NAME

csib - batch image generator for chemical structures

VERSION

1.34, 2021-10-12

SYNOPSYS

csib [-align *none/x/y/diagonal*] [-antialiasing 0/1/2/3] [-annotationfontsize points] [-aroresolver 0/1] [-asymbol *symbol/xsymbol/label/index/box/compact/residue*] [-atomcolor colorname/*type*] [-atomcolormode *direct/invert/circle/voronoi/ splitbonds*] [-background colorname/ *transparent*] [-bead 0/1] [-boldcolorbonds 0/1] [-bondcolor colorname/*split*] [-bonddistance n] [-bondscale n] [-border pixels] [-cleardirectory] [-colorbonds *speclist*] [-copyright text] [-coretemplates file_list] [-count nrecs] [-crop nborderpixels] [-csymbol *none*/*special*/*all*] [-dashes 0/1] [-directcolorfile *filename*] [-directory dirname] [-embed *none*/*smiles*] [-expand 0/1] [-feedback n] [-fixbridgeheadstereo 0/1] [-font fontfile] [-footer text] [-footercolor colorname] [-footerfontsize points] [-footerproperty property] [-forcecarbonsymbol 0/1] [-format *gif*/*png/png8/eps/mif/pict/svg/emf/wmf/pwmf/bmp24/bmp8/cbmp8/bmp4/ cbmp4/bmp1/pdf*] [-frame 0/1] [-fragmentcoloring *atoms*/ *bonds*/*both*] [-fragmentfile *filename*] [-fragmentmatchmode *overpaint/exclude*] [-fragmentpriority *record/size/ reverserecord/reversesize*] [-groupcolorfile *filename*] [-gui] [-hcolor colorname] [-header text] [-headercolor colorname] [-headerfontsize points] [-headerproperty property] [-height pixels] [-highlightcolor color] [-highlightfile filename] [-highlightmatch *relaxed/strict*] [-highlightsmiles smarts_string] [-hsymbol *none*/ *special*//*all*] [-httpheader 0/1/2] [-hydrogens *add/asis/strip/stripall*] [-inputformat fmt] [-interlace 0/1] [-isotopecolorfile *filename*] [-linewidth pixel] [-logofile filename] [-logoscale factor] [-metadata 0/1] [-name *record/name/stdout/*pattern] [-offset records] [-pseudo3d 0/1] [-querysymbol string] [-recalc 0/1] [-sdfile 0/1] [-showcharge 0/1] [-showchirality *none/simple/extended*] [showcrossedbonds 0/1] [-showisotope 0/1] [-showradical 0/1] [-showstereo 0/1] [-showstereogroups 0/1] [-showstereoh 0/1] [-symbolfontsize points] [-template SMILES/SMARTS] [-templatealign *none/x/y/ diagonal/rotate/ redraw/combined*] [-templatefile filename] [-templatematch *strict/relaxed*] [-version] [-wedges 0/1] [-wedgepairs *no/all/ hydrogen/unconnected*] [-wedgestyle *default/ opposite*] [-width pixels] ?files?

DESCRIPTION

csib is a batch-mode processor for chemical structure files. It reads files and produce pixel images (GIF, PNG, BMP) or vector drawings (EPS, WMF, EMF, PICT, SVG) of the structures found. Multiple files can be specified on the command line. If no files are given, or the file name “-” is used, input is read from the standard input channel.

By default, the images of every file are put into a separate subdirectory. The name of the subdirectory is the file name without any suffix, in the current working directory, not where the file resides. If the file does not have a suffix, the suffix _img is added to the directory name. The target directory can be overridden by the *-directory* option. In this case, all images are put into that directory.

The names of the images are by default constructed from the root name of the respective input file (without directory part, and without suffix), followed by the record count, and the standard image file format suffix (.gif, .png, .wmf, etc.). Thus, an input file */usr/local/test/test.sdf* by default produces images named *test_00001.gif*, *test_00002.gif,* etc. in subdirectory *test* in the current working directory. See the **-naming** option for an alternative naming scheme. If data is read from standard input, the default image directory is *stdin_img*, and the files are named *stdin_00001.gif* (or .png, .wmf), and so on.

The program automatically detects the file format of the input files. About 40 standard exchange formats (including SD-files and SMILES with extensions) are supported. A set of input files does not need to be encoded in a single common format. Additional file format readers can be added as dynamically loadable modules.

Structures are automatically scaled to fit completely into images of arbitrary size. Structures which possess up to a configurable number of standard-length bonds in x-direction are scaled by a constant factor, drawn with a constant bond length, and centered in the image. Larger structures are dynamically shrunk. This algorithm generates nice-looking plots of average molecules with plot sizes directly proportional to the structure size, while avoiding plots with incomplete drawings not completely shown on the image. The length of a standard bond is intelligently determined from a plot coordinate analysis. The fonts for the displays of atom symbols are automatically selected according to the scaling factor. If the scaling factor becomes very small, the atom symbols are replaced by small, appropriately colored boxes.

The color of atom symbols depends, if not explicitly specified in the input files, or set to a fixed color by options such as **-atomcolor** and **-hcolor**, on the background color of the image. A brighter color table is applied on dark backgrounds, while a more subdued coloring scheme is selected on white, light gray or transparent backgrounds.

For pixel images, the program provides two different methods to select fonts to render all texts (atom symbols, charges, header and footer lines, etc.). First, it has a small collection of rather blocky built-in fonts. These are used by default. Alternatively, any TrueType font can be used. The default Linux TrueType fonts *Arial*, *Arialn* (narrow), *Arialb* (bold) and *Arialnb* (narrow bold) are included in the standard distribution of the full CACTVS toolkit. More fonts can be added by copying them into the *fonts* distribution directory. For vector images (EPS, EMF, WMF) with embedded font information the selected fonts need not to be actually present on the image generation computer, but should then of course be found on the display computer. However, better clipping data can be computed if the fonts are also present on the computer where the images are generated.

