SpaceMACS Command Line Documentation 1.0

Sascha Jung

November 15, 2023

Contents

1 Introduction

All links, references, table of contents lines etc. in this pdf are clickable.

Please note that this package is a command line package.

SpaceMACS is a command line tool to find maximum common substructures (i.e., chemical substructures) in combinatorial chemical spaces ("fragment spaces") of multi-billion size and beyond. It is also possible to perform "classical" substructure searches with SpaceMACS.

Maximum common substructure (MCS) searches are a common problem in cheminformatics. SpaceMACS calculates the MCS between a given query and compounds in **non-enumerated** fragment spaces.[1] This makes it possible to retrieve compounds with a common substructure from up to trillion-sized chemical spaces on standard, modest hardware.

SpaceMACS is a perfect complement to our SpaceLight[2] tool that performs "near-neighbor" similarities and to our fuzzy Feature Tree[3] technology which has a pronounced strength in detecting distant neighbors with chemical similarity (scaffold hops). SpaceMACS will find those compounds with a **maximum common substructure** or with a **full, exact substructure** in a chemical space.

SpaceMACS on the command line lets you

- perform maximum common substructure searches in chemical spaces to identify analogs to your query that share a maximum chemical motif
- conduct classical substructure searches in chemical spaces to identify molecules that contain an exact scaffold
- perform (maximum common) substructure searches in classical enumerated libraries
- visualize the common substructure between query and hit molecule

SpaceMACS traverses huge chemical spaces using Lego-like chemical reaction combinatorics behind the scenes (Figure 1). To conduct quick calculations, we formalize reactions and encode them as linking reactions with associated fragments. Every link position in a fragment has a dedicated linker type which

[1] `https://www.biosolveit.de/infiniSee/#chemical_spaces`
[2] `https://www.biosolveit.de/download/?product=spacelight`
[3] `https://www.biosolveit.de/download/?product=ftrees`

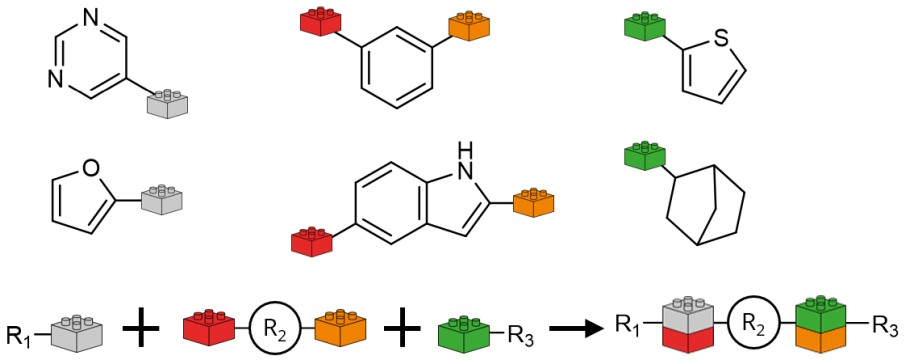

Figure 1: *Example of a formalized reaction and associated fragments encoded with "Lego-like" linkers.*

assures that only chemically meaningful combinations can be formed via its link compatibilities. In the example in Figure 1 the grey brick can only be attached to the red one and the green brick can only be attached to the orange one. SpaceMACS now calculates the (maximum common) substructure on the fragments and enumerates those hit molecules for which the overall MCS is maximized.

Summarizing, with SpaceMACS you can not only search much bigger spaces than with other methods, but you also require much less time — and will be orders of magnitude faster, making it possible to search and mine scaffolds from vast spaces even on modest, standard hardware.

2 Jump Start: Finding Hits With Common Substructures in a Chemical Space

Your license is all set? You are familiar with command line usage? You unpacked the installation archive and found a fragment space file? Then here is a typical query call to search for the MCS and get hit molecules ordered by decreasing MCS size to the query scaffold:

```
./spacemacs -i <path/to/query.sdf> -s <path/to/chemical_space.space>
```

The query can be in SD, smiles (.smi or .smiles file, containing line-separated SMILES), MOL and MOL2 format and the file can have multiple entries. For

quick searches, the input can just as well be a SMILES string in double quotation marks, for example:

```
./spacemacs -i "CC(C)C(=O)N" -s <path/to/chemical_space.space>
```

With the above calls, the 100 hit molecules with the largest common substructure size (MCS size) are written as SMILES to your console/shell (STDOUT). To write your results to an output file (CSV or SD file), additionally use the either the `-o` (separate output file for every query) or `-O` option (single output file with all results), for example:

```
./spacemacs -i "CC(C)C(=O)N" -s <path/to/chemical_space.space> -o <path/to/output.csv>
```

The output file can be either a CSV file or a SD file. It will contain detailed information for every result molecule: result rank, search type, MCS size, result size, query size, MCS similarity, result name, query name, query smiles, space name, reaction name and reagents used to construct the result compound (see section 4.2 for detailed information).

