SpaceMACS Command Line Documentation Version 1.3

Sascha Jung & Robert Schmidt

August 19, 2024

Contents

1 Introduction

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

Please note that this package is a command line package.

SpaceMACS is a command line tool to find compounds with maximum common substructures (MCS) to a query in combinatorial chemical spaces ("fragment spaces") of multi-billion size and beyond. It is also possible to perform "classical" substructure searches to identify molecules with an exact scaffold. SpacMACS is also part of our flagship platform infiniSee and operates in the background of the Motif Matcher mode (see Section 7).

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 (Section 6.1)
- use a Tanimoto-like similarity metric for MCS searches (MCS similarity, Section 6.2)
- conduct classical substructure searches in chemical spaces to identify molecules that contain an exact scaffold (Section 6.3)
- identify compounds with defined substitution patterns (R-group search, Section 6.4)
- perform (maximum common) substructure searches in classical enumerated libraries
- visualize the common substructure between query and hit molecule (Figure 2)

SpaceMACS traverses huge Chemical Spaces using "Lego-like" chemical reaction combinatorics behind the scenes. To conduct quick calculations, we formalize reactions and encode them as linking reactions with associated fragments (see Figure 1). The fragments carry specific linkers, represented by lego bricks in Figure 1. The fragments can only be combined in a chemically meaningful and synthetically accessible way, which is determined by the reaction definitions. For example, fragments with a grey brick can only be attached to fragments with a red brick, those carrying a green brick only to those with an orange brick. SpaceMACS now calculates the MCS on the fragments and enumerates those hit molecules for which the overall MCS is maximized. Detailed information on the basic ideas of the algorithm can be found in the original publication by Schmidt et al.[2]

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.

[1] `https://www.biosolveit.de/chemical-spaces/`
[2] `https://www.biosolveit.de/download/?product=spacelight`
[3] `https://www.biosolveit.de/download/?product=ftrees`

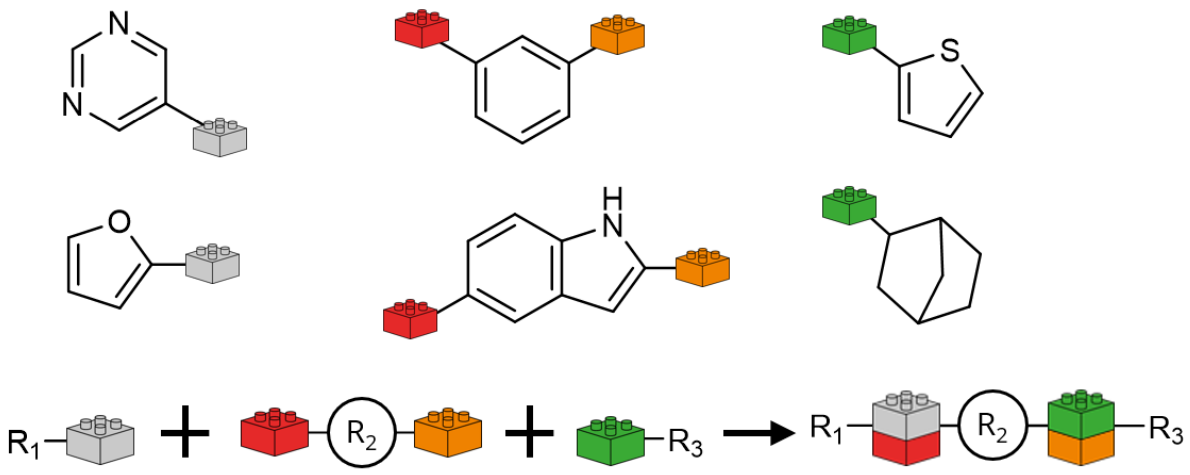

Figure 1: *Example of a formalized reaction and associated fragments encoded with "Lego-like" linkers.*

2 Technical Prerequisites

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

- The **SpaceMACS 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: `mailto: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 `mailto: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.

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

Your license is all set? You unpacked the installation archive? You downloaded a chemical space file (`.space` file, from `https://www.biosolveit.de/chemical-spaces/`)? 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 an SD file (`.sdf`), a SMILES file (`.smi` or `.smiles`, containing line-separated SMILES), a `.mol` or `.mol2` file with one or multiple entries. For quick searches, the input can just as well be a SMILES string enclosed by quotation marks, for example:

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

By default, the 100 hit molecules with the largest common substructure to the query are written as SMILES to your console/shell (STDOUT). To write your results to an output file (`.csv` or `.sdf`), use either the `-o / --output-files` option (writes a separate output file for every query contained in your input file) or the `-O / --single-output-files` option (writes a single output file with the concatenated results for all queries), for example:

```
./spacemacs -i <path/to/queries.sdf> -s <path/to/chemical_space.space>
            -o <path/to/output.csv>
```

The output file contains the structure of the result molecules as well as detailed information on the used search type, MCS size and MCS similarity score (see page 7 for more information). You can adjust the number of results by using the `--max-nof-solutions` option:

