.2DCOOR

Generating 2D Coordinates of Molecules

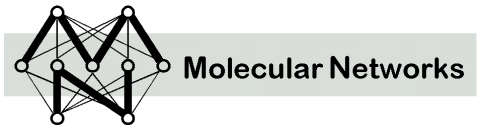
User Manual

Version 1.0

for program version 1.0 (or higher)

November 2003

Molecular Networks GmbH – Computerchemie
Nägelsbachstraße 25, 91052 Erlangen, Germany



Molecular Networks GmbH
Computerchemie
Nägelsbachstr. 25
91052 Erlangen
Germany
Phone: +49-(0)9131-815668
Fax: +49-(0)9131-815669
Email: info@mol-net.de
WWW: www.mol-net.de

Table of Contents

1. General Information about MN.2DCOOR

2D depictions of chemical compounds are 2-dimensional representations of structures which are similar to that ones a chemist would sketch. By glancing at the 2D depiction of a molecule a chemist can identify the presence and location of reactive groups, perceive its overall physicochemical property and assess structural similarities between compounds. MN.2DCOOR generates 2D depiction like atomic coordinates. The 2D structure is generated from the atom types and connectivity information only. MN.2DCOOR is a reliable tool perfectly suited for generating 2D structures of publishing quality.

The program MN.2DCOOR

- consist of one of the best 2D coordinates generators obtainable
- provides options for the homogeneous arrangement of ring systems
- is fully stereo-aware, in chains and rings
- includes an intelligent pseudo-energy optimizer
- converts datasets with 99.9% conversion rate
- handles datasets of hundreds of thousands of chemical structures
- supports the SD file format for reading the input files
- supports the SD file format for saving the output files

2. Installation

2.1. Requirements

MN.2DCOOR is available for common UNIX platforms (x86 Linux, Sun Solaris, SGI IRIX, DEC AlphaStation). It is also available for Microsoft Windows NT4/2000/XP.
The program runs in a batch mode.

2.2. Installation Steps for UNIX Operating Systems (IRIX, Solaris, Linux)

1.) Create a subdirectory, e.g., mn_2dcoor
(for system administrators when installing software locally, e.g. `/usr/local/bin/mn_2dcoor`).
2.) Copy the file `mn_2dcoor_<version>.<os>.gz` to the subdirectory `mn_2dcoor`
3.) Unpack the distribution by executing the gunzip command:
`gunzip mn_2dcoor_<version>.<os>.gz`
4.) Rename the file `mn_2dcoor_<version>.<os>` to `mn_2dcoor`.
Please note: `mn_2dcoor_<version>.<os>` is a binary file.
5.) Add the `mn_2dcoor` subdirectory name to the environment variable PATH in your .login or .cshrc files (.profile or .bashrc).

Launch MN.2DCOOR with the command

```
mn_2dcoor –version or /usr/local/bin/mn_2dcoor/mn_2dcoor –
version
```

2.3. Installation Steps for Microsoft Windows Operating Systems (NT4/2000/XP)

Although administrator privileges are not necessary, we recommend logging in as administrator. Double-click on the executable setup program and follow the instructions on the screen.
After successful installation there is no need to reboot your PC.

3. Uninstallation

3.1. Uninstallation Steps for UNIX Operating Systems (IRIX, Solaris, Linux)

Log in as root and delete the file `mn_2dcoor` in your installation directory carefully (default path during installation was `/usr/local/bin/mn_2dcoor/`).

3.2. Uninstallation Steps for Microsoft Windows Operating Systems (NT4/2000/XP)

Log in as administrator, launch the uninstaller and follow the on-screen instructions.

4. Problems and Help!

If you have any difficulties with the installation of MN.2DCOOR or if any problems occur while running MN.2DCOOR, please send all your inquiries to the following address:

Molecular Networks GmbH Computerchemie
Nägelsbachstr. 25
91052 Erlangen
Germany,

or contact us by email support@mol-net.de,
or by fax +49-(0)9131 - 81 56 69.

Please mention the program version of MN.2DCOOR (`mn_2dcoor -version`), include your input file and the output file on an MS/DOS diskette (3½") or send it to us by email. These files will help us to analyze the problem; if your system displays any error messages, please add them to your report.
You can also use the report form at the end of this manual.

