

Recount Meta-Analysis

Jeff Leek and Margaret Taub

June 7, 2016

Contents

Analysis	1
Load libraries we will need	1
Get data sets	1
Load the data	2
Load the metadata file	2
Get additional GEI data for these samples	2
Merge the data sets	3
Subset to normal control individuals	3
Analysis comparing blood to colon	4
Compare concordance at the top (CAT) plots	6
Reproducibility	7

Analysis

To illustrate the ease of combining data from multiple projects included in **recount** as part of a cross-study meta-analysis, we present here a cross-tissue differential expression (DE) analysis comparing gene expression between colon and whole blood. We perform an initial analysis on samples from a variety of studies found in the **recount** project and then compare the results we obtain to those from analysis of GTEx data in the same tissues.

Load libraries we will need

```
library('dplyr')
library('recount')

##
## This data.table install has not detected OpenMP support. It will work but slower in single threaded mode

library('magrittr')
library('limma')
library('edgeR')
library('ffpe')
library('RSkittleBrewer')
library('SummarizedExperiment')
library('devtools')
trop <- RSkittleBrewer::RSkittleBrewer('tropical')
```

Get data sets

As an initial analysis, colon samples labeled as controls were taken from studies SRP029880 (a study of colorectal cancer (Kim, Kim, Kim, Roh, et al., 2014), n=19) and SRP042228 (a study of Crohn's disease

(Denson, 2014), n=41). Whole blood samples labeled as controls were taken from SRP059039 (a study of virus-caused diarrhea, unpublished, n=24), SRP059172 (a study of blood biomarkers for brucellosis, unpublished, n=47) and SRP062966 (a study of lupus, unpublished, n=18).

We start by downloading `.Rdata` files containing `SummarizedExperiment` objects for each of these studies, which contain RNA-seq counts at the gene level for each individual in the study, as well as accompanying sample-specific phenotype data.

```
colon_proj <- c('SRP029880', 'SRP042228')
if(any(!file.exists(file.path(colon_proj, 'rse_gene.Rdata')))) {
  sapply(colon_proj, download_study)
}

blood_proj <- c('SRP059039', 'SRP059172', 'SRP062966')
if(any(!file.exists(file.path(blood_proj, 'rse_gene.Rdata')))) {
  sapply(blood_proj, download_study)
}

proj <- c(colon_proj, blood_proj)
```

Load the data

Once the data has been downloaded, we load all five data sets into R and calculate the number of genes (number of rows) and samples (number of columns) for each data set.

```
dat <- lapply(proj, function(x) {
  load(file.path(x, 'rse_gene.Rdata'))
  return(rse_gene)
})
proj
```

```
## [1] "SRP029880" "SRP042228" "SRP059039" "SRP059172" "SRP062966"
```

```
sapply(dat, dim)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 23779 23779 23779 23779 23779
## [2,]    54   314   205   169   117
```

Load the metadata file

Since the data contained in the `SummarizedExperiment` objects for each study are somewhat limited, we also load a data frame containing metadata for each sample, including Gene Expression Omnibus (GEO) sample identifiers. We will use this in order to identify and extract the control subjects from each study for analysis.

```
metadata <- all_metadata('sra')
```

```
## 2016-11-30 23:38:38 downloading the metadata to /var/folders/cx/n9s558kx6fb7jf5z_pgszgb80000gn/T//Rt
```

