ExomeCNV User Guide

From Nelsonlab

Contents

Overview

ExomeCNV (http://cran.r-project.org/web/packages/ExomeCNV/index.html) is an R package tailored to detection of CNV (Copy-Number Variants) and LOH (Loss of Heterozygosity) from exome sequencing data. It exploits the unique discrete feature of exon definitions and incredible cross-sample consistency of depth-of-coverage. ExomeCNV is most suitable when paired samples (e.g. tumor-normal pair) are available. Both of the paired samples should be processed and sequenced in a similar manner (e.g. same library prep, sequencer, average depth-of-coverage, etc.).

The inputs necessary for ExomeCNV are 1) bam/pileup file 2) exome definition file (.bed) and 3) a conservative approximation of sample admixture rate.

System Requirements

ExomeCNV is developed using Oracle Grid Engine (aka Sun Grid Engine) batch-queuing system. Thus most of our example are provided using OGE syntax. However, the very core of ExomeCNV does not rely on this and can be adapted to run anywhere.

The system should have the following programs installed:

- R
- Perl
- Python
- Samtools (we recommend v0.1.16 as newer version may not support pileup function/files)
- GATK (optional, but it will make your life easier)

And chromosomes the input files are assumed to follow this format "chr#" where # is the chromosome number (e.g. chr1, chr2, ...).

CNV Calling Procedure

There are a few steps in using ExomeCNV to identify CNV. The steps are outlined here:

Convert bam file into coverage files

There are two ways to do this: 1) use GATK (Genome Analysis Toolkit) DepthOfCoverage or 2) use our script.

GATK DepthOfCoverage

If you have GATK set up in your system, you are golden. Use this command to produce a coverage file:

```
/usr/java/latest/bin/java \
   -jar GenomeAnalysisTK.jar \
   -T DepthOfCoverage \
   -omitBaseOutput \
   -omitLocusTable \
   -R human_g1k_v37.fasta \
   -I <my.bam> \
   -L <exome.interval_list> \
   -o <output.coverage>
```

where human_g1k_v37.fasta is the reference genome sequences that can be downloaded from ftp://ftp.sanger.ac.uk/pub/1000genomes/tk2/main_project_reference/, and the exome.interval_list follows the format:

```
chr#:start-end
```

For example, these are the first few lines of our file:

```
1:35136-35176
1:35275-35483
1:35719-35738
1:69089-70010
1:367657-368599
```

The GATK output should look like: http://genome.ucla.edu/~fah/ExomeCNV/data/sampleCoverage.gatk.txt

When reading in coverage in R, use the `read.coverage.gatk(file)` command.

bam2coverage scripts

You need to download a set of scripts from http://genome.ucla.edu/~fah/ExomeCNV/script/makeCoverageScript.tar.gz, extract them, and run shell script bam2coverage.sh like so:

```
sh bam2coverage.sh <my.bam> <exome.bed> <hg.fa> <src>
```

This may takes a while depending on how large your pileup/bam file is. <exome.bed> refers to the exome definition in bed format, <hg.fa> refers to the reference genome file used in mapping the raw reads, and <src> is the directory in which the perl scripts are stored. These should be consistent with the files used in producing the bam file.

Run ExomeCNV functions

The three primary steps of ExomeCNV are

1. Calculate log coverage ratio between case and control
2. Call CNV/LOH for each exon individually
3. Combine exonic CNV/LOH into segments using Circular Binary Segmentation (CBS)

After this you can plot your results or export them.

