2015 EBI Cancer Genomics workshop - CNV analysis

Mathieu Bourgey, Ph.D

MCGill University and Genome Quebec Innovation Center

mathieu.bourgey@mcgill.ca

Introduction

This workshop will show you how to proceed when to detect Copy Number Variants (CNV) using either Next Generation Sequencing (NGS) or SNParray data

The initial structure of your folders should look like this:

```
UKworkshop/
|-- src/                    # scripts
|-- NGS_CNV/                # Analysis folder for NGS data
|-- SNP_CNV/                # Analysis folder for SNParray data
|-- SNParray/               # BAF and logR files per sample
|-- alignment/              # bam files generated previously
    `-- normal              # Normal sample directory
    `-- tumor               # Tumor sample directory
```

Data

in this hands-on two types of data are available: SNParray and NGS data

NGS data

approximately 1M of Illumina paired-end 100 reads covering the region *chr19 : 50500375 - 52501256* from a whole genome experiment for germline and matched tumor samples.

For each sample:

- 1 alignment file (.bam generated in the previous hands-on) on the human reference sequence

SNParray data

Illumina data for more than 650000 SNPs located genome wide for germline and matched tumor samples

For each sample:

- 1 file containing the LogRatio (LRR) mesurement of each probe: the signal intensities compared to a collection of reference hybridizations
- 1 file containing the B Allele Frequency (BAF) mesurement of each probe: the proportion of the total allele signal (A + B) explained by a single allele (A)

What are the advantages and limitations of using each type of technology ? *solution technology*

NGS data analysis

We will start from two alignment files of the tumor and germline samples form the same individual and we will apply a manual read depth procedure to estimate the tumoral copy number.

What are the steps to proceed this analysis ? *solution NgsAnalysisSummary*

NGS data analysis for CNV detection

Starting from a fastq files (1-9) or from BAM files (4-9):

1. Sequence trimming

Removing low quality bases

2. Genome alignment

Locate individual reads to the reference genome

3. Alignment refinement

Realign around INDELs, remove duplicates, etc...

4. Bin creation and count

The choice of the bin size has a major importance (moving or overlaping windows)

We use the software BVAtools to run this sep of the analysis.

Environment setup Open a terminal and set-up the environment in order to be able running BVAtools:

```
export BVATOOLS_JAR=/home/training/Applications/bvatools-1.6/bvatools-1.6-full.jar
```

BVAtools overview Take a look at the BVAtools usage:

```
java -jar $BVATOOLS_JAR

java -jar $BVATOOLS_JAR bincounter
```

Which parameters should we used ?
solution BinCount

Feel free to test different sets of parameters.

```
cd /home/training/ebicancerworkshop201507/NGS_CNV
java -jar $BVATOOLS_JAR bincounter \
  --minMapQ 35 \
  --refbam ../alignment/normal/normal.sorted.dup.bam \
  --bam ../alignment/tumor/tumor.sorted.dup.bam \
  --norm chr \
  --windows 1000 \
> sample_binCount_1kb.tsv
```

The output file should looks like:

```
$ head sample_binCount_1kb.tsv
chr start   end sample_raw  ref_raw sample_normalized   ref_normalized  ln(sample/ref)
chr1    0   1999    0   0   0.0 0.0 NaN
```

Why first lines show no coverage data ?
solution BinCount 2

5. Bin count correction (optional)

Two different type of correction are usually applied to the dat:

1. GC content
2. Mappability

It won't be done today because it needs a whole genome data to properly work

6. Computing LRR

The analysis of binned data and CNV calls will be don eusing the R tool and the DNAcopy package from Bioconductor

Open R

```
cd $HOME/ebicancerworkshop201507/CNV/NGS
R
```

Load DNAcopy

```
library("DNAcopy")
```

A short presentation DnaCopy Here is the minimal list of functions in the DNAcopy package that are used to run a basic CNV calling algorithm

Function	Explanation
CNA	Creates a *Copy Number Array* data object
DNAcopy	R object resulting of the segmentation step
smooth.CNA	Smooth the signal to reduce outliers points
segment	Find segments harboring similar signal using the CBS algorithm
plot.DNAcopy	optional: plot the result of the segmentation
segments.summary	optional: provides statistics from segements

Load binned Data

```
data=read.table("sample_binCount_1kb.tsv",header=T)
```

```
head(data)
```

```
  chr start  end sample_raw ref_raw sample_normalized ref_normalized
1   1     0  199          0       0                 0              0
2   1   200  399          0       0                 0              0
3   1   400  599          0       0                 0              0
4   1   600  799          0       0                 0              0
5   1   800  999          0       0                 0              0
6   1  1000 1199          0       0                 0              0
  ln.sample.ref.
1            NaN
2            NaN
3            NaN
4            NaN
5            NaN
6            NaN
```

Clean data to remove region with no coverage

```
dataClean=data[data[,4] > 0 | data[,5] > 0,1:5]
head(dataClean)
```

```
         chr    start      end sample_raw ref_raw
13549734  19 50500200 50500399         70     120
13549735  19 50500400 50500599        185     328
13549736  19 50500600 50500799        188     349
13549737  19 50500800 50500999        174     363
13549738  19 50501000 50501199        146     312
13549739  19 50501200 50501399        198     341
```