Get additional GEI data for these samples

Now we go through and collect GEO information for the samples. We label them with their respective tissue and identify which samples are supposed to be normal controls, with labels varying from data set to data set. We also label the data sets with their tissue of origin.

```

if(!file.exists('charvec.Rdata')) {
  charvec <- vector('list', 5)
  dir.create('geoinfo', showWarnings = FALSE)
  for(i in 1:5){
    index <- match(colData(dat[[i]])$run, metadata$run)
    colData(dat[[i]])$geo <- metadata$geo_accession[index]
    info <- sapply(colData(dat[[i]])$geo, geo_info, destdir = 'geoinfo')
    charvec[[i]] <- sapply(info, geo_characteristics)
  }
  save(charvec, file = 'charvec.Rdata')
} else {
  load('charvec.Rdata')
}

## first data set - normals called 'normal-looking surrounding colonic epithelium'
colData(dat[[1]])$normal <- grepl('normal', unlist(charvec[1])[(1:54) * 2 - 1])
colData(dat[[1]])$tissue <- 'colon'

## second data set - normals called 'not ibd'
colData(dat[[2]])$normal <- grepl('not ibd', tolower(unlist(charvec[[2]][5, ])))
colData(dat[[2]])$tissue <- 'colon'

## third data set - normals called 'Control'
colData(dat[[3]])$normal <- grepl('Control', unlist(charvec[[3]][2, ]))
colData(dat[[3]])$tissue <- 'blood'

## fourth data set - normals called 'Control'
colData(dat[[4]])$normal <- grepl('Control', unlist(charvec[[4]][1, ]))
colData(dat[[4]])$tissue <- 'blood'

## fifth data set - normals called 'healthy'
colData(dat[[5]])$normal <- grepl('healthy', unlist(charvec[[5]][1, ]))
colData(dat[[5]])$tissue <- 'blood'

```

Merge the data sets

We merge the data sets into one ranged SummarizedExperiment.

```
mdat <- do.call(cbind, dat)
```

Subset to normal control individuals

Using the label determined above, we find out how many samples are normal in each study and subset to just the normal samples for further analysis.

```
table(colData(mdat)$normal, colData(mdat)$project)
```

```
##
##          SRP029880 SRP042228 SRP059039 SRP059172 SRP062966
##  FALSE          35         273         181         122          99
##   TRUE           19          41          24          47          18
```

```
ndat <- mdat[, colData(mdat)$normal]
```

Analysis comparing blood to colon

Here we do a differential expression analysis comparing blood to colon using *voom* (Law, Chen, Shi, and Smyth, 2014). We start by normalizing the counts and then consider only the genes that have an average normalized count of at least 5 across the data set.

```
ndat <- scale_counts(ndat)
ndat_counts <- assays(ndat)[[1]]
keep <- rowMeans(ndat_counts) > 5
ndat_counts = ndat_counts[keep, ]
design <- model.matrix(~colData(ndat)$tissue)
dge <- DGEList(counts = ndat_counts)
dge <- calcNormFactors(dge)
v <- voom(dge, design, plot=FALSE)
fit <- lmFit(v, design)
fit <- eBayes(fit)
topTable(fit)
```

```
## Removing intercept from test coefficients
```

		logFC	AveExpr	t	P.Value	adj.P.Val	B
##	7103	15.409396	0.4924358	79.43601	2.020597e-125	3.560898e-121	272.2428
##	10083	13.253543	-0.3745418	75.54867	3.368731e-122	2.968357e-118	265.4188
##	1909	7.783661	-2.5134139	75.11897	7.820641e-122	4.594105e-118	263.9306
##	56667	14.923360	0.6573342	73.53603	1.811337e-120	7.980298e-117	261.7541
##	25878	11.098638	-1.1683531	72.98350	5.507110e-120	1.941036e-116	260.6763
##	57381	8.703755	-1.8892183	70.64164	6.712078e-118	1.971449e-114	256.0142
##	133584	7.382474	-2.6339197	70.55688	8.008934e-118	2.016306e-114	255.2011
##	1441	-8.823323	8.1510433	-68.43403	7.140208e-116	1.143926e-112	254.6341
##	1015	14.208157	0.9677960	69.54431	6.710833e-117	1.314056e-113	254.2764
##	441094	6.910093	-2.8053456	69.62619	5.644920e-117	1.243505e-113	253.1423