These steps are illustrated here (http://genome.ucla.edu/~fah/ExomeCNV/demo/demo.R)

If your system is equipped with OGE/SGE (Oracle/Sun Grid Engine), you may perform parallel processing as demonstrated in http://genome.ucla.edu/~fah/ExomeCNV/demo/demo_sge.R

First load the package

```
library(ExomeCNV)
```

Calculate log coverage ratio

```
chr.list = c("chr19","chr20","chr21")

suffix = ".coverage"

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/normal."
normal = read.all.coverage(prefix, suffix, chr.list, header=T)

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/tumor."
tumor  = read.all.coverage(prefix, suffix, chr.list, header=T)

demo.logR = calculate.logR(normal, tumor)
```

Call CNV for each exon

Then call CNV on each exon (using classify.eCNV), one chromosome at a time. We recommend high min.spec (0.9999) and option="spec" to be conservative against false positive. This is because whatever is called at exon level will persist through merging step (Step 3).

```
demo.eCNV = c()
for (i in 1:length(chr.list)) {
        idx = (normal$chr == chr.list[i])
        ecnv = classify.eCNV(normal=normal[idx,], tumor=tumor[idx,], logR=demo.logR[idx], min.spec=0.9999, min.sens=0.9999,
option="spec", c=0.5, l=70)
        demo.eCNV = rbind(demo.eCNV, ecnv)
}
```

Combine exonic CNV into larger segments

Here, we use lower min.spec and min.sens and option="auc" to be less conservative and allow for more discovery.

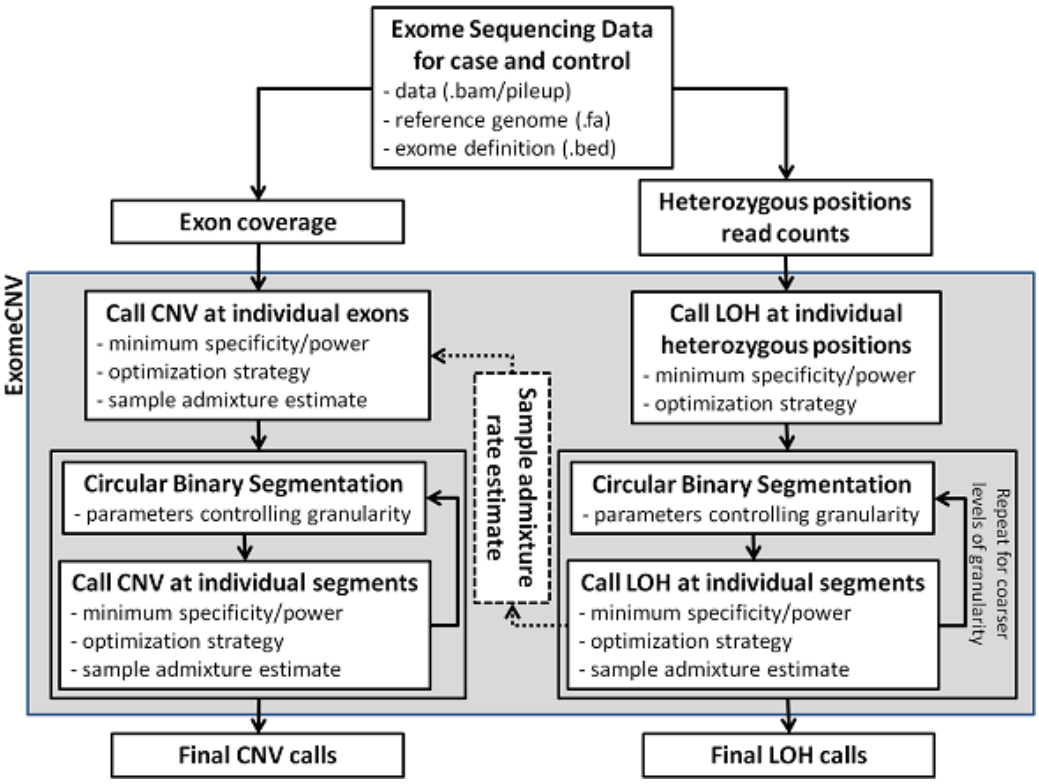
```
library(DNAcopy)
demo.cnv = multi.CNV.analyze(normal, tumor, logR=demo.logR, all.cnv.ls=list(demo.eCNV), coverage.cutoff=5, min.spec=0.99,
min.sens=0.99, option="auc", c=0.5)
```

