Easymap documentation

I.1.- What is easymap

`easymap` is an all-in-one software that allows mapping mutations with high-throughput reads in model organisms with a reference sequence available (e.g. Drosophila, Arabidopsis, Caenorhabditis). It is specifically designed for EMS-induced mutants (GC→AT transitions) and for mutants harbouring large insertions (e.g. transposable elements, T-DNAs). For EMS-induced mutations, it uses linkage analysis to markers within a sample in a phenotyped F2 population. Currently it only supports recessive mutations. For large insertions, it uses a custom approach based on capturing an aligning insertion flanking reads. It can map hemizygous insertions.

Many sections of this documentation apply to the analysis of only one of the two types of mutations. This is the case when we mention the `snp` (for linkage analysis of EMS-induced mutations) and `ins` (for tag-mediated mapping of large insertions) modes.

I.2.- How to install easymap

I.2.1.- Option A. Install easymap to be run only through the command line interface.

`easymap` is available for UNIX-based systems that have Python 2. You can clone it from Github with the following command:

```
1    $ git clone <url>
```

You can also download it from <here> and decompress it in your desired directory.
Go to the easymap root folder and run `easymap-setup.sh`:

```
1    $ sudo chmod XXX easymap-setup.sh
2    $ ./easymap-setup-std.sh
```

I.2.2.- Option B. Install easymap to be run through both the command line interface and the web interface.

If you already have a server (e.g. Apache) that runs PHP, simply move to any folder under the root folder of your server and follow the steps in Option A.

If you need to install a server and/or the PHP interpreter, here we describe how to install Apache and PHP5:

```
$ sudo chmod XXX easymap-setup.sh
$ ./easymap-setup-web.sh
```

Passwords

```
$ apt-get www
$ apt-get php
```

Next, give www-data user (this is the Apache server user) write and execute permissions in the easymap directory:

```
$ chmod 777 easymap www-data (look for a safer alternative)
```

Finally, configure php.ini file to increase the file size limit:

```
$ nano php.ini
```

I.3.- Limit resources available to users (optional)

If you are installing easymap to be used by other users, either via the web graphical interface (http) or through the command line interface, and want to limit the resources available to them, configure the following file:

```
$ nano config/config
user_projects-bytes-limit:0
max-simultaneous-jobs:0
```

`user_projects-bytes-limit` is the maximum number of bytes allowed in the folder `easymap/user_projects`. `max-simultaneous-jobs` is the maximum amount of jobs that can be running at any given time. The default value (0) is interpreted by `easymap` as unlimited. To limit their memory usage, set these parameters to any positive integer.

I.4.- Input files you need to run easymap

Ideally, all the text files that you provide to easymap should be formatted with UNIX line separators (`\n`). This is the case with FASTQ reads obtained directly from high-throughput sequencers or with FASTA and GFF3 files downloaded from biological databases. However, if you edit a file in a Windows or Mac machine, you will need to check your file is still readable by easymap by opening it and inspecting it. To do so, click on the button “Preview” to the right of the name of the file.

Input files that are always required:

-High throughput sequencing reads of your problem sample (and control, when required) in FASTQ format (https:// en.wikipedia.org/wiki/FASTQ_format). If your reads are single-ended, you must provide one file (`sample.fq`), whereas if they are paired-ended, you must provide two (e.g. `sample_f.fq`, `sample_r.fq`). FASTQ files have quality information associated with each nucleotide call. There are different encodings for this information, and `easymap` needs the encoding to be “Sanger”. To check whether your reads meet this requirement, and to convert them if necessary, the easiest way is to analyze them with `FastQC` and convert them with `FastQ groomer`. Both tools can be found in the public Galaxy server (https://usegalaxy.org/). If you are using Illumina data produced in the past few years, your read qualities will be Sanger-encoded. Anyway, `easymap` performs an analysis of the input reads at the beginning of each execution and warns the user if their encoding is not Sanger. To map an EMS-derived mutant, `easymap` needs two read sets (single-end reads: `sample.fq` and `control.fq`; paired-end reads: `sample_f.fq`, `sample_r.fq` and `control_f.fq`, `control_r.fq`). All the requirements specified above apply to every file supplied. Regarding read depth, `easymap` will analyze datasets that render any read depth, but values lower than 10X (for `ins` mode) and 25X (`snp` mode) will compromise the accuracy of the results. `easymap` also checks and reports to the user the quality of the nucleotide calls in the reads provided. Mind that low quality calls can also compromise the results (https:// en.wikipedia.org/wiki/FASTQ_format).