The 100 compounds with the largest common substructure are written to the output by default. You can adjust this by using the `--max-nof-solutions` option with your desired number:

```
./spacemacs -i "CC(C)C(=O)N" -s <path/to/chemical_space.space> -o <path/to/output.csv>
            --max-nof-solutions 300
```

To identify compounds containing the exact, full structure of the query you can switch to a classical substructure search with the `-t / --search-type` option:

```
./spacemacs -i "CC(C)C(=O)N" -s <path/to/chemical_space.space> -t 0
```

All this was too dense? Then let's take it step by step in the paragraphs below.

3 Technical Prerequisites

SpaceMACS is a command line application. SpaceMACS needs the following to run:

- The **application package** (from `https://www.biosolveit.de/download/?product=spacemacs`); depending on your operating system, some libraries may have to be installed (get in touch with us if that is the case: support@biosolveit.com; and please mention any errors/warnings that you see in your mail)
- A **shell** (Linux/Unix) or a terminal (macos), or a command line environment (Windows; e.g.: cmd.exe)
- A valid **license** (from license@biosolveit.com)

The license setup instructions will come with the license that we will send out — or that has already been sent out to you.

A "test license" that you can request online and that is sent to you instantaneously can simply be placed next to the executable (spacemacs.exe, spacemacs, or SpaceMACS — depending on your operating system). For MacOS please read on...

MacOS Specialties On MacOS, the executable will typically reside inside the *.app package:

```
/Applications/SpaceMACS.app/Contents/MacOS/SpaceMACS
```

To place the short term test license there, you will have to go into the *.app package using a right mouse click (or CTRL-click) on SpaceMACS.app in the Finder, and click on "Show package contents". In there, you will see the Contents/ subfolder, in there the MacOS subfolder, and in there, the SpaceMACS executable. If you are about to use the **test license**, place is right there, next to the executable. A longer term license will be handled separately, we will tell you how when we send that very license.

When you call SpaceMACS for the first time, go to the Finder, and navigate to the Applications folder. Do a right(!) click on SpaceMACS.app, and — if applicable — confirm that you want to open the program. It will flash up once, and you are good to go at the terminal prompt from there on.

To make the first step, call SpaceMACS within your shell/terminal/environment.

4 Command Line Options

4.1 Overview

An overview of all command line options is available by calling SpaceMACS with `--help`:

```
./spacemacs --help

 Program options:
-i [ --input ] arg                    Input query molecule file or single input molecule as smiles.
                                      Supported file types are *.smi, *.smiles, *.mol, *.mol2 and *.sdf.
-s [ --search-files ] arg             Paths to library input molecule files for similarity scoring or to
                                      Fragment Space FSF files or Fragment Spaces. Supported file types are
                                      *.smi, *.smiles, *.mol, *.mol2, *.sdf, *.space, *.zip and *.fsf.
                                      Note: The .flf and fragment files specified in the FSF have to be in
                                      the appropriate relative paths.
-o [ --output-files ] arg             Output base files (suffixes are required). For each query molecule,
                                      the results are written to a separate output file. Supported file
                                      types are *.csv and *.sdf.
-O [ --single-output-files ] arg      Output files (suffixes are required). All results are written to a
                                      single output file. Supported file types are *.csv and *.sdf.
-m [ --match-image-base-file ] arg    Output base file name for matching images (suffix required).
                                      Supported file types are *.pdf, *.png and *.svg.
                                      Note: For each match a separate file is created.
-t [ --search-type ] arg (=1)         Type of the performed mcs search:
                                          0 Substructure search
                                          1 MCS search maximizing the number of mapped atoms.

Configuration:
--max-nof-results arg (=100)          Maximum number of top-ranking result molecules [1 to 1000000].
--expand-alternative-results [=arg(=1)]
                                      Write alternative results based on alternative reaction paths.

General options:
-h [ --help ]                         Print this help message
--license-info                        Print license info
--thread-count arg                    Maximum number of threads used for calculations. The default is to
                                      use all available cores.
--version                             Print version info
-v [ --verbosity ] arg (=2)           Set verbosity level
                                           0 [silent]
                                           1 [error]
                                           2 [warning]
                                           3 [workflow]
                                           4 [steps]
```

The abbreviated, one-letter options are preceded with one dash `-`. The longer, named options are preceded with two dashes: `--`. If an option needs an argument (arg), you can include or omit the equals sign.