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

To find 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 1
```

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 file containing molecules or SMARTS. Supported file types
                                      are *.smi, *.smiles, *.mol, *.mol2, *.sdf, *.sma and *.smarts.
                                      Alternatively, a single SMILES (e.g., 'c1ccccc1') or SMARTS (e.g.,
                                      '[CR0][#7D3]') query can be given as an argument.
-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 (=2)         Type of the performed search:
                                          0 R-group search: Modified substructure search, finding
                                            substructures with defined substitution patterns.
                                            E.g. 'c1ccccc1[R*]' as SMILES or '[n,c]1ccccc1[?R?]' as SMARTS.
                                          1 Substructure search finding results with the exact
                                            substructure.
                                          2 MCS search finding results with the maximum common
                                            substructure.
                                          3 Similarity search using a Tanimoto inspired MCS similarity
                                            metric.

Configuration:
--max-nof-results arg (=100)          Maximum number of top-ranking result molecules [1 to 1000000].
--min-similarity-threshold arg (=0)   Similarity threshold below which molecules are discarded [0.0 to
                                      1.0]. This requires '--search-type 3'.
--expand-alternative-results [=arg(=1)]
                                      Write alternative results based on alternative reaction paths.
--min-result-size arg                 Specify the minimum number of heavy atoms that a result molecule
                                      must have.
--max-result-size arg                 Specify the maximum number of heavy atoms that a result molecule
                                      is allowed to have.

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 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       ASA
Rank:  1  mcs size: 13  O=C(Oc1c(cccc1)C(=O)N(C)C)c2c(OC(=O)C)cccc2 WXVL021____AN0032
Rank:  2  mcs size: 13  O=C(Oc1c(cccc1)C(=O)N(C(C)C)C)c2c(OC(=O)C)cccc2 WXVL021____AN0073
Rank:  3  mcs size: 13  O=C(Oc1c(cccc1)C(=O)N(C2CC2)C)c3c(OC(=O)C)cccc3 WXVL021____AN0276
Rank:  4  mcs size: 13  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, 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 the `--max-nof-results` option (see page 10).

Of course, you can also write the result molecules to an output file either with the `-o` option (one separate output file per query) or with the `-O` option (a single output file with results molecules for all queries). See page 7 for more details.

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`) or SMARTS (`.sma` or `.smarts`). Instead of a query file you can also specify a single SMILES or SMARTS string enclosed by quotation marks. It is also possible to use the `-i` option multiple times in a row (see examples below).
NOTE: Recursive SMARTS (e.g. `[C$(C(=[OD1])[OD1])]`) are not supported!
NOTE: The `-i` option is required.

Examples:

```
spacemacs -i myquery.sdf

spacemacs -i myquery.sdf -i mydrugs.smi

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

spacemacs -i "[c,n]:1ccc(cc1)*"
```

-s [--search-files] arg Specify the chemical space file or library file to be searched. Also multiple files can be specified at once or the `-s` option can be used several times in a row. And even libraries and chemical spaces can be mixed (see examples below). Supported file types for libraries are SDF, SMILES (`.smi` and `.smiles` containing line-separated SMILES) and MOL2. For chemical spaces, `.space` and `.fsf` can be used. Please note: If you specify a `.fsf` file you have to make sure that the corresponding `.flf` files and fragment files (`.smi`) are in the appropriate 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. The output will be written either as SD file (`.sdf`) or `.csv` file — or both. Specify the desired output file type by the file extension. The SD file will contain additional information for every result molecule in dedicated SD data fields. The CSV file will contain the result molecules as SMILES together with additional information (see below).
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 `--single-output-files` option below.
NOTE: If you do not specify an output file, reduced output will be written to the command line (STDOUT, see Section 4.2).

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` and `myoutputtable_{querynumber}.csv`

The output file(s) contain additional information for every result molecule:

- **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 6.2)
- **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)

–O [––single-output-files] arg Specify the name for the output files as argument. The output will be written either as SD file (`.sdf`) or `.csv` file — or both. Specify the desired output file type by the file extension. The SD file will contain additional information for every result molecule in dedicated SD data fields. The CSV file will contain the result molecules as SMILES together with additional information (see page 7).
As a difference to the `--output-files` option (see above), 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 CSV file and a SD file at the same time (see last example below).
NOTE: If you do not specify an output file, reduced output will be written to the command line (STDOUT, see Section 4.2).

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 specified, images will be written as `.png`, `.pdf`, or vector-based `.svg` files.
NOTE: Generation of match images leads to extended runtimes!
NOTE: Match images cannot be created for SMARTS queries!

Example:

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

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

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

Figure 2 shows an example match image for Acetylsalicylic Acid (ASA) as query.

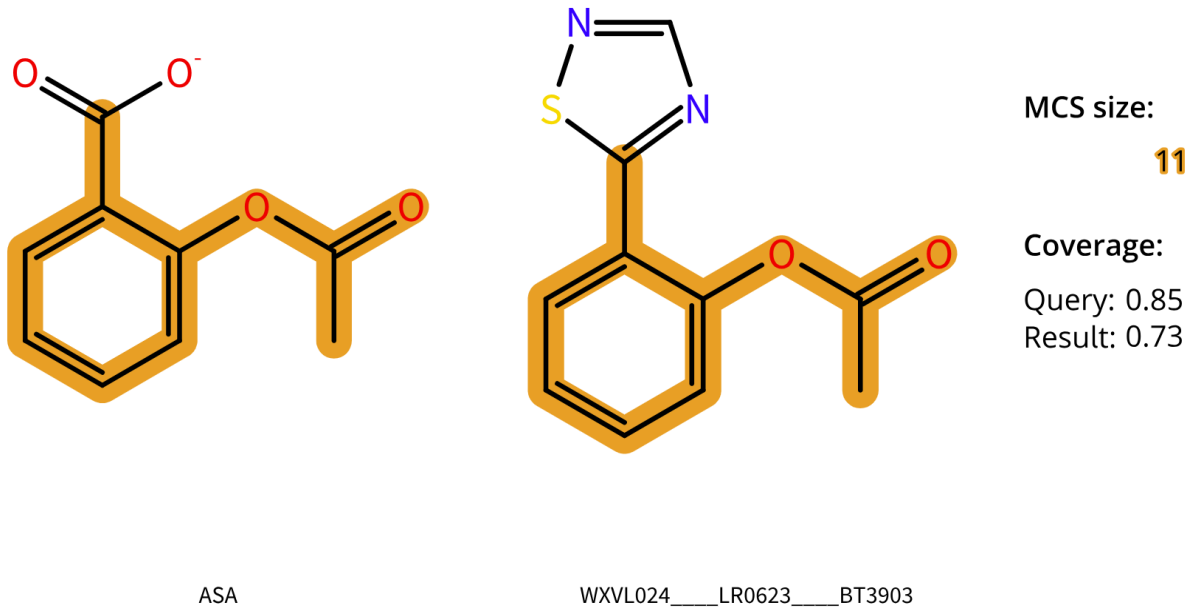

Figure 2: *Example of a match image. The query is on the left side, the hit molecule on the right side. The common substructure is highlighted in orange.*

The common substructure of query (left molecule) and result (right molecule) is highlighted in orange. The size of the common substructure (MCS size = number of heavy atoms of the common substructure) is annotated and highlighted in the legend on the right side. Additionally, the query coverage (MCS size divided by the number of query heavy atoms) and result coverage (MCS size divided by the number of result heavy atoms) are annotated, expressing the percentage of the query or result covered by the MCS, respectively.

–t [––search-type] arg(=2) Specify the search type. Default value is 2, i.e. a MCS size search is performed. See Section 6 for more information.

0 **R-group search.** Modified substructure search to identify defined substitution patterns. Requires special input. See Section 6.4 for detailed information.

1 **Substructure search.** Exact substructure search, i.e. the query must be contained as a complete substructure in the result molecules. See Section 6.3 for detailed information.

2 **MCS size search.** Maximum common substructure search using the size of the common substructure as metric. The results are ranked by decreasing MCS size. See Section 6.1 for detailed information. The default search type.

3 **MCS similarity search.** Maximum common substructure search using a Tanimoto-like similarity metric. The results are ranked by decreasing MCS similarity value. See Section 6.2 for detailed information.

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 parameter controls the TOP number of results that are either sorted by decreasing MCS size or by decreasing MCS similarity (see Section 6 for detailed information). The output is limited to 1,000,000 result molecules.

Example:

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

––min-similarity-threshold arg(=0) Takes a number between 0 and 1 as argument. This parameter adjusts the minimum similarity threshold below which the result molecules are discarded. By default, the value is 0, e.g. no result molecules are discarded. This option can only be used in combination with the MCS similarity search type (`-t=3`, see page 9 and Section 6.2).

Example:

```
spacemacs --min-similarity-threshold 0.7
```

––expand-alternative-results Expands alternative reactions for a result molecule. In chemical spaces, the same molecule can be formed in different reactions with different reagents/building blocks. If you use this option, these different possibilities are written to the output. Alternative results all have the same rank and the same similarity score but have different names and different reagents. You can also use this option to find identical results in different spaces if you specify multiple spaces at the same time (see `-s` option).

Example:

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

––min-result-size arg Specify the minimum number of heavy atoms that a result molecule *must* have. Takes an integer as argument. Limits the result set to those compounds with at least the specified number of heavy atoms. Results with a lower number of heavy atoms are discarded.

Example:

```
spacemacs --min-result-size 15
```

––max-result-size arg Specify the maximum number of heavy atoms that a result molecule *is allowed to have*. Takes an integer as argument. Limits the result set to those compounds that have at most the specified number of heavy atoms. Results with a higher number of heavy atoms are discarded.

Example:

```
spacemacs --max-result-size 30
```

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.

Example:

```
spacemacs ––help
```

––license-info Shows command line information about the license setup you currently use. If you have any problems with your license, send an email to `mailto:support@biosolveit.com` and include this information.

Example:

```
spacemacs ––license-info
```

––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 if you want to run other computations on your computer at the same time, or if you share the compute resource.

Example:

```
spacemacs ––thread-count 4
```

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

Example:

```
spacemacs ––version
```

–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.

Example:

```
spacemacs –v 0
```

5 Maximum Common Substructure in SpaceMACS

The generic MCS problem in cheminformatics 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 maximum common induced substructure** (cMCIS) 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)

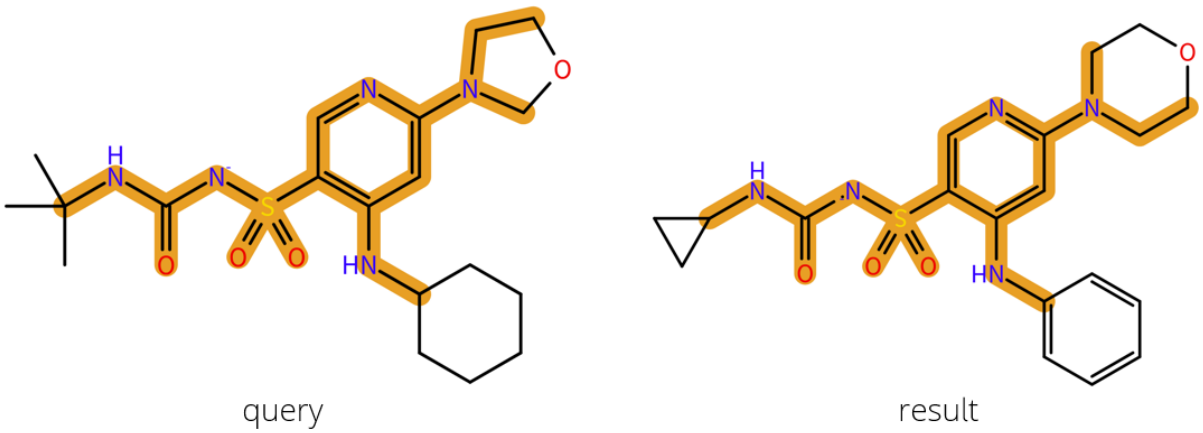

Figure 3: *Example for the MCS between a query and result molecule calculated with the SpaceMACS algorithm.*

The characteristics of the MCS can be illustrated with the example in Figure 3. The common substructure of query and result molecule is highlighted in orange. The cyclopropyl group of the result molecule is not part of the MCS, which demonstrates that the corresponding acyclic bonds of the query's *tert*-butyl group are not mapped to the result 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 a mapping between the "entry" ring atom of the cyclopropyl group and the central atom of the *tert*-butly group. The MCS terminates here because mapping of cyclic bonds to acyclic bonds is forbidden. The lacking mapping of the query's cyclohexane ring to the result's phenyl ring demonstrates that aliphatic cyclic bonds are not mapped to aromatic bonds. Again, the MCS terminates at the "entry" atom of the rings.

The non-mapped oxygen atom of the aliphatic heterocycle (5-membered oxazolidine ring in the query, 6-membered morpholine ring in the result) is an example for the induced behavior of the MCS. The oxygen of the oxazolidine ring is adjacent to two mapped carbon atoms, while the oxygen of the morpholine ring is only next to one mapped carbon atom. In the context of an induced MCS algorithm, such atoms are considered "different" and are not part of the MCS.

6 Different Search Types

6.1 MCS Size Search

By default, SpaceMACS performs a maximum common substructure (MCS) search (`--search-type=2`, see page 9). Hit molecules are ranked in decreasing order by the size of the common substructure (MCS size = number of heavy atoms that are part of the MCS between query and result) as the primary sorting criterion. Result molecules with the same MCS size are ranked in ascending order according to their size (result size = number of heavy atoms of the result molecule) as the second sorting criterion. If two or more result molecules have the same number of heavy atoms and the same MCS size, then they are ranked according to their SMILES representation as the third sorting criterion.

