Using limma for microarray and RNA-Seq analysis

Humberto Ortiz-Zuazaga

March 7, 2013

Bioconductor

- Bioconductor `http://bioconductor.org/`
- Software suite for analysis of biological data
- emphasis on microarray and other high-throughput datasets

Loading packages

Load the `affy` and `limma` packages.

```
> library(limma)
> library(affy)
```

Targets file

A simple text file with tab separated columns can describe the microarray samples.

```
> targets <- readTargets("targets.txt")
> targets
```

	FileName	Target
1	OC-1_(HuGene-1_0-st-v1).CEL	pos
2	OC-5_(HuGene-1_0-st-v1).CEL	pos
3	OC-6_(HuGene-1_0-st-v1).CEL	pos
4	OC-7_(HuGene-1_0-st-v1).CEL	pos
5	OC-8_(HuGene-1_0-st-v1).CEL	pos
6	OC-10_(HuGene-1_0-st-v1).CEL	pos
7	OC-11_(HuGene-1_0-st-v1).CEL	neg
8	OC-12_(HuGene-1_0-st-v1).CEL	neg
9	OC-13_(HuGene-1_0-st-v1).CEL	neg
10	OC-14_(HuGene-1_0-st-v1).CEL	neg
11	OC-15_(HuGene-1_0-st-v1).CEL	neg

Reading the data

```
> ab <- ReadAffy(filenames=targets$FileName)
```

Sample data

For this presentation, we will use sample data provided with bioconductor.

```
> require(affydata)
```

```
     Package
[1,] "affydata"
     LibPath
[1,] "/Library/Frameworks/R.framework/Versions/2.15/Resourc
     Item       Title
[1,] "Dilution" "AffyBatch instance Dilution"
```

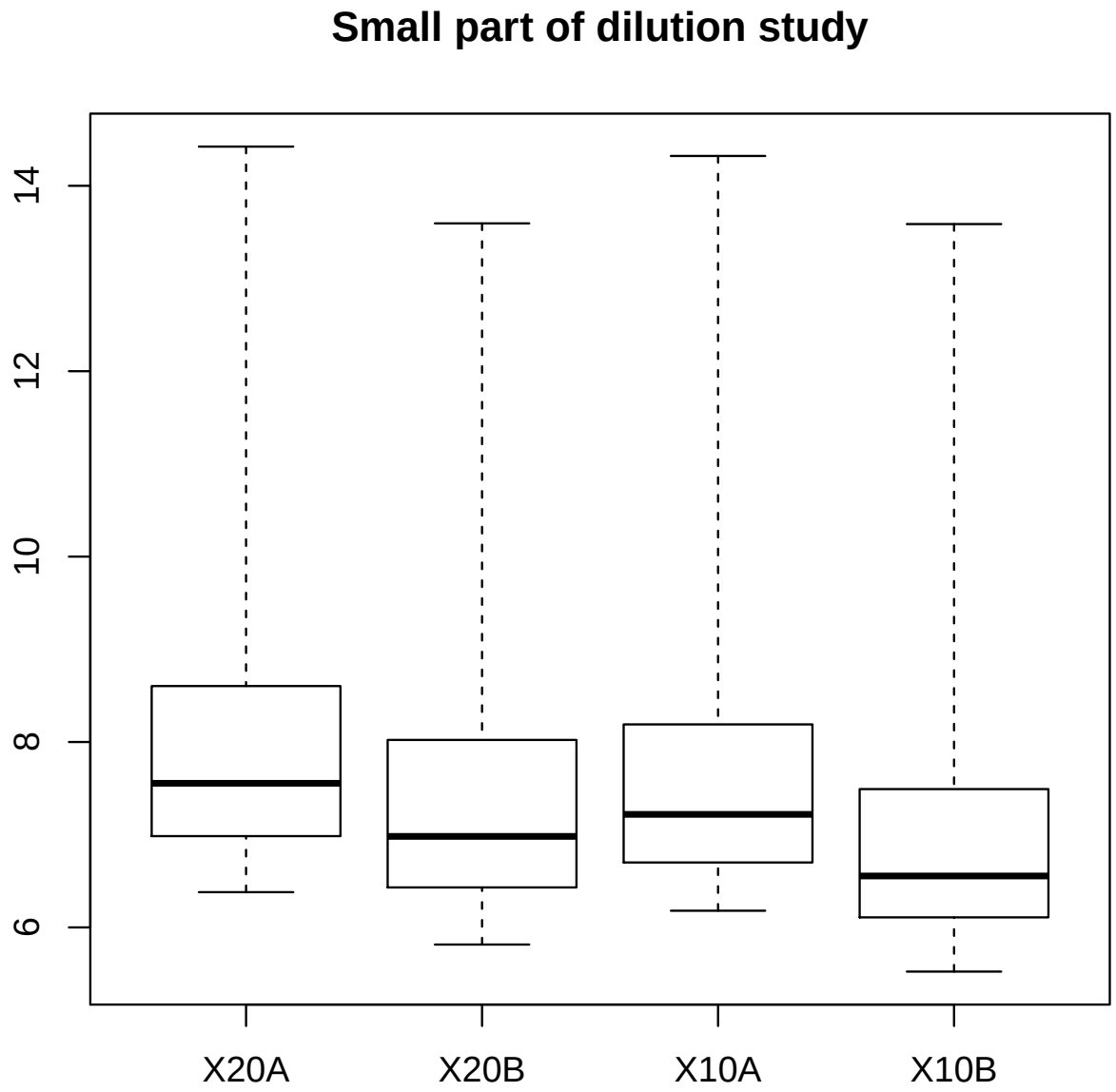
```
> data(Dilution)
> ab <- Dilution
```

