EXCAVATOR2: detecting copy number variants from Whole-Exome Sequencing data

version 1.1

Romina D'Aurizio, Lorenzo Tattini and Alberto Magi

August 17, 2016

Contents

1 Introduction

1.1 Scope

Copy Number Variants (CNVs) are structural rearrangements involving DNA segments of at least 50bp that can be present with an altered copy number compared to the reference genome. EXCAVATOR2 allows to identify genomic CNVs (overlapping or non-overlapping exons) from Whole-Exome Sequencing data by integrating the analysis of In-targets and Off-targets reads.

1.2 Citation

Please cite the following articles when using EXCAVATOR2:

- D'Aurizio R, *et al.*"Enhanced copy number variants detection from whole-exome sequencing data using EXCAVATOR2."Nucl. Acids Res. first published online August 9, 2016 doi:10.1093/nar/gkw695

- Magi A, *et al.*"EXCAVATOR: detecting copy number variants from whole-exome sequencing data." Genome Biology 2013,14:R120 doi:10.1186/gb-2013-14-10-r120.

1.3 About EXCAVATOR2

EXCAVATOR2 is a collection of bash, R and Fortran scripts and codes that analyses WES data to identify CNVs. It extends the Read Count approach to the whole genome sequence and exploits the Shifting Level Model (SLM) algorithm [1][2] to segment the two combined profiles. Last the FastCall algorithm [3] allows to classify each segmented regions into five possible states (two-copy deletion, one-copy deletion, normal, one-copy duplication and multiple-copy amplification).

1.4 How to get help

All package documentation can be found in the `docs` subfolder of EXCAVATOR2.
For any questions about the tool please email us at: romina.daurizio@gmail.com and albertomagi@gmail.com

1.5 Conventions

In this manual the commands, command line options, executable programs or anything else dealing with your shell, `is written with this character`. The shell prompt is presented with the symbol >. e.g.,

```
> cd $HOME/EXCAVATOR2
```

Furthermore, we use the following syntax:

[1] Magi A, *et al.* A shifting level model algorithm that identifies aberrations in array-cgh data. Biostatistics, 11(2):265Ð80, Apr 2010

[2] Magi A, *et al.* Detecting common copy number variants in high-throughput sequencing data by using jointSLM algorithm. Nucleic Acids Res, 39(10):e65, May 2011

[3] Benelli M, *et al.* A very fast and accurate method for calling aberrations in array-cgh data. Biostatistics, 11(3):515Ð8, Jul 2010.

```
lib>  mycommand
```

whenever we want to run a command from a specific folder (in this case the `lib` folder.)

In case we have to write a *very* long command line we will split it with "\\" going to a new line in the text.

1.6 Before to start

Before you run EXCAVATOR2 please consider that:

- *the coherence of chromosomes annotation among all files*. Namely, .bam files and the input .bed file with target design must show same chromosomes encodes (i.e. both as `chrN` or both as `N`);
- *target .bed file must be sorted* by chromosome number and region coordinate and *no overlapping regions must be present* in your target file. In case some overlapping regions are found you should merge those overlapping each other into a single region.
  You can use: `sort -k1,1 -k2,2n *.bed | bedtools merge`
- *target initialisation can not be run in parallel on a machine.* E.g., if the user has to initialise two targets running two `TargetPerla.pl`, the second must be run once the first has successfully completed all the calculations!

Moreover, EXCAVATOR2 program folder contains several subfolders:

- *changing folders organisation or file names will result in package malfunctions.*

1.7 Quick start

```
EXCAVATOR2> perl TargetPerla.pl SourceTarget.txt myTarget.bed MyTarget_w50K \\
             50000 hg19
EXCAVATOR2> perl EXCAVATORDataPrepare.pl ExperimentalFilePrepare.w50K.txt \\
             --processors 6 --target MyTarget_w50K --assembly hg19
EXCAVATOR2> perl EXCAVATORDataAnalysis.pl ExperimentalFileAnalysis.w50K.txt \\
             --processors 3 --target MyTarget_w50K --assembly hg19 \\
             --output /.../OutEXCAVATOR2/Results_MyProject_w50K --mode paired \\
```

2 EXCAVATOR2 Installation Guide

2.1 Requirements

EXCAVATOR2 was conceived for running on 64-bit UNIX desktop machines with at least 4 CPUs and 4 GB RAM.

