

Library analysis output

Tomas Bjorklund

Thu Nov 5 23:24:45 2015

This workflow brings together FastQ files containing barcodes and 5'/3' ends of a suitable insert and alignmen them using Bowtie2. It also includes starcode based false barcode reduction and a MapReduce based hierarchical clustering

```
library(compiler)
enableJIT(3)
```

```
## [1] 0
```

```
suppressPackageStartupMessages(library(knitr))
suppressPackageStartupMessages(library(ShortRead))
```

```
## Note: no visible binding for global variable '.Data'
## Note: no visible binding for global variable '.Data'
## Note: no visible binding for global variable '.Data'
## Note: no visible binding for global variable '.Data'
## Note: no visible binding for global variable '.Data'
## Note: no visible binding for global variable '.Data'
## Note: no visible binding for global variable '.xData'
## Note: no visible binding for global variable '.->tasks'
## Note: no visible binding for global variable '.Data'
## Note: no visible binding for global variable '.->jobname'
## Note: no visible binding for global variable '.->catch.errors'
## Note: no visible binding for global variable '.->stop.on.error'
## Note: no visible binding for global variable '.->progressbar'
## Note: no visible binding for global variable '.->cluster'
## Note: no visible binding for global variable '.->.clusterargs'
## Note: no visible binding for global variable '.->.controlled'
## Note: no visible binding for global variable '.->timeout'
## Note: no visible binding for global variable '.->log'
## Note: no visible binding for global variable '.->logdir'
## Note: no visible binding for global variable '.->resultdir'
## Note: no visible global function definition for 'initFields'
## Note: no visible global function definition for 'callSuper'
## Note: no visible global function definition for 'initFields'
## Note: no visible global function definition for 'callSuper'
## Note: no visible binding for global variable '.refClassDef'
## Note: no visible global function definition for 'callSuper'
## Note: no visible global function definition for 'callSuper'
## Note: no visible binding for global variable '.->tasks'
## Note: no visible binding for global variable '.->jobname'
## Note: no visible binding for global variable '.->catch.errors'
## Note: no visible binding for global variable '.->stop.on.error'
## Note: no visible binding for global variable '.->progressbar'
## Note: no visible binding for global variable '.->cluster'
## Note: no visible binding for global variable '.->.clusterargs'
## Note: no visible binding for global variable '.->.controlled'
## Note: no visible binding for global variable '.->timeout'
## Note: no visible binding for global variable '.->log'
## Note: no visible binding for global variable '.->logdir'
```

[illegible]

```
## Note: no visible binding for global variable 'xp'
## Note: no visible binding for global variable '.link_to_cached_object'
## Note: no visible binding for global variable 'contents'
## Note: no visible binding for global variable 'NAMES'
## Note: no visible binding for global variable 'NAMES'
## Note: no visible binding for global variable 'partitioning'
## Note: no visible binding for global variable 'partitioning'
## Note: no visible binding for global variable 'cnirl'
## Note: no visible binding for global variable 'cnirl'
## Note: no visible binding for global variable 'NAMES'
## Note: no visible binding for global variable 'NAMES'
## Note: no visible binding for global variable 'listData'
## Note: no visible binding for global variable 'listData'
```

```
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(ggbio))
```

```
## Note: no visible binding for global variable 'NAMES'
```

```
suppressPackageStartupMessages(library(beanplot))
suppressPackageStartupMessages(library(parallel))
suppressPackageStartupMessages(library(doParallel))
suppressPackageStartupMessages(library(data.table))
suppressPackageStartupMessages(library(scales)) #Gives the log2 ability to ggplot2
suppressPackageStartupMessages(library(formatR))
suppressPackageStartupMessages(library(BSgenome))
suppressPackageStartupMessages(library(Rsamtools))
suppressPackageStartupMessages(library(rtracklayer))
suppressPackageStartupMessages(library(GenomicFeatures))
```

```
## Note: no visible global function definition for 'winMenuNames'
## Note: no visible global function definition for 'winMenuAdd'
## Note: no visible global function definition for 'winMenuAdd'
## Note: no visible global function definition for 'winMenuAddItem'
```

```
suppressPackageStartupMessages(library(GenomicAlignments))
suppressPackageStartupMessages(library(GenomicRanges))
suppressPackageStartupMessages(library(biovizBase))
suppressPackageStartupMessages(library(Gviz))
suppressPackageStartupMessages(library(plyr))
suppressPackageStartupMessages(library(devtools))
suppressPackageStartupMessages(library(Hmisc))
```