OPTIONS

Color names can be specified either as an X11 color identifier, or as an RGB specification (for example, #00FF00 or #0F0 for 100% green).

-align *none/x/y/diagonal*

Change the alignment of the 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 extent is aligned with the x or y axis can sometimes improve the visual appearance. Diagonal alignment is along a 30 degrees angle. Structures can also be aligned with a substructure template. This procedure is accessible through the **-template** set of options.

-antialiasing 0/1/2/3

This flag controls whether textual content (atom symbols and annotations, header and footer lines) are rendered with anti-aliased fonts when writing the pixel-based output formats (everything except WMF/EMF/EPS/PICT/SVG), and/or lines are drawn in a similarly anti-aliased fashion. Bit 0 (values 0/1) controls the setting for font anti-aliasing, and bit 1 (values 0/2) the setting for lines. The option value can be set to any sum of these values. By default, this option is fully enabled (value 3). It does not have any effect if the built-in fonts instead of external TrueType fonts are used.

-annotationfontsize points

By default (or if you set this number to less than 0) the program chooses suitable font sizes for annotations (charges, etc.) automatically. It is possible to override this choice with this option. Note that the number of available font sizes is limited if you use the built-in fonts instead of external TrueType fonts. The program chooses the best built-in font (from the list 6pt, 7pt, 8pt, 10pt) in this case. The desired point size may be a floating point number.

-aroresolver *0/1*

If this flag is set, aromatic query bonds in MDL molfile variants are interpreted as real aromatic bonds, and a Kekulé structure is generated for the display. If this flag is not set, these query bonds (which by MDL's definition are not an *aromatic bond* type and must not be used outside ISIS query definitions) are displayed as single bonds and internally treated in a completely different fashion than real bonds (no electron counting, etc.)

-asymbol *symbol/xsymbol/label/index/box/compact/residue*

Select the type of symbol to print. For normal atoms, the display types *symbol* and *xsymbol* are equivalent, but *xsymbol* produces a more detailed text for certain types of query atoms, such as atom lists. With the *label* style, atom labels replace the atomic symbols. These are either taken from the input file (if the files stores this information) or correspond otherwise to the internal atom ordering, which is always the same as in the file. *Index* displays the index of the atom in the atom list, starting with one. In the box style, hetero atoms are depicted as a small rectangles. The *compact* mode displays hydrogen atoms on hetero atoms and other special hydrogens as a common symbol, such as OH or NH_2. Subscripting of hydrogen counts only works with Unicode fonts and in the EPS format (see also the paragraph on the **-font** option). The *residue* mode substitutes PDB residue labels as atom symbols, if these are present. If no residue data is found, the default *xsymbol* mode is invoked.

-atomcolor colorname/*type*

Select a global color for all atom symbols. If the default color *type* is chosen, the color of atom symbols is determined individually. If the input data contains color information, it is used. Otherwise, a standard element-specific coloring scheme is applied if no external file coloring option (**-directcolorfile**, **-groupcolorfile**, **-isotopecolorfile**) is selected. If external coloring information is used, the default atom color is uniform black. For hydrogen atoms, the hydrogen color specified with the **-hcolor** option overrides both a global atom color and an individual atom color, if it is not an empty string.

-atomcolormode *direct/invert/circle/voronoi/splitbonds*

The parameter determines how atoms for which a specific color was selected via the **-directcolorfile**, **-groupcolorfile** or **-isotopecolorfile** options are drawn. The default is *direct*. In this mode, the atom color is simply set to the color specification, and its symbol is later drawn with this color. Hydrogen and carbon atoms for which symbols are not displayed by default are forced to plot with symbols if they are colored. Mode *invert* causes the atom symbols to be drawn in reverse, i.e. as a background color atom symbol on top of a box colored in the selected color. This style also works in cases where no symbol is drawn, such as in case of carbon atoms which are only displayed as bond nodes. The *voronoi* mode is similar to *invert*, but instead of a simple box a Voronoi polygon centered around the 2D display coordinates of the selected atom is filled with color and plotted underneath the atom symbol and bond lines. The Voronoi polygons of adjacent atoms are smoothly joined and do not overlap, allowing a seamless coloring of multiple atom groups. Mode *circle* plots colored circles with a radius slightly larger than half a standard bond length underneath the selected atoms. The *splitbonds* mode does not color the selected atoms but rather the bonds the atom is participating in. The bond halves closer to the selected atom is drawn in the selected color. In case the other atom of a bond treated this way is not colored, or colored differently, the bond is drawn with two different colors for each half. Wedge bonds and other bond types which are not drawn with a simple lines are also split into multiple polygons etc. if necessary.

-**background** colorname/*transparent*

Determines the background color of the image. If the special value *transparent* is chosen, display programs show whatever is behind the image in those regions which are not covered by the plot or logo.

-**bead** 0/1

This option is active by default. If set, certain generic atom types (R, polymer, search classes) are depicted as two-color circles with the shape of beads. This feature is intended to be used in conjunction with polymer-support chemistry.

-**boldcolorbonds** 0/1

If this flag is set, bonds which are colored by the action of one of the coloring directives -**groupcolorfile**, -**directcolorfile**, or -**isotopecolorfile** are also plotted with bold lines. In case a bond is split-colored, the boldness attribute is always applied to both halves.

-**bondcolor** colorname/*split*

Select a global color for bond lines. In contrast to atom colors, individual bond colors are currently not supported. The special value *split* splits the bond into two halves. Each half bond is colored in the same color as its associated atom symbol, with the exception of carbon atoms. In case of half bonds to carbon atoms, the split color is either black or white, automatically selected according to the image background. The default bond color is black.