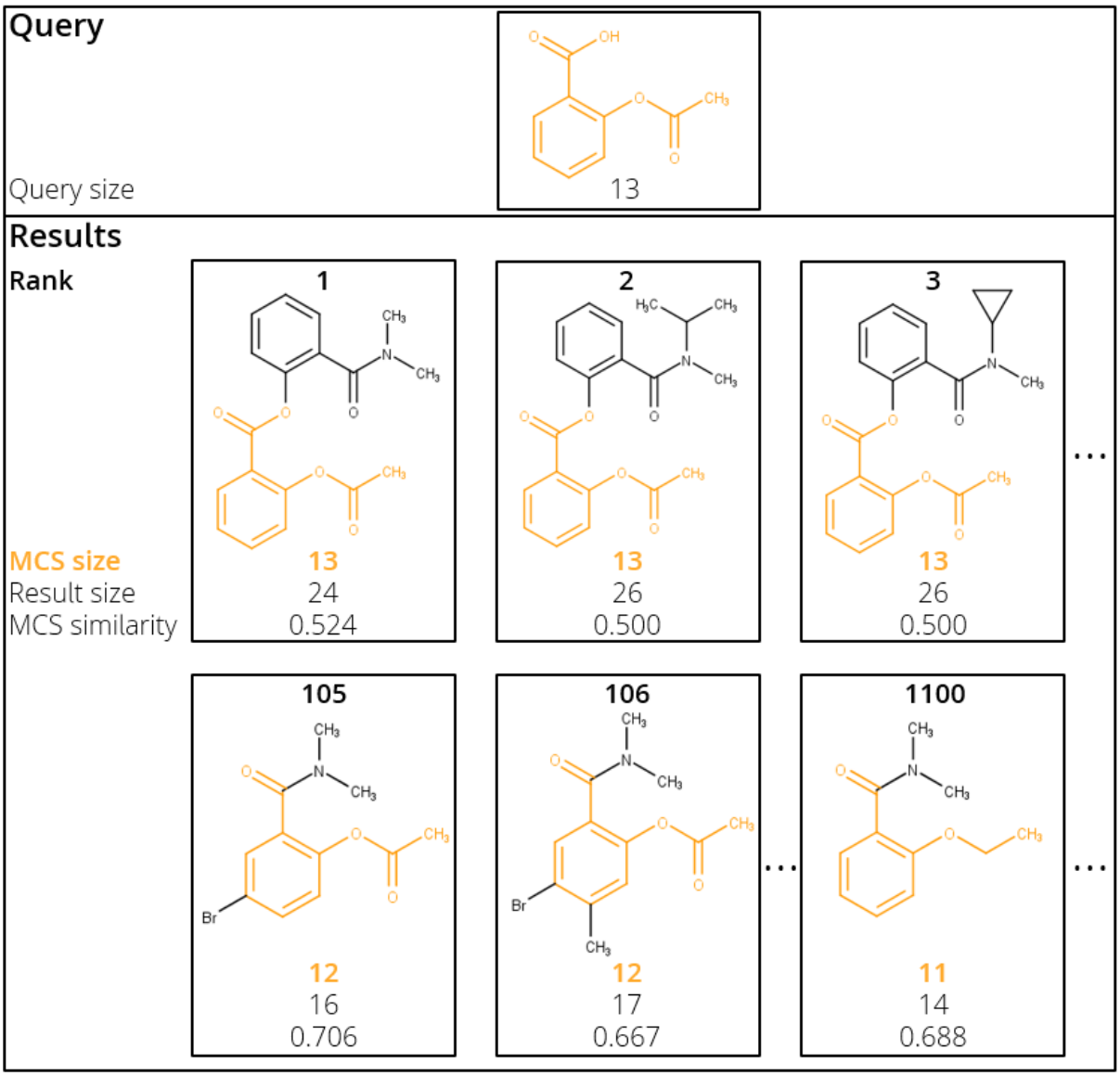

Figure 4: *Example results for a MCS size search.*

Figure 4 shows some example results for a MCS size search with Acetylsalicylic acid (ASA) as the query. On rank 1 is the smallest result (smallest result size) with the largest MCS (MCS size, highlighted in orange). In this case, the MCS size is even equal to the query size, so we have an exact substructure match here (see also Section 6.3). The same applies to the result on rank 2 which has the same MCS size, but is larger (two more heavy atoms). On rank 3, we have a result where both MCS and result size

are the same compared to rank 2, e.g. the ranking here only depends on the SMILES representation of the molecules. If we go further down the result list, we find the first result that is only a partial substructure match (MCS size < query size) on rank 105, i.e. all results from rank 4 to 104 (not shown) are full matches increasing in size. Actually, the result on rank 105 has the highest MCS similarity value and will be the top-ranked result in an MCS similarity search (see Section 6.2 and Figure 5). The result on rank 106 has again one more heavy atom but the same MCS size as result 105. On rank 1100, we find the first compound with the next smaller MCS size. And so on and so forth...

You can use the MCS size search to find compounds that share a maximized chemical motif with your query on the top ranks!

6.2 MCS Similarity Search

Instead of the MCS size as search criterion (see Section 6.1), it is also possible to use a MCS similarity metric (`--search-type=3`, see page 9). It is inspired by the Tanimoto score from fingerprint-based similarity searches. The MCS similarity 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)

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. If you use the MCS similarity search type, the result molecules are ranked according to the MCS similarity value. If two or more result molecules have the same similarity score, then they are ranked according to their SMILES representation. Figure 5 shows some example results for a MCS similarity search with Acetylsalicylic acid (ASA) as the query. Compare the ranking of the results to a MCS size search for the same query (Figure 4).