Sequencing files

```
knitr::kable(config, format = "markdown")
```

Parameter	Value
dataDir	.././Shared/NGS\ data/Original\ sequencing\ files/TB20151026-26037026
in.name.P5	psc-lib-1-2UndetOld_S1_L001_R1_001.fastq.gz
in.name.P7	psc-lib-1-2UndetOld_S1_L001_R2_001.fastq.gz
name.out	2015-11-05_AAVlibrary_complete

Parameter	Value
paired.alignment	TRUE
bb.dir	../Templates/adapters/pscAAV_firstLib
fragmentTemplate	../Shared/NGS\ data/bowtieIndices/libIndex
sc.param	0
run.subset	FALSE
align.P7	FALSE
max.cores	32
subset.count	500000

```
dataDir <- config$Value[1]
in.name.P5 <- file.path(dataDir, config$Value[2])
in.name.P7 <- file.path(dataDir, config$Value[3])
name.out <- config$Value[4]
paired.alignment <- as.logical(config$Value[5])
```

Analysis parameters

```
bb.dir <- config$Value[6]
fragmentTemplate <- config$Value[7]
output.table$SC <- config$Value[8]
run.subset <- as.logical(config$Value[9])
align.p7 <- as.logical(config$Value[10])
max.cores <- as.integer(config$Value[11])
subset.count <- as.integer(config$Value[12])
```

Script execution

```
strt<-Sys.time()

id.backbone.L <- file.path(bb.dir, "Ltrim.fa")
id.backbone.R <- file.path(bb.dir, "Rtrim.fa")
id.BC.L <- file.path(bb.dir, "BC-L.fa")
id.BC.R <- file.path(bb.dir, "BC-R.fa")
id.uncut <- file.path(bb.dir, "uncut.fa")
```

Selection of real amplicons

```
out.name.P5 <- tempfile(pattern = "P5_", tmpdir = tempdir(), fileext = ".fastq.gz")
out.name.P7 <- tempfile(pattern = "P7_", tmpdir = tempdir(), fileext = ".fastq.gz")
command.args <- paste("-Xmx12g overwrite=true k=15 rcomp=f skipr2=t qhdist=0 maskmiddle=f hammingdistance=2",
  " in=", in.name.P5,
  " in2=", in.name.P7,
  " outm=", out.name.P5,
  " outm2=", out.name.P7,
  " fliteral=", "GTATGTTGTTCTGGAGCGGGAGGGTGCTATTTGCCTAGCGATAA", sep = " ") #Length 48-72
# postLoxP on P5: GTATGTTGTTCTGGAGCGGGAGGGTGCTATTTGCCTAGCGATAAGCTGATGTAGCC
```

```
# GFP from P7: CCTGCTGGAGTTCGTGACCGCCGCCGGGATCACTCTCGGCATGGACGAGCTGTACAAGTAA
# Cap from P7: AGACAAGCAGCTACCGCAGATGTCAACACACAAGGCGTTCTTCCAGGCATGGTCTGG

sys.out <- system2(path.expand("~/bbmap/bbduk2.sh"), args=command.args, stdout=TRUE, stderr=TRUE) #

sys.out <- as.data.frame(sys.out)

colnames(sys.out) <- c("bbduk2 Identification of real amplicons")
invisible(sys.out[" "] <- " ")
lengthOut <- (nrow(sys.out))
knitr::kable(sys.out[3:lengthOut,], format = "markdown")
```

	bbduk2 Identification of real amplicons
3	
4	BBDuk2 version 34.79
5	Set ORDERED to true
6	Set threads to 32
7	k=15
8	hamming distance=2
9	kfiltering using 1 literal.
10	
11	Initial:
12	Memory: free=12090m, used=258m
13	
14	Added 30721 kmers; time: 0.070 seconds.
15	Memory: free=11639m, used=709m
16	
17	Input is being processed as paired
18	Started output streams: 0.071 seconds.
19	Processing time: 83.207 seconds.
20	
21	Input: 23191088 reads 3490215687 bases.
22	Contaminants: 23095890 reads (99.59%) 3475908371 bases (99.59%)
23	Result: 95198 reads (0.41%) 14307316 bases (0.41%)
24	
25	Time: 83.356 seconds.
26	Reads Processed: 23191k 278.22k reads/sec
27	Bases Processed: 3490m 41.87m bases/sec

```
in.name.P5 <- out.name.P5
in.name.P7 <- out.name.P7
```

