DECoN v2.0.1

Documentation

1 INTRODUCTION

DECoN (Detection of Exon Copy Number) is a sensitive and specific tool for detection of whole exon deletion/duplications in targeted sequencing data. DECoN provides quality checks and visualization to enhance utility for the clinical setting. It is based on ExomeDepth[1] and is most suitable for use on targeted panels run on batches of samples. This tool has been developed through a collaboration between the Institute of Cancer Research, London and the Wellcome Trust Centre for Human Genetics, University of Oxford.

DECoN is implemented in R[2] and has a strict version control using packrat[3]. It therefore will not be affected by future changes to any packages or their dependencies.

2 DEPENDENCIES

DECoN is implemented in R and requires

- R v4.2.0 or later.
- The capacity to build packages from source. See Section 3 for detailed instructions.
- A modern internet browser such as Firefox, Chrome, or IE v.10 or later.
- An internet connection for installation.

3 INSTALLATION GUIDE

DECoN is available for Mac OS X, Linux, and Windows and can be downloaded from URL

3.1 Building packages from source

The capacity to build R packages from source is required. Instructions for building packages from source in either a Mac/Linux environment or a Windows environment are provided below.

3.1.1 Mac/Linux

Building packages from source in a Mac or Linux environment requires gcc and gfortran compilers. The Mac gfortran compiler is available from http://cran.r-project.org/bin/macosx/tools/. gcc is available as part of xtools, the command line tools.

3.1.2 Windows

Building packages from source in a Windows environment can be accomplished with Rtools. Detailed instructions for downloading and installing Rtools can be found in the Appendix.

3.2 Installation

Installation of DECoN requires the following steps:

- Unpack the tarball to a local directory
- Run the setup script:
    - In a Mac/Linux environment – from the directory containing the DECoN scripts, run setup.sh from the command line.
    - In a Windows environment – run the setup.bat file.

The setup script downloads and installs all required packages and dependencies. A log file, called setup.log, is automatically created.

DECoN implements strict version control over all packages and dependencies used by changing the local default R settings. It is recommended to unpack the tarball to a new directory containing only DECoN.

4 RUNNING DECoN

Running DECoN requires four sequential steps:

1. Reading the BAM files (Section 4.1)
2. Running quality checks (Section 4.2)
3. Creating deletion/duplication calls (Section 4.3)
4. Visualizing the calls (Section 4.4)

A summary .RData file is created after steps 1 and 3 which is required for the following steps. This approach allows the user to perform individual steps without having to re-run preceding steps. For instance, a user can make calls with a different set of parameters or change quality thresholds for failing samples without having to re-process the BAM files.

In Mac/Linux, each step is launched via the command line. In Windows, there are executable files for each step which are clicked to launch. The user is then prompted to specify the required inputs. Detailed examples for a Mac/Linux and a Windows environment are provided in Sections 7.1 and 7.2, respectively.

4.1 Reading BAM files

This step calculates a coverage metric called the fragment per kilobase and million base pairs (FPKM) for each exon specified in the BED file. This is done for each of the BAM files.

4.1.1 Inputs

This step has four required inputs:

- BAM files – These can either be specified in a text file or a path to a directory containing all the BAM files can be supplied. The text file must contain a list of the BAM files to be read in, with each file name on a separate line and each file name ending in .bam. If a path is supplied, all BAM files in the directory will be read. DECoN expects each BAM file to have a .bai file in the same location as the .bam file, with a .bai extension instead of .bam as the file name, e.g. *directory/sample.bam* and *directory/sample.bai*.
- BED file – the targeted BED file to be used for analysis. This file does not have a header and must have four columns corresponding to:
    - Chromosome
    - Start position
    - End position
    - Gene
- FASTA file – the reference genome FASTA file to be used with the data.
- Output prefix – the prefix for the summary output .RData file. If none is supplied, the default value is DECoN.

4.1.2 Running DECoN

In Windows, click the *ReadInBams.bat* executable. You will be prompted to enter inputs.