5. Release Notes

5.1. Version 1.0

First release of MN.2DCOOR

6. Getting Started

6.1. UNIX operating systems

The example file `ringsystem0.sdf` submitted with the distribution contains the structure information of one molecule in SD format. The structure only contains atom connectivity as all 2D coordinates are set to NULL. Copy this example file into your working directory and type the following command:

```
mn_2dcoor -outfile ringsystem2d.sdf ringsystem0.sdf
```

MN.2DCOOR now creates the output file named `ringsystem2d.sdf` written to the same directory where the file `ringsystem0.sdf` is located. Figure 1 shows the content of this file.

6.2. Microsoft Windows operating systems

The example file `ringsystem0.sdf` submitted with the distribution contains the structure information of one molecule in SD format. The structure only contains atom connectivity as all 2D coordinates are set to NULL. Copy this example file into your working directory and open a DOS shell. Change the working directory to the directory where you installed `mn_2dcoor` by using the `cd` command then type the following command:

```
mn_2dcoor -outfile ringsystem2d.sdf ringsystem0.sdf
```

MN.2DCOOR now creates the output file named `ringsystem2d.sdf` written to the same directory where the file `ringsystem0.sdf` is located. Figure 1 shows the content of this file.

If you have no permission writing to the directory in which the program was installed, set the **-directory** option for specifying another directory:

UNIX: `mn_2dcoor -directory /tmp -outfile ringsystem2d.sdf ringsystem0.sdf`

Windows: `mn_2dcoor -directory C:/temp urate.sdf -outfile ringsystem2d.sdf  ringsystem0.sdf`

```
C18H28O4
OLmn2dcoor12290316412D 0   0.00000     0.00000

 50 50  0  0  1  0  0  0  0  0999 V2000
    3.5000    2.5981    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    3.0000    1.7321    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.5000    0.8660    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
...
24 49  1  0  0  0  0
24 50  1  0  0  0  0
M  END
$$$$
```

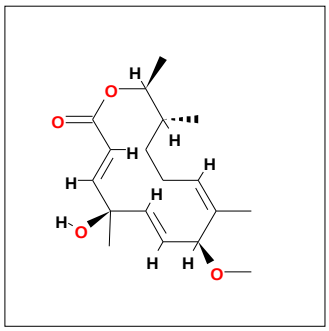


Figure 1: Content of the output file `ringsystem2d.sdf`

7. Program Use

7.1. Synopsis

The general synopsis for using MN.2DCOOR is:

```
mn_2dcoor [ -option(s) ] [ infile ]
```

An overview of the various options is given in Table 1 and in a more detailed one in the following chapter. `Infile` is the input file name. If no file name is given, the program reads from standard input.

[-**align** none/x/y/diagonal]	changes the alignment of the 2D structures
[-**count** maxrecs]	limits the amount of processed structures
[-**directory** dirname]	specifies the output directory
[-**feedback** 0/n]	prints a control message after processing a block of n items
[-**format** fmt]	specifies the output format name
[-**h**] or [-**help**]	shows a brief help message about the usage of the program
[-**stat**]	writes statistical information about the number of successfully converted records and conversion failures
[-**outfile** filename]	defines the name of the output file
[-**template** SMILES/SMARTS]	specifies a SMILES string to be used as template for structure layout
[-**templatealign** none/x/y/diagonal/ rotate/redraw/besteffort/combined]	aligns the layout of the structure according to a substructure template
[-**templatefile** file]	specifies the name of a file which contains a substructure template
[-**templatematch** strict/relaxed]	influences the match operation of fragments for the template alignment
[-**version**]	prints version and licensing information

Table 1: Overview of all options

Executing the program without any option will write structures with (re)calculated 2D coordinates to the output file.

7.2. General Program Features

The file type of the input file is automatically recognized. If no input file is specified, or the file name „-“ is used, the program reads from standard input.
If you are running MN.2DCOOR under a UNIX operating system, there are some more features reading input files (see chapter 8 “Extended Features Only Available for the UNIX Operating Systems” for more details).
The file name of the output file is either explicitly set with the **-outfile** option or automatically derived from the input file and the given output file format. The special filename `stdout` can be used to direct output to the standard output channel.

