SpaceLight Command Line Documentation
Version 1.3

Sascha Jung & Marcus Gastreich

March 7, 2024

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.

SpaceLight is a command line package to navigate combinatorial chemical spaces ("fragment spaces") of multi-billion size and beyond using classical similarity descriptors (fingerprints). SpaceLight is also part of our flagship platform infiniSee and operates in the background of the Analog Hunter mode (`https://www.biosolveit.de/download/?product=infinisee`).

SpaceLight is a perfect complement to our fuzzy Feature Trees technology (FTrees, `https://www.biosolveit.de/download/?product=ftrees`). Whereas the FTrees descriptor has a pronounced strength in detecting **distant** neighbors with chemical similarity, SpaceLight will find the **close-by** neighbors in a chemical space.

SpaceLight lets you conduct

- very fast, **2D similarity searching** across vast combinatorial Chemical Spaces [3] for your query molecules

- similarity searches using a variety of different fingerprint types and sizes (ECFP and CSFPs)

- similarity searches in enumerated library files (SMILES, SDF or MOL2)

SpaceLight traverses huge chemical spaces using Lego-like chemical reaction combinatorics behind the scenes. To conduct quick calculations, we formalize reactions and encode them as pseudo-linking reactions. Per reaction, one so-called "topology graph" is stored and every node of this graph contains formalized, virtual building blocks, or as we sometimes call them, "reaction fate foreseeing" fragments (Figure 1). For more information on space generation, please have a look at our CoLibri package (`https://www.biosolveit.de/products/#CoLibri`).

The search then uses fast combinatorial algorithms and can deliver results in a few seconds only — even for very large, multi-billion sized spaces and beyond.

The implemented 2D similarity descriptors [1] are specially adapted versions of descriptors that you may know from "2D / Tanimoto" similarity calculations from other tools. They need to be specially prepared beforehand for every chemical space. Spaces ready to be used with SpaceLight can be downloaded
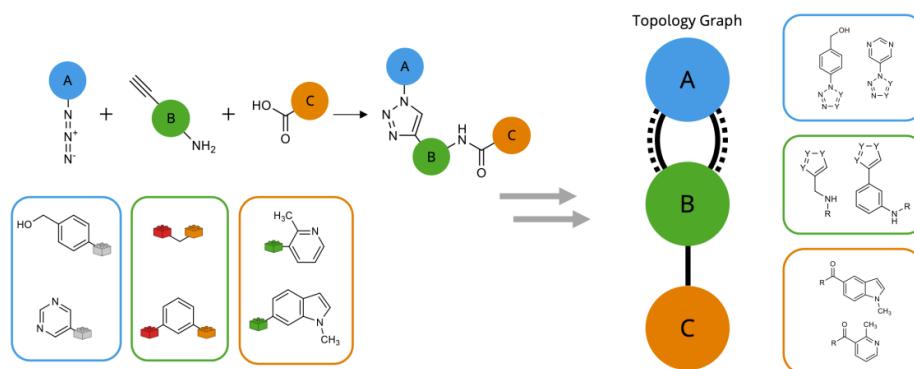
Figure 1: *Example of topology graph representing a reaction with combinatorial fragments.*

from our website (`https://www.biosolveit.de/infiniSee/#chemical_spaces`).
Alternatively, you can prepare your own spaces with the CoLibri package (see above).

Summarizing, with SpaceLight you can conduct 2D similarity searches with well-known fingerprints (e.g. ECFP4) in much bigger spaces than with other methods, and you also require much less time, making it possible to search vast spaces even on modest, standard hardware. Additionally, you can also search traditional "medium sized" enumerated libraries.

2 Technical Prerequisites

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

- The **SpaceLight package**
  (`https://biosolveit.de/download/?product=spacelight`)
  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 environ-
  ment (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 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 (spacelight.exe, spacelight, or SpaceLight — depending on your
operating system). For macos please read on...

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

`/Applications/SpaceLight.app/Contents/MacOS/SpaceLight`

To place the short term test license there, you will have to go into the *.app package
using a right mouse click on SpaceLight.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 SpaceLight executable. If you are about to use the **test license**, place
it 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 SpaceLight for the first time, go to the Finder, and navigate to the Ap-
plications folder. Do a right(!) click on SpaceLight.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 SpaceLight within your shell/terminal/environment.