From here we can plot the results and export outputs.

```
do.plot.eCNV(demo.eCNV, lim.quantile=0.99, style="idx", line.plot=F)
do.plot.eCNV(demo.cnv, lim.quantile=0.99, style="bp", bg.cnv=demo.eCNV, line.plot=T)
write.output(demo.eCNV, demo.cnv, "demo")
```

Parameter Setting

As shown in the overview figure, each step of ExomeCNV involves a number of user parameters. The user is free to set the parameters to fit a particular application, but here are a few tips for general application.



CNV/LOH Calling

The important parameters in CNV/LOH calling step are:

- minimum specificity / sensitivity
- optimization strategy (specificity, sensitivity, or area-under-curve (AUC))
- sample admixture estimate

Minimum Specificity / Sensitivity

These are the required minimum power and specificity (1 - false positive rate) required for ExomeCNV to make a call. If an exon/segment does not have sufficient coverage to achieve the desired power/specificity, ExomeCNV will not make a call on that exon/segment. These parameters are coded as min.spec and min.sens in the functions classify.eCNV and multi.CNV.analyze, as alpha in LOH.analyze, and as test.alpha/min.spec in multi.LOH.analyze. Also note here that LOH calling does not have min.sens. As shown in the overview figure, CNV/LOH calls are made at two separate steps: (First round) exon-level/position-level calls and (Second round) segment-level calls. The recommended settings for these min.spec/min.sens differ at different steps.

Since merging prioritizes calls made in the first round of CNV/LOH calls over later rounds, we advise that the user set high min.spec to avoid false positive. In other words, because of the way merging proceeds, false positives will persist and accumulate through merging, thus we should set higher min.spec to avoid false positive in the earlier rounds of CNV/LOH calls. The user should also set optimization strategy to "specificity" in the first round of calls for the same reason.

Optimization Strategy

Because there could be multiple cutoff values that satisfy min.sens and min.spec (see above), we could pick the cutoff value that optimizes for sensitivity, specificity, or an average between the two (i.e. area-under-curve, AUC = (sensitivity + specificity)/2). The choice depends on the application, but for the same reason that we should set high min.spec in the first round of calls, the user might want to optimize for "specificity" in the first round of CNV/LOH calls. In the segment-level calls, the user may elect to user "AUC" option to get the "best of both worlds," balancing between specificity and sensitivity.

It is also recommended that if the user uses the option "AUC", min.sense and min.spec should be set to the same value. By experience, when "AUC" option is used and min.sens is not equal to min.spec, the results tend to be very noisy.

Sample Admixture Rate

It is common in the case of tumor biopsy to have some heterogeneity in the sample. Particularly, some normal (non-mutated) tissues may be present in the sample at certain rate. A 30-50% admixture is not at all uncommon. The sensitivity and precision of CNV detection are dependent on a good estimate of the admixture rate, and a more conservative estimate tends to lead to fewer false positives. By default, the admixture rate "c" is set at 0.3. If the results appear to be noisy, users may consider increasing "c" up to 0.5 or more.

Circular Binary Segmentation (CBS)

In the functions multi.CNV.analyze and multi.LOH.analyze, ExomeCNV performs segmentation using CBS (implemented in DNAcopy package). There are a few parameters for CBS controlling granularity of segmentation. Intuitively, a more granulated segmentation will produce more of smaller segments. The two parameters that controls granularity of CBS are:

- sd.undo: the number of standard deviations between means to keep a breakpoint
  - The higher sd.undo is, the coarser the segments will become. The default values are set to 1 and 2.
- alpha: significance levels for the test to accept change-points
  - The lower alpha is, the coarse the segments will become (harder to accept a breakpoint). The default values are set to 0.05 and 0.01.