-**bonddistance** n

Set a scaling factor for the sideways shift of the lines of multiple bonds. This is a floating-point parameter. The default value is 2.0. Useful values are in the range between 1 and 3.

-**bondscale** n

This parameter determines how many standard-length bonds should fit on the image in x-direction. Structures which possess less bonds are centered on the image. Structures which have more bonds are shrunk until they fit tightly into the image. A larger value lets the average structure appear smaller, but more structures fit in the image without rescaling, maintaining their correct size relationships. Smaller values make plots larger, but rescaling is necessary for more compounds. The default value is 8.

-**border** pixels

The parameter determines the width of the border from the center of the outmost atoms of a structure which fits tightly into the display area to the outer border of the image. Note that atoms with plotted symbols require a few pixels in all directions around the atom center, so setting this parameter to very small values should be avoided. The default is 12 pixels.

-cleardirectory

If this option is set, the target directory (either specified by the **-directory** option, or automatically chosen) is completely cleared before the image generation commences.

-colorbonds specification_list

This option can be used for direct coloring of bonds via their bond labels (starting with 1). The argument is a nested Tcl-style list. Every sub-list consists of a valid color name or other color specification, followed by an arbitrary number of bond label elements. In case a bond label is listed multiple times, its last occurrence determines the coloring. Bonds not listed in this argument are colored using the default scheme, or may be colored by the effects of other options.

When reading a file, bonds are created in the order they are encountered. If a file record contains explicit bond labels, they are extracted and assigned as bond labels, even if that means the first bond label is not one, or the label list is not numerically ascending or has holes. In the absence of explicit labels, the first bond of a structure is assigned label one, and the other bonds form a numerically ascending sequence. When reading files without bond information, or complicated mixtures of implicit and explicit bonds, no assumption should be made about the values of the bond labels. In case postprocessing of a structure results in additional bonds being created, their labels start with the highest explicit or automatically assigned label from the file plus one. In case bonds are deleted, their label is unused, but the original labels of the other bonds are kept.

Example: -colorbonds '{red 1} {"light blue" 2 3} {#FF0080 4}'

-copyright text

This is a hidden copyright message, which is be embedded in the images. It is stored in a comment field and can be extracted by viewers which are capable of displaying this kind of additional meta-information. It is also possible to use the standard Unix command *strings* to extract this information.

-coretemplates file_list

A list of files with structure fragments to augment the built-in set of level 2 2D ring system templates. These are not the same as the templates used for aligning sequences of compounds in a common fashion (**-templatealign** option) which are used at a higher level of processing. The core templates are used directly in the low-level layout of complex ring systems. Multiple files can be listed with this parameter, and files can be multi-record. All recognized file formats which contain basic structure data and 2D coordinates are acceptable. A maximum of 100 user-defined core templates in all files is currently supported. Additional files or records are ignored.

The core templates are simple structure fragments with specified 2D coordinates. The coordinates are automatically scaled and do not need to adhere to specific value ranges and scaling. Elements are ignored in matching the templates, so typically only an all-carbon structure framework is supplied. Single bonds in that pattern match any bond in the processed structures, including multiple and aromatic bonds. Other bond orders need to match exactly. This is useful to ensure, for example, that a specific double bond in a macrocycle is always placed in the same position. Level 2 templates must consist of a single fragment and must contain only ring atoms. They can only match complete ring systems of the structures being processed. This is more restrictive than for high-level alignment templates. Level 2 templates override the more elementary built-in level 1 templates but have lower precedence than user-specified alignment templates. In case the processed structures contain multiple ring systems, more than one template may be applied to different sections of the molecule, and even if a high-level alignment template matches, other parts of the processed structures may still be drawn using these templates.

-count nrecs

This option sets a maximum to the number of records which should be processed from each file. By default no limit is set. In combination with the **-offset** option, this parameter can be used to selectively render a part of a file.

-crop nborderpixels

By default, the image size is precisely defined by the **-height** and **-width** parameters (default: 200x200 pixels). The **-crop** option can be used to remove unused border space on pixel-based output format. This option is ignored for vector formats (EPS, EMF, WMF, SVG, PICT). In a first step, the number of continuous pixel columns and rows which contain only background pixels are determined, counting from the outsides, and these rows and columns are then removed. Next, a background-colored frame of the specified **-crop** width is added. It is possible to expand the image beyond its initial size with this option, but normally images become smaller and loose their aspect ratio. If this parameter has a negative value (the default), no cropping takes place.

-csymbol *none/special/all*

This parameter sets the carbon atom display style. Either no symbols are plotted (but those atoms are still nodes for bonds), only symbols for carbon atoms in certain special environments (such as in alkynes or cyanides, or when charged, etc.) are displayed, or all carbon atoms are labelled with a *C*. The default is *special*. This parameter is ignored if the global **-asymbol** style is *label, index or box*.

-dashes 0/1

This option controls whether dashed bonds are actually drawn in this style, or always in solid style. The default is 1, meaning that dash attributes are rendered. This option interacts with the **-wedges** option. Resetting this option alone, without also resetting **-showstereo**, is not useful.