GTEx analysis: comparing same tissues

In order to have something to compare to the results generated on the data sets selected above, we do a differential expression analysis comparing blood to colon samples from GTEx, project SRP012682. We carry out the same steps as above for the analysis, but in this case we also control for batch as included in the GTEx metadata.

```
## Download the GTEx data
if(!file.exists(file.path('SRP012682', 'rse_gene.Rdata'))){
  download_study('SRP012682')
}
load(file.path('SRP012682', 'rse_gene.Rdata'))
```

```
## Download GTEx metadata
gtex_metadata <- all_metadata('gtex')
```

```
## 2016-11-30 23:39:07 downloading the metadata to /var/folders/cx/n9s558kx6fb7jf5z_pgszgb80000gn/T//Rt
```

```
## Subset to samples from tissues of interest
gtex_blood <- rse_gene[, subset(gtex_metadata, smtsd == 'Whole Blood')$run]
```

```
colData(gtex_blood)$tissue <- 'wholeblood'
gtex_colon <- rse_gene[, subset(gtex_metadata, smts == 'Colon')$run]
colData(gtex_colon)$tissue <- 'colon'

## Combine tissue subsets and include batch label
gtex_both <- do.call(cbind, list(gtex_blood, gtex_colon))
colData(gtex_both)$batch <- gtex_metadata[match(colData(gtex_both)$run,
  gtex_metadata$run), 'smgebtch']

## Scale and filter data to the same genes kept in initial analysis
gtex_both <- scale_counts(gtex_both)
gtex_both_counts <- assays(gtex_both)[[1]]
gtex_both_counts <- gtex_both_counts[keep, ]

## Carry out differential expression analysis
design_gtex <- model.matrix(~colData(gtex_both)$tissue +
  colData(gtex_both)$batch)
dge_gtex <- DGEList(counts = gtex_both_counts)
dge_gtex <- calcNormFactors(dge_gtex)
v_gtex <- voom(dge_gtex, design_gtex, plot=FALSE)
fit_gtex <- lmFit(v_gtex, design_gtex)
fit_gtex <- eBayes(fit_gtex)
topTable(fit_gtex, coef = 2)
```

##		logFC	AveExpr	t	P.Value	adj.P.Val	B
##	6793	4.773601	6.779423	90.21488	0	0	874.9992
##	409	6.040177	7.723587	88.28947	0	0	861.2107
##	4542	7.490221	7.258075	84.59393	0	0	834.0351
##	4688	8.381934	6.579985	84.59066	0	0	833.9083
##	3936	7.980207	8.364888	84.02830	0	0	829.8006
##	1793	-5.244754	2.974639	-83.29415	0	0	823.9264
##	101	7.478347	6.054686	83.16901	0	0	823.2335
##	752	6.065599	7.218798	81.74749	0	0	812.4860
##	8514	4.617038	6.310878	80.81177	0	0	805.2694
##	51312	6.423430	7.944806	80.73634	0	0	804.6646

GTEx analysis: comparing different tissues

To have a way of gauging how much overlap we expect to see between the results from the five combined data sets and those from GTEx data, we also carry out a differential analysis using GTEx data comparing blood to lung. We expect to see much less concordance between our initial results and the GTEx blood-lung comparison than we see between our initial results and the GTEx blood-colon comparison.

```
## Subset lung data
gtex_lung <- rse_gene[, subset(gtex_metadata, smts=='Lung')$run]
colData(gtex_lung)$tissue <- 'lung'

## Combine lung and blood data and label with batch
gtex_both_lung <- do.call(cbind, list(gtex_blood, gtex_lung))
colData(gtex_both_lung)$batch <- gtex_metadata[
  match(
    colData(gtex_both_lung)$run,
    gtex_metadata$run
```

```

    ), 'smgebtch']

## Scale and filter to correct gene set
gtex_both_lung <- scale_counts(gtex_both_lung)
gtex_both_lung_counts <- assays(gtex_both_lung)[[1]]
gtex_both_lung_counts <- gtex_both_lung_counts[keep,]

## Carry out differential expression analysis
design_gtex_lung <- model.matrix(~colData(gtex_both_lung)$tissue +
    colData(gtex_both_lung)$batch)
dge_gtex_lung <- DGEList(counts = gtex_both_lung_counts)
dge_gtex_lung <- calcNormFactors(dge_gtex_lung)
v_gtex_lung <- voom(dge_gtex_lung, design_gtex_lung, plot = FALSE)
fit_gtex_lung <- lmFit(v_gtex_lung, design_gtex_lung)
fit_gtex_lung <- eBayes(fit_gtex_lung)
topTable(fit_gtex_lung, coef = 2)

