# CNV-segHMM

December 28, 2010

## 1 Introduction

CNV-segHMM is an implementation of the two-stage Hidden Markov Model approach for CNV-seq data analysis. The first stage HMM utilizes sliding window based read count-ratios to estimate the location of candidate CNV regions. The second stage will examine each candidate CNV boundaries from the first stages separately. Each individual mapped reads around the candidate boundaries will be considered, thus the resolution could theoretically reach the average distance between two adjacently mapped reads.

## 2 Installation

CNV-segHMM should work on any modern Linux or UNIX operating systems, but we only tested the package under the following configuration:

- Ubuntu (10.04)
- Perl (5.10.1)
- R (2.11.1)

Several R packages are required, which can be installed from within R by typing:

```
install.packages(c(‘‘RHmm’’, ‘‘ggplot2’’))
```

To install CNV-segHMM, you simply download the package from `http://code.google.com/p/cnv-seghmm` and put the two files (`segHMM.pl` and `lib.segHMM.r`) in the same directory.

## 3 Input Format

You need prepare two input files for CNV-segHMM — the best hit location files for one test and one reference sample. (If you only have one sample, you can try to simulate a set of sequencing data using the same sequencing and mapping method as your real data). The format for the best hit location file is very simple:

1. Each line represent the location of one mapped read
2. Each line must contain a chromosome id followed by the mapped position, separated by a TAB. No extra spaces are allowed
3. The mapped locations must be sorted, from beginning of a chromosome to the end
4. Only one chromosome is allowed for each input file

For example:

```
22 14430318
22 14430323
22 14430327
22 14430344
22 14430346
22 14430364
22 14430367
22 14430381
```

There are two sample data files available at the project download page: `KB1.chr22.hits.bz2` and `ABT.chr22.hits.bz2`. The are the best hit map locations on chromosome 22 from two recently sequenced Bushmen genome[**?**].

## 4 Usage

You can get the usage of `segHMM.pl` by run the script without any argument:

```
$ PATH-TO/segHMM.pl
usage: segHMM.pl [options]
### for stage 1 ###
```

```
--test = test.hits.file
    (sorted, only one chromosome)
--ref = ref.hits.file
    (sorted, only one chromosome)
--log2-threshold = number
   (default=0.6)
--p-value = number
   (default=1e-05)
--bigger-window = number
   (default=1.5)
--minimum-windows-required = number
   (default=3)
--cw    (default; simple consecutive window annotation)
--no-cw
--hmm   (default; HMM segmentation)
--no-hmm
--cbs     (CBS segmentation)
--no-cbs  (default)
### for stage 2 ###
--refine = stage.1.log
   (all other options will be overwritten)
### others ###
--myR = path to your R program
   (default: "/usr/bin/env R")
--help
```

## 4.1 Stage 1

Two of the parameters are required: `--test` and `--ref`, which are the best hit location files for the two samples to be compared. All other parameters have default values.

## 4.2 Stage 2

Only one parameter is required for the second stage: `--refine`, which is the path to the log file (with file extension of `*.log`) created by the first stage.

# 5 Tutorial

(Download the sample data from the project home page)

## 5.1 Stage 1

```
$ ~/cnv/segHMM/segHMM.pl --test KB1.chr22.hits --ref ABT.chr22.hits
... minimum reads for log2>= 0.6 should be 715(test) and 188(ref)
... minimum reads for log2<=-0.6 should be 448(test) and 118(ref)
... each window should contain, after X 1.5:
   minimum 1072 test reads or minimum 282 ref reads
check out the log for a list of output:
KB1.chr22.hits-vs-ABT.c...g2-0.6.pvalue-1e-05.minw-3.cw.hmm.log
Loading required package: RHmm
Loading required package: MASS
Loading required package: nlme
Loading required package: plyr
Simple consecutive windows ...
found 35 CNV segments
HMM-ing ...
found 59 CNV segments
```