7.3. Supported file formats for input files

The program will automatically detect the file format of the input files. Two standard exchange formats are supported. Thus, there is no need for a parameter specifying the input format. A set of input files does not need to have a common format.
The supported file formats are listed in the table below.

Full Format Name	Default Input-Extension	Read	Comment
MDL Molfile	mol	Yes	
MDL SDF	sdf	Yes	

Table 2: Overview of the supported input file formats

7.4. Program Features in More Detail

-format <abbreviation of the output format>

The parameter **format** is specified for selecting the **output** format of the file conversion.
The supported file formats are listed in the table below.

Full Format Name	Default Output-Extension	Write	Comment
MDL Molfile	mol	Yes	
MDL SDF	sdf	Yes	

Table 3: Overview of the supported output file formats

Please use the abbreviation of the format names for specifying your desired file format using the -**format** option. If no output file is specified, the output has the same name (but with an updated suffix) and is written in the same directory as the input file. The extension of the resulting output file is sometimes different to the given abbreviation (see the previous table). If the output file is specified explicitly with the -**outfile** parameter, this file name including the chosen suffix, will be used.

Default value:

Parameter without a default value

Example:

Generating a MDL SD-file with (re)calculated 2D coordinates:

```
mn_2dcoor -format sdf ringsystem0.sdf
```

Remarks:

If this option is not used, an attempt is made to guess the output file format from its suffix by the given *outfile* parameter.

-outfile <filename.extension>

The parameter **outfile** defines the name of the output file. MN.2DCOOR automatically recognizes the desired output format, thus in most cases it is not necessary to specify the output format.

If your are using MN.2DCOOR on a UNIX operating system the output file name can also be an anonymous ftp URL.

Default value:

Parameter without a default value

Example:

Generating a MDL SD-file with (re)calculated 2D coordinates:

```
mn_2dcoor -outfile ringsystem2d.sdf ringsystem0.sdf
```

-directory <dirname>

This parameter sets the target directory. If the directory does not yet exist, it will be created.

Default value:

The directory of the output files is the same as of the corresponding input files, or the current directory, if the input file names do not contain directory information.

Example:

Generating a MDL SD-file saved in a given directory:

UNIX: `mn_2dcoor -outfile ringsystem2d.sdf -directory /tmp ./examples/ringsystem0.sdf`

Windows: `mn_2dcoor -outfile ringsystem2d.sdf -directory C:/Temp ./examples/ringsystem0.sdf`

-feedback 0/n

If the parameter **feedback** is set to a value larger than zero, a control message is printed after processing a block of n items (structures or reactions). The current record number and the object name are printed on the standard error channel. Only structures which are actually written out are counted.

Default value:

It is not active by default.

Example:

Generating a MDL SD-file printing dots for every one hundred records:

```
mn_2dcoor -outfile maybridge2d.sdf -feedback 100
./examples/maybridge.sdf
```

-stat

If this flag is set, statistical information about the number of successfully converted records and conversion failures is written to the standard error channel.

Default value:

This flag is deactivated by default.

Example:

Generating a MDL SD-file showing the statistical information of the conversion:

```
mn_2dcoor -outfile maybridge2d.sdf -stat
./examples/maybridge.sdf
```

Output:

Convert file ./examples/maybridge.sdf
Successfully read 62622 records, failed 0
Successfully wrote 62622 records, failed 0.

-align none/x/y/diagonal

This parameter changes the alignment of the 2D structure layout. By default, structure coordinates are generated in a layout where common ring systems are in their familiar orientations. In case of rectangular image sizes, a rotation of the structure so that the largest coordinate is aligned with the x or y axis can sometimes improve the visual appearance. Diagonal alignment is along a 30 degrees angle.

Default value:

Parameter without a default value

Example:

Generating a MDL SD-file where the largest coordinate extent is aligned with the y axis:

```
mn_2dcoor -outfile ccc_y.sdf -align y ./examples/ccc.sdf
```

Remarks:

This option can be used both for newly computed 2D plot coordinates or coordinates read from a file. Structures can also be aligned to a substructure template. This procedure is accessible through the **-template** set of options.