```

	##	logFC	AveExpr	t	P.Value	adj.P.Val	B
	## 221395	-7.675466	3.4109146	-106.91629	0	0	996.3729
	## 6943	-8.792146	1.6860275	-101.86151	0	0	964.1776
	## 5754	-5.150188	2.7899033	-97.54954	0	0	935.9110
	## 5420	-6.092697	4.0348220	-97.15255	0	0	933.2634
	## 6909	-8.130718	2.9581907	-97.16373	0	0	933.1846
	## 599	-2.835088	4.1595326	-96.50413	0	0	928.9989
	## 10418	-6.396727	3.2092320	-94.48530	0	0	914.9644
	## 9368	3.712945	6.1694693	94.43085	0	0	914.8108
	## 207107	-8.226899	-0.5998609	-93.51892	0	0	908.0309
	## 10160	-5.323208	3.0386054	-93.39156	0	0	907.3923

Compare concordance at the top (CAT) plots

To look for overlap among the most differentially expressed genes across the three analyses, we construct concordance at the top (CAT) plots. We make three comparisons: our five-study results compared to GTEx blood-colon results (same tissue comparison), our five-study results compared to GTEx blood-lung results (different tissue comparison) and our five-study results compared to a ranked set of coefficients for the batch variable in the GTEx blood-lung results (which should not show any biologically meaningful correspondence).

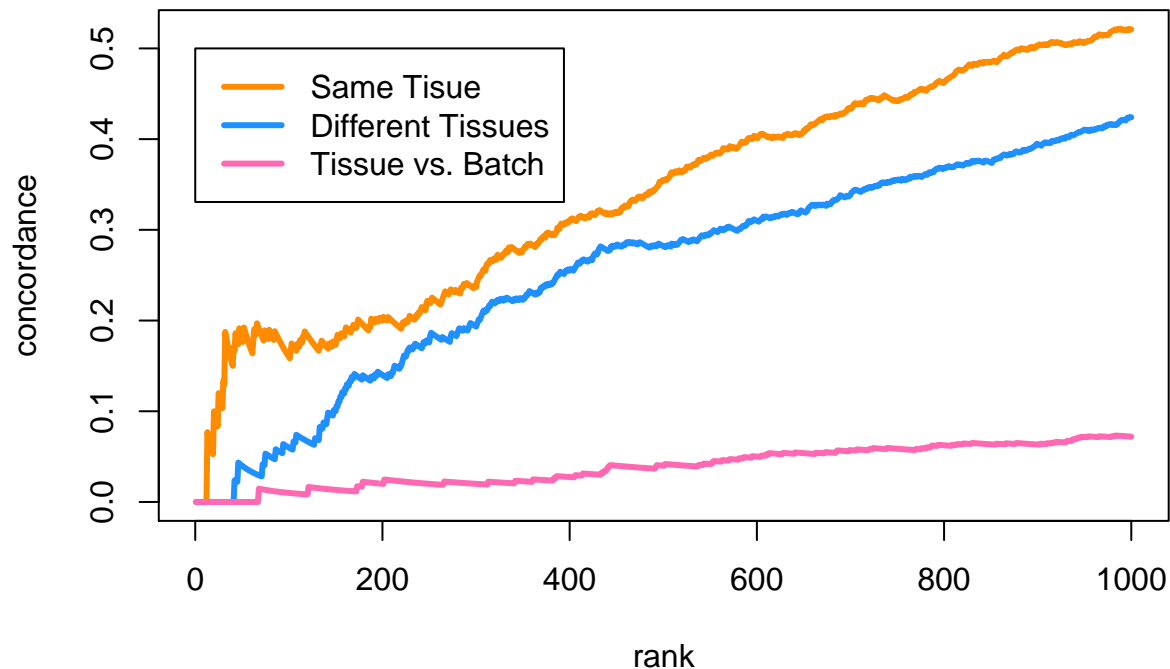
```

## Comparison of initial analysis to GTEx, same tissues
cat_sra_gtex <- CATplot(
  -rank(fit$coefficients[, 2]),
  -rank(-fit_gtex$coefficients[, 2]), maxrank = 1000, ylim = c(0,1), make.plot=FALSE)
## Comparison of initial analysis to GTEx, different tissues
cat_sra_gtex_lung = CATplot(
  -rank(fit$coefficients[, 2]),
  -rank(-fit_gtex_lung$coefficients[, 2]), maxrank = 1000, ylim = c(0,1), make.plot=FALSE)
## Comparison of initial analysis to GTEx, coefficient of batch in model
cat_sra_gtex_batch = CATplot(
  -rank(fit$coefficients[, 2]),
  -rank(-fit_gtex_lung$coefficients[, 3]), maxrank = 1000, ylim = c(0,1), make.plot=FALSE)

plot(cat_sra_gtex, type = 'l', col = trop[1], lwd = 3)
lines(cat_sra_gtex_lung, type = 'l', col = trop[2], lwd = 3)
lines(cat_sra_gtex_batch, type = 'l', col = trop[3], lwd = 3)