-**directcolorfile** filename

This option is used to color atoms and/or bonds on a per-record basis. The color file is a simple ASCII file, which contains one or two integers, plus a color name per line. These entries are separated by whitespace. The color part may be either an X11 well-known name from the X11 color database, such as *red*, or any shade in *#rrggbb* encoding, with 1 to 4 bytes of color data per channel. Empty lines and lines beginning with a pound character are ignored in the color table file. Color names containing spaces are allowed - everything after the first one or two integers is interpreted as the color part. Lines which contain only a single integer before the color name are interpreted as atom coloring instructions. The integer is the atom index (beginning with zero) in the atom list. If there are two integers, it is decoded as a bond connecting two atoms with the given indices. Generally, atoms remain in the sequence they were originally stored in when a file is read. Added hydrogens are appended to the rear of the atom list. However, in case atoms are deleted, sequence gaps are filled, and in this case the sequence may become different from the original file, although atoms are never shuffled. Atoms and bonds identified by the current coloring record are colored with the specified color. If bonds are selected, their color is set directly. In case of atoms, the atoms, or bonds on these atoms, are colored according to the setting of the **-atomcolormode** parameter. After processing an input structure, the current color record is discarded and the next one read. Color records are separated by lines which start with a dash, or a vertical bar. The color file may span multiple input files if it contains sufficient records. If the color file is exhausted before the structure errors are, the remaining structures are not colored. In case an illegal atom index is used, or a bond between atoms cannot be found, or a color cannot be decoded, a warning is printed on the standard error channel, but processing continues.

-directory dirname

The default image directory selection process described in the introductory section is superseded if a named target directory is specified with this option. The directory must either already exist and be writable, or is must be possible to create it with the right permissions. The target directory can optionally be automatically cleared with the **-cleardirectory** option. Note that the directory specified with this option is not used if the **-name** option directly sets a directory component of the output file name. However, the **-cleardirectory** option only works on the directory specified with this option. Relative directories, such as *../output*, or references to the local directory (.) are allowed.

-embed *none/smiles*

The program support the embedding of structure information into the images. If structure information is stored, it is possible to reclaim the molecule information without going through a lengthy and error-prone chemical OCR procedure. The *smiles* style stores the structure data as SMILES string. This string can be extracted either by suitable software with the CACTVS toolkit, or simply by standard Unix tools such as *grep* and *strings*. By default, no structure code is embedded.

-expand 0/1

If this option is activated, an attempt is made to expand identified superatoms in the input data, such as “COOH” or “CO_2Et”. Currently, a built-in table with about 150 fragments is used, which cannot be extended by user-defined fragments. Expanded fragments are laid out in the opposite direction of the bond connecting them to the framework. This only works reasonably well in case of fragments that are connected only by a single bond to the rest of the structure. Recomputation of layout coordinates is recommended in these cases. Superatoms which cannot be resolved are passed through unchanged and output as such in files which allow this. In case of output formats without superatom support, un-expanded superatoms are omitted. Some file formats, such as the ISIS skc format, may internally store selected (but generally not all, even within a single drawing there may be closed and pre-expanded fragments side by side) fragments already in expanded form, even if they do not show up expanded in native applications such as ISIS/Draw. If such a pre-expanded fragment is encountered, its expanded form is automatically used in this application without need to use this option. However, even in such cases, this option can be helpful to expand those fragments which are not pre-expanded.

-feedback n

If this option is set, the progress of the program is reported on standard error. The name of the currently processed file is printed, and a for every *n* completed images the current record number and the compound name (its formula if no name is found) is output. If the parameter is omitted, or set to zero, no feedback is produced.

-fixbridgeheadstereo 0/1

If this flag is set, the program tries to detect problematic stereo center displays at bridgehead atoms. In case of a bicyclic ring system, the bridgehead atom to the front always has a solid up wedge, and the rear atom a dashed wedge. If necessary, the structure is flipped vertically - which may result in a drawing which appears to be drawn with a flipped template in case templates are used. This flag has an effect only if new 2D coordinates are computed. By default it is not set. In any case, the local stereochemistry at all atoms is correct - but if these atoms are part of a complex ring system with bridges, the overall geometry may be questionable.

-font fontfile

By default, this option is set to the empty string, which indicates that the default font should be used. For pixel-based rendering it is an in-memory version of the Arial Narrow (arialn) font.

As a replacement of the built-in default fonts, any TrueType font can be used instead. These fonts can be scaled to an arbitrary point size, and they can be rendered with anti-aliasing (see option **-antialiasing**). The font file argument can either be a full path to a *.ttf* file, or the simple body of the file name (with or without the suffix). If the font file is not found as a fully qualified file, a search path is traversed, which includes the *fonts/* subdirectory in the standard installation. This directory is therefore the preferred place to store font files. For example, an *Arial Narrow* font in that directory could both be specified as */usr/local/lib/cactvs/fonts/arialn.ttf* or simply *arialn*.

If the *compact* atom symbol display style is used, with contracted functional groups which potentially employ subscript characters in their symbol, an extended Unicode font which provides these characters must be used. The collection of *DejaVu* fonts included with the software contain all necessary characters. If a standard Arial font or other font which only contains ISO Latin1 characters is used, empty boxes are rendered as placeholders for missing characters.

The above explanation only applies to pixel-based formats. WMF/EMF, EPS and other vector formats always use an Arial/Helvetica font. The font rendering when showing such an image is performed by the display application, not this software.

-footer text

This is a free text which is centered on the bottom of every image. Compound data can be automatically inserted into the footer with the **-footerproperty** option.

-footercolor colorname

Selects the color for the text written with the **-footer** option. The default color is black.

-footerfontsize points

The font size used for the footer line. If this parameter is set to a negative number, the default size, which is the same as that of the atom symbols, is used.

-footerproperty propertyname

If no explicit footer is set with the **-footer** option, this option can be used to transfer the data associated with a CACTVS property into the footer field. Useful properties are for example E_NAME, E_FORMULA, E_WEIGHT, or E_SMILES. If the data is not yet present, but a method is available to compute the data from available information, it is automatically invoked. Currently, the footer property must be of the ensemble property attachment type, and subfield extraction is not yet supported.