-Reference genome in FASTA format (https://en.wikipedia.org/wiki/FASTA_format). If your genome has multiple contigs, its sequence can be provided as a single (e.g. `genome.fa`) or multiple

FASTA files (`1.genome.fa`, `2.genome.fa` and `3.genome.fa`, etc). The FASTA headers of the contigs must be present in the GFF3 file you provide, so easymap can link the information on both files. Genomic FASTA files downloaded from NCBI normally contain long headers that refer to the name of the accession in the sequence database instead of the name of the contig in the genome. If that is the case, you need to manually reheader the files to match the name of the contigs in the GFF3 file. `easymap` compares the input FASTA and GFF3 files at the beginning of each execution and warns the user if the headers in the FASTA file are not in GFF3 file.

-Gene structural annotation of your reference genome in GFF3 format (http://www.ensembl.org/info/website/upload/gff3.html). `easymap` comes with several GFF3 annotations for the reference strains of common model organisms. If you manipulate GFF3 files, be sure that this does not introduce additional characters such as " when saving the file, and that the line separators are UNIX-like.

Input files that are required only for certain analyses:

-Reference sequence of the insertion sequence in FASTA format. Only required in mode 'large insertions' (`--worflow ins`). The sequence must contain at least the full length of the insertion, but can contain other sequences (e.g. the whole sequence of the vector used to engineer a transgene).

Optional input files:

-Gene functional annotation of your reference genome. There is no standard format for this information, so `easymap` asks for the simplest possible file: A tab-separated text file with at least 2 columns, the first being the gene identifiers (e.g. At1g01010 for *Arabidopsis thaliana*), and the remaining columns containing their description (e.g. gene symbols, functional information, etc.). Except the first one, columns can have blank records. You can create tab-delimited text files in excel saving as "text (tab-delimited) (*.txt)" [windows newline symbol!?]. If you do not provide an annotation file, `easymap` can equally run, but will not be able to include gene functional information in the final report. The gene identifiers in the first column of this file must match those in the GFF3 file. This file, or a very similar one that can be modified to fit the format specified above, is typically available from model organism's databases.

I.5.- Running easymap through the web interface

Point your web browser to `easymap`. If your machine has `easymap` installed: http://localhost/path/to/easymap/interface. If you are in another computer within the same network (e.g. inside your University), you need the network IP where `easymap` is installed (e.g. http:// http://10.1.28.91/path/to/easymap/interface).

Place your input files in `easymap`. See section [What input files you need to run `easymap`]. For that, click on **[Manage input files]** in the main menu, and then on **[Select files]**. The browsing window of your operative system will appear to let you select multiple files at once. It accepts files with the following extensions: `.fa`, `.fas`, `.fasta`, `.fq`, `.fastq`, `.gff`, `.txt`. If any of your files have a different extension, simply change it to one of the accepted ones. This does not change the file content. Once you click on accept, you will see the files listed ready for upload. Click on **[Upload files]**. `easymap` accepts uncompressed files of any size. However, the amount of space available can be limited by the server. See section I.2 (How to install easymap). When you click on **[Upload files]** you will see that the progress percentage of a file increases and stops at 100%. After that, it make take some minutes until the file appears listed under “Current files in disk” in the “Manage input files” and the “Run new project” pages.

To run an analysis, click on **[Run new project]** in the main menu. In this view, first give a name to the job. `easymap` will automatically append a timestamp to the name to make it unique. Next, choose an analysis workflow between `[Linkage-analysis mapping]` and `[Tagged-sequence mapping]`. Finally, choose whether you want to analyze your own experimental data (`[Experimental data]`) or want to perform a test with simulated reads (`[Simulated data]`).