-template <SMILES/(SMARTS)>

This parameter specifies a SMILES string to be used as template for structure layout. The structures of the output file will all be aligned in the same way. In a limited way the substructure template can also be given in SMARTS notation.

Default value:

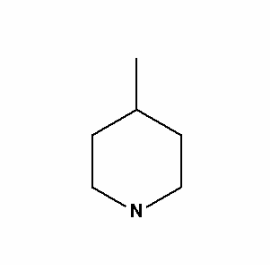
Parameter without a default value

Example:

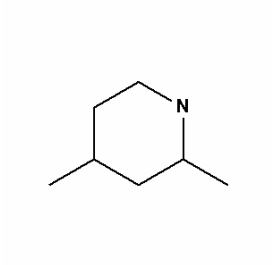
Generating a MDL SD-file using a substructure as template for aligning:

```
mn_2dcoor -outfile n_rings_aligned.sdf -template
[C]1[C][C][N][C][C]1
-templatealign redraw ./examples/n_rings.sdf
```

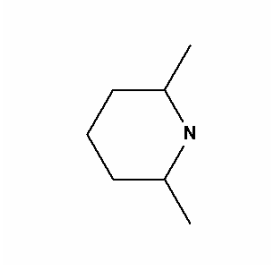
Pictures:



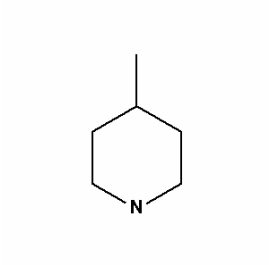
input structure no. 1



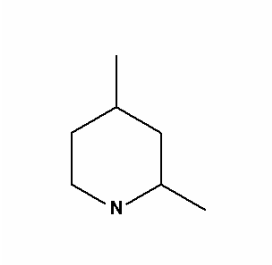
input structure no. 2



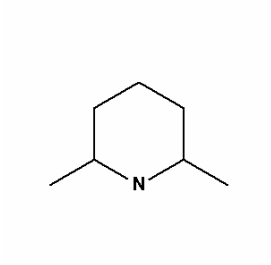
input structure no. 3



output structure no. 1



output structure no. 2



output structure no. 3

Remarks:

This option is used in combination with the **-templatealign** option.

-templatealign none/x/y/diagonal/rotate/redraw/besteffort/combined

Align the layout of the image according to a substructure template, which was specified by the **-template** or **-templatefile** options. If no substructure is present, this parameter is ignored. The substructure is matched on all structures. If it does not match, no error is generated and processing continues as if this parameter had not been specified or set to *none*. The *redraw* option implicitly sets additional substructure flags which will allow matching of substructure ring atoms and bonds only on corresponding structure atoms and bonds which are in the same class of ring system. With this option, a ring system must be matched completely. Thus, for example, a phenyl ring will not match a naphthalene ring if this option is chosen. The other match variants do not have this limitation. If the substructure does not possess 2D coordinates and the *rotate* or *redraw* parameter types are selected, coordinates will be computed by the standard 2D layout procedure. The first successful match of the substructure is used as template. The *x*, *y*, and *diagonal* parameters will align the major axis of the matched atoms of the structure to the x and y axis or on a 30 degrees angle to the x axis, respectively. For these options, no substructure 2D layout coordinates are used. The *rotate* variant will rotate the structure by multiples of 30 degrees with and without a coordinate flip. From among those 24 orientations, the one with the best overlay to the substructure coordinates is chosen. The *redraw* variant will regenerate the 2D layout coordinates, using the matched fragment with its coordinates transferred from the substructure as the nucleus for the layout. In this style, all

matched structure coordinates will have exactly the same relative coordinates as the substructure atoms, but the standard bond length will be scaled to one.
Finally, the mode *besteffort* (*combined* is identical to *besteffort*) combines the redraw and rotate modes - if a match in mode *redraw* fails, the match attempt is automatically repeated in mode *rotate*, which has relaxed match conditions with respect to ring system checks.

