Introduction to Next-Generation Sequencing Data and Analysis

Utah State University – Spring 2012

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 2012 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

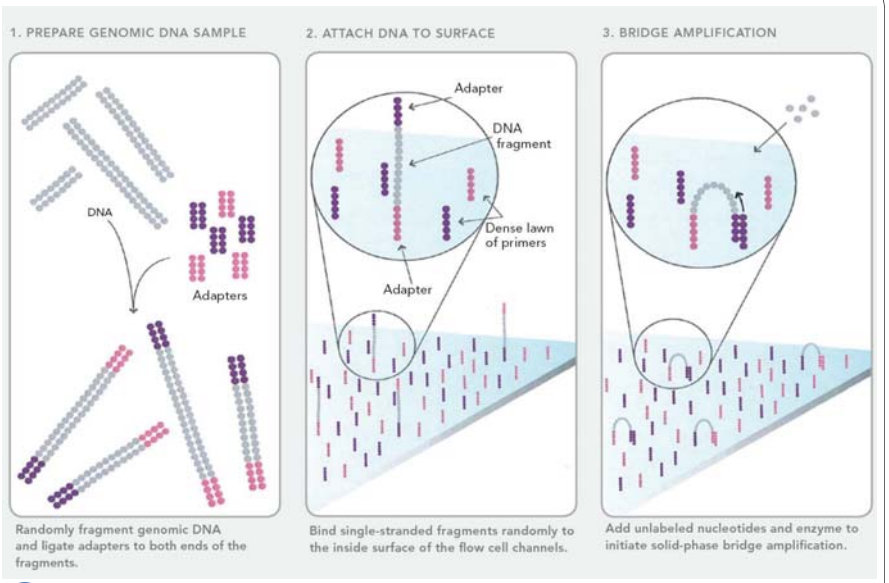
- Published 2010 (Brooks et al., Genome Research)
- Drosophila melanogaster
- 3 samples "treated" by knock-down of "pasilla" gene (thought to be involved in regulation of splicing)

	T1	T2	T3	U1	U2	U3	U4
FBgn0000003	0	1	1	0	0	0	0
FBgn0000008	118	139	77	89	142	84	76
FBgn0000014	0	10	0	1	1	0	0
FBgn0000015	0	0	0	0	0	1	2
FBgn0000017	4852	4853	3710	4640	7754	4026	3425
FBgn0000018	572	497	322	552	663	272	321

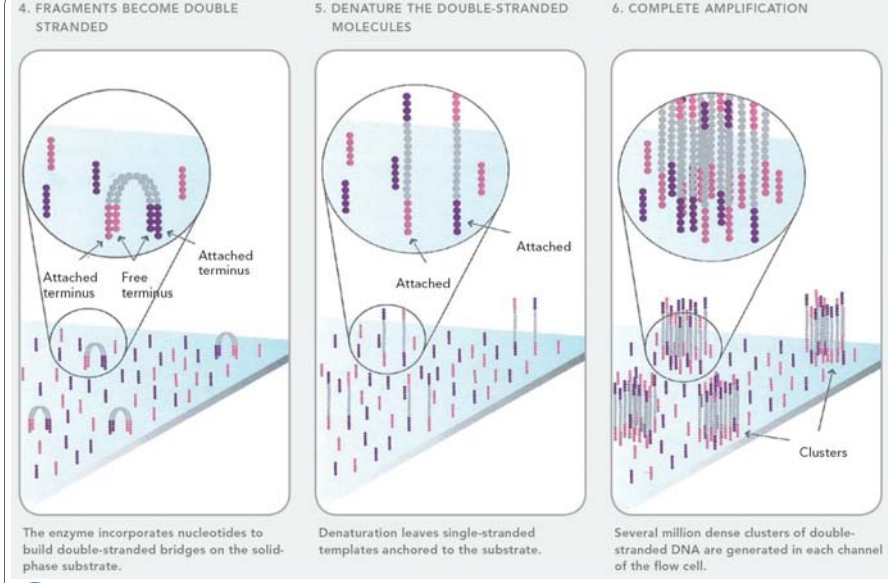

```
library(pasilla); data(pasillaGenes)
eset <- counts(pasillaGenes)
colnames(eset) <- c('T1','T2','T3','U1','U2','U3','U4')
head(eset)
```


Here, RNA-Seq:

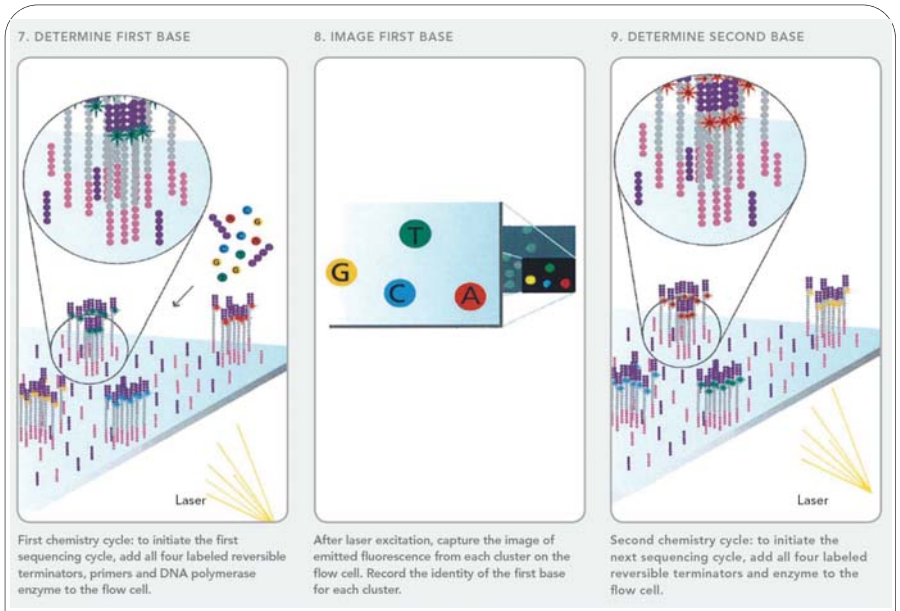
- similar biological objective to microarrays
  - recall central dogma: DNA → mRNA → protein → action
  - quantify [mRNA] transcript abundance
- Isolate RNA from cells, fragment at random positions, and copy into cDNA
- Attach adapters to ends of cDNA fragments, and bind to flow cell (Illumina has glass 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



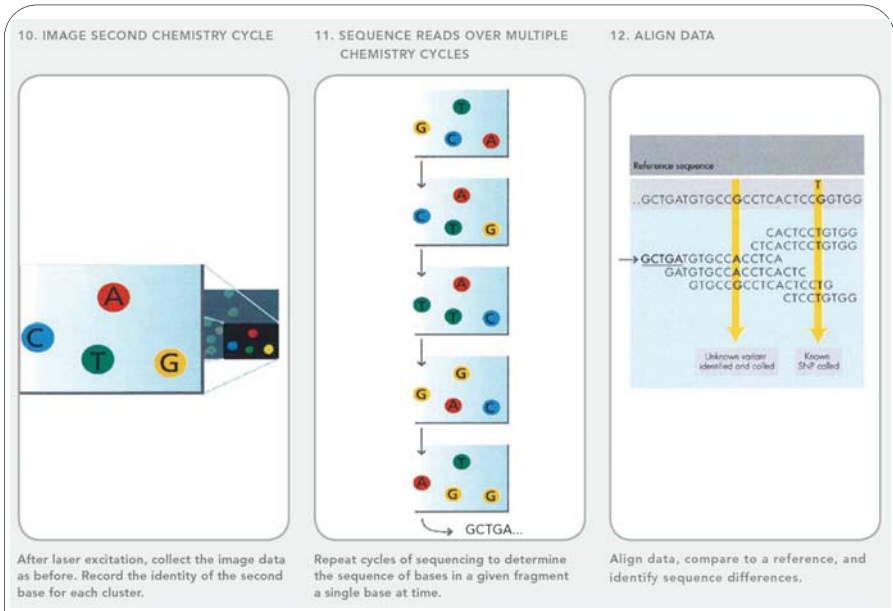
9 (originally illumina.com download)



10 (originally illumina.com download)



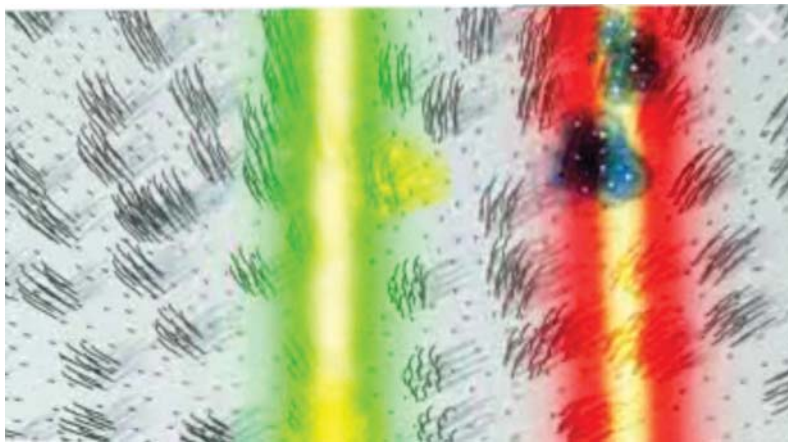
11 (orginally illumina.com download)