Extraction of a subset

```
if (run.subset){
  suppressWarnings(sampler <- FastqSampler(gsub("([\\])", "", in.name.P5), subset.count, readerBlockSize=1e6)
  set.seed(123); tmp.P5 <- yield(sampler)
  in.name.P5 <- tempfile(pattern = "P5_", tmpdir = tempdir(), fileext = ".fastq.gz")
  writeFastq(tmp.P5, in.name.P5, compress=TRUE)
  rm(tmp.P5)
  suppressWarnings(sampler <- FastqSampler(gsub("([\\])", "", in.name.P7), subset.count, readerBlockSize=1e6)
```

```

set.seed(123); tmp.P7 <- yield(sampler)
in.name.P7 <- tempfile(pattern = "P7_", tmpdir = tmpdir(), fileext = ".fastq.gz")
writeFastq(tmp.P7, in.name.P7, compress=TRUE)
rm(tmp.P7)
}

output.table$Reads <- as.integer(system(paste("gunzip -c ", shQuote(gsub("([\\])", "", in.name.P5)),
      " | echo $((`wc -l`/4)) 2>&1", sep = ""), intern = TRUE,
      ignore.stdout = FALSE)) #Stores the read count utilized
print(paste("Utilized sequences:", output.table$Reads[1]))

```

```
[1] "Utilized sequences: 11547945"
```

Extraction of barcodes

```

out.name.P5 <- tempfile(pattern = "P5_", tmpdir = tmpdir(), fileext = ".fastq.gz")
out.name.P7 <- tempfile(pattern = "P7_", tmpdir = tmpdir(), fileext = ".fastq.gz")
command.args <- paste("-Xmx12g overwrite=true k=10 rcomp=f skipr2=t qhdist=0 maskmiddle=t hammingdistance=1
      " in=", in.name.P5,
      " in2=", in.name.P7,
      " outm=", out.name.P5,
      " outm2=", out.name.P7,
      " fliteral=", "ATAACTTCGTATAATGTATGC", sep = "") #Length 48-72 bp k=18 mink=10 qhdist=1

sys.out <- system2(path.expand("~/bbmap/bbduk2.sh"), args=command.args, stdout=TRUE, stderr=TRUE) #

sys.out <- as.data.frame(sys.out)

colnames(sys.out) <- c("bbduk2 Identification of real barcodes")
invisible(sys.out[" "] <- " ")
lengthOut <- (nrow(sys.out))
knitr::kable(sys.out[3:lengthOut,], format = "markdown")

```

	bbduk2 Identification of real barcodes
3	
4	BBDuk2 version 34.79
5	Set ORDERED to true
6	Set threads to 32
7	k=10
8	maskMiddle=true
9	hamming distance=1
10	kfiltering using 1 literal.
11	
12	Initial:
13	Memory: free=12090m, used=258m
14	
15	Added 336 kmers; time: 0.027 seconds.
16	Memory: free=11639m, used=709m
17	
18	Input is being processed as paired
19	Started output streams: 0.026 seconds.
20	Processing time: 81.580 seconds.

	bbduk2 Identification of real barcodes
--	--

```

21
22  Input: 23095890 reads 3475908371 bases.
23  Contaminants: 23057674 reads (99.83%) 3470622796 bases (99.85%)
24  Result: 38216 reads (0.17%) 5285575 bases (0.15%)
25
26  Time: 81.639 seconds.
27  Reads Processed: 23095k 282.90k reads/sec
28  Bases Processed: 3475m 42.58m bases/sec

```

```

in.name.P5 <- out.name.P5
in.name.P7 <- out.name.P7

out.name.P5 <- tempfile(pattern = "BC_", tmpdir = tempdir(), fileext = ".fastq.gz")

sys.out <- system(paste("~/bbmap/bbduk2.sh overwrite=true k=15 mink=15 hammingdistance=1 findbestmatch=t ",
  "rcomp=f findbestmatch=f qhdist=0 minavgquality=0 maxns=0 minlength=18 ",
  "maxlength=22 threads=", detectCores(), " in=", shQuote(in.name.P5),
  " out=", out.name.P5, " lliteral=", "GGCCTAGCGCCGCTTTACTT",
  " rliteral=", "ATAACTTCGTATAATGTATGC",
  " 2>&1", sep = ""), intern = TRUE, ignore.stdout = FALSE) ## fliteral=",id.uncut,
sys.out <- as.data.frame(sys.out)

in.name.P5 <- out.name.P5

colnames(sys.out) <- c("bbduk2 Extraction of barcodes")
invisible(sys.out[" "] <- " ")
lengthOut <- (nrow(sys.out))
knitr::kable(sys.out[3:lengthOut,], format = "markdown")