-format *gif/png/png8/emf/wmf/pwmf/pict/svg/eps/mif/bmp24/bmp8/cbmp8/bmp4/cbmp4/*

bmp1/swf/pdf

Select an output format. Both GIF and PNG can store additional, invisible data, such as a copyright message (**-copyright**), encoded structure information (**-embed**) and image metadata (**-metadata**). However, care must be taken when post-processing these images: many image manipulation programs do not support the handling of auxiliary data, which may therefore get lost when writing back these images from those programs.

The difference between the *png* and *png8* formats is that the latter is forced to be an 8-bit colormap image. By default, images which use line antialiasing (see **-antialiasing** option are internally true-color. In the case of GIF output, color reduction to an 8-bit image is mandatory, because GIF images are always colormap-based. For PNG, this step is optional, and by default the full color depth is preserved. For normal structure drawings, anti-aliased lines plus 8-10 different anti-aliased element base colors plus a couple of additional pure colors (background etc.) can be stored in an 8-bit colormap without loss of color precision. Colormap-based PNG images are typically only half the size of the true-color version, so for most applications *png8* is the preferred format. This program automatically inserts invisible dummy data into PNG images when needed in order to circumvent datasize-related rendering bugs in IE.

The *pwmf* format is a *placeable* metafile. This format is generally preferable to raw metafile data if the drawing is imported as files into graphics applications or word processor documents. MS Internet Explorer can display *pwmf* images embedded in HTML Web pages, but not plain *wmf*. MS Word and other MS Office applications can import both.

The *pict* format is a Mac vector-based image format, similar to *wmf*/*emf* on Windows.

The *svg* format are standards-conforming XML-based Scalable Vector Graphics. Some Web browsers (such as Firefox or Opera, but not IE6/7) can render this data directly, which results in nicely scalable and printable structure and reaction images on Web pages. Unfortunately, this cannot be used with all browsers.

The *mif* format is the native graphical exchange format of the Adobe FrameMaker word processing program. While FrameMaker can also import *wmf* and pixel images, this is the preferred format because it allows in-document editing, such as highlighting of bonds etc. which is not possible with other formats without the use of external and properly configured editor applications.

eps is the embedded PostScript format, suitable both for direct printing on PostScript-capable printers and import into certain graphics applications. *pdf* is the well-known Adobe print format, but it is also suited for Web applications where it is displayed in an embedded Acrobat reader window and print nicely.

The various Windows bitmap (*bmp*) formats differ in their color depth, which is directly related to the maximum number of colors present in the image (1 bit = 2 colors, 4 bit = 16 colors, 8 bit = 256 colors). A smaller color depth saves significant storage space, but if the color depth is too small, color information may be lost. Usually, 8 or 4 bits are a good choice. For black and white drawings, the storage-efficient 1-bit bitmaps are sufficient. Unfortunately, many simple graphics applications do not support all bitmap variants. Generally, the 24 bit bitmap is most portable, but these files are large. Compressed 8 and 4-bit images (*cbmp4*, *cbmp8*) are significantly smaller than the un-compressed variants, but once again may face portability problems.

-forcecarbonsymbol 0/1

In some cases, such as when underlying an atom symbol with a colored marker circle, this forces carbon atoms without other special attributes and which are only drawn as bond nodes but without an explicit atom symbol, to be rendered with a symbol. This behavior can be disabled when setting the option value to zero. The default is one, i.e. the non-standard symbol rendering will take place.

-frame 0/1

If this option is set, a thin black line is drawn around the perimeter of the image. This is useful for example to defeat auto-cropping mechanisms in later image-processing steps.

-fragmentcoloring *atoms/bonds/both*

Specify whether atoms and/or bonds which are matched by fragment substructures specified with the **-fragmentfile** option are colored with the fragment color. The default is *both*.

-fragmentfile filename

Specify a file with SMARTS or Recursive SMARTS definitions for highlighting parts of the displayed structure with custom colors. The file is expected to contain on each line a SMARTS expression and, separated by whitespace, a color definition. The color definition can either be a name from the X11 color database, or a hex RGB notation such as #FF0000. In case the color name contains spaces, it must be quoted. A maximum of 10000 fragments are read from this file. For each processed structure or reaction an attempt is made to match all fragments in a specific order (see **-fragmentpriority** flag). If a fragment matches, the matched atoms and/or bonds (see **-fragmentcoloring** option) are colored. In case of recursive SMARTS definitions only the parts matching in the base recursion level are painted. Thus, it is for example possible to color ring systems with specific substituent patterns without also highlighting the substituents.

-fragmentmatchmode *overpaint/exclude*

This option selects the match procedure for substructure fragments specified with **-fragmentfile**. The default is *overpaint*. In this mode, a later matching fragment paints over any atoms and/or bonds already matched by another fragment. This can lead to a partial alteration of the coloring of an earlier match if the later match does partially, but not completely overlap the earlier match. In the *exclude* mode, any atoms in the processed structures which have been matched by a fragment are blocked from matching any later fragment. In this mode, overlaps cannot happen. The priority of the fragments can be adjusted with the **-fragmentpriority** option.

-fragmentpriority *record/reverserecord/size/reversesize*

This option determines the priority of the substructure fragments. The default is *record*, meaning that the order in the input file directly corresponds to the order the fragments are applied, with the first record having the highest priority. If the **-fragmentmatchmode** option is *overpaint*, the first fragment is matched last, and, if it matches, it paints over any previous matched atoms and/or bonds in the processed structure (see **-fragmentcoloring**). In fragment match mode *exclude*, the fragment is matched first and, if it matches, blocks all matched structure atoms from further matching by any other fragment. If the *size* priority is selected, the substructures are sorted by total atom count (in the base level, for recursive SMARTS patterns) and the largest structure has the highest priority. The *reversexxx* modes invert the sort order.