In order to work properly EXCAVATOR2 needs R (version $\geq$ 2.14.0) and the Hmisc library, SAMtools (version $\geq$ 0.1.17), and Perl (version $\geq$ 5.8.8) to be correctly installed on your system. The former

can be downloaded at CRAN (http://cran.r-project.org), while SAMtools can be found at SourceForge (http://samtools.sourceforge.net). Perl is native in almost any Unix machine. Before installing EXCAVATOR2 make sure they are all installed on your machine and their executable files have been exported in your `PATH`. If you experience any problem with any of them you should contact your system administrator. Installation of any of these softwares requires superuser privileges.

To check for R, SAMtools and Perl you can type on your shell the following commands:

```
> R
```

Press CTRL+D to quit R.

```
> samtools
```

```
> perl -v
```

2.2 Installing EXCAVATOR2

In order to install EXCAVATOR2:

1. Open the compressed EXCAVATOR2 package.
2. Move the uncompresssed EXCAVATOR2 folder and its subfolders (any alteration of the folders tree will result in EXCAVATOR2 malfunction) to any folder you can access on your computer. The path to the EXCAVATOR2 folder (the program path) will be required for the calculations performed by EXCAVATOR2.
3. With the command `R CMD SHLIB`, compile Fortran subroutines in the folder `/.../EXCAVATOR/lib/F77`. There are two .f files. Both must be compiled. Compilation is thus as easy as:

   ```
   F77>  R CMD SHLIB F4R.f
   F77>  R CMD SHLIB FastJointSLMLibraryI.f
   ```

2.2.1 For Mac OS X users

Mac OS X users may get compilation errors compiling Fortran subroutines with GNU Compiler Collection (GCC) shipped with Xcode. If any error occur, please download Universal GNU Fortran available at CRAN. Moreover, you need to delete default `bigWigAverageOverBed` binary file and rename the `bigWigAverageOverBed.macOSX.x86_64` file to `bigWigAverageOverBed` in the folder `lib/OtherLibrary/` .

2.3 Exporting EXCAVATOR2 folder

You may also want to run your analyses from any folder in your filesystem. In this case you need to export the path to EXCAVATOR2 folder. E.g. if you are a bash user you can type:

```
>export PATH=$PATH:/.../any/folder/to/EXCAVATOR2
```

2.4 System Subfolders and Files

EXCAVATOR2 program folder contains several subfolders:

docs - All package documentation can be found in the `docs` subfolder of EXCAVATOR2.

data - The `data` subfolder contains target-related data and centromere positions for any assembly you initialized. Once you initialize a target file (with `TargetPerla.pl`) all the data produced will be stored in a `data` subfolder as .RData compressed files. Please note that data calculated from different assemblies are organized in different subfolders.

lib - The `lib` subfolder includes all the scripts, codes and subroutines necessary to perform calculations.

Perl files that manage the analysis workflow are stored in the main program folder EXCAVATOR2. Furthermore, you will find a support file (`SourceTarget.txt`) for TargetPerla.pl and a file defining the parameter used by HSLM and FastCall algorithms (`ParameterFile.txt`).

The EXCAVATOR folder can be safety used as a working directory for Perl modules execution.

3 EXCAVATOR2 Workflow

The entire workflow of EXCAVATOR2 can be executed by means of 3 Perl scripts (or *modules*). Each module of EXCAVATOR2 can be invoked by means of a Perl script:

1. TargetPerla.pl
2. EXCAVATORDataPrepare.pl
3. EXCAVATORDataAnalysis.pl

In the following sections you will find a brief description of each module and how to invoke.

3.1 TargetPerla.pl

TargetPerla.pl is the module for target initialization. It creates a pseudo-target file with coordinates from the user-specified target input file (.bed) and it calculates GC content and mappability values for both In-target and Off-target regions.

TargetPerla.pl requires 5 arguments: the path to a source file (e.g. `SourceTarget.txt`), the path to the target input file, a "target name", the window size (i.e. 10000, 20000 or 500000) and the assembly (allowed options are: hg19 and hg38).
Example of cmd is:

```
perl TargetPerla.pl SourceTarget.txt myTarget.bed MyTarget_w50000 50000 hg19
```

The default source file is `SourceTarget.txt` that is placed in the main program folder. SourceTarget.txt is a space delimitated file that contains the absolute paths to a bigWig file (`.bw`) for the calculations of mappability and a `.fasta` file of the user-defined reference assembly for GC-content calculations.