```

	bbduk2 Extraction of barcodes
--	-------------------------------

```

3
4  BBDuk2 version 34.79
5  Set threads to 32
6  k=15
7  maskMiddle=true
8  hamming distance=1
9  right-ktrimming using 1 literal.
10 left-ktrimming using 1 literal.
11
12 Initial:
13 Memory: free=77523m, used=2507m
14
15 Added 301 kmers; time: 0.030 seconds.
16 Memory: free=74600m, used=5430m
17
18 Added 301 kmers; time: 0.002 seconds.
19 Memory: free=74182m, used=5848m
20
21 Input is being processed as unpaired
22 Started output streams: 0.022 seconds.
23 Processing time: 30.369 seconds.
24

```

	bbduk2 Extraction of barcodes
<hr/>	
25	Input: 11528837 reads 1735623565 bases.
26	KTrimmed: 22982216 reads (199.35%) 1501448690 bases (86.51%)
27	Low quality discards: 4 reads (0.00%) 80 bases (0.00%)
28	Result: 11280473 reads (97.85%) 225471114 bases (12.99%)
29	
30	Time: 30.440 seconds.
31	Reads Processed: 11528k 378.74k reads/sec
32	Bases Processed: 1735m 57.02m bases/sec

```
rm(sys.out)
```

```
reads.BC <- readFastq(in.name.P5)
```

Note: no visible binding for global variable 'xp_list'

Note: no visible binding for global variable '.link_to_cached_object_list'

Note: no visible binding for global variable 'NAMES'

Note: no visible binding for global variable 'pool'

Note: no visible binding for global variable 'pool'

```
sread(reads.BC)
```

```
A DNASTringSet instance of length 11280473
      width seq
[1]      20 CTACCAGCCCAGAATGACGG
[2]      20 GTAGGCGCCACGAACGATG
[3]      20 GATTCTGGCCGGATGCACTC
[4]      20 GCTCAAAGGCTTACTGATTC
[5]      20 GAGGCCGTGCAGGTGTGTGC
...      ...
[11280469] 20 CTGGGTCCATTTCATTCGCGT
[11280470] 20 ATGCGTGGGCATAAAGGAAC
[11280471] 20 GTGCCAGGAATTAAGCCAGC
[11280472] 20 CCGCGCGTCCTCGAACAATT
[11280473] 20 GCAGGTGGCCATATAACAAT
```

```
output.table$OrigBC <- length(unique(sread(reads.BC)))
unique(sread(reads.BC))
```

```
A DNASTringSet instance of length 3904547
      width seq
[1]      20 CTACCAGCCCAGAATGACGG
[2]      20 GTAGGCGCCACGAACGATG
[3]      20 GATTCTGGCCGGATGCACTC
[4]      20 GCTCAAAGGCTTACTGATTC
[5]      20 GAGGCCGTGCAGGTGTGTGC
...      ...
[3904543] 20 CCACATTGACGCATGGGCAT
[3904544] 20 GTGCACTCACGGGATGATTG
[3904545] 20 CTGGGTCCATTTCATTCGCGT
[3904546] 20 GTGCCAGGAATTAAGCCAGC
[3904547] 20 GCAGGTGGCCATATAACAAT
```



```
barcodeTable <- data.table(ID=as.character(ShortRead::id(reads.BC)), BC=as.character(sread(reads.BC)))

##"CATTACGCGCTCGCGTAAGC" %in% names(frag.ranges.matched)
```

Extraction of fragments

```
out.name.P7 <- tempfile(pattern = "P7_", tmpdir = tempdir(), fileext = ".fastq.gz")
command.args <- paste("-Xmx12g overwrite=true k=10 mink=18 rcomp=f qhdist=0 maskmiddle=t hammingdistance=1",
  " in=", in.name.P7,
  " out=", out.name.P7,
  " lliteral=", "AGCAACCTCCAGAGAGGCAAC",
  " rliteral=", "AGACAAGCAGCTACCGCAGATGTCAACACACAAGGCGTTCTTCCAGGCATGGTCTGG", sep = " ")

sys.out <- system2(path.expand("~/bbmap/bbduk2.sh"), args=command.args, stdout=TRUE, stderr=TRUE) #

sys.out <- as.data.frame(sys.out)

colnames(sys.out) <- c("bbduk2 Identification of real amplicons")
invisible(sys.out[" "] <- " ")
lengthOut <- (nrow(sys.out))
knitr::kable(sys.out[3:lengthOut,], format = "markdown")
```

