

## Documentation

Once you have downloaded a binary version of MOMO, you have to make the file executable:

```
chmod a+x MOMO-x86_64.ApplImage
```

and then you can run it through the command line:

```
./MOMO-x86_64.ApplImage
```

The MOMO software has several running modes that can be used by passing different combinations of flags and parameters by command line.

Running `./MOMO-x86_64.ApplImage -h` shows the help with all the possible flags:

<code>-h [ --help ]</code>	Produce help message.
<code>-N [ --network-file ] arg</code>	Input network file (MANDATORY).
<code>-B [ --bounds-file ] arg</code>	Input file with the bounds for the output.
<code>-o [ --output-file ] arg (=solutions.txt)</code>	Output file.
<code>-t [ --optimizer-timeout ] arg (=3600)</code>	Timeout for the optimization procedure (in seconds).
<code>--maximize arg</code>	Reactions to maximize.
<code>--minimize arg</code>	Reactions to minimize.
<code>--reactions-to-fix arg</code>	Reactions to fix. Format <code>reactionID=flux[,reactionID=flux]</code> (useful for the enumeration).
<code>--number-deletions arg (=0)</code>	Consider deletions sets of informed size.
<code>--enumerate</code>	Enumerates deletions. In this case the objective functions are ignored.
<code>-D [ --deletions-candidates-file ] arg</code>	Input file with the reactions that be considered for deletion.
<code>-R [ --reference-flux-file ] arg</code>	Input file with the fluxes that will be used as reference for the MOMA approach during enumeration.
<code>-m [ --optimizer-internal-memory ] arg (=2048)</code>	Internal memory used by the optimizer before using disk space (in Mb).

The network file (-N flag) and the file with the upper and lower bounds (-B flag) should be always provided when invoking MOMO.

### Optimise a flux

In order to optimise one flux (for example, the biomass reaction) you should invoke MOMO with at least the following flags:

`--maximize` (or `--minimize`, depending how you want to optimize).

The argument for this flag should be the id of the reaction that you want to optimize according to the SBML file used.

Example:

```
./MOMO-x86_64.ApplImage -N yeast_5.01_model.xml -B bounds.txt --maximize=r_2110 -o fba.out
```

In this example, we set the metabolic network as the file "yeast\_5.01\_model.xml", the bounds for each reaction of the SBML file are given by the file "bounds.txt", we set the optimization function to be the reaction with the id `r_2110` and we set the output file to be "fba.out".

## Optimise several fluxes considering deletions

In order to optimise multiple fluxes (for example, the biomass and ethanol productions) you should invoke MOMO with at least the following flags:

--maximize- (or ---minimize-, depending how you want to optimize);

The argument for this flag should be the ids of the reactions that you want to optimize according to the SBML file used, separated by commas.

--number-deletions;

The argument for this flag should be an integer  $\geq 1$ , representing the size of the deletion set. If the parameter is 1, we have single deletions, if it is 2 then we have pairs of deletions, and so on.

Example:

```
./MOMO-x86_64.AppImage -N yeast_5.01_model.xml -B bounds.txt --maximize=r_1761,r_2110 --number-deletions=1 -R fba.out -D candidateReactions.txt -o pareto.out
```

## Enumerate deletions

In order to enumerate all deletions for a fixed production state (for example, given that biomass is equal to 0.4 and that the ethanol is equal to 6) should invoke MOMO with at least the following flags:

--enumerate

--number-deletions;

The argument for this flag should be an integer  $\geq 1$ , representing the size of the deletion set.

If the parameter is 1, we have single deletions, if it is 2 then we have pairs of deletions, and so on.

Important:

--maximize and --minimize should not be used in the enumeration mode.

Example:

```
./MOMO-x86_64.AppImage -N yeast_5.01_model.xml -B bounds.txt -D candidateReactions.txt -R fba.out -o enumeration.out --number-deletions=1 --reactions-to-fix r_2110=0.1,r_1761=17.4 --enumerate
```

In this example, we set the metabolic network as the file “yeast\_5.01\_model.xml”, the bounds for each reaction of the SBML file are given by the file “bounds.txt”, we selected to enumerate single deletions (--enumerate and --number-deletions=1) and we fix the reaction r\_2110 (biomass) to 0.1 and the reaction r\_1761 (ethanol) to 17.4.

This means that we want to enumerate all single deletions that enables a biomass of 0.1 and an ethanol production of 17.4.

## Experiments

Here you can download the data to reproduce the experiments from the paper.

In the file available at this address: <http://team.inria.fr/erable/files/2020/11/experiments2.tar.gz>, you will find:

- A copy of the Yeast 5.01 metabolic network in SBML file (yeast\_5.01\_model.xml).
- A file with the bounds used in our experiments (yeast\_5.01\_model.inputs.bounds.txt)
- A file with a list of deletion (reaction) candidates (deletionsCandidates.txt)

- A text file with the full commands to run MOMO (commands.txt).

### Compiling the source code

In order to compile the source code, you will have to install the following dependencies:

- glib (with glibmm)
- SCIP and PolySCIP versions 3.2.1 (SCIP Optimization Suite v. 3.2.1)
- libxml version 2.6
- libboost\_program\_options

You will also need the more recent version of GNU G++ compiler and Make.

Except for SCIP and PolySCIP, all other dependencies can be easily installed using the software manager of your favorite Linux distribution.

In Ubuntu for example, it suffices to install the packages.

For the SCIP Optimization Suite, you will have to download the source code from <https://www.scipopt.org/> and follow their instructions to compile SCIP and PolySCIP.

After all this you have to generate a library from PolySCIP code and finally compile MOMO.

This part is a little bit technical but in the file available at this address: <http://team.inria.fr/erable/files/2020/11/Ubuntu.pdf>, you can find all the instructions to reproduce this steps and compile MOMO in an Ubuntu machine (tested with 18.04 and 19.04).

### References

Ricardo Andrade, Mahdi Doostmohammadi, João L. Santos, Marie-France Sagot, Nuno P. Mira, Susana Vinga. MOMO – Multi-Objective Metabolic mixed integer Optimization: application to yeast strain engineering. *BMC Bioinformatics*, 21(1):1-13, 2020, <https://hal.inria.fr/hal-02490353/file/Andrade-2020.pdf>.