3 Jump Start: Finding Analogs in Large Chemical Spaces

Your license is all set? You unpacked the installation archive? You downloaded or prepared a fragment space file (`.space` file from `https://www.biosolveit.de/infiniSee/#chemical_spaces` or `.tfsdb` file)? Then here is a typical call to search a query against a fragment space:

```
./spacelight -i <path/query.sdf> -s <path/fragmentspace.space>
```

```
./spacelight -i <path/query.sdf> -s <path/fragmentspace.tfsdb>
```

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:

```
./spacelight -i "CC(C)C(=O)N" -s <path/fragmentspace.space>
```

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

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

The output file contains the structure of the result molecules as well as detailed information on the used fingerprint type and similarity score (see page 9 for more information). You can adjust the number of results by using the `--max-nof-solutions` option:

```
./spacelight -i "CC(C)C(=O)N" -s <path/fragmentspace.space> -o <path/output.csv>
        --max-nof-solutions 30
```

The fragment connected subgraph fingerprint (fCSFP4) is used as the default similarity descriptor (see page 13).[1] The Tanimoto similarity is used as similarity measure. The similarity values are normalized between 0 (no similarity) and

5

1 (identical, in the framework of the descriptor). The default descriptor is **not** stereo-aware, but it captures elements, connectivity, valence, and ring membership. As such it will find "near-neighbor" similar compounds in a chemical space. You can limit the output to those results which exceed a certain similarity by using the `--min-similarity-threshold` option:

```
./spacelight -i "CC(C)C(=O)N" -s <path/fragmentspace.space> -o <path/output.csv>
        --min-similarity-threshold 0.7
```

Instead of CSFP fingerprints, you can also use the well-known extended connectivity fingerprints (ECFP, see page 13 and Section 5):

```
./spacelight -i "CC(C)C(=O)N" -s <path/fragmentspace.space> -f ecfp4
```

4 Command Line Options

4.1 Overview

An overview of all command line options is available by calling SpaceLight with
`--help`.

```
./spacelight --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
                                      Topological Fragment Space database files or Fragment Spaces.
                                      Supported file types are *.smi, *.smiles, *.mol, *.mol2, *.sdf,
                                      *.tfsdb, *.space and *.zip.
-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.
--gen-mapping-output [=arg(=1)]       Generates command line SpaceLight similarity descriptors and annotates
                                      them in the output file.

Configuration:
-f [ --fingerprint ] arg (=fCSFP4)    Fingerprints for searching: ECFP and CSFP variants are supported.
                                      Supported ECFP variants are ECFP0 to ECFP8. Three CSFP subtypes are
                                      available: fCSFP, iCSFP and tCSFP in variants from 1 to 5. E.g. ECFP4,
                                      fCSFP5, iCSFP4 or tCSFP3.
--min-similarity-threshold arg (=0)   Similarity threshold below which molecules are discarded [0.0 to 1.0].
--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 logical 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 `-` whereas
the longer, named options are preceded with two dashes: `--`. If an option
needs an argument (arg), you can include or omit the equals sign. Adapt the
command line usage to your operating system and shell.

4.2   Minimal Required Options

This section describes the arguments you must specify at minimum to success-fully run a similarity search. First, you must provide the path to a file containing the query compounds. All well-known data formats are supported (MOL, SDF, SMILES, 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. Alterna-tively, 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/spacelight/executable> -i <path/to/queries> -s <path/to/space_file>
```

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

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

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

```
./spacelight -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(CCCC)C O(CCCC)C
Rank:   1 sim: 0.706 O(CCCOC)C EN300-1717039
Rank:   2 sim: 0.577 O(CCC[NH2+]CCCC)C m_270004cba____8288582____9143638
Rank:   3 sim: 0.577 S(CCCOC)CCCC m_62bba____875776____9108904
Rank:   4 sim: 0.536 O(CCCC[NH2+]CCCC)C m_270004cba____8290272____9143638
Rank:   5 sim: 0.536 S(CCCCOC)CCCC m_62bba____875776____10159086
...
```

For every query molecule, the respective result molecules are printed to the console with information on the rank, similarity score, SMILES representation and name of the molecule. By default, 100 results 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 13).

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 9 for more information.

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. It is also possible to use the **-i** option multiple times in a row (see examples below).
NOTE: The `-i` option is required.