-groupcolorfile filename

This option allows the coloring of atoms and bonds present in named groups. Named groups may for example be read from native CACTVS files, MDL Molfiles or SDfiles (V2000 “M SMT” keywords, V3000 named collections). The group color file is a simple ASCII file. Each line contains a group name, and, separated by whitespace, an X11 color name (either a well-known name from the X11 color database, such as *red*, or any shade in *#rrggbb* encoding, with 1 to 4 bytes of color data per channel). Empty lines and lines beginning with a pound character are ignored in the color table file. Color names containing spaces are allowed - everything after the group name is interpreted as the color part. The groups encountered in every processed structure are checked against the group color table. If a group name matches, the group elements are colored. If the group contains explicit bonds, the bond color is set directly. If it contains atoms, the atoms in the group, or the bonds between them, are colored according to the setting of the **-atomcolormode** parameter.

-gui

This option starts the program with a graphical user interface. It is the same as with the *csig* program. Both programs are actually the same code. The **-gui** option disables the **-feedback** option, since it provides its own feedback mechanisms. This option is not supported in the stand-alone version of the program, which is distributed without the standard CACTVS toolkit environment.

-hcolor colorname

Specify an override color for hydrogen atoms. If the color name is not an empty string, it overrides both the global atom color, or an individual hydrogen color taken from file or the standard element color table. The default hydrogen color is black.

-header text

This is a free text which is centered on the top of every image. Compound data can be automatically inserted into the header with the **-headerproperty** option.

-headercolor colorname

Selects the color for the text written with the **-header** option. The default color is black.

-headerfontsize points

The font size used for the header line. If this parameter is set to a negative number, the default size, which is the same as that of the atom symbols, is used.

-headerproperty propertyname

If no explicit header is set with the **-header** option, this option can be used to transfer the data associated with a CACTVS property into the header field. Useful properties are for example E_NAME, E_FORMULA, E_WEIGHT, or E_SMILES. If the data is not yet present, but a method is available to compute the data from available information, it is automatically invoked. Currently, the header property must be of the ensemble property attachment type, and subfield extraction is not yet supported.

-height pixels

The height of the image in pixels. If the **-crop** option is used, this parameter determines only the height of the initial drawing area, not necessarily the height of the final image. The default are 200 pixels.

-**highlightcolor** color

Specify a color used for highlighted atoms and bonds. The default is red. The information about the atoms and bonds selected for highlighting can either be present in the read data, or may be added by matched substructures (options -**highlightsmiles**,

-highlightfile).

-**highlightfile** filename

Specify a file with substructure fragments. All records from this file are read and matched against the table structures. Implicit hydrogen addition is disabled while reading the file, but no explicit hydrogen stripping is performed, so substructures with extraneous hydrogen atoms may not match. If the table contains image columns, the matched atoms and bonds are highlighted. Only the first match of the substructure is shown, but if multiple substructures match the same table compound, instances of all matching substructures are highlighted.

-**highlightmatch** *strict/relaxed*

This flag influences the match operation of highlight substructures on structures. In mode *strict*, aliphatic substructure atoms do not match aromatic structure parts. In *relaxed* mode, which is the default, aliphatic substructure atoms also match aromatic systems. This setting applies only to atoms without additional explicit 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 thus match aromatic systems. This option can be used to counter this convention.

-**highlightsmiles** smarts_string

This is an alternative to the -**highlightfile** option. Here, a SMARTS substructure definition is decoded and used as substructure for highlighting purposes. Only a single SMARTS structure may be used.

-hsymbol *none/special/all*

Defines how hydrogen atoms are plotted. Mode *none* suppresses them all. In contrast to carbon atoms, the bonds to suppressed hydrogen atoms also vanish. Mode *all* plots them all, and mode *special* displays only hydrogen atoms in a few selected environments, such as on aldehydes, or when bonded to hetero atoms. The default is *special*.

-**httpheader** 0/1/2

This flag controls whether an HTTP header is inserted into the output before the image data proper. The default value is zero, meaning that no such header is inserted. Header mode one adds a standard MIME header with data type and size information. This is suitable for return from a standard CGI application. Mode two additionally inserts a 200 HTTP status code and is useful for NPH CGI scripts. This flag is typically used in combination with direct image data output to the standard output channel (see -**name** option).

-**hydrogens** *add/asis/strip/stripall*

Explicitly specify the handling of hydrogens. Since many file formats do not really inform you whether the hydrogen set in the file is complete or not, the default setting is *add*, which results in a chemically fully defined compound or reaction which are best suited for all internal processing. Nevertheless, in case hydrogens need to be completely ignored, or the user is certain that the implicit or explicit hydrogen set is complete, the modes *asis* (no change of hydrogens), *strip* (remove hydrogens usually not shown) or *stripall* (remove all hydrogens) can be used. Mode *asis* enables for example the proper display of radicals from SMILES strings, which would suffer hydrogen addition with loss of radical information in the default mode.

-ident property

Identify a property (native CACTVS property or input file data field) which serves as the record identifier. For example, if an SD input file, which contains a data field *NSC* is read, setting **-ident** to *NSC* allows, via a "**-name** nsc%i" invocation parameter, to construct an image file name which contains the NSC number as convenient identifier part of the file name

-inputformat fmt

This option may be used to specify the input format explicitly. The recognized format names are the same as for the output format (**-format**). In most circumstances, this option should be omitted, because the input format is reliably auto-detected from file contents. The suffix of the name of the input file is also used as additional information to break ties. The danger of format name mismatches is avoided if the default automatic mechanism is used. However, in certain circumstances, it is possible that the input format is ambiguous. The most common example is when single SMILES strings are read via standard input. In this case, there is no file name, and certain SMILES strings are also syntactically correct SLN strings, albeit sometimes representing a different structure. In such cases, the input format ("smiles" or "sln") could be explicitly set to enforce one interpretation.