4.2 Minimal Required Options

This section describes the arguments you must specify at minimum to successfully run a search with SpaceMACS. First, you must provide the path to a file containing the query compounds. All well-known data formats are supported (MOL, SDF, SMILES file, MOL2). Instead of a file containing the query molecules you can also specify a single SMILES string enclosed by double quotation marks. The SMILES string or the molecule file must be passed to the `-i` option. Additionally, you need to specify the path to a fragment space (.space file) to be searched. Alternatively, you can also specify a library file (SDF MOL2, SMILES file) to perform an enumerated search instead of a space search. Either the space file or library file have to be specified with the `-s` option. The minimal search prompt then has the following general form:

```
<path/to/spacemacs/executable> -i <path/to/queries> -s <path/to/space_file>
```

When you specify the required paths the search prompt might look like the following:

```
./spacemacs -i my_queries.sdf -s my_space.space
```

Or, if you specified a SMILES string instead of a file:

```
./spacemacs -i "CC(C)C(=O)N" -s my_space.space
```

In the examples above, the output is printed on the console by default, which will look similar to the following example:

```
Query: [O-]C(=O)c1c(OC(=O)C)cccc1       ASS
Rank:  1 mcs size: 13 sim: 0.542 O=C(Oc1c(cccc1)C(=O)N(C)C)c2c(OC(=O)C)cccc2 WXVL021____AN0032
Rank:  2 mcs size: 13 sim: 0.5 O=C(Oc1c(cccc1)C(=O)N(C(C)C)C)c2c(OC(=O)C)cccc2 WXVL021____AN0073
Rank:  3 mcs size: 13 sim: 0.5 O=C(Oc1c(cccc1)C(=O)N(C2CC2)C)c3c(OC(=O)C)cccc3 WXVL021____AN0276
Rank:  4 mcs size: 13 sim: 0.5 O=C(Oc1c(cccc1)C(=O)N(CC#N)C)c2c(OC(=O)C)cccc2 WXVL021____AN2942
...
```

For every query molecule, the respective hit molecules are printed to the console with information on the rank, the MCS size, the MCS similarity, SMILES representation and name of the molecule. By default, 100 hit molecules per query are printed to the console. If you want to increase or decrease that number, please have a look at section 4.4. Of course, you can also specify an

output file to which the hit molecules will be written by using the `-o` option (one separate output file per query) or `-O` option (a single output file containing the hit molecules for all queries, see Section 4.3 for more details). You can specify either a SD file or CSV file, e.g.

```
./spacemacs -i "CC(C)C(=O)N" -s my_space.space -o my_output.csv
```

The output file(s) contain detailed information for every result molecule in addition to its SMILES (in CSV file) or MOL block (in SD file) representation:

- **result-rank**: rank among all hit molecules for that query
- **search-type**: type of search (MCSSize or Substructure, see Section 6)
- **mcs-size**: size of the common substructure between query and hit molecule (number of heavy atoms)
- **result-size**: size of the result/hit molecule (number of heavy atoms)
- **query-size**: size of the query molecule (number of heavy atoms)
- **mcs-similarity**: Tanimoto MCS similarity (see Section 6.1)
- **result-name**: name of the result molecule
- **query-name**: name of the query molecule
- **query-smiles**: SMILES of the query molecule
- **space**: name of the searched space(s) or library
- **reaction-name**: name of the reaction that constructs the result molecule
- **reagent(1-5)-name**: name of building block (1-5) from which the result molecule is constructed (relevant only for space searches, number depends on reaction type)
- **reagent(1-5)-smiles**: SMILES of building block (1-5) from which the result molecule is constructed (relevant only for space searches, number depends on reaction type)

4.3 Program Options

-i / ––input (arg) Specify a file containing the query molecules. Supported file formats are SDF, MOL and MOL2. You can also provide a text file containing multiple line-separated SMILES (file extension must be .smi or .smiles). Instead of a query file you can also specify a single SMILES string enclosed by quotation marks, e.g. "CNCCC1=CC=CC(=C1)C(O)=O".
NOTE: The `-i` option is required.