The top-ranked result by MCS similarity is found on rank 105 by MCS size, the second-ranked result by MCS similarity just on rank 1100 by MCS size, i.e. both results are not in the result set by MCS size with default settings (maximum number of results = 100, see page 10)! **The MCS similarity metric favors results which have a smaller MCS but a comparable size to the query over those with a larger MCS but completely different size!** The top-ranked result by MCS size has 24 heavy atoms and is therefore significantly larger than the query (13 heavy atoms), but the MCS is maximal (MCS size = query size). The top-ranked result by MCS similarity has only 16 heavy atoms and is therefore much more comparable in size to the query, but also the MCS is slightly smaller (12 heavy atoms). By taking the query size into account, the MCS similarity metric somehow balances the MCS size and the result size. This leads to compounds on the top ranks that share a maximized chemical motif while being similar to the query in terms of a traditional screening metric. Therefore, the MCS similarity connects traditional screening similarity with a maximum common substructure.

You can use the MCS similarity search to find compounds that have a maximized chemical motif and a comparable size to your query on the top ranks!

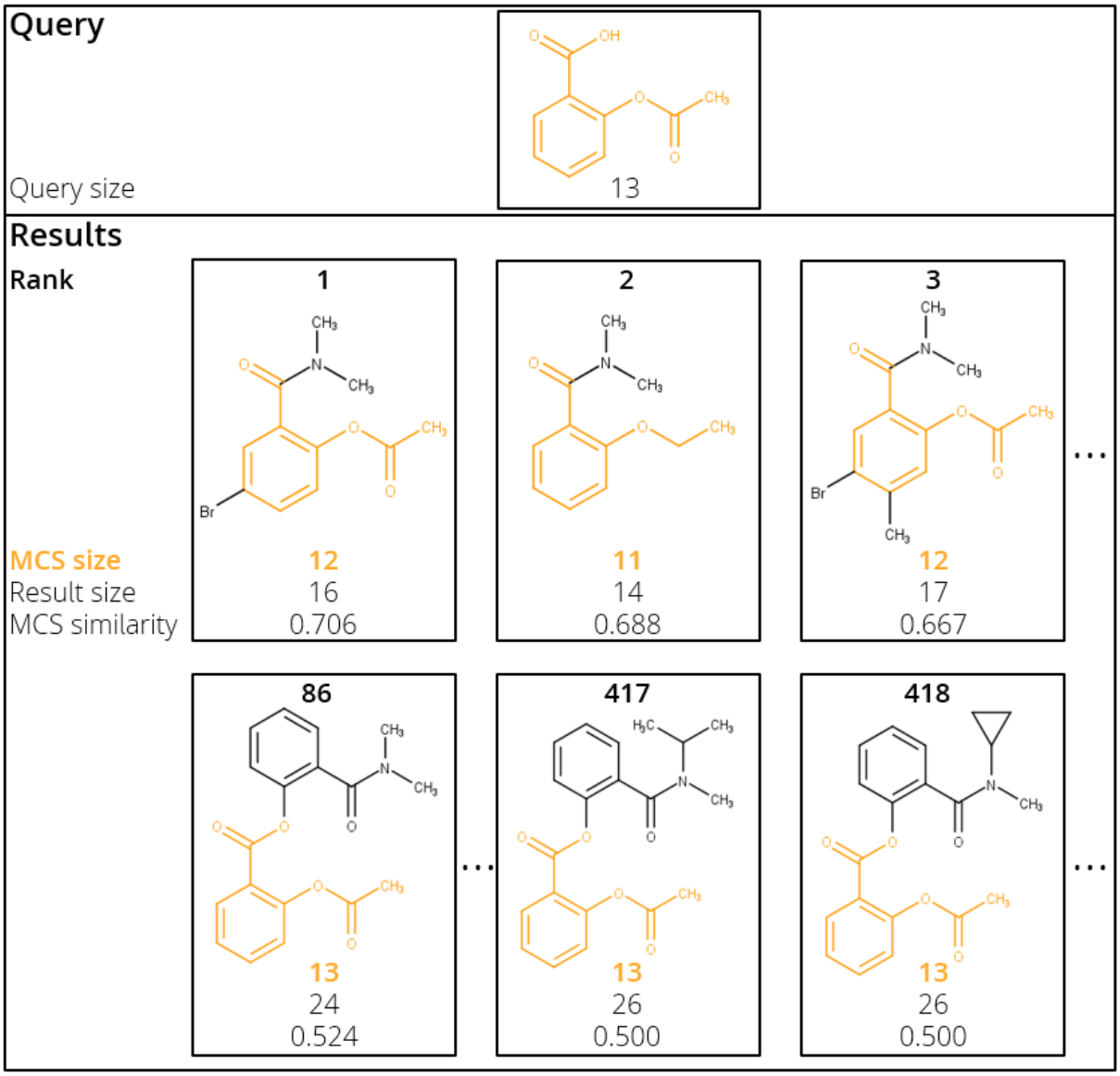

Figure 5: *Example results for a MCS similarity search.*

6.3 Exact Substructure Search

SpaceMACS can perform a "traditional" substructure search to find only those compounds that fully contain the query scaffold (`--search-type=1`, see page 9). 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 result molecule (query size = MCS size). Please be aware that, if there are no results that fully contain your searched pattern, there will be no results reported at all (and no output file(s) will be generated). If there are multiple result 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. Result molecules with the same number of heavy atoms are ranked according to their (internal) SMILES representation as the second