Normalization and summarization

```
> probeNames(ab)[1:10]

 [1] "100_g_at" "100_g_at" "100_g_at" "100_g_at" "100_g_at"
 [7] "100_g_at" "100_g_at" "100_g_at" "100_g_at"

> eset <- rma(ab)

Background correcting
Normalizing
Calculating Expression

> featureNames(eset)[1:10]

 [1] "100_g_at"  "1000_at"   "1001_at"   "1002_f_at" "1003_
 [7] "1005_at"   "1006_at"   "1007_s_at" "1008_f_at"
```

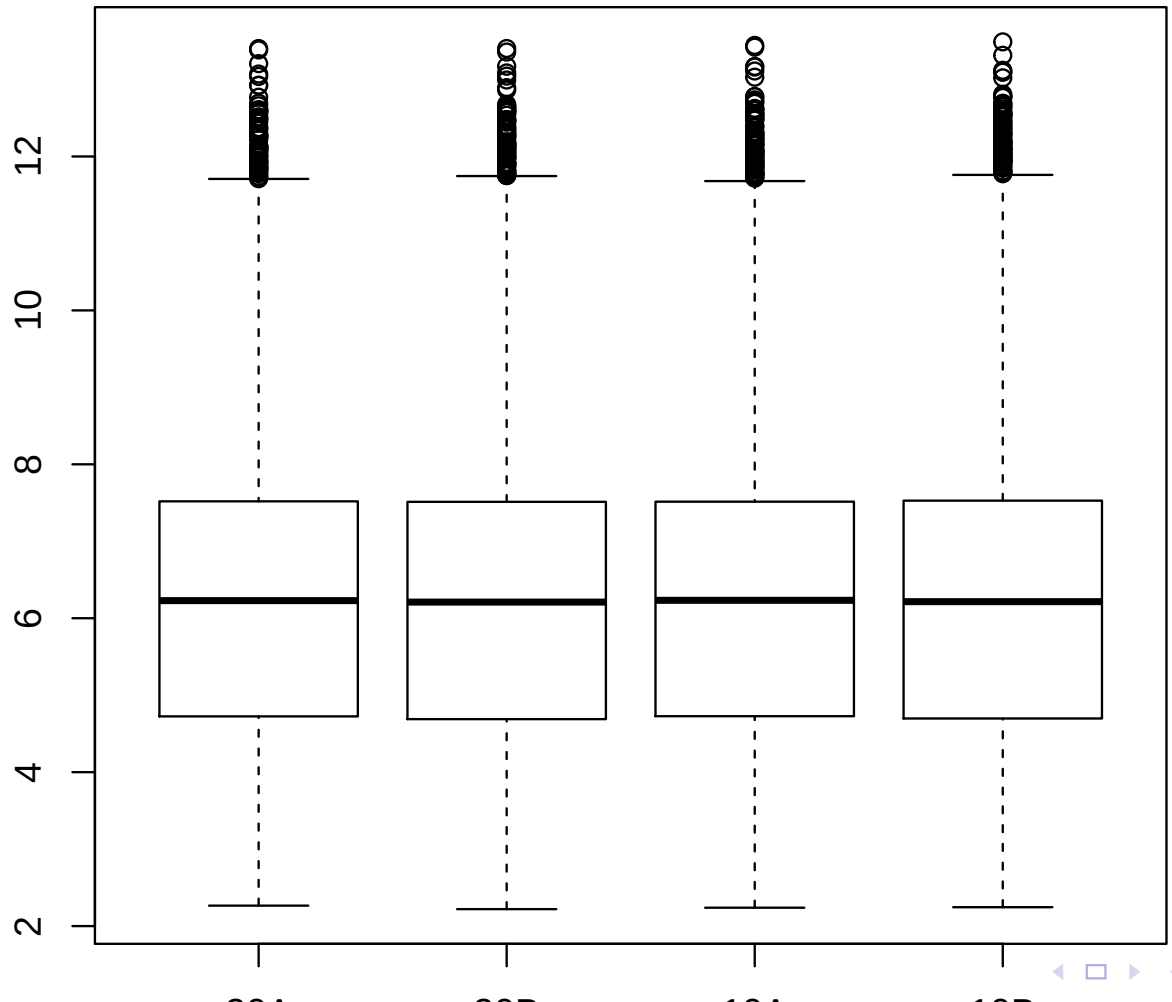
Boxplot before normalization

```
> boxplot(ab)
```



