

# MOTUS manual

September 23, 2008

## 1 Presentation

**Purpose:** MOTUS is a software for searching and inferring coloured motifs in coloured graphs. This document is an initial guide to use this software.

**Definitions:** A coloured graph is a vertex-labelled graph. A coloured motif is defined as a multiset of colours (*i.e.* vertex labels). An occurrence of a coloured motif in a coloured graph is a set of connected vertices such that there is a bijection between its colours and the colours of the motif.

**Options:** This software can be used to perform **motif search** (input: graph and motif, output: all the occurrences of the motif in the graph) or **motif inference** (input: graph and motif size, output: all motifs of given size and their occurrences)

**Scope:** MOTUS has been originally developed for structural analysis of metabolic networks but can be used for any vertex-labelled graph. Two input formats are supported, see section 3 for more details. MOTUS also has advanced options specific to metabolic data.

**Web:** Finally, MOTUS is also available as a web service called MOTUSweb: <http://pbil.univ-lyon1.fr/software/motus/>

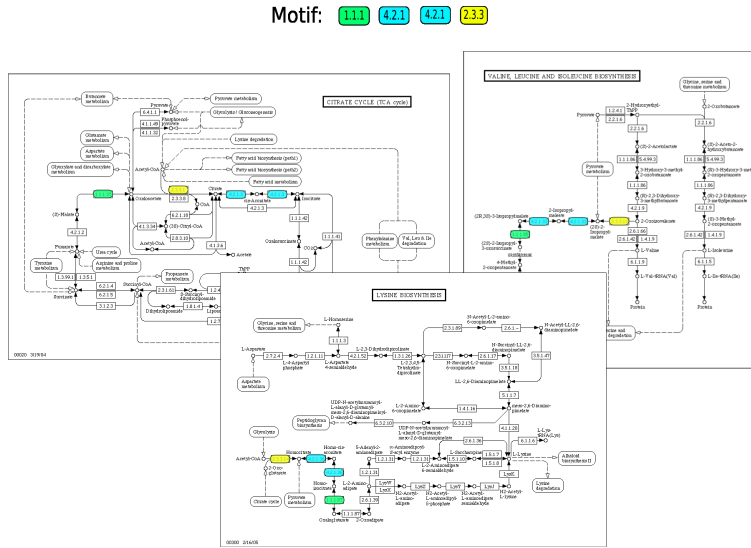


Figure 1: Coloured motif and its occurrences in the metabolic network of the bacterium *E. coli*.

**Third-party** MOTUS is distributed with Nauty, a program for computing automorphism groups of graphs and digraphs. Nauty has been developed by Brendan McKay who authorizes to include his package in other packages “with the exception of sale for profit or application with nontrivial military significance”. The web page of Nauty is: <http://cs.anu.edu.au/~bdm/nauty/>.

## 2 Installation

### 2.1 Requirements

MOTUS has only been tested on linux platforms, although it should work properly on MacOS. Only a c++ compiler is required.

### 2.2 Compilation and installation

At the location where you download the file:

1. tar xvzf MOTUS1.0.tar.gz

2. `cd MOTUS1.0`
3. `make`

## 2.3 Running MOTUS for the first time

- `cd motus`
- `./bin/mo-mo-motus`

A help screen will appear and propose various options. In order to get familiar with the software, you may start by using available datasets with standard options.

Two basic examples are:

```
./bin/mo-mo-motus -d data/primCpdsSmmReactionsCompounds.col -n 3  
./bin/mo-mo-motus -g data/ERn100p0.1 -l data/labelsn100c10 -n 3
```

## 3 Input

There are two types of inputs for using MOTUS:

- Metabolic network in MOTUS format
- Any undirected vertex-labelled graph in edge list format

### 3.1 MOTUS format for metabolic networks

Example datasets are available in the data directory. Fields that need to be specified in the current version of MOTUS are:

- Reaction (identifier and name)
- Compound (identifier, name and status (substrate/product))
- EC number

Other fields may be given any default value such as NA. When the EC number is not known, default value NA should be used.

MOTUS enables to remove the  $n$  most frequent metabolites of the dataset ( $n$  being a user-defined parameter).

Examples:

```
./bin/mo-mo-motus -d data/primCpdsSmmReactionsCompounds.col -ub  
10
```

Other types of preprocessing of the dataset, such as removal of secondary metabolites should be performed prior to analysis by MOTUS.

### 3.2 Edge list format for general vertex-labelled undirected graph

Two files should be given as argument, one for the topology and one for the labels. The topology should be given as a list of edges, vertices being represented by integers. Vertices with no edges do not need to be specified explicitly.

In order to avoid any troubleshooting, you should comply to the following rules:

- Use integers from 0 to n-1 to identify the graph vertices
- An edge should be defined as a pair of integers delimited by blank space
- The labels file should have exactly n lines

Example:

```
./bin/mo-mo-motus -g data/ERn100p0.1 -l data/labelsn100c10 -e 1 2 3
```

### 3.3 Modes

There are essentially 3 ways of using MOTUS:

- Search mode: given a motif, get all occurrences of the motif
- Inference mode: given a motif size (and optionnally a cut-off for EC numbers), find all repeated motifs in the network
- Statistics mode: give general statistics on the dataset

### 3.3.1 Search mode

In search mode, the motif may be specified either as a multiset of (partial) EC numbers (Input type 1) or as a multiset of integers (Input type 2).

Examples:

```
./bin/mo-mo-motus -g data/ERn100p0.1 -l data/labelsn100c10 -e 1 2 3
./bin/mo-mo-motus -d data/primCpdsSmmReactionsCompounds.col -e 1.1
4.2.1 2.3.3
```

### 3.3.2 Inference mode

This mode is controlled by the option “-n” which should be set to the size of the motifs to enumerate. Optionnally, a cut-off for EC numbers can be specified (Input type 1). This is controlled by the option “-s”. Example:

```
./bin/mo-mo-motus -d data/primCpdsSmmReactionsCompounds.col -n 3 -s
2
./bin/mo-mo-motus -g data/ERn100p0.1 -l data/labelsn100c10 -n 3
```

### 3.3.3 Statistics mode

This mode is controlled by the option “-st” which should be set to 1.

Example:

```
./bin/mo-mo-motus -d data/primCpdsSmmReactionsCompounds.col -st 1
```

## 4 Handling the topology

It is possible to cluster the results according to topology, that is, when you search for a coloured motif, MOTUS will first get all the occurrences of the coloured motif and then group the occurrences that share the same topology. This is equivalent to searching for all topological coloured motifs that have this set of colours. (Note that the advantage is that you have all the information at once).

This option is controlled by the parameter “-t” which should be set to 1 if the results should be clustered by topology (topological unordered coloured motifs).

Note that it is possible to group by topology, while maintaining the order of the colours (topological ordered coloured motif). This option is controlled

by the parameter “-order” which should be set to 1 if the results should be clustered by order. Note that it does not make any sense to sort by order if the occurrences have not been sorted by topology first.

In order to know which topology each group of occurrences has, MOTUS outputs the adjacency matrix.

Examples:

```
./bin/mo-mo-motus -g data/ERn100p0.1 -l data/labelsn100c10 -e 1 2 3 -t 1
```

```
./bin/mo-mo-motus -g data/ERn100p0.1 -l data/labelsn100c10 -e 1 2 3 -t 1
```

```
-order 1
```