In Mac/Linux, run the following command from the directory containing the DECoN scripts:

Rscript ReadInBams.R --bams *bams.file* --bed *bed.file* --fasta *fasta.file* --out *output.prefix*

4.1.3 Output

This step outputs a summary .RData file prefixed with the output prefix specified in the input which contains sample read depths and sample names taken from the bam files,.

4.2 Running quality checks

This step takes a summary .RData file outputted by the first step and identifies any samples or exons which should be considered failed. Both exons and samples are evaluated based on their median coverage level. When coverage is low, accuracy of detection will be compromised and caution should be exercised

when interpreting results. Samples are also evaluated based on their correlation with other samples. Samples which do not have a high correlation with other samples in the set are likely to have suboptimal detection across the entire target.

4.2.1 Inputs

This step has one required file input and a number of threshold inputs which can be set by the user:

- Summary RData file (required) – A summary RData file containing the FPKM for each exon of an analyzed BED file, created in Section 4.1.
- Minimum correlation threshold – the minimum correlation between a test sample and any other sample for the test sample to be considered well-correlated. The default value is 0.98.
- Minimum coverage threshold – the minimum median coverage for any sample or exon. The default value is 100.
- Exon numbering (optional) – a file containing exon numberings with clinical annotation for at least one value in the analyzed BED file. This is a tab-separated file with four columns labelled with headings:
    - **Chromosome**
    - **Start** – start position from the analyzed BED file
    - **End** – end position from the analyzed BED file
    - **Clinical.Exon** – clinical exon name
- *BRCA* reporting – Boolean value indicating whether an output file containing only failed samples or exons in *BRCA1* and *BRCA2* should be generated. The default value is FALSE.
- Output prefix – the prefix for the output files. If none is supplied, the default value is DECoN.

4.2.2 Running DECoN

In Windows, click the *IdentifyFailures.bat* executable. You will be prompted to enter inputs.

In Mac/Linux, run the following command from the directory containing the DECoN scripts:

Rscript IdentifyFailures.R --Rdata *summary.file* --mincorr *.98* --mincov *100* --exons *clinicalNumbers.file* --out *output.prefix*

4.2.3 Outputs

If all samples and exons pass the required thresholds, no output is created. If any failed samples and/or exons are identified, a tab-separated text file ending in *_Failures.txt* is created with five columns:

- Sample – the name of the sample which failed. If an exon has failed in every sample, this column has value "All".
- Exon – the number of the exon which failed, in the order of the analyzed BED file. If the sample has poor correlation, this column has value "All".
- Types – the type of failure, either "Whole sample" if the sample fails the correlation or coverage threshold or "Whole exon" if the exon fails the coverage threshold.
- Gene – the name of the gene, from the Gene column of the analyzed BED file.
- Details – metric information used to determine failure.

If the *BRCA* option is TRUE and failures are identified which affect *BRCA1* or *BRCA2*, an additional file ending in *_b1b2_Failures.txt* is created. This is the same format as the *_Failures.txt* file and contains the subset of information pertaining to *BRCA1* and *BRCA2*.

4.3 Creating deletion/duplication calls

This step makes whole exon deletion/duplication calls in each sample by selecting reference samples from all other samples contained in the input summary .RData file. The correlation between samples and the number of samples used as a reference are calculated and used as quality measures to provide confidence in both positive and negative calls.

4.3.1 Inputs

This step has one required file input and a number of parameter inputs which can be set by the user:

- Summary RData file (required) – A summary RData file containing the FPKM for each exon of an analyzed BED file, created in Section 4.1.
- Transition probability – the transition probability between normal copy number state and either deletion or duplication state in the hidden Markov model. The default value is set to 0.01.
- Exon numbering (optional) – a file containing exon numberings with clinical annotation for at least one value in the analyzed BED file. This is a tab-separated file with four columns labelled with headings:
    - **Chromosome**
    - **Start** – start position from the analyzed BED file
    - **End** – end position from the analyzed BED file
    - **Clinical.Exon** – clinical exon name
- *BRCA* reporting – Boolean value indicating whether an output file containing only failed samples or exons in *BRCA1* and *BRCA2* should be generated. The default value is FALSE.
- Output prefix – the prefix for the output files. If none is supplied, the default value is DECoN.

