EXCAVATOR version 2.1

Lorenzo Tattini, Matteo Benelli, Alberto Magi

November 15, 2013

Contents

1 Preamble

1.1 About this Document

Here you will find anything we wanted you to know about installing and running EXCAVATOR on your computer. If this is not enough for you please feel free to contact us for more details.

For details concerning the package please refer to (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). Here you can find a link to the paper.

1.2 About the User

In writing this manual and EXCAVATOR's scipts we have assumed the following about you:

1. You are well trained in UNIX.
2. You have a UNIX computer.
3. You know — at least — something about Next Generation Sequencing (NGS).
4. You want EXCAVATOR to ask your UNIX computer to perform calculations with your NGS data.

1.3 About the Authors

Here (this is a hyperlink) you can find a link to web page (hopefully up-to-date) of our lab. You can also email us at:

lorenzotattini@gmail.com
albertomagi@gmail.com

1.4 Conventions

This is the standard character for anything we have to write, with the exception of commands, command line options, executable programs or anything else dealing with your shell, `which is written with this character`. Furthermore, we represent any shell prompt with the symbol `>`. E.g., if we want you to change your working directory to `$HOME/EXCAVATOR` we may ask to type on your terminal (and then press `Enter`):

```
> cd $HOME/EXCAVATOR
```

In case we have to write a *very* long command line we will split it, going to a new line in the text. E.g., you may have to move a file to a far away folder:

```
> mv /Programing/Data/Raw/BamFiles/PreciousExperiments/Exomes/
Tuscan/ExTc.bam /Another/Far/Away/Folder/ExTc.bam
```

In this case we will complete the command line in the second text line. Furthermore, the following syntax:

```
lib> mycommand
```

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

1.5 About EXCAVATOR

EXCAVATOR is a collection of bash, R and Fortran scripts and codes that allows for the analysis and plotting of exomic data. All the calculations are executed by means of two Perl scripts (or *modules*): TargetPerla.pl and ReadPerla.pl.

EXCAVATOR was conceived for running on UNIX desktop machines with at least 4 CPUs and 4 GB RAM.

2 EXCAVATOR Installation Guide

2.1 Requirements

In order to work properly EXCAVATOR 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 EXCAVATOR 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.

In order 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 EXCAVATOR

In order to install EXCAVATOR:

1. Open the compressed EXCAVATOR package.
2. Move the uncompresssed EXCAVATOR folder and its subfolders (any alteration of the folders tree will result in EXCAVATOR malfunction) to any folder you can access on your computer. The path to the EXCAVATOR folder (the program path) will be required for the calculations performed by EXCAVATOR.

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 Compiling Fortran Subroutines in Mac OS X

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.

3 System Subfolders and Files

EXCAVATOR program folder contains several subfolders. Changing folders organization or file names will result in package malfunctions. The first-level subfolders are reported in Figure 1.

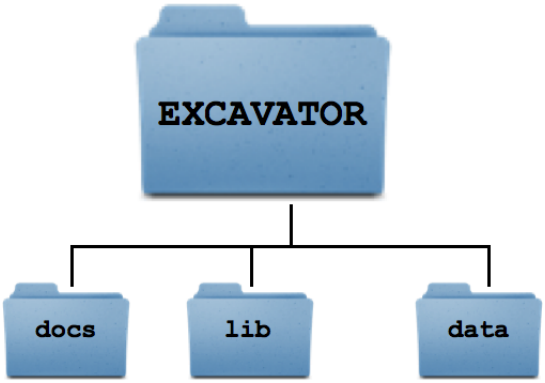

Figure 1: EXCAVATOR first-level subfolders.

All package documentation can be found in the `docs` subfolder. 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. The `lib` subfolder includes all the scripts, codes and subroutines necessary to perform calculations.

Both Perl files that manage scripts execution are stored in the main program folder EXCAVATOR. Furthermore, you will find two support files for Perl modules and a file defining the parameter used by HSLM and FastCall algorithms.

`SourceTarget.txt` is a support file for TargetPerla.pl, while `ReadInput.txt` is a support file for ReadPerla.pl. The EXCAVATOR folder can be safety used as a working directory for Perl modules execution.

4 EXCAVATOR Workflow

Each module of EXCAVATOR can be invoked by means of a Perl script. Here you will find a brief description of the operations performed by each Perl module.

