Introduction to DNA-Seq processing for cancer data - CNVs

By Mathieu Bourgey, Ph.D
https://bitbucket.org/mugqic/mugqic_pipelines

==========

==========

In this workshop, we will present the main steps that are commonly used to process and to analyze cancer sequencing data. We will focus on whole genome data and SNParray data and we will provide command lines that allow detecting Copy Number Variations (CNV).

Introduction

Goals of this session are:

1. to perform a copy number variation analysis (CNV) on a normal/tumour pair of alignment files (BAMs) produced by the mapping of Illumina short read sequencing data.
2. to perform a copy number variation analysis (CNV) on a normal/tumour pair of SNParray data files (BAF and LRR) produced by the processing of Illumina chip.

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

_____

_____

NGS data analysis

In this session we will produce the main steps of the analysis of NGS data in order to detectc CNV and also to estimate cellularity, ploidy of the tuor sample. Sequenza is the tool we will use to perform this analysis. Sequenza is able to perform the CNV analysis in both WGS and WES data.

It consists of a two Python pre-processing steps followed by a third step in R to infer the samples estimates and to plot the results for interpretation.

In a second timel we will also be using IGV to visualise and manually inspect the copy number variation we inferred in the first part for validation purposes.

Data Source

We will be working on a CageKid sample pair, patient C0053. The CageKid project is part of ICGC and is focused on renal cancer in many of it's forms. The raw data can be found on EGA and calls, RNA and DNA, can be found on the ICGC portal. For more details about CageKid

To ensure reasonable analysis times, we will perform the analysis on a heavily subset pair of BAM files. These files contain just 60Mb of chromosome 2

Prepare the Environment

We will use a dataset derived from whole genome sequencing of a clear-cell renal carcinoma patient (Kidney cancer)

```
## set environement
export SEQUENZA_UTILS=/home/training/R/x86_64-pc-linux-gnu-library/3.3/sequenza/exec/sequenz
export REF=/home/training/ebicancerworkshop201607/reference

cd $HOME/ebicancerworkshop201607/CNV/NGS
```

Software requirements

These are all already installed, but here are the original links.

- R
- sequenza R package
- samtools
- python 2.7
- IGV

Original Setup

The initial structure of your folders should look like this:

```
<ROOT>
|-- C0053/                    # fastqs from the center (down sampled)
    `-- normal                      # The blood sample directory
     `-- normal_chr2_60Mb.bam  # Alignment file as  generated through the SNV session
        `-- normal_chr2_60Mb.bai   # Index of the alignment file
    `-- tumor                       # The tumor sample directory
     `-- tumor_chr2_60Mb.bam   # Alignment file as  generated through the SNV session
        `-- tumor_chr2_60Mb.bai    # Index of the alignment file
|-- saved_results                   # Pre-computed files
`-- scripts                         # cheat sheet folder
```

Generate a seqz file

A seqz file contains genotype information, alleles and mutation frequency, and other features. This file is used as input for the R-based part of Sequenza.

The seqz file could be generated from a pileUp file (as shown in SNV session) or directly from the bam file.

Should we start from the mpileUp of the SNV session ?

_____

_____

transforming bam files in seqz file

As we haven't already generated the pileup files, and we are not interested in storing the pileup for further use, we can use the function `bam2seqz` which converting on the fly to pileup using samtools without storing the pileup file.

What the impact of converting data on the file ?

_____

_____

```
## sequenza preprocessing step 1 - bam 2 seqz format
mkdir -p sequenza

${SEQUENZA_UTILS} bam2seqz \
 -n C0053/normal/normal_chr2_60Mb.bam \
 -t C0053/tumor/tumor_chr2_60Mb.bam \
 --fasta ${REF}/Homo_sapiens.GRCh37.fa  \
```

```
 -gc ${REF}/Homo_sapiens.GRCh37.gc50Base.txt.gz \
 -q 20 \
 -N 20 \
 -C 2:106000000-166000000 | gzip > \
 sequenza/C0053.seqz.gz
```

To reduce the size of the seqz file, we'll use of a binning function provided in sequenza-utils.py. This binning decreases the memory requirement to load the data into R, and it also speeds up the processing of the sample.

```
## sequenza preprocessing step 2 - seqz binning 500bp
${SEQUENZA_UTILS} seqz-binning \
 -w 500 \
 -s sequenza/C0053.seqz.gz | gzip > \
 sequenza/C0053.seqz.bin500.gz
```

We can look at the first few lines of the output in the file 'sequenza/C0053.seqz.gz with:

```
zless -S sequenza/C0053.seqz.gz
```

This output has one line for each position in the BAMs and includes information on the position, depths, allele frequencies, zygosity, GC in the location.