Examples:

```
spacelight -i myquery.sdf

spacelight -i myquery.sdf -i mydrugs.smi

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

-s [--search-files] arg   Specify a topological fragment space file (`.tfsdb`) or a library file (`.sdf`, `.mol2`, `.smi`, `.smiles`) or a space file (`.space`). This file is searched for close analogs to the query molecules given with the **-i** option. You can also search multiple spaces or library files at once by using the **-s** option several times in a row. And even libraries and chemical spaces can be mixed (see examples below).
NOTE: The `-s` option is required.

Examples:

```
spacelight -s mylibrary.sdf

spacelight -s mylibrary.sdf -s myspace.space

spacelight -s space1.space -s space2.space
```

-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 (see below). 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 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, see Section 4.2).

Examples:

```
spacelight -o myoutput.sdf
```

```
spacelight -o myoutputtable.csv
```

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

The latter example outputs both, one SD file 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 results for that very query.

- **fingerprint**: fingerprint descriptor used for the search

- **similarity**: Tanimoto similarity value (derived from query molecule fingerprint versus result molecule fingerprint)

- **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)

- **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)

–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 (see below). The CSV file will contain the result molecules as SMILES together with additional information (see below).
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:

```
spacelight -O singleoutput.sdf
```

```
spacelight -O singleoutput.csv
```

```
spacelight -O singleoutput.sdf singleoutput.csv
```


–m [––match-image-base-file] arg   Specify a base name for the match image output files as argument. This will generate (one per hit molecule, so potentially many!) output images that explain the matching of query versus hit molecule in 2D pictures. 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.

Example:

```
spacelight -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 matching image. The local similarities are listed and color-coded on the right hand side. The query molecule is located on the
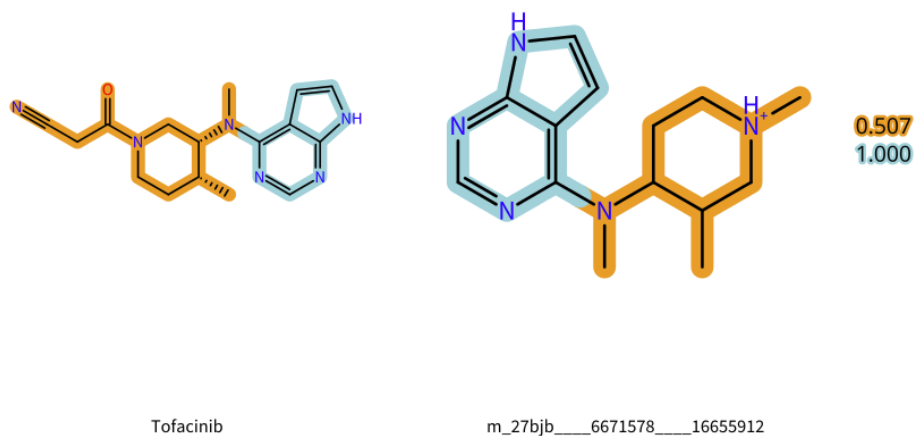
Figure 2: *Example of a match image. The query is on the left side, the hit molecule on the right side.*

left side (Tofacinib), the hit molecule on the right side. The hit molecule is colored according to the fragments/building blocks from which it is constructed. As you can see, the aromatic heterocyclic system (cyan) is identical (similarity 1) between query and hit, the part with aliphatic ring (orange) is less similar (similarity 0.507).

--gen-mapping-output   If you specify this option, for every hit molecule the local similarity values (annotated in the *similarity-descriptor* column/tag) and the corresponding substructures (annotated as semicolon-separated SMILES in the *similarity-descriptor-smiles* column/tag) are written to the output files. This is the same information that is visualized for the hit molecule in a match image (see Figure 2 and the description for the `--match-image-base-file` option above). NOTE: Calculation of local/fragment similarities increases the runtime!

Example:

```
spacelight --gen-mapping-output
```

4.4   Configuration

With the options described in this section you can adjust the algorithmic pa-
rameters of SpaceLight. For example, you can modify the fingerprint used to
derive similarity, adjust the minimal similarity threshold below which result mo-
lecules are discarded and change the maximum number of results generated
for each query molecule. All parameters in this section have default values
which are round-bracketed in the following.

–f [––fingerprint] arg(=fCSFP4)   With this option, you can adjust the fin-
gerprint type and size used for the determination of similarity between query
and result molecules. For more information on the available descriptors see
Section 5. Shortly, you can choose between three different versions of the con-
nected subgraph fingerprint (CSFP [1]), each version with features containing
from 1 up to a maximum of 5 heavy atoms (CSFP1-5): fCSFP1, fCSFP2, fCSFP3,
fCSFP4 (default), fCSFP5; iCSFP1, iCSFP2, iCSFP3, iCSFP4, iCSFP5; tCSFP1, tCSFP2,
tCSFP3, tCSFP4, tCSFP5. Additionally, you can choose the "classical" circular
extended connectivity fingerprint (ECFP [4]) with a maximum diameter of 8:
ECFP0, ECFP2, ECFP4, ECFP6, ECFP8.
NOTE: Specification is case-insensitive.

Examples:

```
spacelight -f ecfp4

spacelight -f fcsfp5

spacelight -f tCSFP3
```

––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.