Analyze the data from an insertional mutant:
Use the drop-down menus in the page to select the files you want to analyze. The reads choosing menu allows selecting one file (if you have single-end reads) or two files (paired-end reads). The reference sequence choosing menu allows to select a single or multiple files, to include genomes that are represented in multiple FASTA files (normally one per contig).

Simulate and analyze the data from an insertional mutant:
If you chose Data source = simulated data, the FASTQ file drop-down menu will be replaced by several fields where you can simulate an insertional mutant and high-throughput reads.

Analyze the data from an EMS-induced mutant:

If you are over the maximum space allowed for `easymap`, or the maximum number of simultaneous jobs running, a red warning banner will appear on the top of the page, and the option to run a new project will be disabled. This is a sanity limit imposed by the administrator of the machine. In the first case, delete one or more projects in the “Manage projects” view to free some space. In the second, wait until one or more jobs finish running.

I.6.- Running easymap through the command line

First, place all your input files in `/easymap/user_data`. All file must be decompressed. When you specify your input files in the `easymap` command, simply type the name of the file, not the path to it. If your genome reference is in multiple fasta files, rename them so all have the same prefix separated by a dot (e.g. `ref.chr1.fa`, `ref.chr2.fa`, `ref.chr3.fa`, etc.) and use them by typing the basename up to but not including the dot.

`easymap` has two modes: `ins` and `snp`, for large insertions and point mutation mapping, respectively. To map large insertions, ensure that you have these files: `genome.fa`, `insertion.fa`, `genome.gff`, `reads.fq` (or `reads-f.fq` and `reads-r.fq`) and run the following command:

```
1  easymap -w ins -P str -r str -i <file> -S <file> -g <file> [-a <file>]
```

Example:

```
1  easymap -w ins -P myProject -r str -i tn5.fa -A myReads.fq -g genes.gff
   -a annotation.txt
```

If you have paired-end reads, specify the names of both files (forward and reverse reads) separated by a comma:

```
1  easymap -w ins -P str -r str -i <file> -S <file_f,file_r> -g <file> [-a
   <file>]
```

The `-a` argument is optional, so if you do not have a functional annotation file, simple omit it.

easymap comes with a simulator module that can simulate many different scenarios. This can be useful for some users. If you want to simulate and analyze reads from a mutant with randomly positioned insertions, use the following command:

```
easymap -w ins -i <insertion_sequence> -g <genome_structural_annotation> [-a <genome_functional_annotation >] -P “project_name” -sm “sim-mut-arguments” -ss “sim-seq-arguments”
```

-sm simply asks for the number of mutations to introduce, and -ss expects a list of subarguments separated by the “+” character (see the examples below and section I.5.1).

Example 1:

```
easymap -w ins -i tn5.fa -g col0.gff -a annotation.txt -P myProject -sm 5 -ss 25+100,0+500,100+1+75+pe
```

In this example, easymap is asked to insert 5 random insertions of the sequence contained in tn5.fa, and then to simulate 25X paired-end reads of 100±0 nt from a library with an insert size of 500±100 bp, with a 1% basecalling error rate and a 75% GC bias strength.

Example 2:

```
easymap -w ins -i tn5.fa -g col0.gff -a annotation.txt -P myProject -sm 5 -ss 15+200,40+500,100+1+75+se
```

In this example, easymap is asked to insert 5 random insertions (the sequence contained in tn5.fa), and then to simulate 15X single-end reads of 200±40 nt, with a 1% basecalling error rate and a 75% GC bias strength. The third argument is ignored because the library is single-end.

In snp mode easymap will analyze the RD distribution of the alignment. To do that the span between the coordinates 100 000 and 200 000 of each contig is analyzed. This assumes that every contig is at least 200 000 bp-long.

Once an easymap job has finished successfully, you can review the results in easymap/user_projects/{date}_project_name/2_output/. See section I.7 for a complete description of the report and datasets.