-interlace 0/1

Both the PNG and GIF image formats support interlacing. An interlaced image can already be displayed by a Web browser application in lower resolution before the full resolution data has arrived. This is a useful feature in WWW environments and similar application scenarios. The disadvantage of interlacing is that it makes the images files larger. GIF images only support interlacing between lines, while PNG interlaces both between lines and columns. Windows bitmaps do not support interlacing. This option is ignored for vector-based formats (WMF,EMF,EPS,PICT,SVG).

-isotopecolorfile filename

This option is used to color atoms which are marked with isotopic labels. The isotope color table file is a simple ASCII file, with an isotope mass (written as an integer) and, separated by whitespace, an X11 color name (either a well-known name from the X11 color database, such as *red*, or any shade in *#rrggbb* encoding, with 1 to 4 bytes of color data per channel). Empty lines and lines beginning with a pound character are ignored in the color table file. Color names containing spaces are allowed - everything after the isotope mass is interpreted as the color part. Atoms in the processed structure which possess an explicit isotope label matching a color definition in the isotope color file are colored according to the setting of the **-atomcolormode** parameter. If this function is used, all isotope label information, even if it was not matched by a color entry, is removed from the structure data after the coloring step. The images therefore do not show any isotope labelling information on atom symbols.

-linewidth pixel

Set the base linewidth of bond lines. The default value is 1.4. Fat bonds are drawn using a format-dependent multiplier of this value. On pixel-based output format, the value is rounded to an integer thickness when the bonds are drawn, but vector formats can benefit from factional adjustments to this parameter.

-logofile filename

If this is not an empty parameter, an attempt is made to read the file as GIF or PNG logo. If the image could be read, it is inserted into the upper left corner of all produced images. If cropping is active, the logo is added *after* cropping, so its placement does not influence the cropping process, but it could then potentially overlap atoms and bonds. Without cropping, the logo is inserted first, and other items plot over it. The logo image can be scaled by the **-logoscale** option. Logo embedding is currently not supported for the EPS, SVG, and EMF formats.

-logoscale factor

This factor can be used to resize a logo file specified by the **-logofile** option. The scale factor is a floating point number. Note that this option should not be used on a regular base - a cleanly rendered logo image in the intended final resolution generally has a better graphical quality than a rescaled logo. The default scaling factor is 1.0.

-metadata *0/1*

If this option is set, the images contain, in an invisible embedded data field, a Dublin-Core-compatible metadata set. This option is effective only for GIF, PNG and SVG output.

-name *record/name/stdout/*pattern

Three default naming schemes for the image output files are predefined. The default style *record* generates file names which are assembled from the root name of the original file, a record count, and the standard image format suffix (.gif, .png, .wmf, etc.). Alternatively, the *name* style constructs file names which consist of the name of the compound as read from a standard file field, plus the image format suffix. If the input file does not contain a name, the structure formula is substituted. Within file names, white space is replaced by underscores.

The *stdout* mode omits the generation of output file names completely and instead redirects all image data output to the standard output channel. This can for example be useful in CGI applications, and is frequently combined with the -**httpheader** option. Obviously, this variant is normally only useful when a single record is rendered. While this program can be used as a CGI application in this fashion, this is not an optimal solution because of the significant overhead to start the program anew for every image. Xemistry can provide an optimized FCGI Web image rendering solution which was specifically developed for this usage scenario.

Any -**name** parameter value other than *record, name* or *stdout* is interpreted as a template, wherein %r is replaced by the current record number, starting with 1 for the first file record, %n by the standard structure name read from the file, %i by the value of the identification tag property (option -**ident**), %f by the molecular formula, %F by the name of the input file without any directory or suffix component, %D by the output directory (parameter -**directory**), and %S by the standard suffix of the selected image type. If the generated name does not contain a suffix, the default image type suffix is automatically appended. Likewise, if no explicit directory component is provided, the standard output directory defined via the -**directory** option is prepended. If the directory component of the output file name is directly set with the -**name** option, it overrides the -**directory** folder. However, none of the standard preprocessing of that directory, such as an attempt at creation if it does not yet exist, or clearing it when -**cleardirectory** is set, applies to a -**name** directory. If a directory specified via -**name** does not exist, the application fails.

-offset records

This parameter controls the entry point into each file. By default it is set to zero, meaning that the rendering starts at the first record. In combination with the **-count** parameter, this option can be used to selectively render a part of a file.

-pseudo3d *0/1*

This parameter (default 0) changes the way new 2D display coordinates are computed. The primary carbon backbone is plotted in a pseudo-3D style. This option has only an effect if new coordinates are generated. Coordinates read from file are not influenced.

-querysymbol string

Set a string to display as atom symbol if the atom is not a normal element or a type with a specific predefined representation. The default symbol for these atoms is a single question mark. The maximum utilized length of this string is eight characters.

-recalc *0/1*

If this parameter is set to 1, a possibly existing 2D coordinate set from the input file is ignored, and new coordinates are computed from the molecular connectivity alone. Computation of coordinates happens automatically if the input structures do not contain 2D coordinates. The computation procedure can be additionally controlled by the **-pseudo3d** and **-align** options.

-sdfile *0/1*

If this flag is set, in addition to the images an 2D SD-file (one multi-record file for every input file) is written. The name of the SD file is the same as the name of the image directory, but with the suffix *_2D.sdf* appended.

-showcharge *0/1*

If this flag is set to 0, atomic charge symbols are not plotted. The default is 1.