sorting criterion. As the query size is always equal to the MCS size, the results are also implicitly sorted by MCS similarity (MCS similarity equation reduces to MCSSize/ResultSize, see Section 6.2). Figure 6 shows some example results for an exact substructure search with Acetylsalicylic acid (ASA) as the query. Compare to the results of an MCS size search (Figure 4) and MCS similarity search (Figure 5) for the same query. The result set contains only molecules that fully contain the ASA motif as a substructure (i.e. it is identical to the top 104 results of the MCS size search in Figure 4).

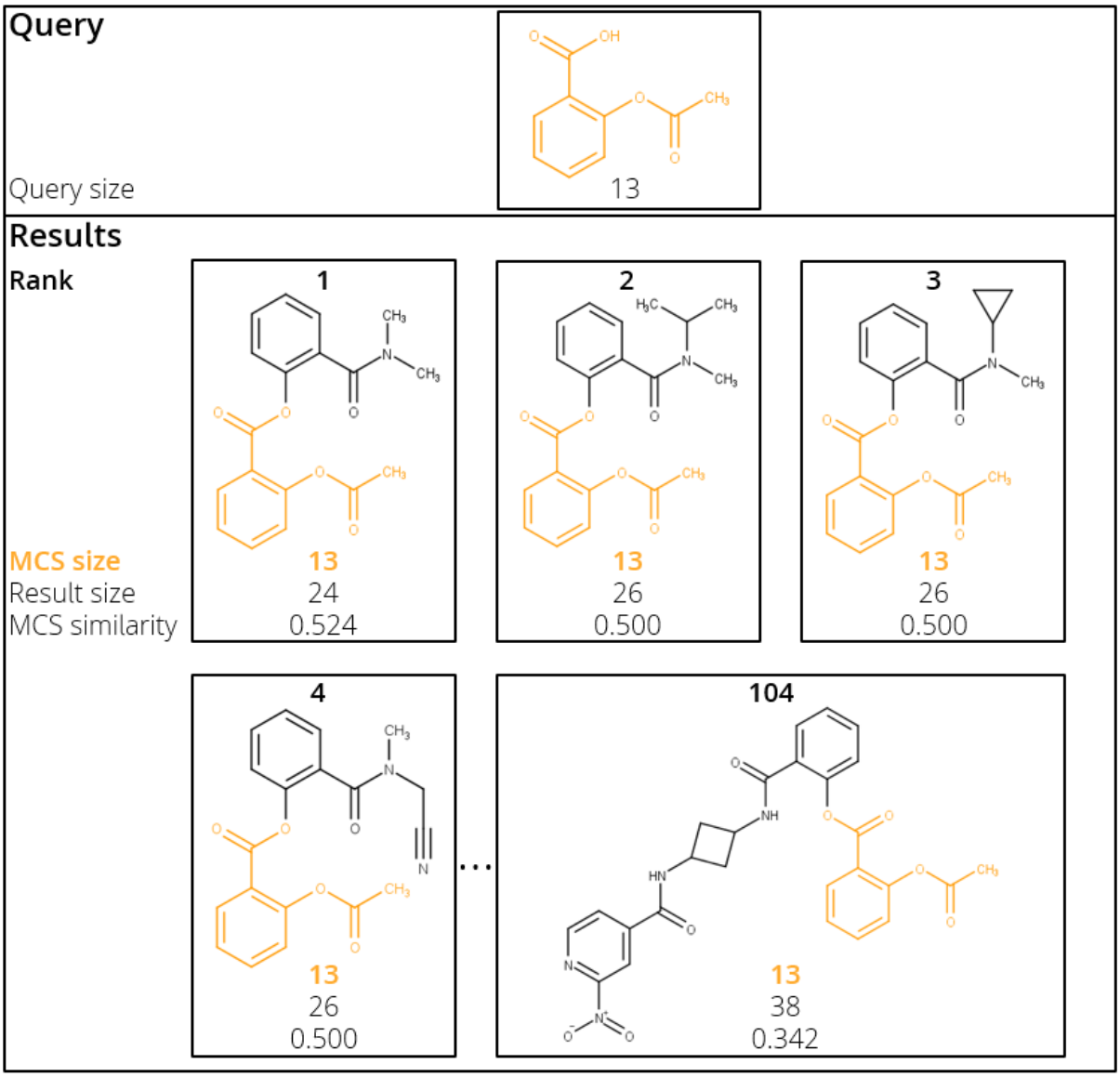

Figure 6: *Example results for an exact substructure search.*

You can use the exact substructure search to find compounds that contain the full chemical motif of your query!

6.4 R-group Search

SpaceMACS can search for substructures with defined substitution patterns (`--search-type=0`, see page 9). To perform an R-group search, special input is required. At least one R-group must be specified in the query (see query in Figure 7). The R-group represents any substituent with at least one heavy atom at this particular position. This is interpreted in such a way that **only at this position** a further substituent is allowed. All other positions must be **exactly** the same as in the query, i.e. the number of hydrogens is also considered here and must be identical to the query. In this sense, **the R-group search is a strictly exact substructure search that allows only modifications at the position of the R**. The results must fully contain the query and are only allowed to have a further substituent at the R position. In Figure 7, Acetylsalicylic acid (ASA) is used as an example again (see Figures 4, 5 and 6 for comparison).

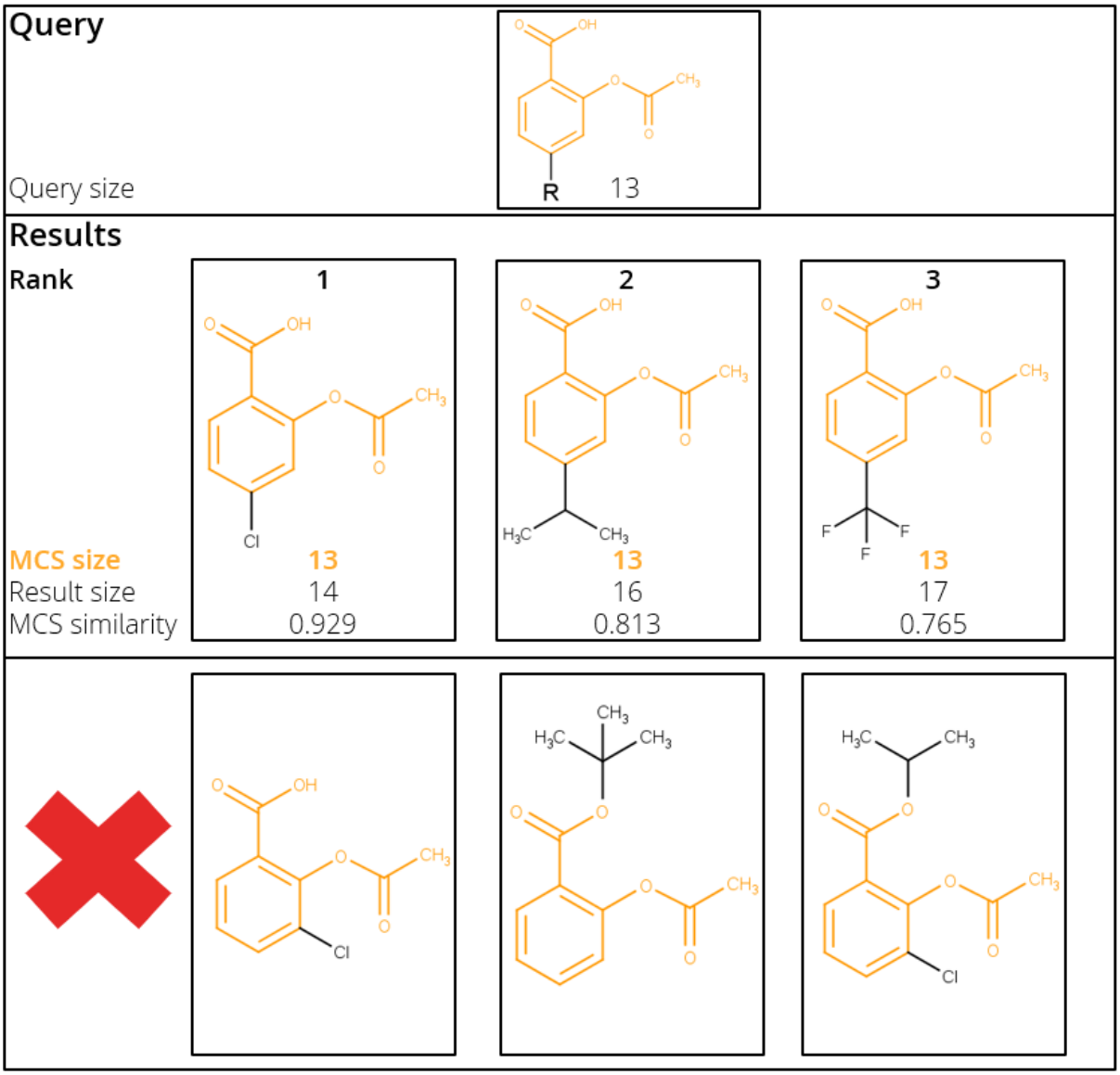

Figure 7: *Example results for an R-group search (top). Examples for compounds that are not valid results of an R-group search (bottom, red X).*