```

```
legend(0, 0.5, legend=c('Same Tissue', 'Different Tissues',  
  'Tissue vs. Batch'), col = trop[1:3], lwd = 3)
```



From the figure above, we can see that the best concordance is for the comparison to GTEx where we analyzed the same pair of tissues. The GTEx analysis with lung looks considerably less overlapping, with the overlap with the batch coefficient well below the other two.

Reproducibility

This analysis report was made possible thanks to:

- R (R Core Team, 2016)
- *BiocStyle* (Oleś, Morgan, and Huber, 2016)
- *derfinder* (Collado-Torres, Nellore, Frazee, Wilks, et al., 2016)
- *devtools* (Wickham and Chang, 2016)
- *dplyr* (Wickham and Francois, 2016)
- *edgeR* (Robinson, McCarthy, and Smyth, 2010)
- *ffpe* (Waldron, L, Ogino, Shuji, Hoshida, Yujin, Shima, Kaori, et al., 2012)
- *knitcitations* (Boettiger, 2015)
- *magrittr* (Bache and Wickham, 2014)
- *recount* (Collado-Torres, Nellore, Kammers, Ellis, et al., 2016)
- *rmarkdown* (Allaire, Cheng, Xie, McPherson, et al., 2016)
- *RSkittleBrewer* (Frazee, 2016)
- *SummarizedExperiment* (Morgan, Obenchain, Hester, and Pagès, 2016)
- *limma* (Law, Chen, Shi, and Smyth, 2014)

Bibliography file

[1] J. Allaire, J. Cheng, Y. Xie, J. McPherson, et al. *rmarkdown*: Dynamic Documents for R. R package version 1.2. 2016. URL: <https://CRAN.R-project.org/package=rmarkdown>.

[2] S. M. Bache and H. Wickham. *magrittr*: A Forward-Pipe Operator for R. R package version 1.5. 2014. URL: <https://CRAN.R-project.org/package=magrittr>.

[3] C. Boettiger. knitcitations: Citations for ‘Knitr’ Markdown Files. R package version 1.0.7. 2015. URL: <https://CRAN.R-project.org/package=knitcitations>.

[4] L. Collado-Torres, A. Nellore, A. C. Frazee, C. Wilks, et al. “Flexible expressed region analysis for RNA-seq with derfinder”. In: Nucl. Acids Res. (2016). DOI: 10.1093/nar/gkw852. URL: <http://nar.oxfordjournals.org/content/early/2016/09/29/nar.gkw852>.

[5] L. Collado-Torres, A. Nellore, K. Kammers, S. E. Ellis, et al. “recount: A large-scale resource of analysis-ready RNA-seq expression data”. In: bioRxiv (2016). DOI: 10.1101/068478. URL: <http://biorxiv.org/content/early/2016/08/08/068478>.

[6] Y. H. A. T. L. T. A. P. J. D. A. M. K. A. D. T. A. R. K. A. R. N. B. A. J. D. N. A. J. R. A. J. M. A. M. B. H. A. A. M. G. A. W. V. C. A. D. R. M. A. S. S. B. A. C. H. A. D. J. K. A. J. S. H. A. S. K. A. T. D. W. A. B. A. A. R. J. X. A. D. G. A. L. A. Denson. “Pediatric Crohn disease patients exhibit specific ileal transcriptome and microbiome signature”. In: The Journal of Clinical Investigation 124 (2014), pp. 3617-3633.