Boxplot after normalization

```
> boxplot(exprs(eset))
> plotMA(eset[,c(1,2)])
```



Describing the design

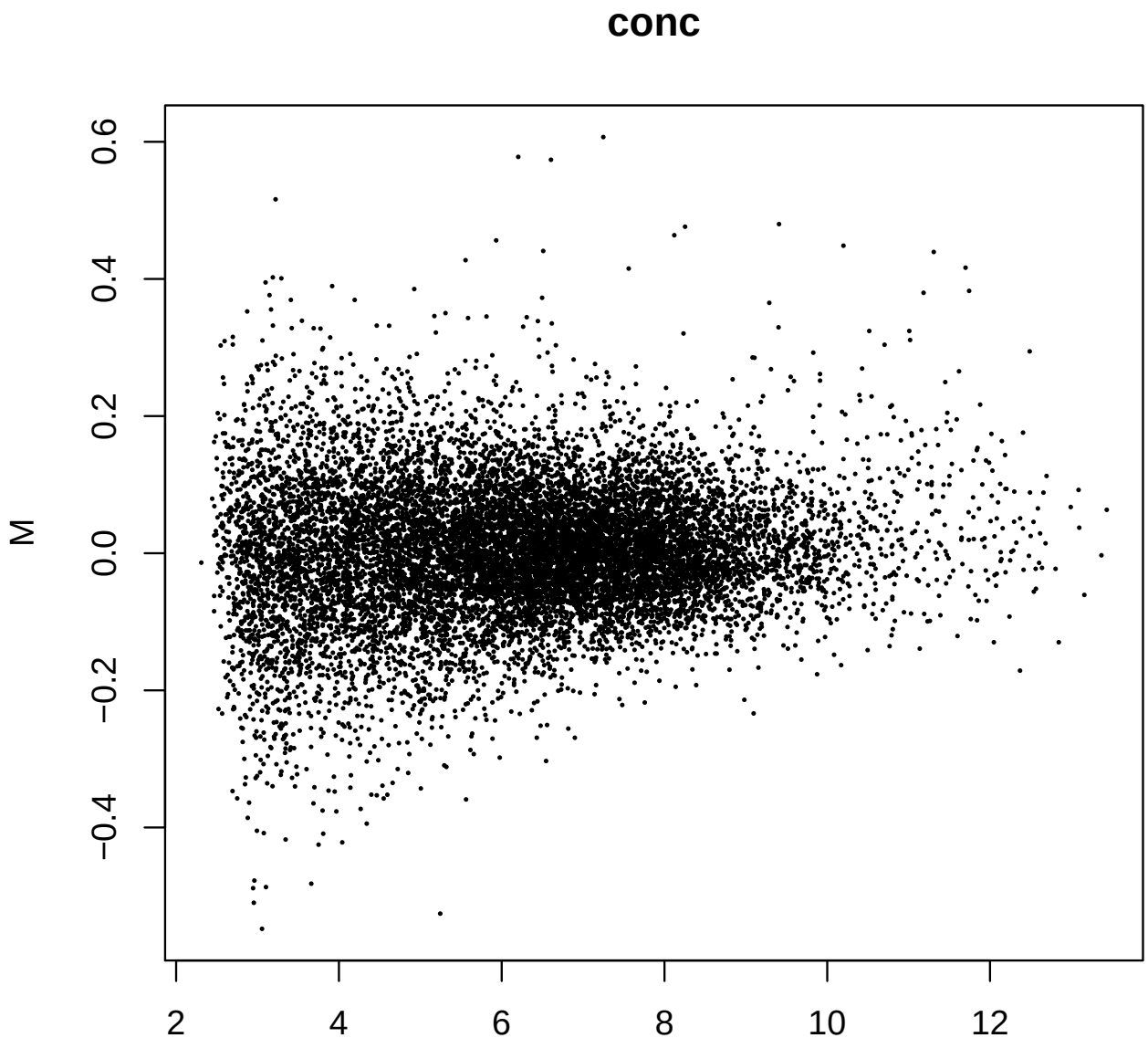
```
> f <- factor(c("C20","C20","C10","C10"))
> design <- model.matrix(~0+f)
> colnames(design) <- c("C20", "C10")
> design
  C20 C10
1   0   1
2   0   1
3   1   0
4   1   0
attr(,"assign")
[1] 1 1
attr(,"contrasts")
attr(,"contrasts")$f
[1] "contr.treatment"
```