Examples:

```
spacemacs -i myquery.sdf

spacemacs -i "CC1=CC=CN=C1"
```

-s / ––search-files (arg) Specify a chemical space file or library file. Also multiple files can be used. And even libraries (enumerated collections of molecules) and chemical spaces (cf. above) can be mixed. Supported file types for libraries are SDF, SMILES and MOL2. For chemical spaces, .space and .fsf can be used. Please note: if you use a .fsf file here you have to make sure that the corresponding .fif files and fragment files (.smi) specified within the .fsf file are in the correct relative paths.
NOTE: The `-s` option is required.

Examples:

```
spacemacs -s mychemicalspace.space

spacemacs -s mychemicalspace1.space  mychemicalspace2.space

spacemacs -s mychemicalspace.fsf

spacemacs -s mylibrary.smi
```

-o / ––output-files (arg) Specify the base name for the output files as argument here. Output will be written either as SDF or in CSV format — or both. Specify the type of output by the file extension. The SDF format will contain the similarities in respective SD data fields. The CSV file will contain the similarities in a comma separated ASCII file and include SMILES for the molecules (see Section 4.2 for detailed output description).
NOTE: If you have multiple queries in your input file, then a separate CSV or SDF file will be written per query! To write all results from multi-query input

files in a single output file, see the `--single-output-files` option below.
NOTE: If you do not specify an output file, reduced output will be written to the command line (STDOUT).

Examples:

```
spacemacs -o myoutput.sdf

spacemacs -o myoutputtable.csv

spacemacs -o myoutput.sdf myoutputtable.csv
```

The latter example outputs both, one sdf and one csv file per query contained in your input file. The names of the output files will have the following general structure:

```
myoutput_{querynumber}.sdf  myoutputtable_{querynumber}.csv
```

-O / --single-output-files (arg) Specify the name for the output files as argument here. Usage is the same as for the `--output-files` option (see above). As a difference, all results from multi-query input files will be written to a single output file (concatenated results). It is also possible to write both a single CSV file and a single SDF file containing all results for all queries at the same time (see last example).
NOTE: If you do not specify an output file, reduced output will be written to the command line (STDOUT).

Examples:

```
spacemacs -O singleoutput.sdf

spacemacs -O singleoutput.csv

spacemacs -O singleoutput.sdf singleoutput.csv
```

-m / --match-image-base-file (arg) Specify a base name for the match image output files as argument here. This will generate (one per hit molecule, so potentially many!) output images that highlight the common substructure between query and hit molecule (see Figure 2). Depending on the file extension you specify, images will be written as png, pdf, or vector-based svg files.
NOTE: Generation of match images leads to extended runtimes.

Example:

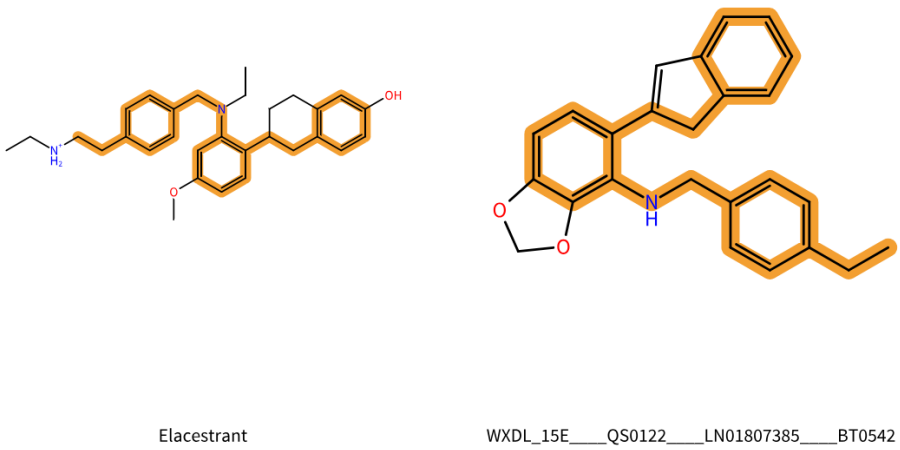

Figure 2: *Example of a match image. The query is on the left side, the hit molecule on the right side. The common substructure is highlighted in orange.*

```
spacemacs -m matching.png
```

This call will generate *one* png file per hit molecule(!), the file names will look like:

`matching_{querynumber}_{hitnumber}.png`.