[7] A. Frazee. RSkittleBrewer: Fun with R Colors. R package version 1.1. 2016. URL: <https://github.com/alyssafrazee/RSkittleBrewer>.

[8] S. Kim, S. Kim, J. Kim, S. A. Roh, et al. “A nineteen gene-based risk score classifier predicts prognosis of colorectal cancer patients”. In: Molecular Oncology 8 (2014), pp. 1653 - 1666.

[9] C. Law, Y. Chen, W. Shi and G. Smyth. “Voom: precision weights unlock linear model analysis tools for RNA-seq read counts”. In: Genome Biology 15 (2014), p. R29.

[10] M. Morgan, V. Obenchain, J. Hester and H. Pagès. SummarizedExperiment: SummarizedExperiment container. R package version 1.4.0. 2016.

[11] A. Oleś, M. Morgan and W. Huber. BiocStyle: Standard styles for vignettes and other Bioconductor documents. R package version 2.2.1. 2016. URL: <https://github.com/Bioconductor/BiocStyle>.

[12] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: <https://www.R-project.org/>.

[13] M. D. Robinson, D. J. McCarthy and G. K. Smyth. “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data”. In: Bioinformatics 26 (2010), pp. -1.

[14] Waldron, L, Ogino, Shuji, Hoshida, Yujin, Shima, Kaori, et al. “Expression profiling of archival tumors for long-term health studies”. In: Clinical Cancer Research 18.22 (2012). PMID: 23136189, pp. 6136–6146. DOI: 10.1158/1078-0432.CCR-12-1915.

[15] H. Wickham and W. Chang. devtools: Tools to Make Developing R Packages Easier. R package version 1.12.0. 2016. URL: <https://CRAN.R-project.org/package=devtools>.

[16] H. Wickham and R. Francois. dplyr: A Grammar of Data Manipulation. R package version 0.5.0. 2016. URL: <https://CRAN.R-project.org/package=dplyr>.

```
## Time spent creating this report:
diff(c(timestart, Sys.time()))
```

```
## Time difference of 13.54219 mins
```

```
## Date this report was generated
message(Sys.time())
```

```
## 2016-11-30 23:51:52
```

```
## Reproducibility info
options(width = 120)
devtools::session_info()
```

```
## Session info -----
```



```
## setting value
## version R version 3.3.2 RC (2016-10-26 r71594)
## system x86_64, darwin13.4.0
## ui X11
## language (EN)
## collate en_US.UTF-8
## tz America/New_York
## date 2016-11-30
```

```
## Packages -----
## package      * version  date      source
## acepack      1.4.1    2016-10-29 CRAN (R 3.3.0)
## affy         1.52.0   2016-10-18 Bioconductor
## affyio       1.44.0   2016-10-18 Bioconductor
## annotate     1.52.0   2016-10-18 Bioconductor
## AnnotationDbi 1.36.0   2016-10-18 Bioconductor
## assertthat   0.1      2013-12-06 cran (@0.1)
## backports    1.0.4    2016-10-24 CRAN (R 3.3.0)
## base64       2.0      2016-05-10 CRAN (R 3.3.0)
## beanplot     1.2      2014-09-19 CRAN (R 3.3.0)
## bibtex       0.4.0    2014-12-31 CRAN (R 3.3.0)
## Biobase      * 2.34.0   2016-10-18 Bioconductor
## BiocGenerics * 0.20.0   2016-10-18 Bioconductor
## BiocInstaller 1.24.0   2016-10-18 Bioconductor
## BiocParallel 1.8.1    2016-10-30 Bioconductor
## BiocStyle    * 2.2.1    2016-11-24 Bioconductor
## biomaRt      2.30.0   2016-10-18 Bioconductor
## Biostrings   2.42.0   2016-10-18 Bioconductor
## bitops       1.0-6    2013-08-17 cran (@1.0-6)
## BSgenome     1.42.0   2016-10-18 Bioconductor
## bumphunter   1.14.0   2016-10-18 Bioconductor
## cluster      2.0.5    2016-10-08 CRAN (R 3.3.2)
## codetools    0.2-15   2016-10-05 CRAN (R 3.3.2)
## colorout     * 1.1-2    2016-10-19 Github (jalvesaq/colorout@6d84420)
## colorspace   1.3-1    2016-11-18 CRAN (R 3.3.2)
## data.table   1.9.8    2016-11-25 CRAN (R 3.3.2)
## DBI          0.5-1    2016-09-10 cran (@0.5-1)
## derfinder    1.8.0    2016-10-18 Bioconductor
## derfinderHelper 1.8.0    2016-10-18 Bioconductor
## devtools     * 1.12.0   2016-06-24 CRAN (R 3.3.0)
## digest       0.6.10   2016-08-02 CRAN (R 3.3.0)
## doRNG        1.6      2014-03-07 CRAN (R 3.3.0)
## downloader   0.4      2015-07-09 CRAN (R 3.3.0)
## dplyr        * 0.5.0    2016-06-24 cran (@0.5.0)
## edgeR        * 3.16.4   2016-11-27 Bioconductor
## evaluate     0.10     2016-10-11 cran (@0.10)
## ffpe         * 1.18.0   2016-10-18 Bioconductor
## foreach      1.4.3    2015-10-13 CRAN (R 3.3.0)
## foreign      0.8-67   2016-09-13 CRAN (R 3.3.2)
## Formula      1.2-1    2015-04-07 CRAN (R 3.3.0)
## genefilter    1.56.0   2016-10-18 Bioconductor
## GenomeInfoDb * 1.10.1   2016-11-04 Bioconductor
## GenomicAlignments 1.10.0   2016-10-18 Bioconductor
## GenomicFeatures 1.26.0   2016-10-18 Bioconductor
```