The bigWig file is a binary file reporting information about mappability, referred to a reference assembly. Mappability files for hg19 and hg38 assemblies are provided with the EXCAVATOR2 package and they are present in the `data` folder. They were created by using the GEM mapper aligner[4] belonging to the GEM suite (`http://gemlibrary.sourceforge.net/`), allowing up to two mismatches and considering sliding windows of 100mer. For any further details concerning mappability calculations please refer also to the original paper [5].

Target input file (.bed, .txt or any plain text file) must be tab-delimited containing at least three fields with:

```
chromosome  start  end
```

Setting the target name as "MyTarget", this module will create a folder (if you are using the hg19 assembly) `/.../EXCAVATOR2/data/targets/hg19/TargetName` containing all target-related .RData subfolders and files.

3.2 EXCAVATORDataPrepare.pl

EXCAVATORDataPrepare.pl is a Perl script managing RC calculations, data normalization and data analysis on multiple .bam files. It requires one argument and three command-line options to run properly. This job can be parallelised by choosing the number of processors to use.
An example of the cmd is:

```
EXCAVATOR2> perl EXCAVATORDataPrepare.pl ExperimentalFilePrepare.w50000.txt \\
  --processors 6 --target MyTarget_w50000 --assembly hg19
```

The argument is the path to an input text file (e.g. ExperimentalFilePrepare.window.txt) that you have to create with details about all .bam files you want to analyse. The options concern the number of processors to use (–processors), the name you want to use for your target (–target) and the human assembly you used for the mapping (–assembly). Available options for assembly are `hg19` and `hg38`.

Before running EXCAVATORDataPrepare.pl you need to create a space delimited file with three fields: the absolute path to the .bam file you want to analyse, the path to the main sample output folder and the sample name. The sample name will be used as a prefix/suffix for output files. Each row in the file contains details about one sample.
An example of a well-formatted ExperimentalFilePrepare.window.txt file with 12 samples is reported in Figure 1.

For each sample, the main output folder specified in the second filed of the ExperimentalFilePrepare.window.txt will be created. This folder will contain three subfolders (`RC`, `RCNorm` and `Images`) with, respectively, the calculated raw WMRC for In- and Off-target regions, the median normalized

[4] Derrien *et al.* Fast Computation and Applications of Genome Mappability. PLoS One 2012, 7
[5] Koelher *et al.* The uniqueome: a mappability resource for short-tag sequencing. Bioinformatics, 27(2):272Đ4, Jan 2011

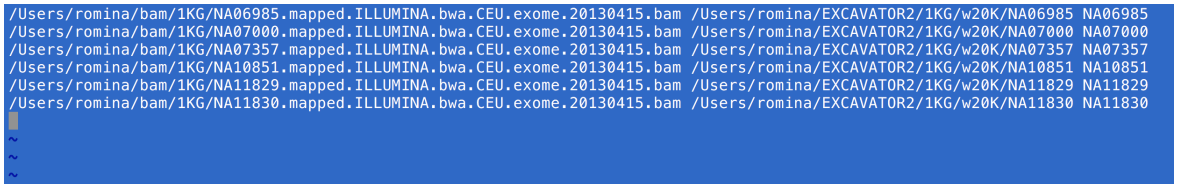

Figure 1: A typical well-formatted input file for EXCAVATORDataPrepare.pl module.

WMRC, and the plots showing the influence of GC content percentage and mappability on WMRC pre- and post-normalization.

3.3 EXCAVATORDataAnalysis.pl

EXCAVATORDataAnalysis.pl is a multi-threading Perl script performing the segmentation of the WMRC (see ..paper for details) by means of the Shifting Level Model algorithm and exploits FastCall algorithm to classify each segmented region as one of the five possible discrete states (2-copy deletion, 1-copy deletion, normal, 1-copy duplication and N-copy amplification). The FastCall calling procedure takes into account sample heterogeneity and exploits the Expectation Maximization algorithm to estimate the parameters of a five gaussian mixture model and to provide the probability that each segment belongs to a specific copy number state.

EXCAVATORDataAnalysis.pl requires one argument and four command-line options to run properly. This is an example of the cmd:

```
EXCAVATOR2> perl EXCAVATORDataAnalysis.pl ExperimentalFileAnalysis.w50K.txt \\
             --processors 6 --target MyTarget_w50K --assembly hg19 \\
             --output /.../OutEXCAVATOR2/Results_MyProject_w50K --mode ...
```

The first argument is a space delimited text file that you need to create and provide. It contains three fields, the second and the third are the same as in ExperimentalFilePrepare.window.txt, while the first is a label which specifies how to handle and compare the samples. Indeed, EXCAVATOR2 can be exploited to identify CNVs in population and cancer genomic studies. You need to specify which kind of analysis you want to perform by selecting “pooling” or “paired” for the mode option. In the first case, all test samples will be compared with the same global control sample which results from summing, region by region, the WMRC of all control samples. Which samples are the test/control need to be specified in the first field of the ExperimentalFileAnalysis.window.txt file using, respectively, `T` or `C` labels.
An example of a well formatted file for “pooling” design is shown in Figure 2.