-t / --search-type (arg) Specify the search type. Default value is 1, i.e. a MCS size search is performed. See Section 6 for more information.

- `0` Substructure search. The query must be fully contained in hit the molecule.
- `1` MCS size search. Maximum common substructure search maximizing the number of mapped atoms between query and hit molecule.

4.4 Configuration

--max-nof-results (arg = 100) Takes a number between 1 and 1,000,000 as argument. This option controls the number of hit molecules per query that will be output. The default is 100, that means 100 hit molecules for every query molecule are written to the output files. The results will be ranked by the MCS

size as the first criterion (from larger to smaller MCS size). If the MCS size is the same between two hit molecules, they are ranked in ascending order according to their number of heavy atoms (result-size). See also Section 6.1.

Example:

```
spacemacs --max-nof-results 1000
```

––expand-alternative-results If you specify this option, the same hit molecule may appear up to 5 times in your output. In chemical spaces, the same molecule can be formed in different reactions with different reagents/building blocks. Up to 5 different possibilities are written to the output files if you use this option. Those molecules will all have the same rank and the same scores but will have different names and different reagents.

Example:

```
spacemacs --expand-alternative-results
```

4.5 General Options

-h / ––help Displays the command line help with short descriptions for every argument option. For more information see Section 4.1.

––license-info Shows command line information about the license setup you currently use.

––thread-count (arg) Specify the maximum number of threads used for your (maximum common) substructure searches. By default, all available logical cores of your computer are used. You may want to reduce the number of threads used if you want to run other computations on your computer at the same time, or if you share the compute resource.

––version Displays information on the version of SpaceMACS on the command line. In quoting SpaceMACS, please mention this version number.

-v / ––verbosity (arg = 2) Set the verbosity level, e.g., the level of console output, with an integer argument. The default value is 2. The following options are available:

0 Silent. No messages will be displayed in the console during the search run. Errors will be ignored whenever possible.

1 Error. Only error messages will be displayed.

2 Warning. The default setting, warnings and error messages will be displayed.

3 Workflow. In addition to errors and warnings, information on the different steps of the search are displayed on the command line.

4 Steps. In addition to the 'Workflow' option, the progress of each step is displayed in detail.

5 Maximum Common Substructure in SpaceMACS

The generic MCS problem in cheminformatics typically has four different variants. The common substructure between two molecules can be either connected or disconnected and additionally, it can be either induced or non-induced. [1] The SpaceMACS algorithm uses the **connected induced MCS** (MCIS) variant with some specific characteristics:

- acyclic bonds are mapped only to acyclic bonds
- cyclic bonds are mapped only to cyclic bonds
- aromatic bonds are mapped only to aromatic bonds
- ring atoms are allowed to be mapped on chain (non-ring) atoms (and vice versa)

The characteristics of the MCS can be illustrated with the example in Figure 3. The common substructure between query and hit molecule is highlighted in orange. The cyclopropyl group of the hit molecule is not fully part of the MCS, which demonstrates that the corresponding acyclic bonds of the query's *tert*-butyl group are not mapped to the hit molecule's cyclic bonds. However, one carbon of the cyclopropyl group (next to the nitrogen) is part of the MCS because ring atoms are mapped to non-ring atoms. This leads, in combination with the strict cyclic/acyclic bond mapping restriction, to the mapping of one

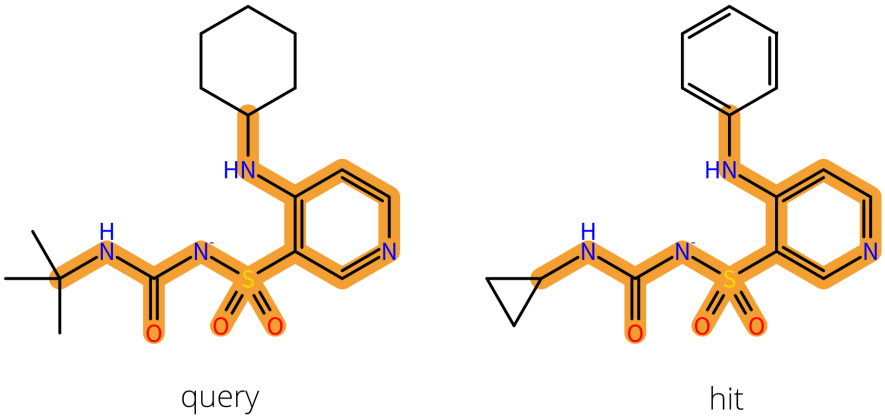

Figure 3: *Example for the MCS between a query and hit molecule calculated with the SpaceMACS algorithm.*

