Canvas 1.8

User Manual

Schrödinger Press

Canvas User Manual Copyright © 2013 Schrödinger, LLC. All rights reserved.

While care has been taken in the preparation of this publication, Schrödinger assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Canvas, CombiGlide, ConfGen, Epik, Glide, Impact, Jaguar, Liaison, LigPrep, Maestro, Phase, Prime, PrimeX, QikProp, QikFit, QikSim, QSite, SiteMap, Strike, and WaterMap are trademarks of Schrödinger, LLC. Schrödinger, BioLuminate, and MacroModel are registered trademarks of Schrödinger, LLC. MCPRO is a trademark of William L. Jorgensen. DESMOND is a trademark of D. E. Shaw Research, LLC. Desmond is used with the permission of D. E. Shaw Research. All rights reserved. This publication may contain the trademarks of other companies.

Schrödinger software includes software and libraries provided by third parties. For details of the copyrights, and terms and conditions associated with such included third party software, see the Legal Notices, or use your browser to open $SCHRODINGER/docs/html/third_party_legal.html (Linux OS) or %SCHRODINGER%\docs\html\third_party_legal.html (Windows OS).

This publication may refer to other third party software not included in or with Schrödinger software ("such other third party software"), and provide links to third party Web sites ("linked sites"). References to such other third party software or linked sites do not constitute an endorsement by Schrödinger, LLC or its affiliates. Use of such other third party software and linked sites may be subject to third party license agreements and fees. Schrödinger, LLC and its affiliates have no responsibility or liability, directly or indirectly, for such other third party software and linked sites, or for damage resulting from the use thereof. Any warranties that we make regarding Schrödinger products and services do not apply to such other third party software or linked sites, or to the interaction between, or interoperability of, Schrödinger products and services and such other third party software.

November 2013

Contents

Document Conventions ix

Chapter 1: Introduction 1

1.1 Running Schrödinger Software 1

1.2 Citing Canvas in Publications 2

Chapter 2: The Canvas Graphical Interface 5

2.1 Starting and Quitting Canvas 5

2.2 The Main Window 5

2.2.1 The Menu Bar 6

2.2.2 The Toolbars 7

2.2.3 Keyboard Shortcuts 9

2.3 Canvas Projects 10

2.3.1 Opening and Closing Projects 10

2.3.2 Importing Structures and Data 11

2.3.3 Exporting Structures and Data 15

2.3.4 The Project View Panel 18

2.3.5 The Messages View Panel 19

2.4 The Canvas Spreadsheet 19

2.4.1 Configuring and Navigating the Spreadsheet 19

2.4.2 Shortcut Menus 20

2.4.3 Selecting and Copying Rows, Columns, and Cells 22

2.4.4 Finding Text 23

2.4.5 Coloring Structures 24

2.4.6 Copying and Pasting Structures 24

2.4.7 Renaming Structures 25

2.4.8 Editing Structures 25

2.4.9 Editing Data Cells 28

2.4.10 Adding and Deleting Rows and Columns 29

2.4.11 Displaying Different Views of the Spreadsheet 29

2.4.12 Creating and Using Custom Views 30
2.4.12.1 Applying a Custom View to the Master View 32
2.4.12.2 Combining Views with Logical Operations 33
2.5 Organizing the Data 34
2.5.1 Sorting 34
2.5.2 Partitioning Rows Into Classes 35
2.5.2.1 Creating Partitions 35
2.5.2.2 Changing the Classes in a Partition 38
2.5.3 Creating a Heat Map 40
2.6 Filtering and Querying the Data 41
2.6.1 Filtering by Property Values 41
2.6.2 Filtering by Property Classes 42
2.6.3 Filtering by Structure 43
2.6.4 Filtering by Substructure 45
2.6.5 Detecting Duplicate Structures 47
2.7 Making Charts of the Data 47
2.7.1 Scatter Plots 48
2.7.2 Histograms 50
2.7.3 Pie Charts 51
2.7.4 Chart Settings 52
2.7.5 Saving Images of Charts 54
2.8 Calculating Statistics 55
2.9 Calculating New Properties from the Data 56
2.10 Viewing Structures in PyMOL 58
2.11 Setting Preferences 59
Chapter 3: Running Applications from Canvas 61
3.1 General Features 61
3.2 Molecular Properties 62
3.2.1 Calculating Molecular Properties 63
3.2.2 Selecting Representative Properties 65

3.3 Fingerprints 68
3.3.1 2D Fingerprints 68
3.3.2 3D Fingerprints from Pharmacophores 70
3.4 Similarity, Dissimilarity, and Clustering 73
3.4.1 Computing a Similarity or Distance Matrix 73
3.4.2 Screening Structures by Similarity or Distance 75
3.4.3 Screening Structures by Shape 76
3.4.4 Selecting Diverse Structures 78
3.4.5 Hole-Filling and Library Optimization 79
3.4.6 Comparing Libraries 81
3.4.7 Clustering Structures by Similarity or Distance 82
3.4.7.1 Hierarchical Clustering 83
3.4.7.2 Leader-Follower and K-Means Clustering 86
3.5 Building a Predictive Model 88
3.5.1 Multiple Linear Regression 89
3.5.2 Partial Least-Squares Regression 91
3.5.3 Kernel-Based Partial Least-Squares Regression 92
3.5.4 Principal Components Regression 95
3.5.5 Bayes Classification 96
3.5.6 Neural Networks 98
3.5.7 Recursive Partitioning 99
3.5.8 Self-Organizing Maps 101
3.6 Other Applications 104
3.6.1 Principal Components Analysis 104
3.6.2 Finding the Maximum Common Substructure 105
3.6.3 Scaffold Decomposition 108
3.6.4 R-Group Analysis 110
3.6.5 Running a 3D Minimization 110
3.7 Running External Applications 111
3.8 Running Python Scripts 114