	bbduk2 Identification of real amplicons
3	
4	BBduk2 version 34.79
5	Set ORDERED to true
6	Set threads to 32
7	k=10
8	maskMiddle=true
9	hamming distance=1
10	right-ktrimming using 1 literal.
11	left-ktrimming using 1 literal.
12	
13	Initial:
14	Memory: free=11959m, used=389m
15	
16	Added 1344 kmers; time: 0.045 seconds.
17	Memory: free=11508m, used=840m
18	
19	Added 336 kmers; time: 0.002 seconds.
20	Memory: free=11508m, used=840m
21	
22	Input is being processed as unpaired
23	Started output streams: 0.024 seconds.
24	Processing time: 32.801 seconds.
25	
26	Input: 11528837 reads 1734999231 bases.
27	KTrimmed: 21791188 reads (189.01%) 1271427101 bases (73.28%)
28	Result: 8383822 reads (72.72%) 448218109 bases (25.83%)
29	
30	Time: 32.884 seconds.
31	Reads Processed: 11528k 350.59k reads/sec
32	Bases Processed: 1734m 52.76m bases/sec

```

in.name.P7 <- out.name.P7

source("retrieveFASTAQID.R")

FastQ1 <- readFastq(out.name.P5)
FastQ2 <- readFastq(out.name.P7)
FastQ1ID <- retrieveFASTAQID(FastQ1, PE=TRUE)
FastQ2ID <- retrieveFASTAQID(FastQ2, PE=TRUE)

hits <- intersect(FastQ2ID,FastQ1ID)

FastQ1Subset <- FastQ1[match(hits,FastQ1ID)]
FastQ2Subset <- FastQ2[match(hits,FastQ2ID)]

system(paste("mv ", out.name.P7, " ./data/fragments_", name.out, ".fastq.gz", sep=""))
system(paste("mv ", out.name.P5, " ./data/barcodes_", name.out, ".fastq.gz", sep=""))

unlink(paste(tempdir(), "/*", sep = ""), recursive = FALSE, force = FALSE) #Cleanup of temp files

print("Total execution time:")

```

```
[1] "Total execution time:"
```

```
print(Sys.time()-strt)
```

Time difference of 9.534845 mins

```
devtools::session_info()
```

Session info -----

```

setting  value
version  R version 3.2.2 (2015-08-14)
system   x86_64, linux-gnu
ui        X11
language (EN)
collate  en_US.UTF-8
tz        <NA>
date     2015-11-05

```

Packages -----

package	* version	date	source
acepack	1.3-3.3	2014-11-24	CRAN (R 3.2.0)
AnnotationDbi	* 1.30.1	2015-09-04	Bioconductor
beanplot	* 1.2	2014-09-19	CRAN (R 3.2.0)
Biobase	* 2.28.0	2015-04-21	Bioconductor
BiocGenerics	* 0.14.0	2015-04-21	Bioconductor
BiocParallel	* 1.2.20	2015-09-04	Bioconductor
biomaRt	2.24.0	2015-04-21	Bioconductor
Biostrings	* 2.36.4	2015-09-04	Bioconductor
biovizBase	* 1.16.0	2015-04-21	Bioconductor