LOH Calling Procedure

Pre-processing

The steps in LOH calling procedure is similar to those in CNV calling. User should prepare BAF information for all heterozygous positions in the exome. The input file should be a tab-delimited file with four columns (and the header):

```
chr     position        coverage        baf
```

For example:

```
chr     position        coverage        baf
chr1    323689  56      15
chr1    325016  23      14
chr1    452701  27      15
chr1    531665  30      12
chr1    564898  26      13
...
```

Examples can be found at http://genome.ucla.edu/~fah/ExomeCNV/data/normal.baf.txt and http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.baf.txt.

The file can be read into R like so:

```
normal = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.baf.txt", header=T)
tumor = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.baf.txt", header=T)
```

The two primary steps for LOH calling are:

1. Calling LOH at each heterozygous position
2. Combine multiple positions into LOH segments

Using GATK to create BAF file

(credit: Dr. Hane Lee)

From VCF file, you can use the perl script for.loh.files.pl (http://genome.ucla.edu/~fah/ExomeCNV/script/makeCoverageScript.tar.gz) which takes:

```
perl for.loh.files.pl <input vcf> <output normal.baf.txt> <output tumor.baf.txt> <normal_col_#> <tumor_col_#>
```

Here is an example:

```
perl for.loh.files.pl sample.vcf for.loh.normal.baf.txt for.loh.tumor.baf.txt 3 2
```

The last two are counting from the columns that contain the genotype information so for instance in http://genome.ucla.edu/~fah/ExomeCNV/data/sample.vcf. The normal_col_# would be 3 as skin is the 3rd sample. The tumor_col_# would be 2 as primary tumor (PT) is the 2nd sample.

If the genotypes for normal and tumor are in separate files, GATK has combinevariants (http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_variantutils_CombineVariants.html) function to merge the two.

This generated "for.loh.normal.baf.txt" and "for.loh.tumor.baf.txt" that can be used as input for exomeCNV LOH run as described below. Note that the demo data used below are different from the ones here.

Calling LOH on each heterozygous position

For details about different statistical tests (method), see below.

```
eLOH = LOH.analyze(normal, tumor, alpha=0.05, method="two.sample.fisher")
```

Combine multiple positions into LOH segments

The function multi.LOH.analyze will perform CBS and call LOH on each segment. User can control fineness of segmentation by controlling sdundo and alpha parameters (these are passed on to DNAcopy) and choose test statistics for LOH calling. See parameter setting discussion below for explanation and recommendation of different types of tests and settings.

```
the.loh = multi.LOH.analyze(normal, tumor, all.loh.ls=list(eLOH), test.alpha=0.001, method="variance.f", sdundo=c(0,0),
alpha=c(0.05,0.01))

do.plot.loh(the.loh, normal, tumor, "two.sample.fisher", plot.style="baf")
write.loh.output(the.loh, "demo.eloh")
```

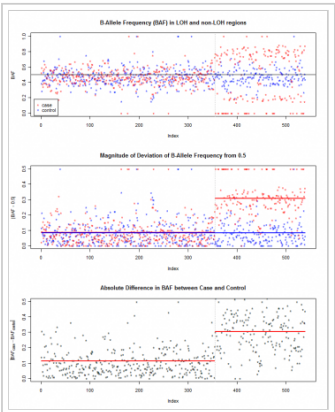
This will give LOH intervals in a format similar to a bed file. If you wish to assign LOH status to all heterozygous positions, expand.loh function may be used:

```
expanded.loh = expand.loh(the.loh, normal)
write.loh.output(expanded.loh, "demo.all")
```

Choice of Test Statistics

Statistical tests that can be used in calling LOH are based on three test statistics:

1. BAF as count statistic
2. Variance of BAF, reflecting the amount of deviation of BAF away from its central value (~0.5)
3. Absolute deviation of BAF from the null value of 0.5
4. Difference between BAF's in case and control samples



Three test statistics based on BAF

Each test statistic allows for different tests and is based on different assumptions.