Note that since many projects might already have been processed with VarScan2, it can be convenient to be able to import such results. For this purpose a simple function is provided within the R package, to convert the output of the somatic and copynumber programs of the VarScan2 suite into the seqz format.

Exploring the seqz file and depth ratio normalization details

After the aligned sequence data have been pre-processed, the sequenza R package handles all the normalization and analysis steps. Thus, the remainder of this vignette will take place in R.

Let's launch R

```
R
```

You should now see the R prompt identified with >.

Load the sequenza package

```
library("sequenza")
```

Read the seqz file

The seqz file can be read all at once, but processing one chromosome at a time is less demanding on computational resources, especially while processing whole genome data, and might be preferable in case of limited computational resources.

Why could we process each chromosome individulally ?

_____

_____

The function `sequenza.extract` is designed to efficiently access the seqz file and take care of normalization steps. The arguments enable customization of a set of actions listed below:

- binning depth ratio and B allele frequency in a desired window size (allowing a desired number of overlapping windows);
- performing a fast, allele specific segmentation using the copynumber package;
- filter mutations by frequency and noise.

```
data.file = "sequenza/C0053.seqz.bin500.gz"
seqzdata = sequenza.extract(data.file)
```

After the raw data is processed, the size of the data is considerably reduced.Typically, the R object resulting from `sequenza.extract` can be stored as a file of a few megabytes, even for whole genome sequencing data.

Inference of cellularity and ploidy

After the raw data is processed, imported into R, and normalized, we can apply the parameter inference implemented in the package. The function `sequenza.fit` performs the inference using the calculated B allele frequency and depth ratio of the obtained segments.

```
CP.example = sequenza.fit(seqzdata)
```

Results of model fitting

The last part of the workflow is to apply the estimated parameters. There is an all-in-one function that plots and saves the results, giving control on file names and output directory

```
sequenza.results(sequenza.extract = seqzdata,
 cp.table = CP.example,
 sample.id = "C0053",
 out.dir="sequenza/results")
```

We can now quit R and explore the generated results

```
q("yes")
```

Sequenza Analysis Results and Visualisation

One of the first and most important estimates that Sequenza provides is the tumour cellularity (the estimated percentage of tumour cells in the tumour genome). This estimate is based on the B allele frequency and depth ratio through the genome and is an important metric to know for interpretation of Sequenza results and for other analyses.

Lets look at the cellularity estimate for our analysis by opening model fit.pdf with the command:

```
evince sequenza/results/C0053_model_fit.pdf
```

What is the graph telling us ?

_____

_____

Close the PDF window to resume the Terminal prompt.

Let's now look at the CNV inferences through our genomic block. Open the genome copy number visualisation file with:

```
evince sequenza/results/C0053_genome_view.pdf
```

This file contains three "pages" of copy number events through the entire genomic block. The first page shows copy numbers of the A (red) and B (blue) alleles, the second page shows overall copy number changes and the third page shows the B allele frequency and depth ratio through genomic block.

What are these graphs telling us ?

We can see how this is a very easy to read output and lets us immediately see the frequency and severity of copy number events through the genome.

Let's compare the small genomic block we ran with the same output from the entire genome which has been pre-computed for you. This is located in the `saved_results/preComputed/` folder and contains the same 13 output files as for the small genomic block.

What are these graphs telling us ?

CNV Visualisation/Confirmation in IGV

Let's see if we can visualise the CNV events. We will now open IGV and see if we can observe the predicted increase in copy number alterations within our genomic region.

```
igv &
```

IGV will take 30 seconds or so to open so just be patient.