Normalize count between tumor and germaline

```
dataNorm=cbind(dataClean,dataClean[,4]/sum(dataClean[,4]),dataClean[,5]/sum(dataClean[,5]))
head(dataNorm)
```

```
         chr    start      end sample_raw ref_raw
13549734  19 50500200 50500399         70     120
13549735  19 50500400 50500599        185     328
13549736  19 50500600 50500799        188     349
13549737  19 50500800 50500999        174     363
13549738  19 50501000 50501199        146     312
13549739  19 50501200 50501399        198     341
         dataClean[, 4]/sum(dataClean[, 4]) dataClean[, 5]/sum(dataClean[, 5])
```

```
13549734                    4.080481e-05                    3.686617e-05
13549735                    1.078413e-04                    1.007675e-04
13549736                    1.095901e-04                    1.072191e-04
13549737                    1.014291e-04                    1.115202e-04
13549738                    8.510718e-05                    9.585203e-05
13549739                    1.154193e-04                    1.047614e-04
```

Estimate the logRatio in each bin

```
Chr=dataNorm[,1]
Pos=dataNorm[,2]
logR=log2((dataNorm[,6]+0.00001)/(dataNorm[,7]+0.000001))
logR[1:6]
```

```
[1]  0.14275958  0.09694868  0.03126659 -0.13555779 -0.16964886  0.13851792
```

7. LRR signal smoothing (optional)

the purpose of this step is to reduce the impact og each single point outliers before doing the analysis

First create a CNA object

```
CNA.object=CNA(logR,Chr,Pos, data.type="logratio",sampleid="TNratio")
CNA.object
```

```
Number of Samples 1
Number of Probes  9818
Data Type         logratio
```

Then smooth the data

```
smoothed.CNA.object=smooth.CNA(CNA.object)
smoothed.CNA.object
```

```
Number of Samples 1
Number of Probes  9818
Data Type         logratio
```

8. LRR signal segmentation

Copy number aberrations (CNA) occur in contiguous regions of the chromosome that often cover multiple bins up to whole chromosome arms or chromosomes. The segmentation split the chromosomes into regions of equal copy number that accounts for the noise in the data

Non-exhaustive available methods:

- Circular Binary Segmentation
- Mean Shift-Based
- Shifting Level Model
- Expectation Maximization
- Hidden Markov Model
- Etc...

There is actually no gold standard for the cancer data

Generate segments using Circular Binary Segmentation

```
segment1=segment(smoothed.CNA.object, verbose=1)
head(segment1$output)
```

```
       ID chrom loc.start  loc.end num.mark seg.mean
1 TNratio    19  50500200 50595200      476   0.0082
2 TNratio    19  50596400 50598600        7  -0.7104
3 TNratio    19  50598800 50609200        9   0.4972
4 TNratio    19  50609400 50636000       15  -0.7169
5 TNratio    19  50636600 50637600        6  -0.0457
6 TNratio    19  50637800 50638800        6   0.5814
```

9. CNV calling from segments

There is many method that can differenciate the copy number of each segments each

Non-exhaustive available methods:

- Thresholds
- Sd deviation
- Poisson distribution
- Z-score
- Event-Wise Testing

- Etc...

As for segmentation there is non perfect method design for cancer data yet because each cancer sample is different

Call CNA for segment using threshold approach

```
duplicationTh=log2(2)
deletionTh=log2(0.5)
minBin=10
CNAcall=segment1$output[segment1$output[,5] >= minBin
  & (segment1$output[,6] <= deletionTh |
  segment1$output[,6] >= duplicationTh) ,]
head(CNAcall)
```

```
        ID chrom loc.start  loc.end num.mark seg.mean
81 TNratio    19  52133600 52149400       80  -5.6947
```

Output your result in a file

```
CNAtype=rep(".",dim(CNAcall)[1])
names(CNAtype)="CNA_type
CNAtype[CNAcall[,6] >= duplicationTh]="DUP"
CNAtype[CNAcall[,6] <= deletionTh]="DEL"
CNAcallfinal=cbind(CNAcall,CNAtype)
write.table(CNAcallfinal,"sampleCNAcall.tsv",sep="\t",quote=F,col.names=T,row.names=F)
q(save="yes")
```

You can then look at the call file using this command:

```
head sampleCNAcall.tsv
```

```
ID  chrom   loc.start   loc.end num.mark    seg.mean    CNAtype
TNratio 19  52133600    52149400    80      -5.6947 DEL
```

This file shows the presence of large deleton event in the chr19 region (19 52133600 52149400).