Fitting a model

```
> cont.matrix <- makeContrasts(conc=C20-C10, levels=design)
> fit <- lmFit(eset, design)
> fit2 <- contrasts.fit(fit, cont.matrix)
> fit.b <- eBayes(fit2)
```

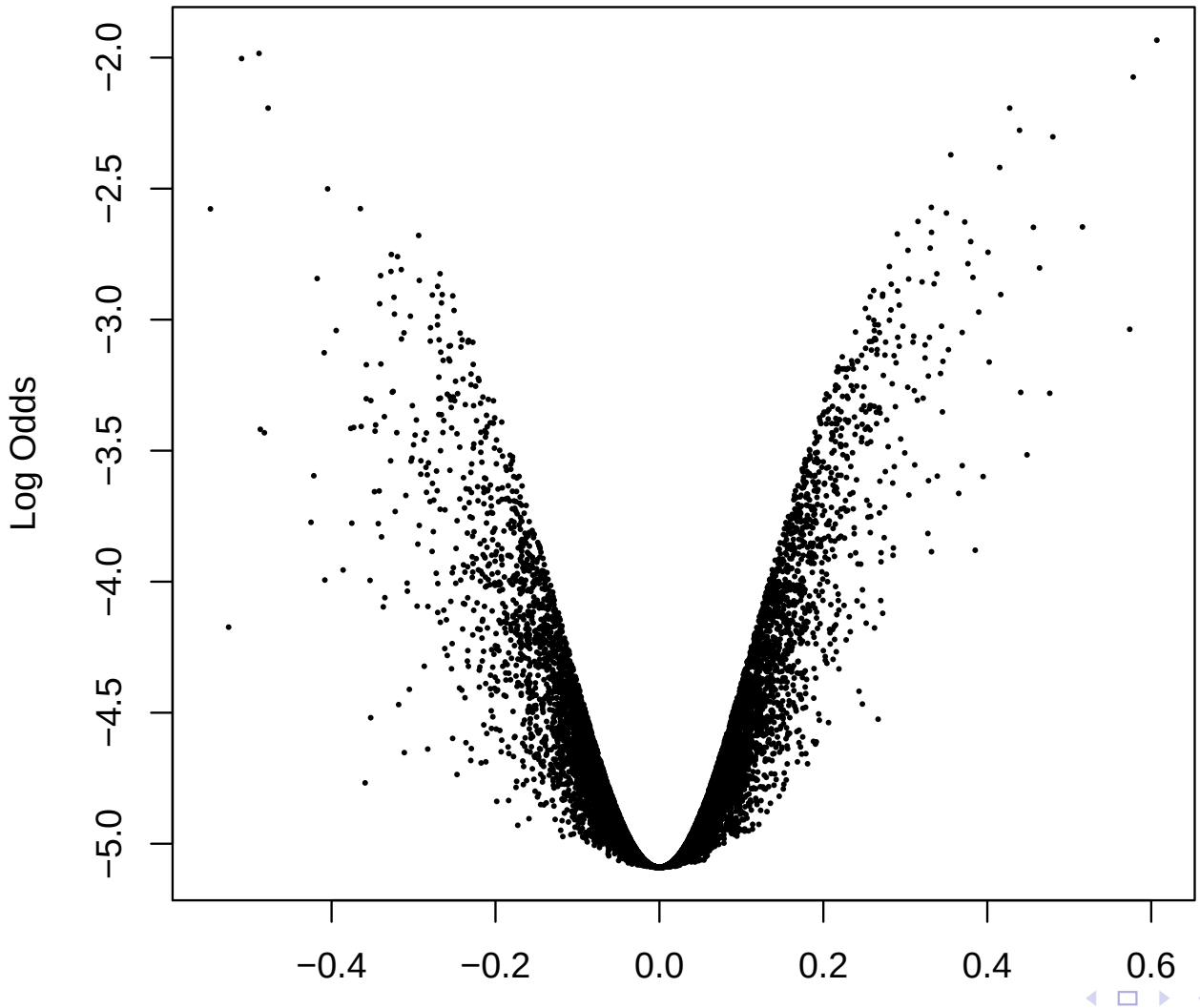
MA plot of model fit

```
> plotMA(fit.b)
```