Example:

```
spacelight --min-similarity-threshold 0.7
```

––max-nof-results arg(=100)   Takes a number between 1 and 1,000,000 as
argument. You can adjust the maximum number of result molecules per query
which will be written to the output files. The default value is a maximum of 100

results per query. The results will always be sorted by descending Tanimoto similarity, so the parameter controls the TOP number of results. The output is limited to 1,000,000 result molecules.

Example:

```
spacelight --max-nof-results 5000
```

––expand-alternative-results Expands alternative reactions for a hit 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:

```
spacelight --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.

Example:

```
spacelight --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:

```
spacelight --license-info
```

––thread-count arg Specify the maximum number of threads used for your similarity 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:

```
spacelight --thread-count 4
```

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

Example:

```
spacelight --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 similarity 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 similarity 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:

```
spacelight -v 0
```

5 Descriptor Choices

SpaceLight has several built-in similarity descriptors (fingerprints, see page 13). Depending on your use case, one descriptor may be more suited than another. The acronym CSFP stands for *Connected Subgraph Fingerprints*, they describe all possible chemical features ("substructures") containing up to 5 heavy atoms for a given molecule.[1] In contrast, the "classical" ECFP descriptors (*Extended Connectivity Fingerprints*) describe molecules with a circular collection of features ("sit on one atom, then collect sphere with radius 1 around you, then sphere with radius 2, and so on").[4] CSFP descriptors are particularly optimized for similarity searches in fragment spaces as their set of features minimizes information loss across fragment boundaries. See the table below that has been taken from the original SpaceLight publication; it may serve as an overview of the respective atomic properties stored for the different descriptors along with the respective features.[2]

		ECFP	fCSFP	iCSFP	tCSFP
chemical substructures	circular	x			
	all		x	x	x
	element	x	x	x	x
	connectivity	x	x		x
	connectivity in substructure		x	x	
	valence	x	x	x	
atom properties	valence in substructure		x	x	
	aromaticity				x
	π electrons				
	formal charge	x			
	weight	x			
	ring membership	x	x		

Summarizing, we suggest to use the following descriptors for the respective use cases:

- fCSFP (fragment CSFP): Use this descriptor when the results should be **highly similar** to the query compounds. This is likely the descriptor that you should be able to relate most to when comparing it with the traditional ECFP descriptors. Element, valence state, and connectivity (within the feature and to the surroundings of the feature) are captured for every atom.

- tCSFP (topological CSFP): Compared to the fCSFP, tCSFP does not take into account the atom connectivity within substructures (features) nor

the valence states, but on the other hand information on aromaticity is stored. This makes the tCSFP less strict in deriving similarity for a given compound pair compared to the fCSFP. Use this fingerprint in cases you do not find close analogs with fCSFP or ECFP fingerprints.

- iCSFP (independent CSFP): This descriptor is recommended for **substructure retrieval**. The iCSFP describes all structural features of a compound by properties of the atoms that are *independent of the surroundings of its substructures*. Therefore, it is suitable when searching for molecules that contain substructures or very similar substructures of a given query compound in an arbitrary order.

- ECFP: Extended Connectivity Fingerprints are widely used in traditional similarity searching. Use ECFPs when you would like to stay close to or compare to another tool that uses this descriptor type.

If you want to hop "deeper" into the chemical space (find "distant neighbors"), then use the "fuzzier" Feature Trees descriptor available in our FTrees tool (`https://www.biosolveit.de/download/?product=ftrees`) and implemented into our infiniSee platform (`https://www.biosolveit.de/infiniSee`).

6 Further Reading, References

The original ideas behind the SpaceLight method are covered in the publication by Louis Bellmann and Matthias Rarey.[2]

If you prefer to control the SpaceLight algorithm through a graphical interface, download our platform infiniSee. Similarity searches with ECFP4 fingerprint can be performed in the Analog Hunter mode.
(`https://www.biosolveit.de/download/?product=infinisee`)

References

[1] Louis Bellmann, Patrick Penner, and Matthias Rarey. Connected subgraph fingerprints: Representing molecules using exhaustive subgraph enumeration. *J. Chem. Inf. Model.*, 59(11):4625–4635, 2019.

[2] Louis Bellmann, Patrick Penner, and Matthias Rarey. Topological Similarity Search in Large Combinatorial Fragment Spaces. *J. Chem. Inf. Model.*, 61(1):238–251, 2021.

[3] Torsten Hoffmann and Marcus Gastreich. The next level in chemical space navigation: going far beyond enumerable compound libraries. *Drug Discovery Today*, 24(5):1148–1156, 2019.

[4] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, 2010.

We wish you great success and much joy with SpaceLight!