Using RNASeqPipelineR

This vignette describes how to use RNASeqPipelineR to process your RNASeq data. The package requires a number of different external programs in order to function. These are listed in the package README.

System requirements

You need the following R packages: data.table GEOquery RSQLite SRAdb

The following command line utilities:
SRA Toolkit (from NCBI http://www.ncbi.nlm.nih.gov/books/NBK158900/)
ascp (Aspera scp client, distributed with Aspera Connect)
RSEM (http://deweylab.biostat.wisc.edu/rsem/)
bowtie2 (http://www.nature.com/nmeth/journal/v9/n4/full/nmeth.1923.html)
BioConductor (http://www.bioconductor.org/)
MiTCR (http://mitcr.milaboratory.com/) fastqc (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) PEAR (http://sco.h-its.org/exelixis/web/software/pear/) for paired end assembly GEOQuery from BioConductor GNU Parallel `annotate` package from BioConductor `pander` R package

It is assumed that the above can be invoked as from the path as: - `ascp`
- `rsem`
- `bowtie2`
- `mitcr`
- `fastqc`
- `pear`
- `parallel`

The following shell script is required for invoking `mitcr` (assuming the jar file is in `/usr/local/lib`)

```
#!/bin/bash
/usr/bin/java -Xmx4g -jar /usr/local/lib/mitcr.jar "$@"
```

and should be in your path.

Assumptions and Caveats

The package is not as robust as it could be and makes all sorts of assumptions (that are not unreasonable, but you need to be careful). - For paired-end data, we assume that paired FASTQ files differ by only one character. - We assume fastq files live in one directory and are not scattered across mutliple subdirectories in the main FASTQ folder. - We assume that command line tools have standard names (listed above) and are in your path. - Error checking is present, but if you encounter a cryptic error message, report it, we'll beef up the error handling. Contact the package maintainer named in the `DESCRIPTION` file. Better yet, open a `github` ticket.

Usage

Create a project

```
library(data.table)
library(GEOquery)
```

```
## Loading required package: Biobase
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
##
## The following object is masked from 'package:stats':
##
##     xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, as.vector, cbind,
##     colnames, do.call, duplicated, eval, evalq, Filter, Find, get,
##     intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rep.int, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unlist, unsplit
##
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Setting options('download.file.method.GEOquery'='auto')
```

```
library(knitr)
library(RNASeqPipelineR)
temp<-tempdir()
createProject("RNASeqPipelineRExample", path=temp, load_from_immport=FALSE)
```

```
## Project successfully created.
```

```
## aspera detected at /Users/gfinak/Applications/Aspera Connect.app/Contents/Resources
```

```
knitr::opts_chunk$set(cache=TRUE)
```

`load_from_immport` is set to false since we are starting with `fastq` files.

If the project already exists, you can load the project configuration.

```
loadProject(project_dir = temp,name="RNASeqPipelineRExample")
```

We move the fastq files that we want to process into place. RNASeqPipelineR just tracks the directories for various parts of the pipeline.
These can be accessed using `getConfig()[["subdirs"]]`. The directory structure for the project is created when the project is created using `creatProject`. The configuration is automatically saved to the project.

```
ext<-system.file("extdata",package = "RNASeqPipelineR")
tocopy<-list.files(path=ext,pattern="*.fastq",full=TRUE)
sapply(tocopy,function(x)file.copy(from = x,to = getConfig()[["subdirs"]][["FASTQ"]]))
```

```
## /Users/gfinak/Library/R/3.1/library/RNASeqPipelineR/extdata/140930_M00932_0040_000000000-A20E9_miseq
##
## /Users/gfinak/Library/R/3.1/library/RNASeqPipelineR/extdata/140930_M00932_0040_000000000-A20E9_miseq
##
```

We run fastQC on the fastq files:

```
#specify the number of cores, RNASeqPipelineR uses GNU parallel.
runFastQC(ncores=2)
```

```
## Finished fastqc process for 2 files.
## Expanding archives
```

Generate a little figure summarizing the fastQC output.

```
#summarize FastQC results
summarizeFastQC()
```