Volcano plot of model fit

```
> volcanoplot(fit.b)
```



References


Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. (2004). affy—analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* 20, 3 (Feb. 2004), 307-315.


R. Gentleman, V. J. Carey, D. M. Bates, B.Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, and others Bioconductor: Open software development for computational biology and bioinformatics (2004). *Genome Biology*, Vol. 5, R80


Smyth, G. K. (2005). Limma: linear models for microarray data. In: 'Bioinformatics and Computational Biology Solutions using R and Bioconductor'. R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds), Springer, New York, pages 397–420.

RNA-Seq packages

```
> library(Biobase)
> library(biomaRt)
> library(edgeR)
```

Load ReCount data

Can load prepared datasets directly from the web:

```
> gilad <- load(
+     url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/gilad_eset.RData"))
> gilad

[1] "gilad.eset"
```

gilad.eset

```
> gilad.eset

ExpressionSet (storageMode: lockedEnvironment)
assayData: 52580 features, 6 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: SRX014818and9 SRX014820and1 ... SRX014828and
  varLabels: sample.id num.tech.reps gender
  varMetadata: labelDescription
featureData
  featureNames: ENSG00000000003 ENSG00000000005 ... LRG_99
    total)
  fvarLabels: gene
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
Annotation:
```

What's in the eset?

```
> phenoData(gilad.eset)$gender
```

```
[1] F F F M M M
Levels: F M
```

```
> exprs(gilad.eset)[1:5,]
```

	SRX014818and9	SRX014820and1	SRX014822and3	S
ENSG00000000003	60	60	16	
ENSG00000000005	0	0	0	
ENSG00000000419	25	9	15	
ENSG00000000457	32	19	21	
ENSG00000000460	1	3	0	
	SRX014826and7	SRX014828and9		
ENSG00000000003	56	37		
ENSG00000000005	0	0		
ENSG00000000419	26	11		
ENSG00000000457	28	28		
ENSG00000000460	1	1		

Removing genes that are not expressed

```
> isexpr <- rowSums(cpm(exprs(gilad.eset))>1) >= 3
> sum(isexpr)
[1] 8069
> gilad.isexpr <- gilad.eset[isexpr,]
```

Normalize the counts

```
> nf <- calcNormFactors(gilad.isexpr)
> groups <- phenoData(gilad.isexpr)$gender
> design <- model.matrix(~ groups)
> y <- voom(exprs(gilad.isexpr),design,plot=TRUE,
+           lib.size=colSums(exprs(gilad.isexpr))*nf,
+           normalize.method="quantile")
```

