsilon_{ij} \text{ iid } N(0,\sigma^2)$$

  - Recall contrast and its moderated test statistic (slides 11 and 12 of Notes 3.4)

$$\Psi = \sum_i w_i \beta_i \qquad \tilde{F} = \frac{1}{\tilde{\sigma}_k^2} \cdot \left(\frac{\hat{\Psi}}{w'Vw} \right) \overset{\sim}{\sim} F_{1,(d_0+d_k)}$$

- But these data are counts – discrete

Now consider Poisson regression (data as counts)

- As with previous models, on a per-gene basis:
  - Let N_i = # of total fragments counted in sample i
  - Let $p_i = \mathrm{P}\{$ fragment matches to gene in sample i $\}$
- Observed # of total reads for gene in sample i :
  - $R_i \sim \mathrm{Poisson}(N_i p_i)$
  - $\mathrm{E}[R_i] = \mathrm{Var}[R_i] = N_i p_i$
- Let T_i = indicator of trt. status (0/1) for sample i
  - Assume $log(p_i) = \beta_0 + \beta_1 T_i$
  - Test for DE using $H_0\!: \beta_1 = 0$

Poisson Regression

- $E[R_i] = N_i p_i = N_i\, exp(\beta_0 + \beta_1 T_i)$
- $log(E[R_i]) = \underbrace{log\ N_i}_{\downarrow} \ + \ \underbrace{\beta_0 + \beta_1 T_i}$

estimate β’s using iterative MLE procedure

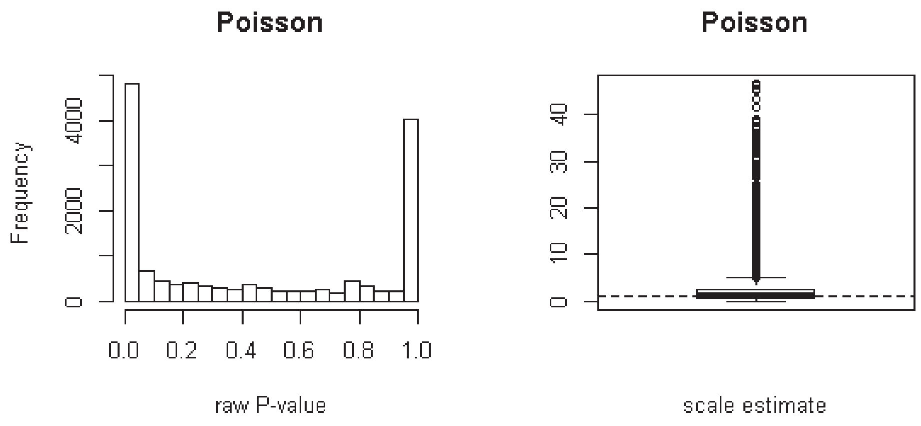
not interesting, but important
– call this the “offset”;
often considered the “exposure” for sample i

- Do this for one gene in R (here, gene 2):

```
trt <- c(1,1,1,0,0,0,0)
R <- eset[2,]
lExposure <- log(colSums(eset))
a1 <- glm(R ~ trt, family=poisson, offset=lExposure)
summary(a1)
```

```
Call:
glm(formula = R ~ trt, family = poisson, offset = lExposure)

Deviance Residuals:
     T1        T2        T3        U1        U2        U3        U4
 0.3690    0.4516   -0.9047   -0.7217    0.5862    2.3048   -2.5286

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.85250    0.06804 -174.19   <2e-16 ***
trt           0.05875    0.10304    0.57    0.569
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 14.053  on 6  degrees of freedom
Residual deviance: 13.729  on 5  degrees of freedom
AIC: 58.17

Number of Fisher Scoring iterations: 4
```

Do this for all genes …



jackpot?