- Plotting of variants – takes one of "All", "Clinical", or "None", will plot either all variants, variants in exons which have clinical numbers provided, or no variants respectively. The default value is "All".
- Plot folder – the folder in which plots are saved, created if it doesn't exist already. Defaults to "DECoNPlots".

Note that the default value for the transition probability is set high to avoid false negatives.

4.3.2 Running DECoN

In Windows, click the *makeCNVcalls.bat* executable. You will be prompted to enter inputs.

In Mac/Linux, run the following command from the directory containing the DECoN scripts:

Rscript makeCNVcalls.R --Rdata *summary.file* --transProb *transition.probability* --exons *clinicalNumbers.file* --BRCA *FALSE* --out *output.prefix* –-plot *All* --plotFolder *DECoNPlots*

4.3.3 Output

This step outputs two files:

- Summary RData file - a summary .RData file containing the FPKM for each sample and exon, all CNV calls, and quality control information
- Table of all calls – a tab-separated text file ending in *_all.txt* detailing all exon deletion/duplication calls. This file has 15 columns:
    - **Sample** – the name of the sample.
    - **Correlation** - the maximum correlation between the sample and any other sample in the full set of BAM files.
    - **N.comp** – the number of samples used as the reference set.
    - **Start.p** – the number of the first affected exon from the analyzed BED file.
    - **End.p** - the number of the last affected exon from the analyzed BED file.
    - **Type** – the type of call. This column has a value of either "deletion" or "duplication".
    - **Nexons** – the number of exons encompassed by the call.
    - **Start** - the start position of the call from the analyzed BED file.
    - **End** - the end position of the call from the analyzed BED file.
    - **Chromosome**
    - **ID** – an identifier of the format **Chromosome:Start-End**.
    - **BF** – the Bayes factor associated with the call, generated by DECoN.
    - **Reads.expected** – the number of expected reads under the probabilistic model.
    - **Reads.observed** – the number of observed reads
    - **Reads.ratio** – the ratio of observed to expected reads.

If the *BRCA* option is TRUE and calls are identified which affect *BRCA1* or *BRCA2*, an additional file ending in *_b1b2.txt* is created. This is the same format as the *_all.txt* file and contains the subset of information pertaining to *BRCA1* and *BRCA2*.
If the plotting option is either "All" or "Clinical", then plots of variants are created. An example is given in the next section.

4.4 Visualizing calls

This step provides interactive exploration and visualization of the exon deletion/duplication calls and the underlying data.

4.4.1 Input

This step has one required input:

- Summary RData file – a summary .RData file containing the FPKM for each sample and exon, all CNV calls, and quality control information, generated in step 4.3.

4.4.2 Running DECoN

In Windows, click the *gui.bat* executable. You will be prompted to enter inputs.

In Mac/Linux, run the following command from the directory containing the DECoN scripts:

Rscript runShiny.R --Rdata *summary.file*

4.4.3 Output

This step launches a web browser with an interactive GUI. Further details and examples are provided in Section 5.4.3.

5 EXAMPLES

Two examples are presented in this section, describing whole exon deletion/duplication detection and visualization using a targeted panel. The input dataset is described in Section 5.1 Example 1 describes analysis of the dataset using DECoN in a Mac or Linux setting and is presented in Section 5.2. Example 2 describes analysis of the same dataset in a Windows setting and is presented in Section 5.3. The resulting outputs are the same in all settings and are described in Section 5.4.