In case you want to identify somatic CNVs in matched tumor/control samples, you need to select the “paired” mode. In the ExperimentalFileAnalysis.window.txt, test samples must be marked with a TX label (where X is an integer number) while control samples must be marked with CY (where Y is an integer number) and samples analysis will be performed comparing each test sample with a particular control sample, with the matching condition being X = Y. Thus sample labeled T1 is compared with control sample C1 and so on. An example is reported in Figure 3.

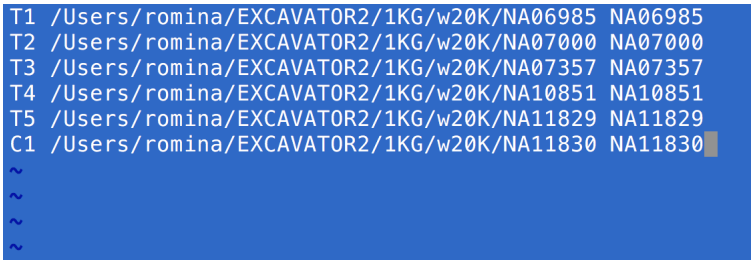

Figure 2: A typical well-formatted input file for EXCAVATORDataAnalysis.pl module and "pooling" mode.

Figure 3: A typical well-formatted input file for EXCAVATORDataAnalysis.pl module and "paired" mode.

Please note that for somatic analysis the number of test samples must match the number of control samples.

The output folder, which is command-line specified (`--output` option), contains two subfolders: `Plots` and `Results` with results for all tested samples in separated sub-subfolders. The `Plots` folder collects .pdf files reporting a scatter plot of the segmented data and statistically significant regions (chromosome by chromosome) for each test sample (follow `/.../Plots/SampleName/`). An example is reported in Figure 4.

The `Results/SampleName/` folder contains .txt and .vcf files with the results produced by HSLM and FastCall. Precisely, FastCall results are summarized in `FastCallResults_SampleName.txt` files (only for test sample). These files report: chromosome, start position, end position, median *log2*-ratio in the segment copy number fraction, copy number value, copy number state and call probability (see Figure 5 for details). Concerning copy number state values: 2-copies deletion are encoded with "-2", while 1-copy deletions are reported as "-1" calls. 1-copy and multiple-copies duplication are reported as "1" and "2" respectively. As an example, part of the file is shown in Figure 5.
For details about FastCall output see [3].

EXCAVATOR2 produces also a .vcf file (`ExcavatorRegionCall_SampleName.vcf`) with details about identified CNVs. The VCF (Variant Call Format) is a text file of nine fields used to store sequence variations. Each row contains details about a CNV: the starting breakpoint is specified in POS field, the end and the length of the CNV are in the INFO field (END and SVLEN id). Details from EXCAVA-