Possible (frequent) problem – overdispersion

- Recall [implicit] assumption for Poisson dist'n:
  - $E[R_i] = Var[R_i] = N_i p_i$
- It can sometimes happen that $Var[R_i] > E[R_i]$
  - common check: add a scale (or dispersion) parameter σ
  - $Var[R_i] = \sigma\ E[R_i]$
  - Estimate σ^2 as χ^2/df
  - Deviance χ^2 a goodness of fit statistic:

$$\chi^2_D = 2\cdot\sum_i \left(R_i \cdot \log\frac{R_i}{\hat{R}_i} \right)$$

```
# Poisson regression for all genes, checking for overdispersion
Poisson.p <- scale <- rep(NA,nrow(eset))
lExposure <- log(colSums(eset))
trt <- c(1,1,1,0,0,0,0)

## this next part takes about 1.5 minutes
print(date()); for(i in 1:nrow(eset))
  {  count <- eset[i,]
     a1 <- glm(count ~ trt, family=poisson, offset=lExposure)
     Poisson.p[i] <- summary(a1)$coeff[2,4]
     scale[i] <- sqrt(a1$deviance/a1$df.resid)
   }; print(date())

par(mfrow=c(2,2))
hist(Poisson.p, main='Poisson', xlab='raw P-value')
boxplot(scale, main='Poisson', xlab='scale estimate');
abline(h=1,lty=2)

mean(scale > 1)
     #   0.640152
```

Can use alternative distribution:

- edgeR package does this:
  - For each gene: $R_i \sim NegativeBinomial$
    - (number of indep. Bernoulli trials to achieve a fixed number of successes)
    - Let $\mu_i = E[R_i]$, and $v_i = Var[R_i]$
    - But low sample sizes prevent reliable estimation of μ_i and v_i
  - Assume $v_i = \mu_i + \alpha\,\mu_i^2$
    - estimate α by pooling information across genes
    - then only one parameter must be estimated for each gene
- But – DESeq package improves on this (see next set of slides – Notes 6.4)

Major Advantages of NGS

- No artifacts of cross-hybridization (noise, background, etc.)
- Better estimation of low-abundance transcripts
- "Dynamic Range"
  - no technical limitation as with intensity observations
  - Aside: this would be violated by quantile normalization [in tails of distributions] – so instead consider RPKM normalization (reads per kilobase of exon model per million)
- Cost expected to improve in coming years

Remaining issues with NGS

- Practical problem with sample preparation – possible low reads for A/T-rich regions
- High error rates – due to sample preparation / amplification and dependence of read quality on base position
- Image quality (bubbles, etc.)
- File size [huge] – expected to soon be cheaper to re-run experiment than to store data
  - but what about sample availability?
  - value in older files (as with .CEL for microarrays)
- Sequence mapping – methods and implementations

Interesting statistical questions

- Fully accounting for all sources of variation
  - slide, lane, etc.
- Error propogation
  - counts estimate transcript abundance
  - alignment
- Accounting for gene length
  - offset?
- Effective statistical computing
  - sifting through massive alignment files

A Rough Timeline of Arrivals

- (1995+) Microarrays
  - require probes fixed in advance – only set up to detect those
- (2005+) Next-Generation Sequencing (NGS)
  - typically involves amplification of genomic material (PCR)
- (2010+) Third-Generation Sequencing
  - "next-next-generation" – Pac Bio, Ion Torrent
  - no amplification needed – can sequence single molecule
  - longer reads possible; still (as of 2013) showing high errors
- (2012+) Nanopore-Based Sequencing
  - Oxford Nanopore, Genia, others
  - bases identified as whole molecule slips through nanoscale hole (like threading a needle); coupled with disposable cartridges; still (as of 2013) under development
- (?+) more …

Differ in how sequencing done; subsequent post-alignment statistical analysis basically same

Conclusions

- NGS a powerful tool for transcriptomics
- Computational challenges
  - storage (sequencing and alignment files)
- Most meaningful to use count-data models
  - Up next: a negative binomial model with DESeq
- Issues (technological and statistical) remain