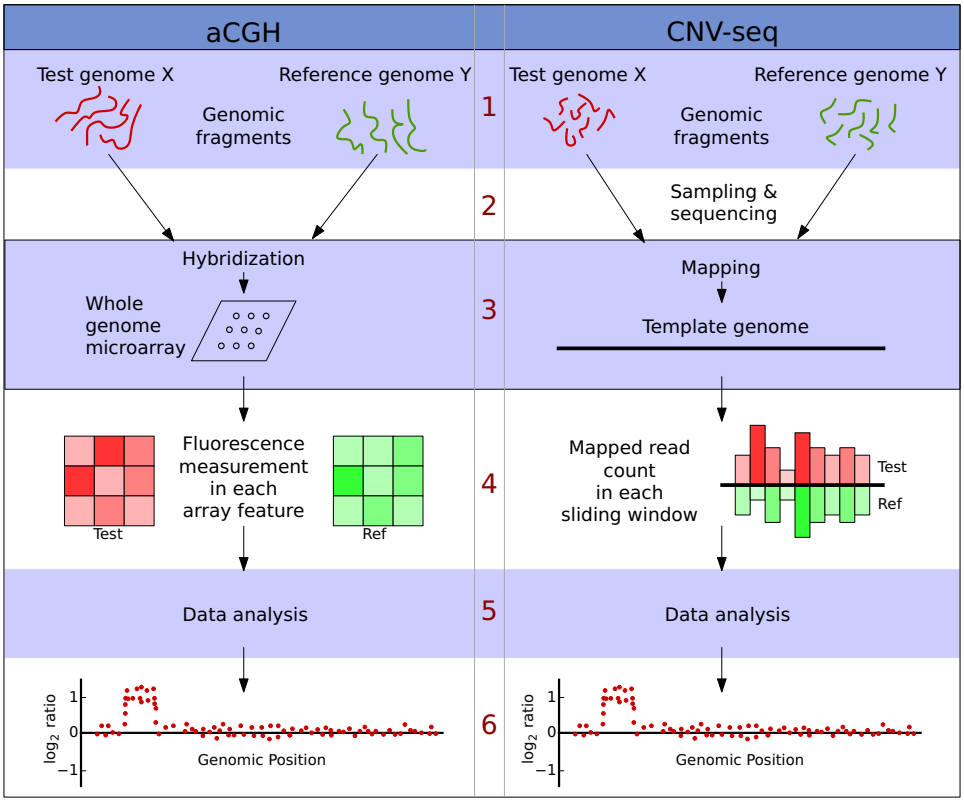
CNV-seq Manual

Xie Chao*

May 26, 2011

1 Introduction



CNV-seq is a method for detecting DNA copy number variation (CNV) using high-throughput sequencing, described by [1]. This method provides a statistical framework to estimate parameters (more details can be found in [1]). The package here is an

*xie at nus.edu.sg or xiechaos at gmail.com

implementation of the CNV-seq method. We tested this package in the following configurations:

```
OS: Mac OS X leopard, Ubuntu Hardy, CentOS 4.3, Debian 5
Perl: 5.8.8, 5.10.0
R: 2.10.1
ggplot2: 0.8.5
```

2 Install

This package contains two Perl scripts and one R package. To use the CNV-seq package, you need to install Perl and R first. Then you can download the CNV-seq package from http://tiger.dbs.nus.edu.sg/CNV-seq. After downloading the package, open a command line console and run:

```
$ tar xzf cnv-seq.tar.gz
$ cd cnv-seq
$ ls
```

There are several Perl scripts (**best-hit.*.pl** and **cnv-seq.pl**) and one R package **cnv** in this directory. To install the two Perl scripts, simply move or copy the them to your desired location. To install the R package run:

```
$ R CMD INSTALL cnv/
```

Please note that the R package **cnv** requires package **ggplot2** for plotting CNV graphs, you may want to install **ggplot2** if you need the plotting function in the **cnv** package.

3 Usage

3.1 best-hit.*.pl

The only requirement for CNV-seq is best-hit location files for each mapped sequence read, in the following format:

```
1       1234
1       23456
chr2    999
chr2    999
X       1234
```

We provide **best-hit.*.pl** for obtaining the best mapping locations for several alignment tools. Currently it only support BLAT psl file and SOLiD matching pipeline (in corona_lite) output as input. To extract map locations from a *.bam file (by SAM tools), try this command:

```
samtools view -F 4 my.bam | perl -lane 'print "$F[2]\t$F[3]"' > my.hits
```

More input formats from various alignment or mapping programs will be included in the future. You need to generate your own best-hit file if you are using other alignment tools.

3.2 cnv-seq.pl

cnv-seq.pl is used to calculate sliding window size, to count number of mapped hits in each window, and to call **cnv** R package to calculate log2 ratios and annotate CNV. You can get the usage of **cnv-seq.pl** by run the script without any arguments:

```
$ ./cnv-seq.pl
  usage: cnv-seq.pl [options]
  --test = test.hits.file
  --ref = ref.hits.file
  --log2-threshold = number
        (default=0.6)
  --p-value = number
        (default=0.001)
  --bigger-window = number
        (default=2, in order to use larger window than minimum)
  --genome = (human, chicken, chrom1, autosome, sex, chromX, chromY)
        (higher priority than --genome-size)
  --genome-size = number
        (in bases, overwiten by --genome option)
  --window-size = number
        (overwrite window size calculation)
  --global-normalization
        (if used, normalization on whole genome,
         instead of on each chromosome)
  --annotate
  --no-annotate
```

```
      (default do annotation)
--minimum-windows-required = number
      (default=4; only for annotation of CNV)
--Rexe = path to your R program
      (default=R)
--help
```

The option `--test` and `--ref` are required. Either `--genome` or `--genome-size` option must be specified too. All other options have default values as shown above. The "test" and "ref" options accept the best hit files outputed from **`best-hit.*.pl`** for the test and reference individuals.