5.1 Input

The input dataset has nine samples to be analyzed in a single batch. The reads were aligned to the hg19 reference genome (*hg19.fa*) and the targeted panel is specified in a 0-based BED file (*Target_Regions.bed*). The reference genome FASTA file, the BED file, and the nine aligned BAM files are all stored in a folder called *test_files*.

5.2 Example 1 – Mac/Linux

In a Mac/Linux environment, the analysis steps are run using the command line. The commands are provided and explained below.

Rscript ReadInBams.R --bams *test_files* --bed *test_files/Target_Regions.bed* --fasta *test_files/hg19.fa* --out *DECoNtest* > ReadInBams.log 2 >&1

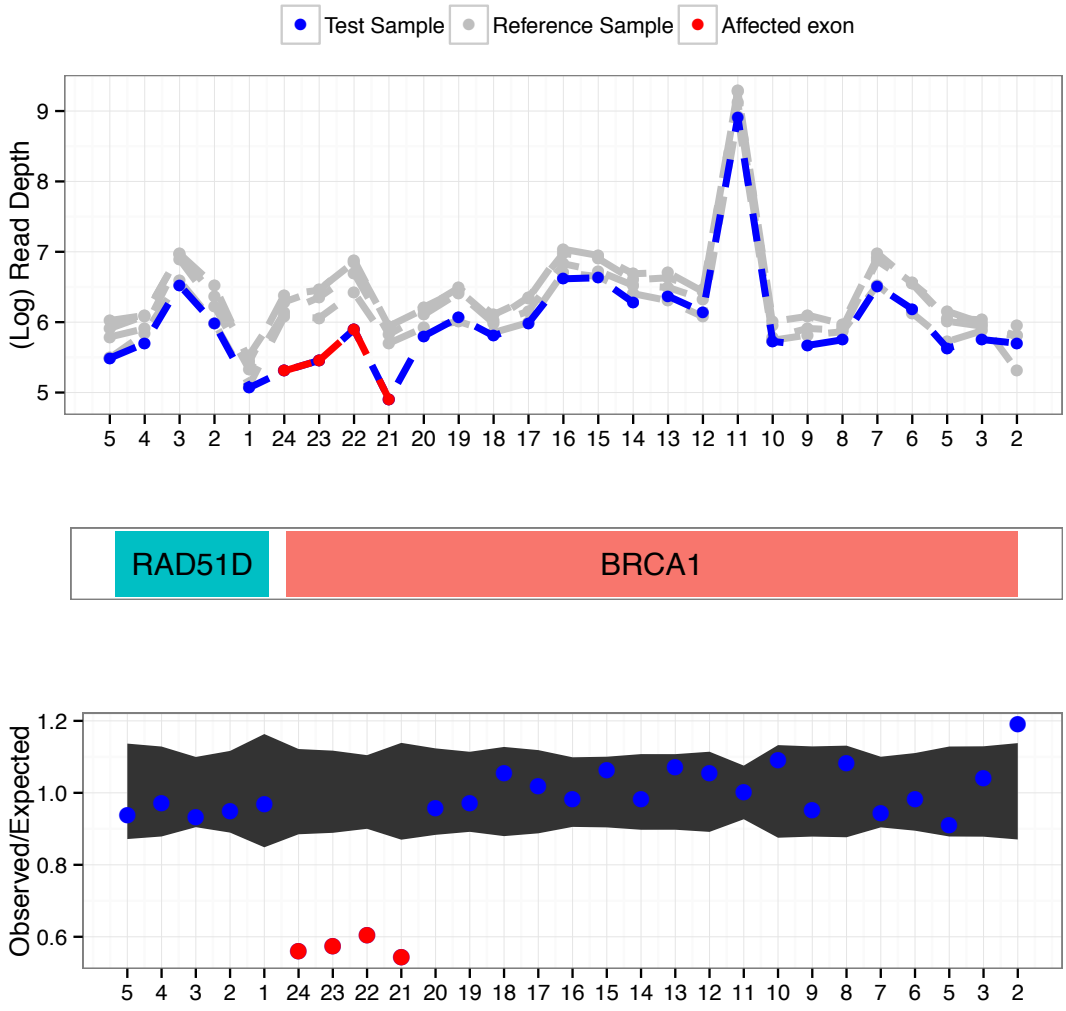
This command reads the BAM files and generates the FPKM for each exon and each sample, outputting a summary .RData file called DECoNtest.RData. The stdout and stderr are redirected to a log file called ReadInBams.log.