If an execution of `easymap` fails, the following message appears on the terminal:

```
$ ERROR: easymap failed to analyze the data. See the log file.
```

If you want to investigate the origin of the error, open the file easymap/user_projects/`{date}_project_name/2_logs/log.log`

List of `easymap` command line arguments:

`-w, --workflow` Type: string (`snp`, `ins`) Required: always	Analysis workflow you want to use. It depends on the type of mutant. Use `snp` for EMS mutants, and `ins` for insertional mutants.
`-p, --project-name` Type: string Required: always	String containing the name you want to give to the job. After the program has finished, you can find the results in `/user_projects/{date}_project_name`.
`-r, --reference-sequence` Type: string Required: always	Basename of the FASTA file or files that contain the reference sequence. If only one file, the name must have the format `name.fa` (e.g. danio_rerio.fa). If more than one, each file must contain only one contig, and the file names must have the format `index.name.fa`, where `index` is the number of the contig in the genome (e.g. `1.danio_rerio.fa`, `2.danio_rerio.fa`, `3.danio_rerio.fa`).
`-i, --insertion-sequence` Type: file locator File format: FASTA Required: only in `ins` mode	Name of the FASTA file that contains the insertion sequence. Accepts files with numbered lines and blank lines. We need to make a script that cleans the sequence, using pBIN-pROK2.fa.txt as template.
`-g, --gff3-file` Type: file locator File format: GFF3 Required: always	Name of the GFF3 file that contains the genome structural annotation. The contig identifiers in this file must match the contig identifiers in the genome reference file(s).
`-a, --annotation-file` Type: file locator File format: custom Required: optional	Name of the file that contains the functional information of the genes in the reference genome. The format is custom. See section *I.3.- What input files you need to run easymap*.
`-rp, --reads-problem`	Name of the FASTQ file(s) that contains the reads of your problem sample (F2 population phenotyped for the recessive

Type: file locator File format: FASTQ Required: not when `-sim` is on	phenotype). If your reads are single-end supply the file name. If paired-end, supply both file names separated by a comma.
`-rc, --reads-control` Type: file locator File format: FASTQ Required: not when `-sim` is on	Name of the FASTQ file(s) that contains the reads of your control sample. If your reads are single-end, supply the file name. If paired-end, supply both file names separated by a comma.
`-cr, --cross-type` Type: string (`bc`, `mc`) Required: only in `snp` mode	Type of cross (backcross or mapcross) performed to create the mapping population.
`-mb, --mutant-background` Type: string (`ref`, `noref`) Required: only in `snp` mode	Genetic background or strain of the mutant (same or different from the reference sequence supplied).
`-co, --control-type` Type: string (`par_mut`, `par_pol`, `f2wt`) Required: only in `snp` mode	Type of sample used to create the control reads and provided to `-rc`. Choose between par_mut (your control reads belong to the mutant before the mutagenesis), par_pol (to the polymorphic strain the mutant was crossed to), or f2wt (to)
`-sim, --simulate-reads` Type: flag	Turns on `easymap` built-in function to simulate the data for a mapping experiment. First, it creates a mutant strain, then creates a mapping population, and finally creates high-throughput reads. It turned on, you must define the simulation parameters with the arguments `-sm`, `-sr`, `-ss`.
`-sm, --sim-mut` Type: string Required: when `-sim` is on	Specifies the number of mutations in the mutant. The reference sequence provided with `-r` is used as template. In ins mode, the insertions sequence provided in `-i` is used as insert. If mode is `ins`, the number is limited to 1 per Mb of reference genome. If mode is `snp`, the number is limited to 100 per Mb of reference genome.
`-sr, --sim-recsel` Type: string Required: only when mode is `ins` and `-sim` is on	Specifies how to create recombinant chromosomes, using the following format: `param1+param2+param3`. `param1`: recombination frequency distribution of each contig provided in `-r`. `param2`: location of the causal mutation in the format `contig,position`. `contig` is an integer that specifies the chromosome number you want to select. `position` is an integer with the position, in the chromosome specified with