3.3 R package cnv

The **`cnv-seq.pl`** will call the R package **cnv** by default, and a tab-delimited file containing the log2 ratios and (optionally) CNV annotation will be ouputed. However, in order to achieve the full power of the **cnv** package, you are strongly recommended to run the **cnv** package from R by yourself. The **cnv** package contains several functions:

```
cnv.cal  <- function (file, log2.threshold = NA,
                chromosomal.normalization = TRUE,
                annotate = FALSE, minimum.window = 4)
cnv.print <- function (cnv, file = "")
cnv.summary <- function (cnv)
plot.cnv <- function (data, chromosome = NA,
                CNV = NA, ...)
plot.cnv.all <- function (data, chrom.gap = 2e+07,
                colour = 5, title = NA, ylim = c(-2,2),
                xlabel = "Chromosome")
plot.cnv.chr <- function (data, chromosome = NA,
                from = NA, to = NA, title = NA,
                ylim = c(-4, 4), glim = c(NA, NA),
                xlabel = "Position (bp)")
plot.cnv.cnv <- function (data, CNV, upstream = NA,
                downstream = NA, ...)
```

4 Demonstration

We provide some sample data for demonstration. You can download Sample1.tar.gz from our website. Sample1.tar.gz contains BLAT output for simulated Solexa reads with 1X coverage on human chromosome 1. (**Warning: samlple1.tar.gz is huge)** After download the file, open command line console, and run:

```
$ cd DOWNLOAD_DIR
$ tar xzf Sample1.tar.gz
$ PATH-TO/best-hit.BLAT.pl ref.psl > ref.hits
$ PATH-TO/best-hit.BLAT.pl test.psl > test.hits
$ ### NOTE: there are several other versions of
$ ###       best-hit.*.pl for different input formats
```

The above lines will generate two output files: test.hits and ref.hits, which are the genomic locations of the best BLAT hits. We also provided the two files (test.hits and ref.hits) on our website as Sample2.tar.gz, which is much smaller than Sample1.tar.gz. After obtaining the two hits files, you can run **`cnv-seq.pl`**:

```
$ cnv-seq.pl --test test.hits --ref ref.hits --genome chrom1
          --log2 0.6 --p 0.001 --bigger-window 1.5  #default
          --annotate --minimum-windows 4            #default
```

This will give you output like this:

```
genome size used for calculation is 247249719
test.hits: 1874797 reads
ref.hits: 1878852 reads
The minimum window size for detecting log2>= 0.6 should be 17676.9728733869
The minimum window size for detecting log2<=-0.6 should be 17692.0064154661
window size to use is 17692.0064154661 x 1.5 = 26538
window size to be used: 26538
read 1874797 test reads, out of 1874797 lines
read 1878852 ref reads, out of 1878852 lines
write read-counts into file: test.hits-vs-ref.hits.log2-0.6.pvalue-0.001.count
R package cnv output: test.hits-vs-ref.hits.log2-0.6.pvalue-0.001.minw-4.cnv
...
[1] "chromosome:  1"
```

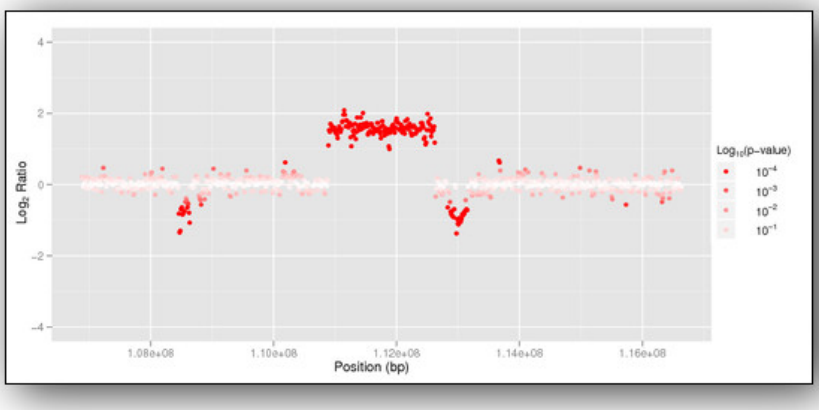
```
[1] "cnv_id:  1  of  50"
[1] "cnv_id:  2  of  50"
[1] "cnv_id:  3  of  50"
[1] "cnv_id:  4  of  50"
...
```

The sliding window size used is 26.5Kb.

This will give a tab delimited file test.hits-vs-ref.hits.log2-0.6.pvalue-0.001.minw-4.cnv. This file contains all information about CNV prediction from the analysis. In order to plot the log2 CNV graph:

```
$ R
# in R command prompt
> library(cnv)
> data <- read.delim("test.hits-vs-ref.hits.log2-0.6.pvalue-0.001.cnv")
> cnv.print(data)
# output ...
> cnv.summary(data)
# output ...
> plot.cnv(data, CNV=4, upstream=4e+6, downstream=4e+6)
> ggsave("sample.pdf")
```

The plot.cnv command will generate a plot like this picture:



References

[1] Chao Xie and Martti Tammi. CNV-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinformatics*, 10(1):80, Mar 2009.