TargetPerla.pl is the module devoted to target initialization. It loads mappability (from a .wig file) and GC content data (from a .fasta file) from a user-defined reference assembly, TargetPerla.pl calculates GC content and mappability values in the regions of a user-specified target file. Reference assembly can be selected, e.g. with the option `--assembly hg18`. If no assembly is specified the default reference assembly (hg19) is loaded. This is equivalent to specify the option `--assembly hg19`. Target input file (.bed, .txt or any plain text file) must be tab-delimited and the fourth (additional) field must be provided. If your target input file reports only the three required fields you must add the forth (also by adding a dummy field). TargetPerla.pl requires, as arguments, the path to a source file (`SourceTarget.txt`), the path to the target input file and a "target name". Setting the target name as "MyTarget", all data calculated will be saved in the "MyTarget" folder in (if you are using the hg19 assembly) `/.../EXCAVATOR/data/targets/hg19/MyTarget`. The default source file is `SourceTarget.txt` and is placed in the main EXCAVATOR folder. `SourceTarget.txt` contains the paths to a .wig file (for the calculations of mappability) and a .fasta file (for GC-content calculations). Paths must be space delimited.

The .wig file is a text file reporting informations about mappability (referred to a reference assembly; don't forget to check its compatibility with the reference your using for your analysis and which was used for read mapping). You can download it from http://grimmond.imb.uq.edu.au/uniqueome/downloads/: for hg18 you need the hg18_uniqueome.coverage.base-space.25.1.Wig.gz file, while for hg19 you need the hg19_uniqueome.coverage.base-space.25.1.Wig.gz file. After the download, unzip it (in any folder you prefer) and place its absolute path in the SourceTarget.txt file. For any detail concerning mappability calculations please refer to the paper by Koelher et al. (Bioinformatics (2011) 27 (2): 272-274. doi: 10.1093/bioinformatics/btq640)

ReadPerla.pl is a Perl script that manage read count (RC) calculations, data normalization and data analysis. This module supports calculations on multiple .bam file. ReadPerla.pl requires two arguments and one command-line option (the "mode" option) to run properly. The first argument is the path to an input text file. The default file name is `ReadInput.txt`. It should be placed in EXCAVATOR main folder. The last argument is the path to the main output folder. An example of a well-formatted `ReadInput.txt` file is reported in Figure 2. This example deals with five experiments designed on the target "SureSelect50" which was initialized on reference assembly hg19 (fields 1 and 2). Field 3 reports the path to the .bam files, while field 4 is the sample name. The sample name will be used for naming sample output folders and as a prefix/suffix for output files. If the sample output folder does not exist, ReadPerla.pl will create it. Samples' output folders will be placed under the *main* output folder which is command-line specified. Single-bam results will be placed in three subfolders (`RC`, `RCNorm` and `Images`) under the output folder specified in field 4 of `ReadInput.txt` (see Section 4.4 for further details concerning output folders structure). Field 5 is the label which specifies how to handle the sample. In the "somatic" mode, test samples must be marked with a TX label (where X

is an integer number) while control samples are marked with CY (where Y is an integer number). When somatic mode option is selected, samples analysis is performed comparing each test samples 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. On the contrary, selecting "pooling" mode all test samples are compared with the same global control sample. The global control sample results from summing — region by region — RC of all control samples. The option `--mapq <integer>` allows the user to select mapping quality for .bam file filtering. If omitted default value is 0.

Finally, `ParameterFile.txt` is a plain text file. It defines the values of the parameters for both HSLM and FastCall algorithms.

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 (Magi A, et al. (2011) Detecting common copy number variants in high-throughput sequencing data by using JointSLM algorithm. Nucleic Acids Res., Feb 14).

For details concerning `D_norm` and `Cellularity` see (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 here.

For details concerning `Threshold d` and `Threshold u` see (Benelli M, et al. A very fast and accurate method for calling aberrations in array-CGH data. Biostat (2010) 11 (3): 515-518. doi: 10.1093/biostatistics/kxq008).

4.1 Before Planning Analyses Beware...

Before you run EXCAVATOR you should take care a fundamental aspect. Chromosomes in the input .bed file (for target initialization) and .bam files must be coherently encoded. Namely, both files must show the same chromosome tags (i.e. both encoding chromosome 11 as `chr11` or both encoding chromosome 11 as `11`).

Figure 2: A typical well-formatted input file.

Please also check the presence of the fourth field in your target input .bed file. Furthermore, target files must sorted by chromosome number and region cohordinate. 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.

4.2 TargetPerla.pl Examples

To run TargetPerla.pl:

```
EXCAVATOR> perl TargetPerla.pl SourceTarget.txt
/home/.../SureSelect50.bed TargetName
```

This will create a folder `/.../EXCAVATOR/data/targets/hg19/TargetName` containing all target-related .RData subfolders and files.