##	GenomicFiles	1.10.3	2016-10-21	Bioconductor
##	GenomicRanges	* 1.26.1	2016-10-20	Bioconductor
##	GEOquery	2.40.0	2016-10-18	Bioconductor
##	ggplot2	2.2.0	2016-11-11	CRAN (R 3.3.2)
##	gridExtra	2.2.1	2016-02-29	CRAN (R 3.3.0)
##	gtable	0.2.0	2016-02-26	CRAN (R 3.3.0)
##	Hmisc	4.0-0	2016-11-01	CRAN (R 3.3.0)
##	htmlTable	1.7	2016-10-19	CRAN (R 3.3.0)
##	htmltools	0.3.5	2016-03-21	cran (@0.3.5)
##	httr	1.2.1	2016-07-03	CRAN (R 3.3.0)
##	illuminaio	0.16.0	2016-10-18	Bioconductor
##	IRanges	* 2.8.1	2016-11-08	Bioconductor
##	iterators	1.0.8	2015-10-13	CRAN (R 3.3.0)
##	jsonlite	1.1	2016-09-14	CRAN (R 3.3.0)
##	KernSmooth	2.23-15	2015-06-29	CRAN (R 3.3.2)
##	knitcitations	* 1.0.7	2015-10-28	CRAN (R 3.3.0)
##	knitr	1.15.1	2016-11-22	CRAN (R 3.3.2)
##	lattice	0.20-34	2016-09-06	CRAN (R 3.3.2)
##	latticeExtra	0.6-28	2016-02-09	CRAN (R 3.3.0)
##	lazyeval	0.2.0	2016-06-12	cran (@0.2.0)
##	limma	* 3.30.6	2016-11-29	Bioconductor
##	locfit	1.5-9.1	2013-04-20	CRAN (R 3.3.0)
##	lubridate	1.6.0	2016-09-13	CRAN (R 3.3.0)
##	lumi	2.26.3	2016-11-15	Bioconductor
##	magrittr	* 1.5	2014-11-22	cran (@1.5)
##	MASS	7.3-45	2016-04-21	CRAN (R 3.3.2)
##	Matrix	1.2-7.1	2016-09-01	CRAN (R 3.3.2)
##	matrixStats	0.51.0	2016-10-09	CRAN (R 3.3.0)
##	mclust	5.2	2016-03-31	CRAN (R 3.3.0)
##	memoise	1.0.0	2016-01-29	CRAN (R 3.3.0)
##	methylumi	2.20.0	2016-10-18	Bioconductor
##	mgcv	1.8-16	2016-11-07	CRAN (R 3.3.0)
##	minfi	1.20.0	2016-10-18	Bioconductor
##	multtest	2.30.0	2016-10-18	Bioconductor
##	munsell	0.4.3	2016-02-13	cran (@0.4.3)
##	nleqslv	3.0.3	2016-08-08	CRAN (R 3.3.0)
##	nlme	3.1-128	2016-05-10	CRAN (R 3.3.2)
##	nnet	7.3-12	2016-02-02	CRAN (R 3.3.2)
##	nor1mix	1.2-2	2016-08-25	CRAN (R 3.3.0)
##	openssl	0.9.5	2016-10-28	CRAN (R 3.3.0)
##	pkgmaker	0.22	2014-05-14	CRAN (R 3.3.0)
##	plyr	1.8.4	2016-06-08	cran (@1.8.4)
##	preprocessCore	1.36.0	2016-10-18	Bioconductor
##	quadprog	1.5-5	2013-04-17	CRAN (R 3.3.0)
##	qvalue	2.6.0	2016-10-18	Bioconductor
##	R6	2.2.0	2016-10-05	CRAN (R 3.3.0)
##	RColorBrewer	1.1-2	2014-12-07	cran (@1.1-2)
##	Rcpp	0.12.8	2016-11-17	CRAN (R 3.3.2)
##	RCurl	1.95-4.8	2016-03-01	cran (@1.95-4.)
##	recount	* 1.0.3	2016-11-27	Bioconductor
##	RefManager	0.13.1	2016-11-13	CRAN (R 3.3.2)
##	registry	0.3	2015-07-08	CRAN (R 3.3.0)
##	rentrez	1.0.4	2016-10-26	CRAN (R 3.3.0)
##	reshape	0.8.6	2016-10-21	CRAN (R 3.3.0)