Reference Genome

You need a reference genome to align the fastqc files against. Place the reference outside of the project directory and tell the pipeline where that reference genome lives: It is expected in a directory called `Reference_Genome`.

```
ext<-system.file("extdata",package = "RNASeqPipelineR")
utils_dir<-list.files(ext,pattern="Utils",full=TRUE)
list.files(utils_dir) #There it is!
```

```
## [1] "Reference_Genome"
```

```
list.files(list.files(utils_dir,full=TRUE),pattern="*") #directory contents
```

```
##  [1] "hg38.1.bt2"          "hg38.2.bt2"          "hg38.3.bt2"
##  [4] "hg38.4.bt2"          "hg38.chrlist"        "hg38.fa"
##  [7] "hg38.grp"            "hg38.idx.fa"         "hg38.n2g.idx.fa"
## [10] "hg38.rev.1.bt2"      "hg38.rev.2.bt2"      "hg38.seq"
## [13] "hg38.ti"             "hg38.transcripts.fa" "knownIsoforms.txt"
## [16] "UCSC.gtf"
```

```
#Configure RNASeqPipelineR to use the reference genome, which has been built to use rsem.
buildReference(path=utils_dir,gtf_file="UCSC.gtf",fasta_file="hg38.fa",name="hg38")
```

```
## Reference Genome Found
```

Alignment

Finally we are ready to do the alignment. We tell `RSEMCalculateExpression` to use 4 cores and that the data are paired-end. In this example we're only aligning one sample so `parallel_threads` is 1, and `bowtie_threads` is 5 so `bowtie2` will use 5 cores.

```
#Align and compute expression counts
RSEMCalculateExpression(parallel_threads = 1,bowtie_threads = 5,paired=TRUE)
```

Construct an expression matrix

Next we combine the assembled files and build expression matrices. This gets stored in a standard location.

```
#Assemble an expression matrix of counts and tpm and save them to output files.
RSEMAssembleExpressionMatrix()
```

```
## Assembling counts matrix
```

```
## Warning: closing unused connection 5
## (/private/var/folders/jh/x0h3v3pd4dd497g3gtzsm8500000gn/T/RtmpSFIDr3/RNASeqPipelineRExample/FASTQ/ar
```

Annotation

We will annotate the features (genes) using Bioconductor's hg38 UCSC annotations.

```
#Annotate using bioconductor
BioCAnnotate(annotation_library="TxDb.Hsapiens.UCSC.hg38.knownGene",force=FALSE)
```

```
## Annotating transcripts.
## Writing feature data to rsem_fdata.csv
```

Now we prepare the annotations for cell libraries. We copy our annotations into place in the project directory.

```
#Copy our annotations into place
ext<-system.file("extdata",package = "RNASeqPipelineR")
annotation_files<-list.files(path=ext,pattern="*.csv",recursive=TRUE,full=TRUE)
file.copy(from=annotation_files,to=file.path(getConfig()[["subdirs"]][["RAWANNOTATIONS"]],basename(anno
```

```
## [1] TRUE
```

Some code to parse the annotations and annotate the matrix. Eventually this will be standardized.

```
#Prepare phenotypic data (stim / unstim)
raw_annotation_path <- getConfig()[["subdirs"]][["RAWANNOTATIONS"]]
files.csv <- list.files(raw_annotation_path,pattern="csv",full=TRUE)
annotations <- fread(files.csv)
counts<-fread(file.path(getConfig()[["subdirs"]][["RSEM"]],"rsem_count_matrix.csv")) #TODO Need a getCo

# The sample names are the columns of the expression matrix. The first column is the gene_id, we don't
samplenames<-colnames(counts)[-1]

#Some custom code to make use of the annotation file
indices<-vector('numeric',length(samplenames))
for(i in seq_along(annotations$SequencingDate[1:3])){
  indices[which(samplenames%like%(annotations$SequencingDate[1:3])[i])] <- i
}
indices[indices==0] <- 4
df1 <- data.table(samplenames,indices)
annotations$indices<-1:4
```

The two tables we'll be merging:

```
kable(df1)
```

samplenames	indices
140930_M00932_0040_000000000-A20E9_miseq_N712_N508_read1_index_N712_N508	4

```
kable(annotations)
```

SequencingDate	SortDate	cellType	Stim	indices
140408	03/26/14	CD4/+154+	Activated	1
140717	07/08/14	CD4+/CD154-	UnActivated	2
140729	07/09/14	CD4+/CD154+	Activated	3
09/12/14	08/05/14	CD4+/CD154-	Unactivated	4

The sample name column is expected to have the name "srr". In the future we'll make this neater but for now it aligns with the sample annotations when using SRA files.

```
#merge on indices
setkey(df1,indices)
setkey(annotations,indices)
annotations <- annotations[df1]
#set the sample name column to be "srr"
setnames(annotations,"samplenames","srr")
```