BAF as count statistic

These tests are only available when calling LOH on a single position (using LOH.analyze).

Options "only.tumor" and "only.normal" use only one sample (case or control) to perform binomial test against null p=0.5. We can model LOH as a binomial event, asking among N reads mapped to the position, how likely is it to observe a certain number of B-allele (BAF).

Options "two.sample.fisher" and "two.sample.prop" are similar to the binomial test for one sample above but instead of testing the observed proportion against the null value of 0.5, they compare the observed proportion between case and control. This can be modeled by binomial distribution (two.sample.prop) or hypergeometric distribution (Fisher's exact test; two.sample.fisher), hence

the two possible tests.

Variance of BAF

Option "variance.f" performs F-test to compare variances of case and control BAF's

Absolute Deviation of BAF from 0.5

Options "deviation.wilcox" and "deviation.t" perform t-test and Wilcoxon Rank Sum (Mann-Whitney) Test, respectively. This is to compare the mean value of the absolute deviation of BAF from 0.5 (i.e. |BAF - 0.5|).

Difference between case and control BAF's

Option "deviation.half.norm" is based on the observation that the distribution of BAF difference between case and control are normally distributed around 0. Thus the absolute value follows folded-normal distribution. Under LOH, the absolute difference will have a higher mean value, and we can measure and test the increase in the difference using half-normal distribution.

Other Test

Cochran-Mantel-Haenszel Statistics

Option "CMH" or "mantelhaen" uses Cochran-Mantel-Haenszel Chi-sq test for common odds ratio equal to 1. It requires that the number of stata N >= 2. In case N = 1, it is equivalent to Pearson's Chi-sq (prop.test). This is useful when trying to call LOH for segments, which contain multiple heterozygous positions, each with its own contigency table. The only problem with this test is that it requires phasing information, which does not always exist. Thus it is not recommended for use.

Limitation

There are a few assumptions that ExomeCNV is making. Here is a list of known limitations:

- ExomeCNV has been used and tested in human only. If there is enough interest in extending ExomeCNV into other organisms, we may extend it further. Please send email to the author at fah@cs.stanford.edu if you have a particular application that you would like to try ExomeCNV on.

FAQ

Q: My data is paired end data. The ends are 50bp and 35bp. The classify.eCNV step asks for read lengths. Should I put 50, 35, or 85?

A: 42. Read length is used to convert depth-of-coverage back into approximate read counts. As long as the number of reads from both ends are roughly equal, using the mean read length should give the right approximate.

Q: I have multiple (unpaired) case and control exomes. How can I use ExomeCNV to detect CNV?

A: ExomeCNV was developed for closely matched paired samples and performs best in that setting. However, one may pool together a set of samples and compare pooled case and pooled control exomes. The function pool.coverage in ExomeCNV package is available for this use. It is important to note that the power calculation might be off and a conservative evaluation of the results is recommended.

Q: I noticed that when running function "read.coverage.gatk(file)" to reformat GATK output, these following two fields "sequenced.base" and "bease.with..10.coveratge" are all with "NA". Do you think "NA" in these two fields will affect the detection results?

A: No, the fields "sequenced.base" and "bases.with..10.coverage" are degenerate and are not important for ExomeCNV to function. They are there for a historical reason.

Tutorial

I have put together a tutorial for a workshop organized at Harvard Medical School. See ExomeCNV Quick Tutorial.

Reference

JF Sathirapongsasuti, H Lee, BAJ Horst, G Brunner, AJ Cochran, S Binder, J Quackenbush, SF Nelson (2011) Exome Sequencing-Based Copy-Number Variation and Loss of Heterozygosity Detection: ExomeCNV, Bioinformatics, Advanced Access: Aug 9, 2011. (http://bioinformatics.oxfordjournals.org/content/early/2011/08/09/bioinformatics.btr462.full.pdf?keytype=ref&ijkey=ZvqeucgNjC5Nco4)