Creation of `parathyroidGenes` and `parathyroidExons`

Michael Love

September 13, 2014

Abstract

This vignette describes the construction of the data objects in the *parathyroid* package, which are a *CountDataSet* and an *ExonCountSet*. Note that the *ExonCountSet* is beginning to be deprecated (as of March 2014 and *DEXSeq* version 1.10.0). Another Bioconductor data package, *parathyroidSE*, describes the creation of *SummarizedExperiment* objects from the same RNA-Seq experiment files.

Contents

1 Dataset description

We downloaded the RNA-Seq data from the publication of Haglund et al. [1]. The paired-end sequencing was performed on primary cultures from parathyroid tumors of 4 patients at 2 time points over 3 conditions (control, treatment with diarylpropionitrile (DPN) and treatment with 4-hydroxytamoxifen (OHT)). DPN is a selective estrogen receptor β 1 agonist and OHT is a selective estrogen receptor modulator. One sample (patient 4, 24 hours, control) was omitted by the paper authors due to low quality.

2 Downloading the data

The raw sequencing data is publicly available from the NCBI Gene Expression Omnibus under accession number GSE37211[1]. The read sequences in FASTQ format were extracted from the NCBI short read archive file (.sra files), using the sra toolkit[2].

3 Aligning and counting reads

The sequenced reads in the FASTQ files were aligned using TopHat version 2.0.4[3] with default parameters to the GRCh37 human reference genome using the Bowtie index available at the Illumina iGenomes page[4]. An example call for alignment (substituting the SRR number for `file`):

```
tophat2 -o file_tophat_out -p 8 genome file_1.fastq file_2.fastq
samtools index file_tophat_out/accepted_hits.bam
samtools view file_tophat_out/accepted_hits.bam | \
    sort -k1,1 -k2,2n > \
file_tophat_out/accepted_hits_sorted.sam
```

For counting reads in genes, we used `htseq-count` from the HTSeq Python package[5]. We counted reads in genes, using the GTF file packaged with the GRCh37 Illumina iGenome, Ensembl release 66 downloaded 9 March 2012, under the `Ensembl` directory. We used the minimum read quality setting `-a 10`, which is the same setting used by the *DEXSeq* python script described in the next paragraph, and `-s no`, because the assay is not strand-specific. An example call for read counting:

```
htseq-count -a 10 -s no file_tophat_out/accepted_hits_sorted.sam \
    Homo_sapiens/Ensembl/GRCh37/Annotation/Genes/genes.gtf > \
    file_tophat_out/file_gene_counts.txt
```

For counting reads in exons, we used Python scripts provided in the *DEXSeq* packge. For features we again used the Ensembl gene annotations. We use the options `-p yes -s no` to indicate the reads are paired-end, and the assay is not strand-specific. An example call for preparing the exon annotations and read counting:

```
python DEXSeq/inst/python_scripts/dexseq_prepare_annotation.py \
    Homo_sapiens/Ensembl/GRCh37/Annotation/Genes/genes.gtf \
    Hsap.GRCh37.DEXSeq.gff
python DEXSeq/inst/python_scripts/dexseq_count.py -p yes -s no \
    Hsap.GRCh37.DEXSeq.gff file_tophat_out/accepted_hits_sorted.sam \
    file_tophat_out/file_exon_counts.txt
```

[1] `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE37211`
[2] `http://www.ncbi.nlm.nih.gov/books/NBK56560/`
[3] `http://tophat.cbcb.umd.edu/`
[4] `http://tophat.cbcb.umd.edu/igenomes.html`
[5] `http://www-huber.embl.de/users/anders/HTSeq/`