On the other hand if you want to perform the same target initialization against assembly hg18 you may use:

```
EXCAVATOR> perl TargetPerla.pl SourceTarget.txt
/home/.../SureSelect50.bed TargetName --assembly hg18
```

Following the path `/.../EXCAVATOR/data/targets/hg18/TargetName` will lead you to your hg18 initialized .RData files.

4.3 ReadPerla.pl Examples

Running ReadPerla.pl is quite simple as all the mandatory inputs are reported in `ReadInput.txt`. ReadPerla.pl can be executed with the following command:

```
EXCAVATOR> perl ReadPerla.pl ReadInput.txt
/home/.../OutputFolder --mode somatic
```

while with the following :

```
EXCAVATOR> perl ReadPerla.pl ReadInput.txt
/home/.../OutputFolder --mode pooling
```

you can run a pooling analysis. Please note that for somatic analysis the number of test samples must match the number of control samples.

4.4 The Output Folders

The output folder — which is command-line specified — contains several subfolders (`Data`, `Plots` and `Results`). The `Data` folder collects single-bam data (read count in the `RC` folder, normalized RC in `RCNorm`). Thus you will find a sub-subfolder for each sample/control bam file. RC pre- and post-normalization can be found in the `Image` folder. An example of the results of data normalization is reported in Figure 3.

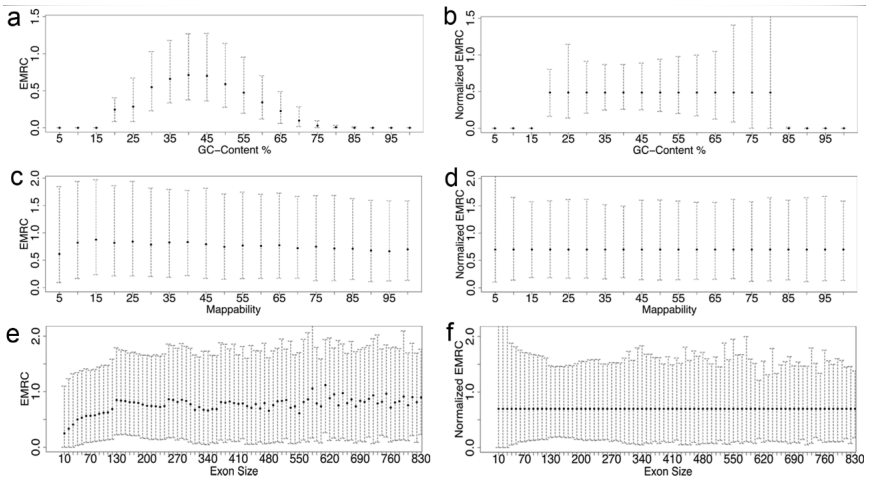

Figure 3: RC data and normalized RC. EXCAVATOR produces six .pdf files, like those reported in panel a-f, for each sample/control. All the files can be found in the "Images" folder of each sample/control.

	Chromosome	Position	Log2R	SegMean
2	chr1	14596	0.107306387010508	-0.00857414781475019
3	chr1	15004	0.75408354879232	-0.00857414781475019
4	chr1	15871	-0.487271818479439	-0.00857414781475019
5	chr1	16686	0.592125337971994	-0.00857414781475019
6	chr1	16956	0.121470295107833	-0.00857414781475019
7	chr1	17300	0.238471308931594	-0.00857414781475019
8	chr1	17674	0.820959825915152	-0.00857414781475019
9	chr1	69549	0.375368108509863	-0.00857414781475019
10	chr1	663162	0.561526340570629	-0.00857414781475019
11	chr1	762244	0.477507234263586	-0.00857414781475019
12	chr1	763109	-0.0636803902495283	-0.00857414781475019
13	chr1	783110	-0.367541956267375	-0.00857414781475019
14	chr1	787398	0.0373337981696762	-0.00857414781475019
15	chr1	788098	0.309406382297003	-0.00857414781475019
16	chr1	788836	0.357573632218451	-0.00857414781475019
17	chr1	789348	0.101941545827792	-0.00857414781475019

Figure 4: HSLM results file showing chromosome, position, *log2*-ratio and segment *log2*-ratio mean values.

4.4.1 Segmentation and Calling Algorithms

On the contrary, folders `Results` and `Plots` contain data related to test samples only. The former collects .txt files with the results produced by HSLM and FastCall.