Here's what we write to the pdata file:

```
kable(annotations)
```

SequencingDate	SortDate	cellType	Stim	indices	srr
09/12/14	08/05/14	CD4+/CD154-	Unactivated	4	140930_M00932_0040_000000000-A20E9_miseq_

```
write.csv(annotations,file=file.path(getConfig()[["subdirs"]][["RSEM"]],"rsem_pdata.csv"),row.names=FAL
```

Additional analysis

Now we'll run MiTCR to annotate the short reads.

```
#MiTCR, annotate TRA and TRB genes using the homo sapiens database.
MiTCR(gene="TRA", species="hs",pset="flex",output_format="txt",paired=FALSE)
MiTCR(gene="TRB", species="hs",pset="flex",output_format="txt",paired=FALSE)
```

Finally we can easily construct an `ExpressionSet`. We'll use the TPM values from RSEM.

```
# For existing project load and grab data
eset<-getExpressionSet(which="tpm")
```

Reporting and tracking

Generate a report that tells us what software was used.

```
#output version info
pipelineReport()
```

```
## Aspera Connect version 3.5.2.95060
##  ascp version 3.5.1.94907
##  Operating System: MacOSX
##  FIPS 140-2-validated crypto ready to configure
##  AES-NI Supported
##  License max rate=(unlimited), account no.=1, license no.=1
##
##
##  fastq-dump : 2.4.2
##
##
## /usr/local/lib/bowtie2-2.2.4/bowtie2-align-s version 2.2.4
##  64-bit
##  Built on Gregs-MBP.pc.scharp.org
##  Tue Dec  9 16:48:53 PST 2014
##  Compiler: Thread model: posix
##  Options: -O3 -m64 -msse2  -funroll-loops -g3 -DPOPCNT_CAPABILITY
##  Sizeof {int, long, long long, void*, size_t, off_t}: {4, 8, 8, 8, 8, 8}
##
## FastQC v0.11.2
##
```

```
## Current version is RSEM v1.2.19
##
## R version 3.1.2 (2014-10-31)
##  Platform: x86_64-apple-darwin14.0.0 (64-bit)
##
##  locale:
##  [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
##  attached base packages:
##  [1] stats4    parallel  stats     graphics  grDevices utils     datasets
##  [8] methods   base
##
##  other attached packages:
##   [1] plyr_1.8.1
##   [2] pander_0.5.1
##   [3] org.Hs.eg.db_3.0.0
##   [4] RSQLite_1.0.0
##   [5] DBI_0.3.1
##   [6] annotate_1.44.0
##   [7] XML_3.98-1.1
##   [8] TxDb.Hsapiens.UCSC.hg38.knownGene_3.0.0
##   [9] GenomicFeatures_1.18.2
##  [10] AnnotationDbi_1.28.1
##  [11] GenomicRanges_1.18.3
##  [12] GenomeInfoDb_1.2.3
##  [13] IRanges_2.0.0
##  [14] S4Vectors_0.4.0
##  [15] clue_0.3-48
##  [16] RNASeqPipelineR_0.2
##  [17] knitr_1.8
##  [18] GEOquery_2.32.0
##  [19] Biobase_2.26.0
##  [20] BiocGenerics_0.12.1
##  [21] data.table_1.9.4
##
##  loaded via a namespace (and not attached):
##   [1] base64enc_0.1-2         BatchJobs_1.5
##   [3] BBmisc_1.8              BiocParallel_1.0.0
##   [5] biomaRt_2.22.0          Biostrings_2.34.0
##   [7] bitops_1.0-6            brew_1.0-6
##   [9] checkmate_1.5.0         chron_2.3-45
##  [11] cluster_1.15.3          codetools_0.2-9
##  [13] colorspace_1.2-4        digest_0.6.5
##  [15] evaluate_0.5.5          fail_1.2
##  [17] foreach_1.4.2           formatR_1.0
##  [19] GenomicAlignments_1.2.1 ggplot2_1.0.0
##  [21] grid_3.1.2              gtable_0.1.2
##  [23] htmltools_0.2.6         iterators_1.0.7
##  [25] MASS_7.3-35             munsell_0.4.2
##  [27] proto_0.3-10            Rcpp_0.11.3
##  [29] RCurl_1.95-4.5          reshape2_1.4.1
##  [31] rmarkdown_0.3.10        Rsamtools_1.18.2
##  [33] rtracklayer_1.26.2      scales_0.2.4
##  [35] sendmailR_1.2-1         SRAdb_1.20.5
```

```
##  [37] stringr_0.6.2           tools_3.1.2
##  [39] xtable_1.7-4            XVector_0.6.0
##  [41] yaml_2.1.13             zlibbioc_1.12.0
```