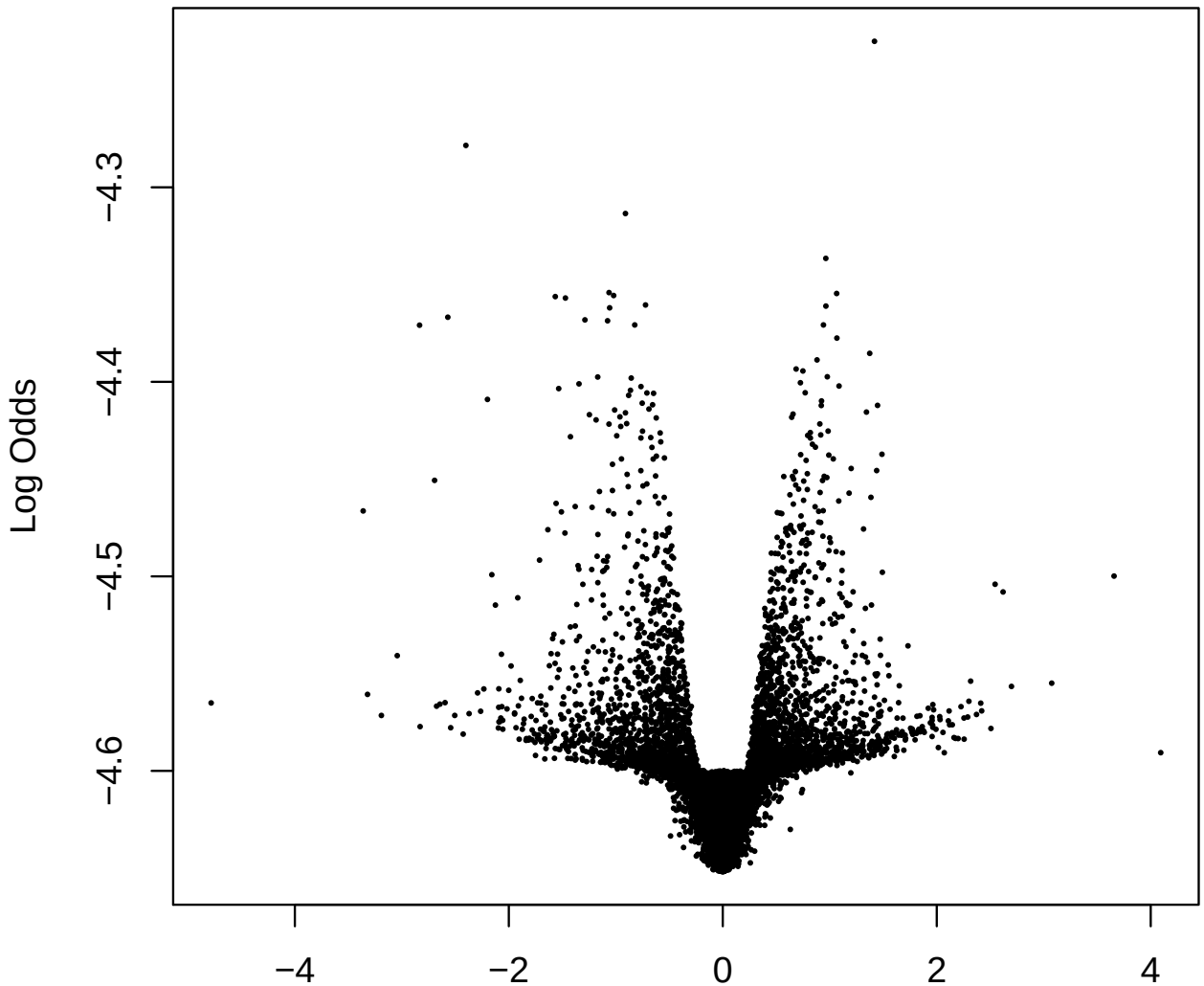
Build a linear model

```
> fit <- lmFit(y,design)
> fit <- eBayes(fit)
> topTable(fit, coef=2)
```

	ID	logFC	AveExpr	t	P.Value	B
286	ENSG00000049239	1.4179955	8.353328	4.932432	0.0006353586	-4.224938
1774	ENSG00000110244	-2.4011726	6.693654	-4.493731	0.0012221106	-4.278461
5957	ENSG00000174718	-0.9099697	7.477112	-3.614733	0.0049090586	-4.313473
191	ENSG00000023330	0.9631780	7.737762	3.342743	0.0076923282	-4.336530
6993	ENSG00000187837	-1.0631848	6.102709	-3.413789	0.0068359089	-4.354120
7743	ENSG00000214456	1.0644053	6.614080	3.275959	0.0085985800	
4913	ENSG00000164626	-1.0201108	6.587511	-3.254500	0.0089125858	
5654	ENSG00000171051	-1.5657336	5.564722	-3.758463	0.0038844280	
3045	ENSG00000133392	-1.4710554	7.721775	-3.134114	0.0109064995	
4827	ENSG00000163513	-0.7220998	7.084751	-3.117506	0.0112154685	

Volcanoplot model fit

```
> volcanoplot(fit,coef=2)
```



References


Frazee AC, Langmead B, Leek JT. ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. BMC Bioinformatics 12:449.


Marioni JC, Mason CE, Mane SM, Stephens M, Gilad Y. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. Genome Res. 2008 Sep;18(9):1509-17.