This command will produce several files:

```
KB1.chr22.hits-vs-...log2-0.6.pvalue-1e-05.minw-3.cw.hmm.cnv.raw
KB1.chr22.hits-vs-...log2-0.6.pvalue-1e-05.minw-3.cw.hmm.count
KB1.chr22.hits-vs-...log2-0.6.pvalue-1e-05.minw-3.cw.hmm.cw
KB1.chr22.hits-vs-...log2-0.6.pvalue-1e-05.minw-3.cw.hmm.hmm.rough
KB1.chr22.hits-vs-...log2-0.6.pvalue-1e-05.minw-3.cw.hmm.log
```

Do not rename them yet, because they are required by the second stage. The `*.count` file contains the raw read counts. The `*.cnv.raw` file contains all information produced in the first stage, some of which will be used when plotting the predicted CNVs later. The `*.cw` file contains CNV regions detected based on consecutive windows. The file `*.hmm.rough` contains the candidate CNV regions detected by the first stage HMM. The regions in this file will be refined in the second stage HMM. The file `*.log` contains a rough description of the files produced by CNV-segHMM.

### 5.2 Stage 2

To refine the rough CNV candidate detected by the first stage:

```
$ PATH-TO/segHMM.pl --refine KB1.chr22.hits-vs-...minw-3.cw.hmm.log
44 1 21300145 21324779 24635 2
33 1 21361985 21420338 58354 1
40 1 21979496 21991802 12307 1.58496250072116
15 1 21998053 22019959 21907 0.584962500721156
Warning message:
In hmm.bound("...hmm.refined.cand/16.cand",  :
  cannot determine bound reliably, check the *.unreliable file
```

Most likely you will see Warning messages like the above. That is because the second stage cannot reliably detect the boundaries for some candidate CNV regions. Those regions are not refined by the second stage, and stored with the original boundaries in the file:

```
KB1.chr22.hits-vs-ABT.chr22.hits.log2-0.6.pvalue-
                  1e-05.minw-3.cw.hmm.hmm.unreliable
```

Check the candidate regions in the *.unreliable file together with raw data carefully. Some of the regions are in low complexity regions and thus should not be classified as CNV, but some of them are part of real CNV broken into pieces by the first stage. At this moment, CNV-segHMM will not try to make over-confident predictions.

Candidate CNV regions that are refined in this stage are stored in the file:

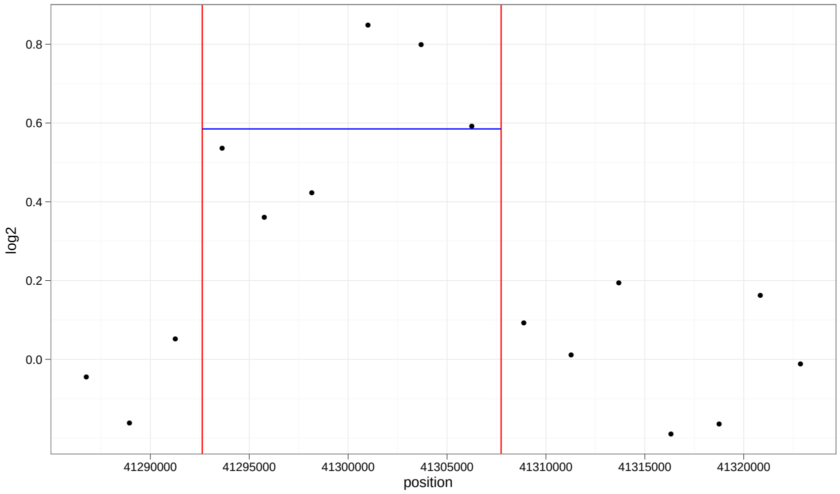
```
KB1.chr22.hits-vs-ABT.chr22.hits.log2-0.6.pvalue-
                     1e-05.minw-3.cw.hmm.hmm.refined
```