-showchirality *none/simple/extended*

If this option is set to *simple*, the images of compounds with defined chirality are tagged with a small *chiral* marker in the upper left corner of structure images. In mode *extended*, the chirality status is described in more detail as *unspecified* (compound is potentially chiral, but chirality is not given), *meso* (all stereo centers are paired by an opposite topologically equivalent stereo center), *chiral* (at least one defined stereo center), or *contradictory* (multiple stereo descriptors clash, or stereo descriptors are given for atoms which cannot be stereo centers). If the option is set to *none* (the default), or the compound is not stereogenic, no tag is inserted into the images.

-showcrossedbonds 0/1

If set, which is the default, mark stereogenic double bonds which do not possess expicit stereochemistry as stereochemically undefined, e.g. draw them as crossed bonds in depictions or attach an explicit ‘stereochemically undefined’ tag in formats which support this concept. If the flag is unset, such annotations are suppressed.

-showisotope 0/1

If this flag is set (the default), isotopically labelled atoms are annotated with their nucleonic number. Heavy hydrogen atoms are displayed as *D* or *T*. If the flag is unset, isotopic information is suppressed.

-showradical *0/1*

If this parameter is set to 0, marks for radical centers are not plotted. The default is 0.

-showstereo *0/1*

If this flag is set to 0, no stereo descriptors are plotted. Note that this flag has no influence on the display of wedge bonds (see **-wedges** and **-dashes** options to control their appearance). It only applies to atomic stereo descriptors such as CIP R or S, which might be present in the input file. These descriptors are not computed if not explicitly present in the input data (although CACTVS can do that in principle). They are only plotted if read from file. The default is 0.

-showstereogroups 0/1

If this flag is on (the default), indicators for membership in MDL-style atom stereo groups (*and/or/abs*) are displayed in the image. If the flag is off, this information is not shown.

-showstereoh 0/1

If this flag is set (the default), hydrogen atoms at stereo centers which are not linked via a wedge bond are explicitly drawn in order to obtain an unambiguous stereocenter display. Hydrogen atoms linked via a wedge bond are always drawn regardless of the value of this flag.

-symbolfontsize points

By default (or if you set this number to less than 0) the program chooses suitable font sizes for atomic symbols automatically. It is possible to override this choice with this option. Note that the number of available font sizes is limited if you use the built-in fonts instead of external TrueType fonts. The program chooses the best built-in font (from the list 6pt, 7pt, 8pt, 10pt) in this case. If this value is set to 0, no symbols are printed. Hetero atoms are then marked by small (colored, if in color mode) squares. The desired point size can be a floating point number.

-template SMILES/SMARTS

A substructure template in SMILES or SMARTS notation. This option is used in combination with the **-templatealign** option.

-templatealign *none/x/y/diagonal/rotate/redraw/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 is 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 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 options, a ring system must be matched completely, so for example a phenyl ring does not match a naphthalene ring is 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 are computed by the standard 2D layout procedure. The first successful match of the substructure is used as template. The *x*, *y*, and *diagonal* parameters aligns 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 rotates 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 regenerates 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 possess exactly the same relative coordinates as the substructure atoms, but the standard bond length is scaled to one. This method yields the cleanest results, provided no parts of the template map onto incomplete ring systems, which prevents a successful template match. The final style *combined* addresses this problem: It first attempts mode *redraw*, and fall back to *rotate* if *redraw* fails for any reason.

-templatefile filename

The name a file which contains a substructure template. This option is used in combination with the **-templatealign** option. Only the first record of the file is read. If the file does not contain 2D coordinates, and these are needed for the selected **-templatealign** option, coordinates are generated.

-templatematch *strict/relaxed*

This flag influences the match operation of fragments on structures. In mode *strict*, aliphatic fragment atoms do not match aromatic structure parts. In *relaxed* mode, which is the default, aliphatic fragment atoms also match aromatic systems. This setting applies only to atoms without additional explicit 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 thus match aromatic systems. This option can be used to counter this convention.

-version

Print program version and licensing information, and exit. No structure processing takes place.

-wedges *0/1*

If this option is set to 0, wedge bonds are not drawn as wedges. If the **-dashes** parameter is set, and the bond is a solid wedge, a bold line is drawn instead, otherwise a dashed line. The default for this parameter is 1.

-wedgepairs *no/all/hydrogen/unconnected*

Add extra wedges when generating 2D coordinates. By default (option value *no*), only a single wedge is used to determine the stereochemistry at atomic centers. In modes *all, unconnected* and *hydrogen*, a second wedge is attached to a bond extending from the stereo center to further disambiguate the stereochemical relationships. The second wedge is preferably attached to a bond leading to a hydrogen ligand, if such neighbor exists. In mode *hydrogen*, only bonds to hydrogen are used. If no such bond exists, no second wedge is drawn. Mode *unconnected* is the same as *all*, with the exception that the extra wedge to the preferred ligand is not drawn in case this ligand already participates in a wedge bond, regardless whether it is located at the tip or the bottom of such a wedge bond. In any case, the wedge is never drawn to a ring atom - if no chain neighbor without a wedge bond exists, the second wedge is omitted in all modes.

-wedgestyle *default/opposite*

If the mode *opposite* is selected, the drawing of the environment of atoms which are located at the tip of two or more wedge bonds of the same class (solid or dashed) is changed. If any of the offending ligands is a single atom, it is moved into a different gap of the neighbor sphere in such a way that the wedge style is changed to the opposite, and the stereochemistry of the center retained. While this feature is useful to meet specific drawing style demands, the graphical quality of the structure layout usually suffers.

-width pixels

The width of the image in pixels. If the **-crop** option is used, this parameter determines only the width of the initial drawing area, not necessarily the width of the final image. The default are 200 pixels.

COPYRIGHT

This program was designed and implemented for the CACTVS system by W. D.