ring atom of the cyclopropyl group to one chain atom of the *tert*-butly group. The MCS terminates here because mapping of cyclic bonds to acyclic bonds is forbidden. The hit molecule's phenyl ring is also not fully part of the MCS. The cyclic aliphatic bonds of the query's cyclohexane ring are not mapped to the aromatic bonds of the hit molecule's phenyl ring.

6 Different Search Types

6.1 Maximum Common Substructure Search

By default, SpaceMACS performs a maximum common substructure search (`--search-type=1`, see Section 4.3). The algorithm identifies hit molecules with a maximized number of mapped heavy atoms to the query (MCS size = number of "matched" atoms between query and hit = the number of heavy atoms that are part of the MCS). The hit molecules are ranked in decreasing order according to their MCS size as the primary sorting criterion. Hit molecules with the same MCS size are ranked in ascending order according to their number of heavy atoms (result-size) as the second sorting criterion. Hit molecules with the same number of heavy atoms and same MCS size are ranked according to their (internal) SMILES representation as the third sorting criterion. Additionally, for all hit molecules a Tanimoto-like similarity value is calculated which is known from fingerprint-based similarity searches. This value is called MCS

similarity and is calculated as follows:

$$\mathrm{MCS}_{\mathrm{similarity}} = \frac{\mathrm{MCSSize}}{\mathrm{ResultSize} + \mathrm{QuerySize} - \mathrm{MCSSize}}$$

- **MCSSize**: size of the common substructure between query and hit molecule (number of heavy atoms)
- **ResultSize**: size of the result/hit molecule (number of heavy atoms)
- **QuerySize**: size of the query molecule (number of heavy atoms)

All values above are annotated in the output files (see Section 4.2 for detailed description of the output). The MCS similarity is, comparable to classical Tanimoto fingerprint similarities, a value between 0 (completely dissimilar, MCS size = 0, no common substructure) and 1 (full identity between query and hit molecule). Therefore, the MCS similarity connects traditional screening similarity with a maximum common substructure. Please note again that the hit molecules are not sorted according to this MCS similarity value but according their MCS size!

6.2 Substructure Search

It is possible to perform a classical substructure search with SpaceMACS to identify compounds in a chemical space that fully contain a specific scaffold (`--search-type=0`, see Section 4.3). In the context of the SpaceMACS algorithm the "traditional" substructure search is a special case of the MCS problem, i.e. a full maximum common substructure match is enforced: The query must be fully contained within the hit molecule (query-size = MCS size). Please be aware that, if the hit molecules do not fully contain your searched pattern, there will be no hits at all reported (and no output file(s) will be generated). If there are multiple hit molecules that fully contain the query, they are ranked in ascending order according to their number of heavy atoms (result-size) as the sorting criterion. Hit molecules with the same number of heavy atoms are ranked according to their (internal) SMILES representation as the second sorting criterion. An MCS similarity is also calculated and annotated for the results from a substructure search (see Section 6.1 for more information on MCS similarity).

7 Further Reading, References

The original ideas behind the SpaceMACS method are covered in the original publication by Robert Schmidt and Matthias Rarey.[2]

More information on the tool is available at `https://www.biosolveit.de/download/?product=spacemacs`

Complementary tools, especially also the graphical platform infiniSee, can be obtained from the BioSolveIT website (`https://biosolveit.com`).

References

[1] Hans Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 1(1):68–79, 2011. `https://doi.org/10.1002/wcms.5`

[2] Robert Schmidt, Raphael Klein, and Matthias Rarey. Maximum Common Substructure Searching in Combinatorial Make-on-Demand Compound Spaces. *Journal of Chemical Information and Modeling*, 2021. `https://doi.org/10.1021/acs.jcim.1c00640`

We wish you great success and much joy with SpaceMACS!