### 5.3 Plotting

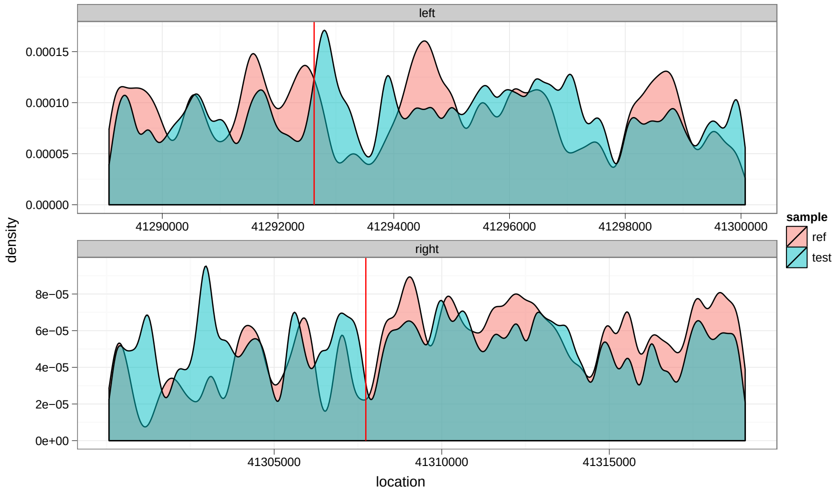
Plotting functions are provided by the R functions. Enter the following commands in R:

```
# load the R functions
source(’~/cnv/segHMM/lib.segHMM.r’)
```

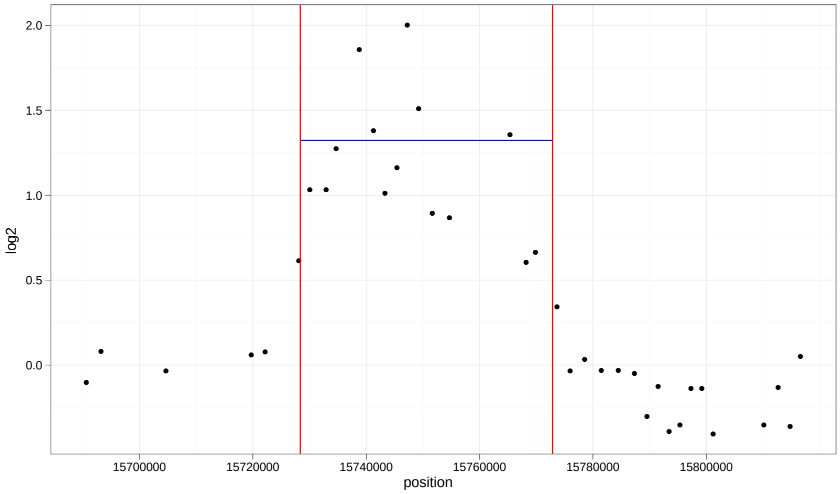
```
# load the refined CNV list
cnv <- read.delim('KB1.chr22.hits-vs-ABT.chr22.hits.log2-
               0.6.pvalue-1e-05.minw-3.cw.hmm.hmm.refined')
# load window-level raw data
raw <- read.delim('KB1.chr22.hits-vs-ABT.chr22.hits.log2-
                 0.6.pvalue-1e-05.minw-3.cw.hmm.cnv.raw')
# specify directory for read-level raw data
cand <- 'KB1.chr22.hits-vs-ABT.chr22.hits.log2-
            0.6.pvalue-1e-05.minw-3.cw.hmm.hmm.refined.cand'
# each CNV has one ID ...
plot.cnv(raw, cnv, 25)
```