12 (orginally illumina.com download)

Cartoons

- Imaging the sequence
  http://www.illumina.com/media/systems/hiseq/preloader.ilmn?iframe

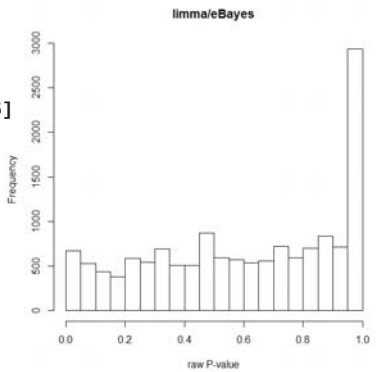


Then align and map …

- 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

Try limma/eBayes

```
# (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
library(limma)
design <- cbind(Intercept=1,trt=c(1,1,1,0,0,0,0))
fit <- lmFit(eset,design)
e.fit <- eBayes(fit)
#   Warning message:
#   Zero sample variances detected,
#      have been offset
top.all <- topTable(e.fit,n=nrow(eset),
   coef=2,adjust="BH")
gn.eB <- top.all$ID[top.all$adj.P.Val<.05]
length(gn.eB) # 5 genes
hist(top.all$P.Value,main='limma/eBayes',
   xlab='raw P-value')
```



(logged counts yield similar result)

But wait …

- limma/eBayes implicitly assumes continuous data

$$\underbrace{Y_{ijk}}_{\text{expression level (log scale)}} = \beta_{k,0} + \underbrace{\beta_{k,1}}_{\text{treatment effect (DE)}} \underbrace{T_{jk}}_{\text{treatment level (could be more than just 2 levels)}} + \varepsilon_{ijk}, \quad Var[\varepsilon_{ijk}] = \sigma_k^2$$

$$\hat{\beta}_{k,1} \mid \beta_{k,1}, \sigma_k^2 \sim N\left(\beta_{k,1}, V_k \hat{\sigma}_k^2\right) \quad \ldots$$

$$\tilde{t}_k = \frac{\hat{\beta}_{k,1}}{\tilde{\sigma}_k \sqrt{V_k}} \sim t_{d_k + d_0}$$

- 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 = P{ fragment matches to gene in sample i }
- Observed # of total reads for gene in sample i :
  - $R_i \sim \text{Poisson}(N_i p_i)$
  - $E[R_i] = \text{Var}[R_i] = N_i p_i$
- 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

- $E[R_i] = N_i p_i = N_i \, exp(\beta_0 + \beta_1 T_i)$
- $log(E[R_i]) = log\, N_i + \beta_0 + \beta_1 T_i$

$log\, N_i$: not interesting, but important – call this the "offset"; often considered the "exposure" for sample i

$\beta_0 + \beta_1 T_i$: estimate β's using iterative MLE procedure

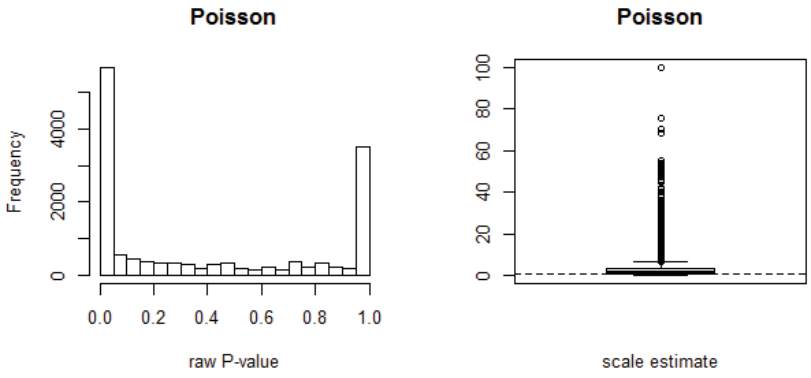
- Do this for one gene in R (here, gene 2):

```
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
-0.08557  1.83437 -2.06690 -1.24502  0.09328  1.64977 -0.34729

Coefficients:
            Estimate Std. Error  z value Pr(>|z|)
(Intercept) -11.90077    0.05057 -235.322   <2e-16 ***
trt           0.11145    0.07451    1.496    0.135
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 14.275  on 6  degrees of freedom
Residual deviance: 12.045  on 5  degrees of freedom
AIC: 61.178

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)
     # .688528
```


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
- (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
- (?+) more …

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