

ExomeDepth

Vincent Plagnol

February 20, 2012

Contents

1	Create count data from BAM files	1
2	Load an example dataset	2
3	Build the most appropriate reference set	2
4	CNV calling	3

1 Create count data from BAM files

The function `getBamCounts` in `ExomeDepth` is set up to parse the BAM files and generate an array of read count, stored in a `GenomicRanges` object/ It is a wrapper around the function `countBamInGRanges.exomeDepth` which is derived from an equivalent function in the `exomeCopy` package. You can refer to the help page of `getBAMCounts` to obtain the full list of options. `getBAMCounts` creates an object of the `GRanges` class which can easily be converted into a data frame or a matrix, which is the preferred format for `ExomeDepth`.

An example of `GenomicRanges` output generated by `getBAMCounts` is provided in this package (chromosome 1 only to keep the size manageable). Here is how this object could for example be used to obtain a more generic data frame:

```
> library(ExomeDepth)
```

```
Package aod, version 1.2
```

```
> data(ExomeCount)
> ExomeCount <- as(ExomeCount[, colnames(ExomeCount)], 'data.frame')
> ExomeCount$chromosome <- gsub(as.character(ExomeCount$space),
+                               pattern = 'chr',
+                               replacement = '') ##remove the annoying chr letters
> print(head(ExomeCount))
```

	space	start	end	width	names	GC
1	1	12012	12058	47	DDX11L10-201_1	0.6041667
2	1	12181	12228	48	DDX11L10-201_2	0.4897959
3	1	12615	12698	84	DDX11L10-201_3	0.5882353
4	1	12977	13053	77	DDX11L10-201_4	0.6025641
5	1	13223	13375	153	DDX11L10-201_5	0.5909091
6	1	13455	13671	217	DDX11L10-201_6	0.5871560

	camfid.032KA_sorted_unique.bam	camfid.033ahw_sorted_unique.bam
1	0	0
2	0	0
3	118	242
4	198	48
5	516	1112
6	272	762

	camfid.035if_sorted_unique.bam	camfid.034pc_sorted_unique.bam	chromosome
1	0	0	1
2	0	0	1
3	116	170	1
4	104	118	1
5	530	682	1
6	336	372	1

Note that to facilitate the generation of read count data the exon positions are available within `ExomeDepth`. This `exons.hg19` data frame can be directly passed as an argument of `getBAMCounts`.

```
> data(exons.hg19)
> print(head(exons.hg19))
```

	chromosome	start	end	name
1	1	12011	12058	DDX11L10-201_1
2	1	12180	12228	DDX11L10-201_2
3	1	12614	12698	DDX11L10-201_3
4	1	12976	13053	DDX11L10-201_4
5	1	13222	13375	DDX11L10-201_5
6	1	13454	13671	DDX11L10-201_6

2 Load an example dataset

We have already loaded a dataset of chromosome 1 data for four exome samples. We run a first test to make sure that the model can be fitted properly. Note the use of the `subset.for.speed` option that subsets some rows purely to speed up this computation.

```
> test <- new('ExomeDepth',
+           test = ExomeCount$camfid.033ahw_sorted_unique.bam,
+           reference = ExomeCount$camfid.035if_sorted_unique.bam,
+           formula = 'cbind(test, reference) ~ 1',
+           subset.for.speed = seq(1, nrow(ExomeCount), 100))
> show(test)
```

```
Number of data points: 256
Formula: cbind(test, reference) ~ 1
Phi parameter: 0.03381291
Likelihood computed
```

3 Build the most appropriate reference set

Moving on toward a more useful computation, the first step is to select the most appropriate reference sample. This step is demonstrated below.

```
> my.test <- ExomeCount$camfid.034pc_sorted_unique.bam
> my.ref.samples <- c('camfid.032KA_sorted_unique.bam', 'camfid.033ahw_sorted_unique.bam', 'camfid.035if_s
> my.reference.set <- as.matrix(ExomeCount[, my.ref.samples])
> my.choice <- select.reference.set (test.counts = my.test,
+                                   reference.counts = my.reference.set,
+                                   bin.length = (ExomeCount$end - ExomeCount$start)/1000,
+                                   n.bins.reduced = 10000)
> print(my.choice[[1]])

[1] "camfid.033ahw_sorted_unique.bam" "camfid.032KA_sorted_unique.bam"
[3] "camfid.035if_sorted_unique.bam"
```

Using the output of this procedure we can construct the reference set.

```
> my.reference.selected <- apply(X = as.matrix( ExomeCount[, my.choice$reference.choice] ),
+                               MAR = 1,
+                               FUN = sum)
```

4 CNV calling

Now the following step is the longest one as the beta-binomial model is applied to the full set of exons:

```
> all.exons <- new('ExomeDepth',
+                 test = my.test,
+                 reference = my.reference.selected,
+                 formula = 'cbind(test, reference) ~ 1')
```

We can now call the CNV by running the underlying hidden Markov model:

```
> all.exons <- CallCNVs(x = all.exons,
+                      transition.probability = 10^-4,
+                      chromosome = ExomeCount$space,
+                      start = ExomeCount$start,
+                      end = ExomeCount$end,
+                      name = ExomeCount$names)
```

Number of hidden states: 3

Number of data points: 25592

Initializing the HMM

Done with the first step of the HMM, now running the trace back

Total number of calls: 20

```
> print(head(all.exons@CNV.calls))
```

	start.p	end.p	type	nexons	start	end	chromosome
1	24	25	deletion	2	89297	91106	1
2	50	64	deletion	15	324290	523834	1
3	552	553	deletion	2	1569583	1570002	1
4	564	568	deletion	5	1592941	1603069	1
5	2259	2262	deletion	4	12976452	12980570	1
6	2297	2301	duplication	5	13328198	13352741	1

	id	BF	reads.expected	reads.observed	reads.ratio
1	chr1:89297-91106	6.53	112	34	0.304
2	chr1:324290-523834	13.50	380	190	0.500
3	chr1:1569583-1570002	5.57	68	24	0.353
4	chr1:1592941-1603069	14.10	1136	434	0.382
5	chr1:12976452-12980570	12.20	780	342	0.438
6	chr1:13328198-13352741	11.30	263	524	1.990