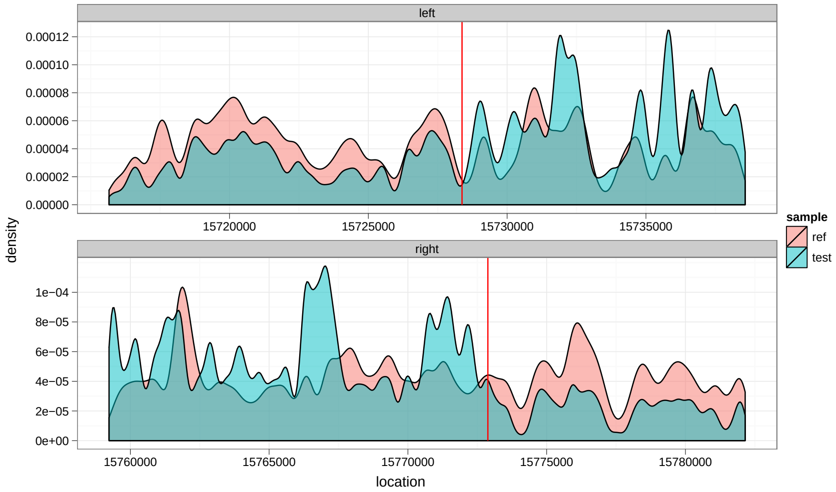
```
plot.bound(cand, cnv, 25)
```



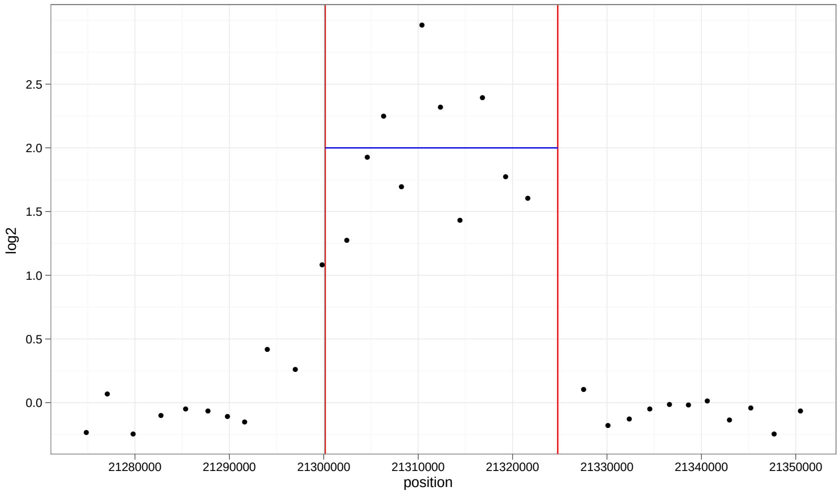
```
plot.cnv(raw, cnv, 37)
```



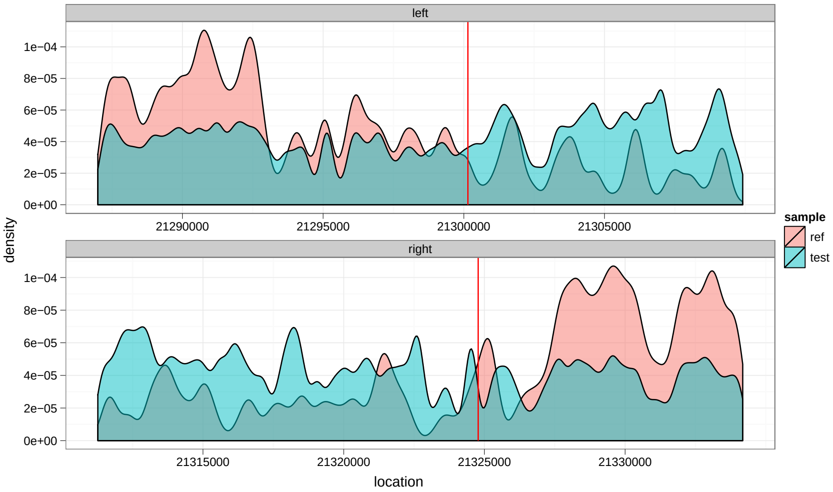
```
plot.bound(cand, cnv, 37)
```



```
plot.cnv(raw, cnv, 44)
```



```
plot.bound(cand, cnv, 44)
```


left
right
density
location
1e−04
8e−05
6e−05
4e−05
2e−05
0e+00
21290000
21295000
21300000
21305000
21315000
21320000
21325000
21330000
sample
ref
test