4 Obtaining sample annotations from GEO

In order to provide phenotypic data for the samples, we used the *GEOquery* package to parse the series matrix file downloaded from the NCBI Gene Expression Omnibus under accession number GSE37211. We included this file as well in the package, and read it in locally in the code below.

```
> library("parathyroid")
> library("GEOquery")
> gse37211 <- getGEO(filename = system.file("extdata/GSE37211_series_matrix.txt",
+     package = "parathyroid", mustWork = TRUE))
> samples <- pData(gse37211)[, c("characteristics_ch1", "characteristics_ch1.2",
+     "characteristics_ch1.3", "relation")]
> colnames(samples) <- c("patient", "treatment", "time", "experiment")
> samples$patient <- sub("patient: (.+)", "\\1", samples$patient)
> samples$treatment <- sub("agent: (.+)", "\\1", samples$treatment)
> samples$time <- sub("time: (.+)", "\\1", samples$time)
> samples$experiment <- sub("SRA: http://www.ncbi.nlm.nih.gov/sra\\?term=(.+)",
+     "\\1", samples$experiment)
> samples
```

	patient	treatment	time	experiment
GSM913873	1	Control	24h	SRX140503
GSM913874	1	Control	48h	SRX140504
GSM913875	1	DPN	24h	SRX140505
GSM913876	1	DPN	48h	SRX140506
GSM913877	1	OHT	24h	SRX140507
GSM913878	1	OHT	48h	SRX140508
GSM913879	2	Control	24h	SRX140509
GSM913880	2	Control	48h	SRX140510
GSM913881	2	DPN	24h	SRX140511
GSM913882	2	DPN	48h	SRX140512
GSM913883	2	OHT	24h	SRX140513
GSM913884	2	OHT	48h	SRX140514
GSM913885	3	Control	24h	SRX140515
GSM913886	3	Control	48h	SRX140516
GSM913887	3	DPN	24h	SRX140517
GSM913888	3	DPN	48h	SRX140518
GSM913889	3	OHT	24h	SRX140519
GSM913890	3	OHT	48h	SRX140520
GSM913891	4	Control	48h	SRX140521
GSM913892	4	DPN	24h	SRX140522
GSM913893	4	DPN	48h	SRX140523
GSM913894	4	OHT	24h	SRX140524
GSM913895	4	OHT	48h	SRX140525

5 Matching GEO experiments with SRA runs

The sample information from GEO must be matched to the individual runs from the Short Read Archive (the FASTQ files), as some samples are spread over multiple sequencing runs. The

run information can be obtained from the Short Read Archive using the *SRAdb* package (note that the first step involves a large download of the SRA metadata database). We included the conversion table in the package.

```
> library("SRAdb")
> sqlfile <- getSRAdbFile()
> sra_con <- dbConnect(SQLite(), sqlfile)
> conversion <- sraConvert(in_acc = samples$experiment, out_type = c("sra",
+     "submission", "study", "sample", "experiment", "run"), sra_con = sra_con)
> write.table(conversion, file = "inst/extdata/conversion.txt")
```

We used the `merge` function to match the sample annotations to the run information. We ordered the *data.frame* `samplesFull` by the run number and then set all columns as factors.

```
> conversion <- read.table(system.file("extdata/conversion.txt",
+     package = "parathyroid", mustWork = TRUE))
> samplesFull <- merge(samples, conversion)
> samplesFull <- samplesFull[order(samplesFull$run), ]
> samplesFull <- as.data.frame(lapply(samplesFull, factor))
```

6 Creating the *CountDataSet* parathyroidGenes

We used the function `newCountDataSetFromHTSeqCount` of the *DESeq* package to read in our gene-level counts. We create a *data.frame* `sampleTable`, with the sample names as the first column and the count files as the second column. The count files are not included in the package due to memory constraints.

```
> library("DESeq")
> genecountfiles <- paste(samplesFull$run, "_gene_counts.txt",
+     sep = "")
> sampleTable <- cbind(samplenames = samplesFull$run, countfiles = genecountfiles,
+     samplesFull)
> parathyroidGenes <- newCountDataSetFromHTSeqCount(sampleTable,
+     directory = ".")
```