In order to detect the boundaries of the genomic regions with altered DNA copy number HSLM perform segmentation on *log2*-ratio data. Calculations of the logarithm of the ratio between RC (exon-corrected) of test and control samples (*log2*-ratio) generates a signal which is mathematically similar to those obtained by RC analysis (see Magi et al., 2012): deletions (or amplifications) are identified as signal decrease (or increase) across multiple consecutive targeted regions. HSLM results are reported in a `HSLMResults_SampleName.txt` file (which can be found following `/.../Results/SampleName/`). An example is reported in Figure 4.

In order to classify each segmented region into a biologically motivated state (2-copies deletion, 1-copy deletion, normal, 1-copy duplication and multiple-copies duplication) we used the FastCall algorithm (Benelli et al., 2010). EX-

CAVATOR calling procedure models the mean of each segment as a mixture of five truncated normal distributions (taking into account sample heterogeneity by means of the `Cellularity` parameter).

FastCall results are summarized in `FastCallResults_SampleName.txt` files (only for test sample) and can be found following `/.../Results/SampleName/`. These files report: chromosome, start position, end position, median *log2*-ratio in the segment (columns 1-4 respectively). 2-copies deletion are encoded with "-2" in `FastCallResults_SampleName.txt` while 1-copy deletions are reported as "-1" calls. 1-copy and multiple-copies duplication are reported as "1" and "2" in the output file.

	Chromosome	Start	End Segment	Call	
2	chr1	12819331	13910423	0.423267159958217	1
3	chr1	17085612	17274922	0.711300362698684	1
4	chr1	46152193	46646804	0.568630289491069	1
5	chr2	96780257	97833468	-0.988005450724877	-1
6	chr2	114036453	114384561	0.419588961672269	1
7	chr4	86860	493360	0.360878537345459	1
8	chr5	98195724	99715865	0.41813742810579	1
9	chr5	180377404	180429728	-7.65953523767341	-2
10	chr6	29718755	29910668	1.06223928181013	2
11	chr7	62752436	64852878	0.354280711668298	1
12	chr8	39233297	39466600	5.85250308562865	2
13	chr8	101661785	101965149	0.739094007503558	1

Figure 5: FastCall results reports chromosome, start and end positions of copy number events detected. Copy-number states encoded in the fifth column refers to 2-copy deletions (-2), 1-copy deletion (-1), 1-copy duplication (1) and multiple-copies duplications (2).

Finally, .pdf files reporting a scatter plot of the segmented data and statistically significant regions (chromosome by chromosome) are stored in `Plots` folder for each test sample (follow `/.../Plots/SampleName/`). An example is reported in Figure 6.

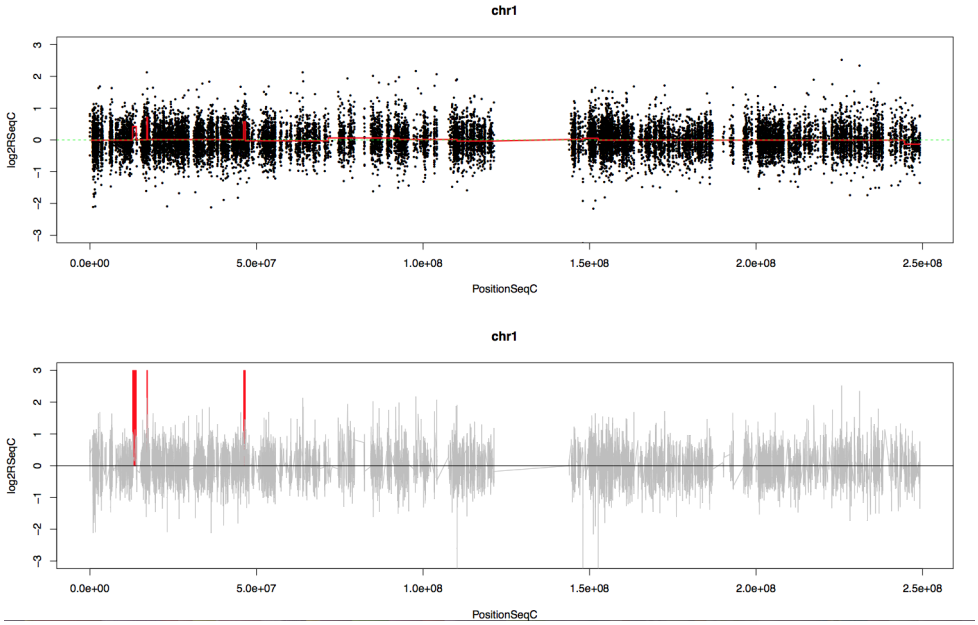

Figure 6: Plots reporting data processed by HSLM and FastCall.