##	reshape2	1.4.2	2016-10-22	CRAN (R 3.3.0)
##	RJSONIO	1.3-0	2014-07-28	cran (@1.3-0)
##	rmarkdown	* 1.2	2016-11-21	CRAN (R 3.3.2)
##	rngtools	1.2.4	2014-03-06	CRAN (R 3.3.0)
##	rpart	4.1-10	2015-06-29	CRAN (R 3.3.2)
##	rprojroot	1.1	2016-10-29	CRAN (R 3.3.0)
##	Rsamtools	1.26.1	2016-10-22	Bioconductor
##	RSkittleBrewer	* 1.1	2016-10-19	Github (alyssafrazee/RSkittleBrewer@0088112)
##	RSQLite	1.1	2016-11-27	CRAN (R 3.3.1)
##	rtracklayer	1.34.1	2016-11-02	Bioconductor
##	S4Vectors	* 0.12.0	2016-10-18	Bioconductor
##	scales	0.4.1	2016-11-09	CRAN (R 3.3.2)
##	sfsmisc	1.1-0	2016-02-23	CRAN (R 3.3.0)
##	siggenes	1.48.0	2016-10-18	Bioconductor
##	stringi	1.1.2	2016-10-01	cran (@1.1.2)
##	stringr	1.1.0	2016-08-19	cran (@1.1.0)
##	SummarizedExperiment	* 1.4.0	2016-10-18	Bioconductor
##	survival	2.40-1	2016-10-30	CRAN (R 3.3.0)
##	tibble	1.2	2016-08-26	cran (@1.2)
##	TTR	* 0.23-1	2016-03-21	CRAN (R 3.3.0)
##	VariantAnnotation	1.20.1	2016-11-14	Bioconductor
##	withr	1.0.2	2016-06-20	CRAN (R 3.3.0)
##	XML	3.98-1.5	2016-11-10	CRAN (R 3.3.2)
##	xtable	1.8-2	2016-02-05	CRAN (R 3.3.0)
##	xts	0.9-7	2014-01-02	CRAN (R 3.3.0)
##	XVector	0.14.0	2016-10-18	Bioconductor
##	yaml	2.1.14	2016-11-12	CRAN (R 3.3.2)
##	zlibbioc	1.20.0	2016-10-18	Bioconductor
##	zoo	1.7-13	2016-05-03	CRAN (R 3.3.0)