bitops	1.0-6	2013-08-17	CRAN (R 3.2.0)
BSgenome	* 1.36.3	2015-09-04	Bioconductor
chron	2.3-47	2015-06-24	CRAN (R 3.2.2)
cluster	2.0.3	2015-07-21	CRAN (R 3.2.2)
codetools	0.2-14	2015-07-15	CRAN (R 3.2.2)
colorspace	1.2-6	2015-03-11	CRAN (R 3.2.0)
data.table	* 1.9.4	2014-10-02	CRAN (R 3.2.0)
DBI	0.3.1	2014-09-24	CRAN (R 3.2.0)
devtools	* 1.9.1	2015-09-11	CRAN (R 3.2.2)
dichromat	2.0-0	2013-01-24	CRAN (R 3.2.0)
digest	0.6.8	2014-12-31	CRAN (R 3.2.0)
doParallel	* 1.0.8	2014-02-28	CRAN (R 3.2.0)
evaluate	0.7.2	2015-08-13	CRAN (R 3.2.2)
foreach	* 1.4.2	2014-04-11	CRAN (R 3.2.0)
foreign	0.8-66	2015-08-19	CRAN (R 3.2.2)
formatR	* 1.2	2015-04-21	CRAN (R 3.2.0)
Formula	* 1.2-1	2015-04-07	CRAN (R 3.2.0)
futile.logger	1.4.1	2015-04-20	CRAN (R 3.2.0)
futile.options	1.0.0	2010-04-06	CRAN (R 3.2.0)
GenomeInfoDb	* 1.4.2	2015-09-04	Bioconductor
GenomicAlignments	* 1.4.1	2015-09-04	Bioconductor
GenomicFeatures	* 1.20.3	2015-09-04	Bioconductor
GenomicRanges	* 1.20.6	2015-09-04	Bioconductor
GGally	0.5.0	2014-12-02	CRAN (R 3.2.0)
ggbio	* 1.16.1	2015-09-04	Bioconductor
ggplot2	* 1.0.1	2015-03-17	CRAN (R 3.2.0)
graph	1.46.0	2015-04-21	Bioconductor
gridExtra	2.0.0	2015-07-14	CRAN (R 3.2.2)
gtable	0.1.2	2012-12-05	CRAN (R 3.2.0)
Gviz	* 1.12.1	2015-09-04	Bioconductor
highr	0.5	2015-04-21	CRAN (R 3.2.0)
Hmisc	* 3.16-0	2015-04-30	CRAN (R 3.2.2)
htmltools	0.2.6	2014-09-08	CRAN (R 3.2.0)
hwriter	1.3.2	2014-09-10	CRAN (R 3.2.0)
IRanges	* 2.2.7	2015-09-04	Bioconductor
iterators	* 1.0.7	2014-04-11	CRAN (R 3.2.0)
knitr	* 1.11	2015-08-14	CRAN (R 3.2.2)
lambda.r	1.1.7	2015-03-20	CRAN (R 3.2.0)
lattice	* 0.20-33	2015-07-14	CRAN (R 3.2.2)
latticeExtra	0.6-26	2013-08-15	CRAN (R 3.2.0)
magrittr	1.5	2014-11-22	CRAN (R 3.2.2)
MASS	7.3-44	2015-08-30	CRAN (R 3.2.2)
matrixStats	0.14.2	2015-06-24	CRAN (R 3.2.2)
memoise	0.2.1	2014-04-22	CRAN (R 3.2.2)
munsell	0.4.2	2013-07-11	CRAN (R 3.2.0)
nnet	7.3-11	2015-08-30	CRAN (R 3.2.2)
OrganismDbi	1.10.0	2015-04-21	Bioconductor
plyr	* 1.8.3	2015-06-12	CRAN (R 3.2.2)
proto	0.3-10	2012-12-22	CRAN (R 3.2.0)
RBGL	1.44.0	2015-04-21	Bioconductor
RColorBrewer	1.1-2	2014-12-07	CRAN (R 3.2.0)
Rcpp	0.12.0	2015-07-25	CRAN (R 3.2.2)
RCurl	1.95-4.7	2015-06-30	CRAN (R 3.2.2)
reshape	0.8.5	2014-04-23	CRAN (R 3.2.0)
reshape2	1.4.1	2014-12-06	CRAN (R 3.2.0)
rmarkdown	0.8	2015-08-30	CRAN (R 3.2.2)
rpart	4.1-10	2015-06-29	CRAN (R 3.2.2)

Rsamtools	* 1.20.4	2015-09-04	Bioconductor
RSQLite	1.0.0	2014-10-25	CRAN (R 3.2.0)
rtracklayer	* 1.28.10	2015-09-04	Bioconductor
S4Vectors	* 0.6.5	2015-09-04	Bioconductor
scales	* 0.3.0	2015-08-25	CRAN (R 3.2.2)
ShortRead	* 1.26.0	2015-04-21	Bioconductor
stringi	0.5-5	2015-06-29	CRAN (R 3.2.2)
stringr	1.0.0	2015-04-30	CRAN (R 3.2.2)
survival	* 2.38-3	2015-07-02	CRAN (R 3.2.2)
VariantAnnotation	1.14.13	2015-09-04	Bioconductor
XML	3.98-1.3	2015-06-30	CRAN (R 3.2.2)
XVector	* 0.8.0	2015-04-21	Bioconductor
yaml	2.1.13	2014-06-12	CRAN (R 3.2.0)
zlibbioc	1.14.0	2015-04-21	Bioconductor