Rscript IdentifyFailures.R --Rdata *DECoNtest.RData* --BRCA TRUE --out *DECoNtest* > IdentifyFailures.log 2>&1

This command uses the summary .RData file and default thresholds to identify failed exons and samples. The stdout and stderr are redirected to a log file called IdentifyFailures.log. The *BRCA* option is set to TRUE thus two text files will be created if any samples or exons fail, called DECoNtest_Failures.txt and DECoNtest_b1b2_Failures.txt. These are described in detail in Section 5.4.1.

Rscript makeCNVcalls.R --Rdata *DECoNtest.RData* --BRCA TRUE --out *DECoNtestCalls* --plot *All* –-plotFolder *DECoNTestPlots* > makeCNVcalls.log 2>&1

This command uses the summary .RData file and default parameters to detect whole exon deletion/duplications. The stdout and stderr are redirected to a log file called makeCNVcalls.log. The *BRCA* option is set to TRUE thus two text files are created, called DECoNtest_all.txt and DECoNtest_b1b2.txt. These are described in detail in Section 5.4.2. Full information is outputted to a summary .RData file called DECoNtestCalls.RData.
Plots of all variants are created in the folder DECoNTestPlots, an example plot is shown below. All exons in the affected gene will be shown, and an additional 5 exons from a neighbouring gene (as specified by the bed file) are also shown if the variant is close to the start or end of the gene, as in the figure below. The read depth for the test sample – the sample containing the variant – along with read depth for all samples used as a reference are shown in the top plot. The bottom plot shows the ratio of observed to expected read depth for the test sample, along with a confidence band.



Rscript runShiny.R --Rdata *DECoNtestCalls.RData* 2>&1

This command launches the interactive GUI in a web browser. Visualization examples are described in Section 5.4.3.

5.3 Example 2 – Windows

In a Windows environment, the analysis steps are simply run by double-clicking the .bat file and responding to the interactive prompts in the terminal window. Stdout and stderr are automatically redirected to log files. The four .bat files to run sequentially are:

ReadInBams.bat
IdentifyFailures.bat
makeCNVcalls.bat
gui.bat

The resulting .txt files and visualization examples are described in Section 5.4.

5.4 Outputs

5.4.1 IdentifyFailures output

Below is the full output from the quality checking step, DECoNtest_Failures.txt:

Sample	Exon	Type	Gene	Details
51	All	Whole sample	All	Low correlation: 0.581, Low median read depth (FPKM): 0
All	359	Whole exon	PMS2	Low median read depth (FPKM): 3
All	940	Whole exon	BUB1B	Low median read depth (FPKM): 0
All	1028	Whole exon	TSC2	Low median read depth (FPKM): 67
All	1331	Whole exon	RHBDF2	Low median read depth (FPKM): 69
All	1336	Whole exon	RHBDF2	Low median read depth (FPKM): 35
All	1356	Whole exon	STK11	Low median read depth (FPKM): 81
All	1359	Whole exon	STK11	Low median read depth (FPKM): 76
All	1394	Whole exon	SMARCB1	Low median read depth (FPKM): 65

Here we see that one sample, 51, has a highest correlation of 0.58 with the other samples and thus fails the default threshold of 0.98. This sample also has poor median coverage across the target. A number of exons from the Target_Regions.bed file have failed the coverage threshold in every sample.

5.4.2 makeCNVcalls output

Table 1 below contains a subset of the output from the variant calling step, DECoNtest_all.txt.