	`contig`, of the causal mutation defined in base pairs. Must be between 1 and the length of the contig. `param3`: positive integer that represents the number of haploid recombinant genomes to create. Example. In a sample with two chromosomes, select the position 5875000 of the second contig and create a mapping population of 200 haploid genomes: `0,14-1,31-2,33-3,15-4,5-5,2/0,24-1,42-2,25-3,6-4,1-5,2+2,5875000+200`
`-ss, --sim-seq` Type: string Required: when `-sim` is on	Specifies how to create high throughput reads, using the following format: `param1+param2+param3+param4+param5+param6`. `param1`: Read depth, or number of times in average that each nucleotide in the sample sequence is read. `param2`: Read length in the format `mean,sd`. `mean` and `sd` must be positive integers. `param3`: Library fragment size in the format `mean,sd`. `mean` and `sd` must be positive integers. `mean` must be bigger than twice the read length specified in `param2`. `param4`: Integer in the interval [0-5] that represents the base calling error rate in percentage. `param5`: Integer in the interval [0-100] that represents the GC content bias strength of the library. Setting this to >0 penalizes the creation of fragments with non-neutral GC content. The less neutral the GC content of a genomic sequence, and the bigger the strength value set by the user, the less probable the sequence is present in the reads created. `param6`: Type of sequencing library. Choose between `se` (single-end library) and `pe` (paired-end library). Example. Simulate paired-end 100-bp-long reads from 500±100-bp-long fragments at 40X, with a base calling error rate of 1% and a GC bias strength of 75: `40+100,0+500,100+1+75+pe`
`-u, --usage` Type: flag	Shows where to find help to run `easymap`.

I.7.- Samples required by `easymap snp`

`easymap snp` always requires two read datasets: the first corresponds to a pool of mutant-phenotype individuals from the F_2 generation of a mapping cross; the second corresponds to a control sample. The sample you can use as control depends on the background of your mutant and the type of cross you perform to obtain the mapping population (backcross versus outcross/mapcross). See Table X to know which samples can be used as the control in each case.

Table X. Compatibility of `easymap` with different experimental setups for mapping by sequencing of SNP mutants based on linkage analysis

Strain in which the mutant was obtained	Cross performed to obtain the mapping population	Samples sequenced	Easymap can analyze the data (Yes / No)
Reference strain	Backcross	F2 mutant pool + mutant parental	Y
		F2 mutant pool + F2 wild-type pool	Y
	Outcross	F2 mutant pool + mutant parental	Y
		F2 mutant pool + polymorphic parental	Y
		F2 mutant pool + F2 wild-type pool	N
Non-reference strain	Backcross	F2 mutant pool + mutant parental	N
		F2 mutant pool + F2 wild-type pool	Y
	Outcross	F2 mutant pool + mutant parental	Y
		F2 mutant pool + polymorphic parental	Y
		F2 mutant pool + F2 wild-type pool	N

In the linkage mapping mode, we recommend a minimum read depth of 30X. Lower values can compromise variant calling. Besides, read depth is a limiting factor to estimate allele frequencies that can limit the power of the analysis even if using many recombinant chromosomes.

I.8.- Explanation of the easymap output report

Primers have Tm between 62 and 66 at standard PCR conditions. Explain the distance between primers and SNP and reason. Explain logic of insertion genotyping. Explain that inner primers are calculated making a consensus sequence of the insertion ends.

Backcross vs Outcross: which one to use when both are possible?

If in the ref strain and enough markers (natural + EMS-induced), backcross makes discrimination between

Recommendations:

Prior to uploading FASTQ files to `easymap`, it is advisable to check their quality with programs such as FASTQC (https://www.bioinformatics.babraham.ac.uk/projects/FASTQc/).

EMS mutations are used as mapping markers. Therefore, it is advisable to create a mapping population with the MX mutant, before backcrossing several times.

Troubleshooting

-No disk space left for the program execution (how to know this from the web interface)

Insertions mode: compatible with sample pooling, minimum recommended RD is 10X

SNPs mode: minimum recommended RD is 40X

```
1	sudo apt-get update
2	sudo apt-get install php7
3	sudo make php7
```

Setting a ftp server to upload files from outside network.