Default value:

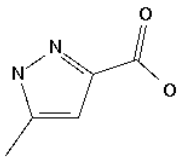
Parameter without a default value

Example:

Generating a MDL SD-file aligning the structure to a reference structure saved in a templatefile:

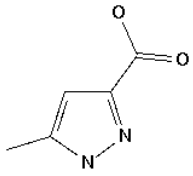
```
mn_2dcoor -outfile pyrazole_aligned.sdf -templatealign redraw
-templatefile ./examples/pyrazole_template.sdf
./examples/pyrazole.sdf
```

Pictures:



input file

structure of the templatefile



output file

Remarks:

This option is used in combination with the **-template** and **-templatefile** option.

-templatefile <filename>

This parameter specifies the name of a file which contains a substructure template. Only the first record of the file is read.

Default value:

Parameter without a default value

Example:

see previous option “templatealign”

Remarks:

This option is used in combination with the **-templatealign** option.

If the file does not contain 2D coordinates, and these are needed for the selected **-templatealign** option, coordinates will be generated.

-templatematch strict/relaxed

This flag influences the match operation of fragments on structures. In mode strict, aliphatic fragment atoms will not match aromatic structure parts. In relaxed mode, the default, aliphatic fragment atoms will also match aromatic systems.

Default value:

This parameter is set to relaxed.

Remarks:

This option is used in combination with other **-template** options.

This setting applies only to atoms without further query attributes. If any atom bears an explicit aromatic or aliphatic query attribute, this attribute has precedence. If a template is specified as SMARTS strings, uppercase atoms are decoded without an explicit aliphatic query attribute and will this match aromatic systems. This option can be used to counter this convention.

-version

If this flag is set, the version and licensing information is printed.

Default value:

This flag is deactivated by default.

Example:

Showing the program version:

```
mn_2dcoor -version
```

-h or -help

If this flag is set, a brief help message about the usage of the program is shown.

Default value:

This flag is deactivated by default.

Example:

Show the help message:

`mn_2dcoor –h` or `mn_2dcoor -help`

8. Extended Features Only Available for the UNIX Operating Systems

Input files can be processed in compressed or gzip-ed form without prior unpacking. The input file name arguments may each be a local file, an URL (http, ftp, gopher, file) or an email message file containing the structure data in the main body or as one or more attachments. URL retrieval and compression can be combined.

9. Frequently Asked Questions (FAQ)

10.Error Messages

MN.2DCOOR does not convert the single input structure although a valid smiles string is given

Running MN.2DCOOR under Windows the very last line of the input structure must be empty.

11.Known Problems and Limitations

12.Technical Support

The MN.2DCOOR Web Site

If you have problems while running MN.2DCOOR please have a look at the Support- and FAQ web site of MN.2DCOOR. The pages are available at http://www.mol-net.de

Reporting Problems

If your problem is not listed in these web pages please report it to the MN.2DCOOR team at Molecular Networks. Please make sure to provide us with all important data for replicating your problem on our machines. Therefore please use the report form on the next page.

Updates

If you have licensed the program MN.2DCOOR with maintenance you will automatically receive updates every time a new release is launched.

Contact Information

Distribution and Maintenance for MN.2DCOOR is handled by Molecular Networks Computerchemie, Erlangen, Germany.

Molecular Networks GmbH
Computerchemie
Nägelsbachstraße 25
91052 Erlangen
Germany

e-Mail: support@mol-net.de
Tel. +49 9131/815668
Fax +49 9131/815669

13.Report Form

In case of problems occurring during installation or running MN.2DCOOR, please complete the following form and send it or fax it to

Molecular Networks GmbH Computerchemie
Nägelsbachstraße 25
91052 Erlangen
Germany
FAX: +49-(0)9131-815669

_____

User:

_____

MN.2DCOOR program and version number (`mn_2dcoor -version`):

_____

Command line to run MN.2DCOOR:

_____

Error and warning messages by MN.2DCOOR:

_____

System messages:

_____

Short description:

_____

Please include the input file and output file generated by MN.2DCOOR on a 3½" diskette written in MS/DOS format or send an e-mail to support@mol-net.de attaching these files. These files will help us to analyze your problems. All data will be treated confidentially.

14.Index