

Dijest Documentation

DELESTRE Clément

July 30, 2014

Contents

1	About dijest	2
1.1	Dependencies	2
2	How it works	2
2.1	Overview	2
2.2	Input files	3
2.3	Output files	3
2.4	R script	3
3	For programmers	4

1 About **dijest**

Dijest is a program that perform *in silico* enzyme digestion, for several enzymes and several nucleotide sequences.

1.1 Dependencies

Dijest is based on *restrict* command that belong to **EMBOSS** package. To extract fragment sequences **dijest** uses *extractseq* command that also belong to **EMBOSS** package. So please check that **EMBOSS** programs are properly installed on your computer.

Dijest can compute a R script, to do this the R language must be installed. More informations on : <http://www.r-project.org/>. Moreover, as for every Java programs, you also need a **JRE** to be able to use it. Please note the one used to test **Dijest** was 1.7.

As for every Java programs, you also need a **JRE** to be able to use it. Please note the one used to test **dijest** was 1.7.

2 How it works

Here we describe how to use **dijest**. Please note this software doesn't have **GUI**¹ so we assume you are able to deal with command lines.

2.1 Overview

As you can see at fig.1, **Dijest** starts by using the **restrict** command. Then, it parses the result file in order to create a flat file containing only fragments length separated by a break line. After this the program can do 3 optional steps :

1. Compute a R script on output file created after parsing (see 2.4).
2. Extract fragment sequences ("-x" or "- -ext" option).
3. Delete the output file created after parsing ("-c" or "- -clean" option).

These steps are done for every enzymes and sequences you give in input files to **dijest** (see 2.2). The figure 1 shows the **dijest** steps for one sequence called S1. Please note that in this figure, "file" represents the file containing the sequence (**FASTA** format) and "enzyme" represents one enzyme used in this step. The blue writings represent unvariables extensions.

¹Graphical User Interface

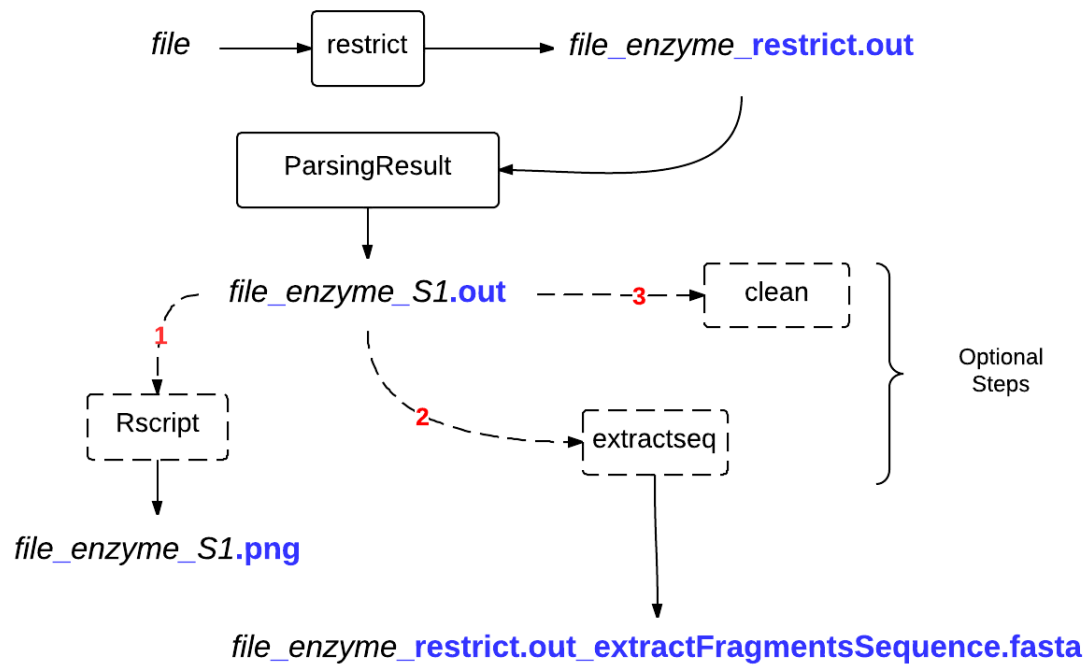


Figure 1: `Dijest` steps for one sequence called S1.

2.2 Input files

`Dijest` needs a FASTA file (specified with "-s" or "- -sequence" option) and an enzyme file ("-e" or "- -enz" option). As you can see at fig.2, each enzyme of the enzyme file must be separated by a line break.

```
EcoRI, EcoRII, ApeKI
EcoRI
```

Figure 2: Example of enzyme file required by `dijest`.

2.3 Output files

As you can see at fig.1, for each sequence and each enzyme, `dijest` will create two type of file :

1. One file with `__restrict.out` extension which contains the result of the `restrict` command.
2. One file with `__.out` extension which contains only fragments length separated by a break line.

Please note that with the option "-c" or "- -clean" you can only delete the second type on file.

2.4 R script

After computing, `dijest` can launch a R script. This later is launched with the fragments size file as argument. The script as to be specified with the "-r" option. There is

already one R script available on `dijest` github² which allow to generate an histogram of fragments size, and save it in `png` format.

3 For programmers

Dependencies

`dijest` use the JSAP library v2.1 downloaded in June 2014, under LGPL licence : <http://www.martiansoftware.com/jsap/> . You need to download it if you want to generate you own JAR file from source code or to modify source code.

Software architecture

The source code can be found here : <https://github.com/AgResearch/dijest/tree/master/src> and the Javadoc can be found here : <https://github.com/AgResearch/dijest/tree/master/Javadoc> .

²<https://github.com/AgResearch/dijest/tree/master/ScriptR>