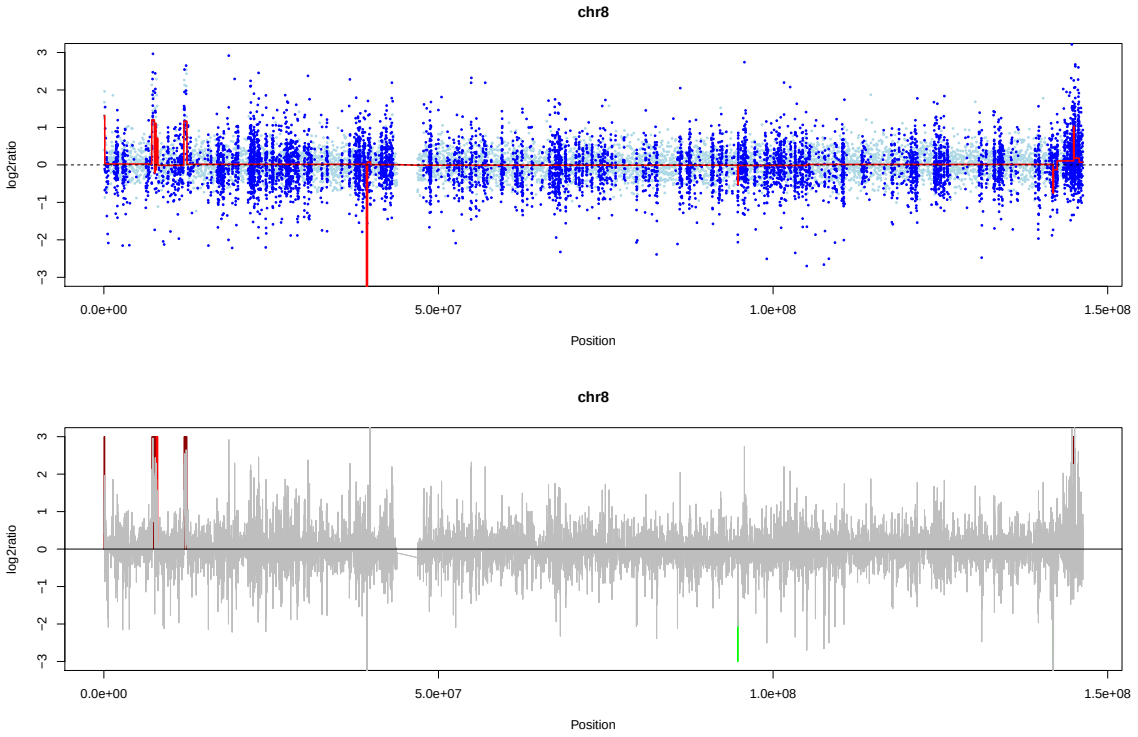

Figure 4: Plots reporting results chromosome by chromosome

TOR2 analysis are store in ninth filed in accordance with the FORMAT field (`GT:CN:CNF:FCL:FCP`). It includes the genotype, the copy number genotype, the copy number genotype fraction, the label and posterior probability inferred by FastCall.

Same, window by window, data is also reported in `ExcavatorWindowCall_SampleName.vcf` while HSML results are in `HSLMResults_SampleName.txt` file.

3.4 Parameters setting

EXCAVATOR2 modules implement HSML and FastCall whose running parameters are stored in the `ParameterFile.txt` in the main folder of the package. For HLSM algorithm the user can set the value of `Omega` in the range $0.0-1.0$, `Theta` $(0.0-1.0)$ and `D_norm`. We suggest to use `Omega` $(0.1-0.5)$, `Theta` $(10^{-7}-10^{-3})$ and `D_norm` $(10^{4}-10^{6})$. For FastCall algorithm the user may set the parameters: `Cellularity` $(0.0-1.0)$ is the fraction of tumor cells, `Threshold d` (recommended $0.2-0.6$) is the lower bound for the truncated gaussian of the neutral (2 copies) state, `Threshold u` (recommended $0.1-0.4$) is the upper bound for the truncated gaussian of the neutral (2 copies) state. For further informations about `Omega` and `Theta` see [2]. For details concerning `D_norm` and `Cellularity` see [6].

[6] Magi A, *et al.* EXCAVATOR: detecting copy number variants from whole-exome sequencing data. Genome Biology 2013, 14:R120 doi:10.1186/gb-2013-14-10-r120. A link to the paper can be found

Chromosome	Start	End	Segment	CNF	CN	Call	ProbCall
chr1	17013801	17272946	0.564376329908218	2.95749623078752	3	1	0.946494901463117
chr1	110230739	110251210	-8.90774029637354	0.00416421335715763	0	-2	0.999999996224918
chr1	144160224	144224227	-0.592136964722077	1.32671918026281	1	-1	0.958668460283003
chr1	148253516	148328426	-0.596863521684494	1.32237969576257	1	-1	0.959668659844766
chr1	196716221	196801147	0.755008193955337	3.37528235477681	3	1	0.93180065155923
chr2	98139739	98162369	1.79754142036572	6.95254617101182	7	2	0.99999999998189
chr3	195404538	195475973	0.625167930785982	3.08478070070835	3	1	0.950271699768718
chr4	9160811	9471046	1.12113964538425	4.35037463670225	4	2	0.880925067267188
chr5	69529518	69808390	-0.781551424690562	1.16348174519222	1	-1	0.971973296600072

Figure 5: FastCall results reports chromosome [Chromosome], start [Start], end [End] (positions of copy number events detected), segment log2-ratio median value [Segment], copy number fraction [CNF], copy number value (rounded to nearest integer) [CN], copy number state [Call], call probability [ProbCall].

For details concerning `Threshold d` and `Threshold u` see [3].