We included experiment data, PubMed ID and protocol data from the NCBI Gene Expression Omnibus.

```
> expdata = new("MIAME", name = "Felix Haglund", lab = "Science for Life Laboratory Stockholm",
+     contact = "Mikael Huss", title = "DPN and Tamoxifen treatments of parathyroid adenoma cells
+     url = "http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE37211",
+     abstract = "Primary hyperparathyroidism (PHPT) is most frequently present in postmenopausal
> pubMedIds(expdata) <- "23024189"
> n <- nrow(samplesFull)
> protocoldata <- AnnotatedDataFrame(data.frame(treatment = rep("cells were plated and treated wi
+     n), growth = rep("Tissue for cell culturing was obtained from four parathyroid adenomas col
+     n), extracted_molecule = rep("total RNA", n), extraction = rep("Illumina TruSeq RNA",
+     n), library_strategy = rep("RNA-Seq", n), library_source = rep("transcriptomic",
+     n), library_selection = rep("cDNA", n), instrument_model = rep("Illumina HiSeq 2000",
+     n), data_processing = rep("Alignment to GRCh37 using TopHat-2.0.4, default settings for pai
```

```
+     n), row.names = samplesFull$run))
> experimentData(parathyroidGenes) <- expdata
> protocolData(parathyroidGenes) <- protocoldata
```

7 Creating the *ExonCountSet* parathyroidExons

We used the `read.HTSeqCounts` function of the *DEXSeq* package to assemble an *ExonCountSet* using the count files and the gene annotation file produced by the Python scripts.

```
> library("DEXSeq")
> exoncountfiles <- paste(samplesFull$run, "_exon_counts.txt",
+     sep = "")
> annotationfile <- "Hsap.GRCh37.DEXSeq.gff"
> design <- data.frame(lapply(samplesFull, factor))
> parathyroidExons <- read.HTSeqCounts(countfiles = exoncountfiles,
+     design = design, flattenedfile = annotationfile)
> sampleNames(parathyroidExons) <- pData(parathyroidExons)$run
> experimentData(parathyroidExons) <- expdata
> protocolData(parathyroidExons) <- protocoldata

> save(parathyroidGenes, file = "data/parathyroidGenes.RData")
> save(parathyroidExons, file = "data/parathyroidExons.RData")
```

8 Session information

```
> toLatex(sessionInfo())
```

- R version 3.1.1 (2014-07-10), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, utils
- Other packages: Biobase 2.25.0, BiocGenerics 0.11.5, BiocParallel 0.99.19, DESeq 1.17.0, DESeq2 1.5.58, DEXSeq 1.11.14, GEOquery 2.31.1, GenomeInfoDb 1.1.19, GenomicRanges 1.17.40, IRanges 1.99.28, Rcpp 0.11.2, RcppArmadillo 0.4.400.0, S4Vectors 0.2.3, lattice 0.20-29, locfit 1.5-9.1, parathyroid 1.1.1
- Loaded via a namespace (and not attached): AnnotationDbi 1.27.9, BBmisc 1.7, BatchJobs 1.3, Biostrings 2.33.14, DBI 0.3.0, Formula 1.1-2, Hmisc 3.14-4, MASS 7.3-34, RColorBrewer 1.0-5, RCurl 1.95-4.3, RSQLite 0.11.4, Rsamtools 1.17.33, XML 3.98-1.1, XVector 0.5.8, annotate 1.43.5, biomaRt 2.21.1, bitops 1.0-6, brew 1.0-6, checkmate 1.4, cluster 1.15.3, codetools 0.2-9, colorspace 1.2-4, digest 0.6.4, fail 1.2, foreach 1.4.2, genefilter 1.47.6, geneplotter 1.43.0, ggplot2 1.0.0, grid 3.1.1, gtable 0.1.2, hwriter 1.3.2, iterators 1.0.7, latticeExtra 0.6-26, munsell 0.4.2, plyr 1.8.1, proto 0.3-10, reshape2 1.4, scales 0.2.4, sendmailR 1.1-2, splines 3.1.1, statmod 1.4.20, stats4 3.1.1, stringr 0.6.2, survival 2.37-7, tools 3.1.1, xtable 1.7-3, zlibbioc 1.11.1

References

[1] Felix Haglund, Ran Ma, Mikael Huss, Luqman Sulaiman, Ming Lu, Inga-Lena Nilsson, Anders Höög, Christofer C. Juhlin, Johan Hartman, and Catharina Larsson. Evidence of a Functional Estrogen Receptor in Parathyroid Adenomas. *Journal of Clinical Endocrinology & Metabolism*, September 2012.