If you reduce the threshold value you will start to see many region that will pop-up all along the candidat region. In fact one of the major issue of most of the CNV caller from NGS data: due to not so uniform coverage, local variation occurs in the read depth which can produce noise in the resulls:

1. scattered calls
2. false positive calls
3. false negative calls

For your information, here is a non-exhaustive list of available softwares for calling CNV using whole genome NGS Data:

Tool	URL
SegSeq	http://www.broad.mit.edu/cancer/pub/solexa_copy_numbers/
CNV-seq	http://tiger.dbs.nus.edu.sg/cnv-seq
RDXplorer	http://rdxplorer.sourceforge.net
BIC-seq	http://compbio.med.harvard.edu/Supplements/PNAS11.html
CNAsega	http://www.compbio.group.cam.ac.uk/software/cnaseg
cn.MOPS	http://www.bioinf.jku.at/software/cnmops/
JointSLMb	http://nar.oxfordjournals.org/content/suppl/2011/02/16/gkr068.DC1/JointSLM_R_Package.z
ReadDepth	http://code.google.com/p/readdepth
rSW-seqa	http://compbio.med.harvard.edu/Supplements/BMCBioinfo10-2.html
CNVnator	http://sv.gersteinlab.org
CNVnorma	http://www.precancer.leeds.ac.uk/cnanorm
CMDS	https://dsgweb.wustl.edu/qunyuan/software/cmds
mrCaNaVar	http://mrcanavar.sourceforge.net
cnvHMM	http://genome.wustl.edu/software/cnvhmm
PopSV	https://github.com/jmonlong/PopSV
SCoNEs	https://bitbucket.org/mugqic/scones

SNParray data analysis

SNParray analysis are very similar to NGS data analysis while incorporating the additional information bring by the SNP: the BAF. In this analysis we will start from one LRR signal file and one BAF signal file for each of the germline and matched tumor samples from an individul.

Many software are avaiable for doing CNV call from SNParray. Here is a non-exhaustive list of softaware that could be used:

1. Proprietary softwares
2. GenomeStudio/CNVpartition - Illumina
3. Genotyping Console/Birdsuite - Affymetrix
4. Affymetrix oriented softwares
5. Genome Alteration Detection Algorithm (GADA)
6. Cokgen
7. Commercial softwares
8. Partek Genomics Suite
9. Golden Helix SNP
10. Freely available general software
11. PennCNV
12. QuantiSNP
13. Freely available cancer oriented software
14. **Allele-Specific Copy number Analysis of Tumors (ASCAT)**
15. OncoSNP

What are the major cancer factors that could biais a CNV analysis ?

solution cancerChallenge

In this practical I choose to use ASCAT because it handles samples with aneuploidy and the presence of normal cells contamination. Moreover ASCAT facilitates detection of tumor cell heterogeneity.

Here is a video showing how the ASCAT software works:

What are the steps to proceed this analysis ?

solution 6.SnpAnalysisSummary

SNParray analysis

we start from filtered LRR and BAF files

during the anlalysis we will need to estimate 3 parrameters in order to have confident CNV calling

1. aberrant cell fraction
2. tumor ploidy
3. absolute allele-specific copy number calls (for each allelic probes of the SNP)

Open R

```
cd ../SNP_CNV/
R
```

Load ASCAT

```
source("../src/ascat-2.2.R")
```

Load data

```
ascat.raw = ascat.loadData("../SNParray/tumor2.LRR.tsv",
  "../SNParray/tumor2.BAF.tsv",
  "../SNParray/normal2.LRR.tsv",
  "../SNParray/normal2.BAF.tsv")
```

Plot raw data

```
ascat.plotRawData(ascat.raw)
```

Perform segmentation

```
ascat.seg = ascat.aspcf(ascat.raw)
```

sPlot segments

```
ascat.plotSegmentedData(ascat.seg)
```

Estimate model parameters

```
ascat.output = ascat.runAscat(ascat.seg)
```

Ouput model parameters

```
params.estimate=data.frame(Sample=names(ascat.output$aberrantcellfraction),
  Aberrant_cell_fraction=round(ascat.output$aberrantcellfraction,2),
  Ploidy=round(ascat.output$ploidy,2))

write.table(params.estimate,
  "sample.Param_estimate.tsv",
  sep="\t",
  quote=F,
  col.names=T,row.names=F)
```

Call CNA

determine the copy number by simply counting the total number of allele reported to the sample ploidy

```
CNA=rep(".",dim(ascat.output$segments)[1])
CNA[rowSums(output.table[,5:6]) > round(ascat.output$ploidy)]="DUP"
CNA[rowSums(output.table[,5:6]) < round(ascat.output$ploidy)]="DEL"
output.table=data.frame(ascat.output$segments,CNA=CNA)
```

Save CNA calls

```
write.table(output.table[output.table$CNA != ".",],
  "sample_CNVcalls.tsv",
  quote=F,
  sep="\t",
  col.names=T,
  row.names=F)

q(save="yes")
```

These files shows the presence of large deletion and duplication evenet all along the genome of this indivduals.

Particularly in the chromosome 12 a very impressive duplication can be observed.

The only limitation of this approach is the size of event that could be detected. Few kb events or less will be missed.

Aknowledgments

The format for this tutorial has been inspired from Mar Gonzalez Porta of Embl-EBI and Louis Letourneau from MUGQIC. This tutorial use materails form the BioDicovery.com website, the Alkan, Coe & Eichler's review article and the Zhao *et al.* artcile. I would like to thank and acknowledge all of them.