In this example, the R-group is located in *p*-position to carboxyl group, i.e. only compounds that fully contain the query motif **and** only have a further substituent at the *p*-position are valid results. The results are ranked according to the same criterion as for the exact substructure search (ascending result size, see Section 6.3). Figure 7 also shows compounds that are not valid results of an R-group search (lower part, marked with the big red X). For example, a substituent in *m*-position to the carboxyl group or at the carboxyl group itself are not valid results in an R-group search (but they would be in an exact substructure search of course, see Section 6.3).

The special input required for the R-group search can be easily prepared with chemical drawing tools like MarvinSketch. Simply draw your structure and add one or more "R" at the desired position(s). Save the drawing as SDF or export as SMILES and you can directly use it as input for R-group searches. In Figure 8, all accepted types of input are listed. For SMILES queries, there are four different possibilities to specify an R-group. It is also possible to use SMARTS queries for R-group searches. In this case, it is mandatory to specify the R-group as `[?R?]`. This syntax cannot be directly exported from standard drawing tools, you have to adjust it manually. In all cases, you can also specify multiple R-groups in a single query.

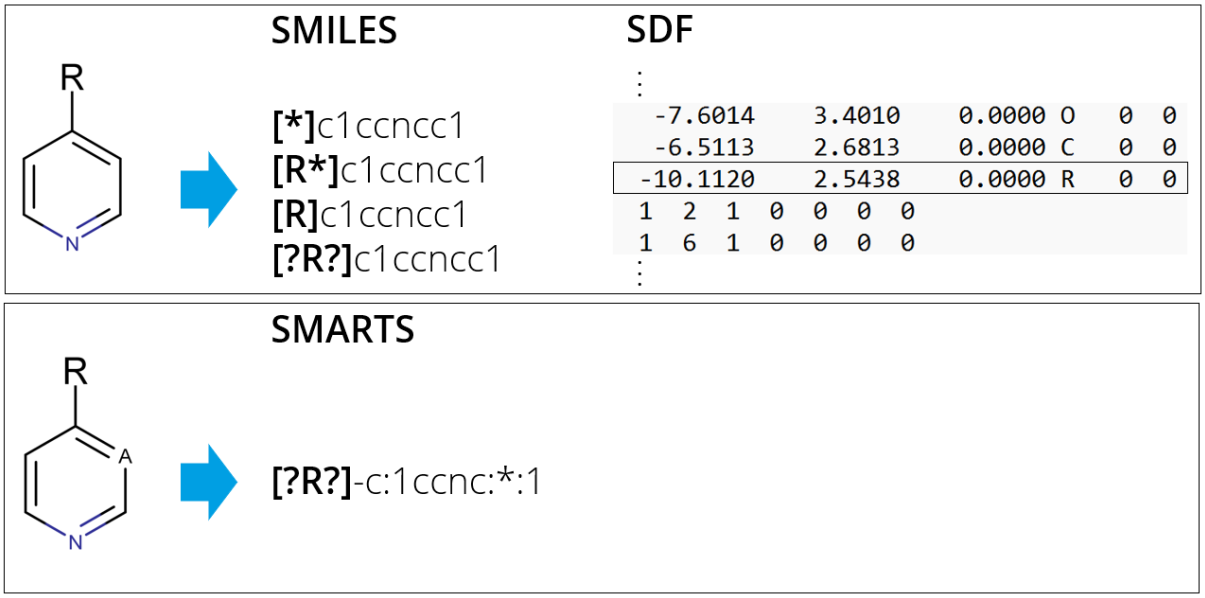

Figure 8: *Example input for R-group searches.*

Additionally, you can use single-, double- and triple-bonded R-groups in your query. The bond is interpreted explicitly, i.e. a query with a single-bonded R-group will only give results with a single-bonded substituent at this particular position, queries with double and triple-bonded R-groups only results with double-bonded or triple-bonded substituents, respectively (see Figure 9).

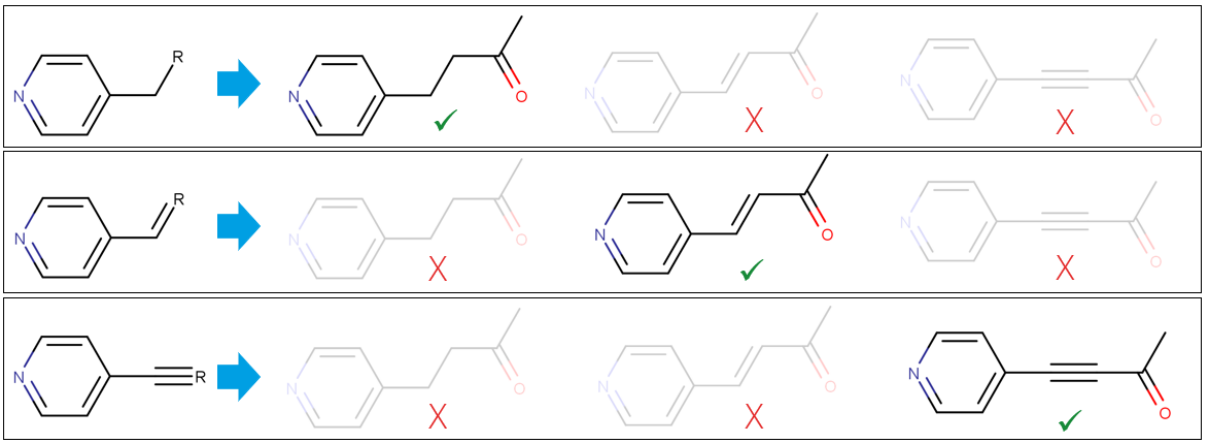

Figure 9: *Example results for queries with differently bound R-groups.*

You can use the R-group search to find compounds that contain the full chemical motif of your query with a defined substitution pattern!

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`

If you prefer to control the SpaceMACS algorithm through a graphical interface, download our platform infiniSee. MCS similarity searches and exact substructure searches can be performed in the Motif Matcher mode.
(`https://www.biosolveit.de/download/?product=infinisee`)

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!