For a events of this size (several Mb), we should not be able to easily observe it just by looking at the raw read alignments. In order to see coverage at large scale I rpre-generate the tdf file of each bam files. This means that we can aggregate the average read depth over relatively large chunks of the genome and compare these values between the normal and tumour genomes.

Once IGV is open just load the normal et tumor bam files and zoom on the region `2:100000000-170000000`

What IGV profiles are telling us ?

How can you explain the peak and drop observed in both nromal and tumor ?

_____

_____

Are IGV profiles in concordence with sequenza results ?

_____

_____

If we look at the tumor profiles wecan see that the 3 copies state correspond to a mean coverage of 60x, the 2 copies to 50x and the 1 copy to 40x.

How could you explain these values ?

_____

_____

SNParray data analysis

SNParray analysis are very similar to NGS data analysis and it is always good to use 2 different technologies to confirm your findings.

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 ?

_____

_____

What are the steps to proceed this analysis ?

_____

_____

Prepare the Environment

We will use a dataset derived from whole genome sequencing of a clear-cell renal carcinoma patient (Kidney cancer)

```
## set environement
cd $HOME/ebicancerworkshop201607/CNV/SNParray
```

Software requirements

These are all already installed, but here are the original links.

- R
- ASCAT R package*

Original Setup

The initial structure of your folders should look like this:

```
<ROOT>
|-- C0053/                    # fastqs from the center (down sampled)
    `-- normal                      # The blood sample directory
        `-- normal_BAF.tsv          # Beta Allele frequency file
        `-- normal_LRR.tsv          # Log R Ratio file
    `-- tumor                       # The tumor sample directory
        `-- tumor_BAF.tsv           # Beta Allele frequency file
```

```
        `-- tumor_LRR.tsv          # Log R Ratio file
|-- saved_results                  # Pre-computed files
`-- scripts                # cheat sheet and builded ASCAT R package
```

SNP data analysis for CNV detection

In our case, the data are in LRR and BAF format so we skip the first processing steps

Probe filtering

This steps aim to filter out SNPs which are found to be homozygous for both tumor and normal.

First let's launch R:

```
R
```

Load the ASCAT in R from the folder scripts

```
library(ASCAT)
```

Load the data into an ASCAT object

```
ascat.bc = ascat.loadData("C0056/tumor/tumor_LRR.tsv","C0056/tumor/tumor_BAF.tsv", "C0056/no
```

Plot the raw data

```
ascat.plotRawData(ascat.bc)
```

Segmenetation of LRR and BAF signal

The next step allows to perform the segmentation of both LRR and BAF signal. The main points of this segmentation is estimate a models of segmentation that should fit between the 2 signals

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

Plot the result off the segmentation

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

Estimation of the model paremters

This function will use the computed segmentation model and estimate the following sample paramters: 1. aberrant cell fraction (cellularity) 2. tumor ploidy 3. absolute allele-specific copy number calls (for each allelic probes of the SNP)

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

Next save these estimates into a file

```
params.estimate=data.frame(Sample=names(ascat.output$aberrantcellfraction),Aberrant_cell_fra
write.table(params.estimate,"sample.Param_estimate.tsv",sep="\t",quote=F,col.names=T,row.nam
```

CNV calling from segments

The last step woll determine the copy number by simply counting the total number of allele reported to the sample general ploidy.

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

And we finally save the CNV results into a file

```
write.table(output.table[output.table$CNA != ".",],"sample_CNVcalls.tsv",quote=F,sep="\t",co
q(save="yes")
```

What are these results telling us ?

_____

_____

The ASCAT analysis have been done on the same sample than the sequenza analysis

Are the two analyses concordent ?

Aknowledgments

I would like to thank and acknowledge Louis Letourneau and Dr. Velimir Gayevskiy for this help and for sharing his material. The format of the tutorial has been inspired from Mar Gonzalez Porta. I also want to acknowledge Joel Fillon, Louis Letrouneau (again), Robert Eveleigh, Edouard Henrion, Francois Lefebvre, Maxime Caron and Guillaume Bourque for the help in building these pipelines and working with all the various datasets.