DOTS-Finder – User Manual

CONTENTS

NAME

dots_finder – Driver OncoGene and Tumor Suppressor Finder

df_liftover – HG18 to HG19 converter for .marf format

VERSION

This manual refers to DOTS-Finder version 1.0 (01-22-2014 at 10:48 Central European Time UTC+01:00)

SYNOPSIS

```
dots_finder -i input.maf -o outputFolder -n Pfx

df_liftover -i inputBuild36.maf -o outputFolder -n Pfx
```

DESCRIPTION

DOTS-Finder is a tool for the discovery of mutational driver genes within a cohort of cancer samples. Given an input file in Mutation Annotation Format (MAF), it tests and ranks the significance of mutations on a gene according to a functional and frequentist method. DOTS-Finder is accompanied by a small tool for conversion from build36 to build37 for those datasets that are still in hg18.

REQUIREMENTS

DOTS-Finder runs on Unix based machines (MacOS, Linux).

DOTS-Finder is written in python and contains some embedded R code. In order to work properly, these freely available languages must be already installed along with their libraries:

* Python 2.7 http://www.python.org/

* R >= 2.0.0 http://r-project.org

    - CRANpackage'multicore' http://cran.r-project.org/web/packages/multicore/index.html

For installation, we required the creation of a personal virtualenv, provided by DOTS-Finder http://www.virtualenv.org/en/latest/

COMMANDS AND OPTIONS

dots_finder

-h, --help	show the help message and exit
-i INPUT, --input	Input one maf/marf/csv file according to the specification in the user manual (see session Input Format)
-o OUTPUT, --output	Output folder where the results will be written
-n NAME, --name	Insert the prefix name for the output files
-c CORES, --cores	Insert the number of cores to be used in R multicore (default=1)

-p --pseudo	Keep pseudogenes and other non-protein coding genes
-l --lax	The OG threshold is set to 0 and the TSG threshold has no lower bound. It must be used just for small datasets without any significant drivers with regular analysis
--keeptmp	Keep temporary files (for developing and debugging)

df_liftover

-h, --help	show the help message and exit
-i INPUT, --input	Input one MARF file in hg18 according to the specification in the user manual (see session Input Format)
-o OUTPUT, --output	Output folder where the results will be written
-n NAME, --name	Insert the prefix name for the output files

EXAMPLES

Assure that python and R are set as executable in PATH and virtualenv is activated.

Invocation of DOTS-Finder from command line:

dots_finder [-h] -i INPUT [-o OUTPUT] [-n NAME] [-c CORES] [-p] [--keeptmp]

Only the input file is required.
In case of absence of –o OUTPUT option, the program will write in the input folder. If the output folder is specified, but the folder doesn’t exist, the program will create it.
In case of absence of –n NAME option the program will use the name of the input file without extension (.maf/.marf)

The temporary files will be stored in /tmp folder with a random alphanumeric suffix. To redirect the temporary folder, add before the invocation

TMP=/your_temporary_folder dots_finder –i etc.

Or use *TMPDIR=/your_temporary_folder* in a line of your shell script before dots_finder invocation.

Invocation of df_liftover from command line:

df_liftover [-h] -i INPUT [-o OUTPUT] [-n NAME]

Only the input file is required.
In case of absence of –o OUTPUT option, the program will write on the input folder. If the output folder is specified, the program will create it in case it doesn't exist.
In case of absence of –n NAME option the program will use the name of the input file without extension (.marf)

INPUT FORMAT

DOTS-Finder accepts only these input formats:

MAF format version 2.3 (10,May,2012) and 2.4 (6,March,2013). The program is also a complete MAF format validator in case of submission to the TCGA. The file specifications can be found here https://wiki.nci.nih.gov/display/TCGA/Mutation+Annotation+Format+(MAF)+Specification

If a MAF file is malformed and doesn't respect the TCGA guidelines the program will raise a warning, but it will not stop the execution unless the 13 columns for MARF are not present.

MARF format. The Mutation Annotation Reduced Format is a short version of the MAF format with just 13 columns instead of the canonical 34. It's a tab separated values format with '\n' as end line. The columns specification is as follow:

	Header Name	Description	Example	Need	Set NA
1	Hugo_Symbol	HUGO symbol for the gene	TP53	No	'Unknown'
2	Entrez_Gene_Id	Gene in Entrez nomenclature	7157	No	'0'
3	NCBI_Build	Genome Identifier	36 or 37	Yes	NOT
4	Chromosome	Chromosome name without "chr" prefix	X,Y,M,1,2,etc.	Yes	NOT
5	Start_Position	1 base coordinate	999	Yes	NOT
6	End_Position	1 base coordinate	1000	Yes	NOT
7	Variant_Classifica	Translational	Missense_Mu	Yes	NOT

	tion	effect of variant allele	tation		
8	Reference_Allele	+ strand reference allele	A , - , ACGT etc.	Yes	NOT
9	Tumor_Seq_Allele1	Tumoral allele 1	C , - etc.	Yes	NOT
10	Tumor_Seq_Allele2	Tumoral allele 2	C , -	No	Equal to column 9
11	dbSNP_RS	Latest rs ID	rs12345	No	'novel'
12	Tumor_Sample_Barcode	Sample unique identifier	PATIENT_1	Yes	NOT
13	Protein_Change	Aminoacid change	p.E123K	No	'.', 'NULL', emptySTR

The "Need" field simply tells you if the value in the column is mandatory or can be set to NA.
The "Set NA" field explains what kind of NA values are accepted for that column. If it's NOT, the column must have a proper value and no NA is accepted.

Note that the strand (+ or -) is not specified since all the MAFs and subsequently MARFs are set to + strand.

With the specification provided above you can easily revert other mutation data files to MARF format.

* VCF (Variant Call Format) http://www.1000genomes.org/node/101 (require annotation of the Variant_Classification)

* Annovar CSV (move all the strands to + and change Variant_Classification). The conversion to MARF is fully explained in Input-Output Format guide. (http://www.openbioinformatics.org/annovar/annovar_gene.html)

* MAF (cut the required 13 columns and change the header)

See special session Input-Output Format for further explanation on the MARF format and format conversions.

OUTPUT FORMAT

See special guide Input-Output Format for the interpretation of the results.

CREDITS

DOTS-Finder uses different database sources in a modified and packaged format that is easily useable for a faster analysis:

* HUGO protein coding gene database http://www.genenames.org/
* RefSeq transcript database http://www.ncbi.nlm.nih.gov/refseq/
* UCSC transcript http://genome.ucsc.edu/
* NCBI GenBank http://www.ncbi.nlm.nih.gov/genbank/
* Codon Usage Database http://www.kazusa.or.jp/codon/
* Conserved Domain Database
http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml
* Uniprot database http://www.uniprot.org/
* COSMIC database v66
http://cancer.sanger.ac.uk/cancergenome/projects/cosmic/
* Cancer Gene Census
http://cancer.sanger.ac.uk/cancergenome/projects/census/
* Mutation Assessor database release 2 http://mutationassessor.org/

BUGS REPORT

For any problem or defects of the software, please send an e-mail to dots-finder[at]iit.it

REFERENCE

Melloni GEM, Ogier AGE, de Pretis S, Mazzarella L, Pelizzola M, Pelicci PG, Riva L. DOTS-Finder: a comprehensive tool for assessing driver genes in cancer genomes. Genome Medicine 2014, 6:44, DOI: 10.1186/gm563