Chapter 4: Using Canvas 115

4.1 Selecting Compounds Not Represented in a Library 115

4.2 Removing Duplicate Structures from a Project 116

4.3 Choosing a Fingerprint 117

4.4 Creating Modal Fingerprints 117

4.5 Tips 118

Chapter 5: Running Applications from the Command Line 119

5.1 Common Syntax Descriptions 119

5.2 2D Fingerprints 122

5.2.1 canvasFPGen 122

5.2.2 canvasFPCombine 127

5.2.3 canvasFPBinary2CSV 128

5.2.4 canvasCSV2FPBinary 129

5.3 3D Pharmacophore Fingerprints 130

5.4 Similarity, Dissimilarity, and Clustering 132

5.4.1 canvasCSV2PW 133

5.4.2 canvasCSVMatrix 133

5.4.3 canvasDBCS 134

5.4.4 canvasLibOpt 136

5.4.5 canvasFPHist 139

5.4.6 canvasFPMatrix 140

5.4.7 canvasHC 143

5.4.8 canvasHCBuild 146

5.4.9 canvasHCSelect 147

5.4.10 canvasTreeDraw 148

5.4.11 canvasKMeans 149

5.4.12 canvasLC 150

5.4.13 canvasPW2CSV 151

5.5 Model Building and Related Applications 152
5.5.1 canvasMDS 154
5.5.2 canvasMLR 155
5.5.3 canvasMolDescriptors 156
5.5.4 canvasPCA 160
5.5.5 canvasPCAGen 160
5.5.6 canvasPCAProj 161
5.5.7 canvasPCAReg 162
5.5.8 canvasPLS 163
5.5.9 canvasKPLS 164
5.5.10 canvasBayes 165
5.5.11 canvasNnet 167
5.5.12 canvasRP 168
5.5.13 canvasSOM and canvasSOMBits 170
5.6 Utilities 172
5.6.1 canvas_app 172
5.6.2 canvasConvert 172
5.6.3 canvasJob 174
5.6.4 canvasProjectDB 176
5.6.5 canvasSDMerge 179
5.6.6 canvasSearch 180
5.6.7 canvasMCS 182
5.6.8 canvasScaffold 184
5.7 Scripting with Canvas 187
References **189**
Getting Help **191**
Index **193**

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	`$SCHRODINGER/maestro`	File names, directory names, commands, environment variables, command input and output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

Links to other locations in the current document or to other PDF documents are colored like this: Document Conventions.

In descriptions of command syntax, the following UNIX conventions are used: braces { } enclose a choice of required items, square brackets [ ] enclose optional items, and the bar symbol | separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

File name, path, and environment variable syntax is generally given with the UNIX conventions. To obtain the Windows conventions, replace the forward slash / with the backslash \ in path or directory names, and replace the $ at the beginning of an environment variable with a % at each end. For example, `$SCHRODINGER/maestro` becomes `%SCHRODINGER%\maestro`.

Keyboard references are given in the Windows convention by default, with Mac equivalents in parentheses, for example CTRL+H (⌘H). Where Mac equivalents are not given, COMMAND should be read in place of CTRL. The convention CTRL-H is not used.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Introduction

Canvas is a cheminformatics package that provides a range of applications for structural and data analysis, including fingerprints, similarity searching, substructure searching, selection by diversity, clustering, building regression and classification models. The Canvas graphical interface is project-oriented and provides chemical structure storage and organization, data analysis and visualization, and access to the applications.

You can run Canvas applications from the graphical interface, from the command line, from KNIME with the Schrödinger KNIME Extensions, and from Python scripts with the Canvas Python API.

A good introduction to cheminformatics can be found in the book by Leach and Gillet [1].

1.1 Running Schrödinger Software

Schrödinger applications can be run from a graphical interface or from the command line. The software writes input and output files to a directory (folder) which is termed the *working directory*. If you run applications from the command line, the directory from which you run the application is the working directory for the job.

Linux:

To run any Schrödinger program on a Linux platform, or start a Schrödinger job on a remote host from a Linux platform, you must first set the `SCHRODINGER` environment variable to the installation directory for your Schrödinger software. To set this variable, enter the following command at a shell prompt:

csh/tcsh: `setenv SCHRODINGER` *installation-directory*

bash/ksh: `export SCHRODINGER=`*installation-directory*

Once you have set the `SCHRODINGER` environment variable, you can run programs and utilities with the following commands:

```
$SCHRODINGER/program &
$SCHRODINGER/utilities/utility &
```

You can start the Canvas interface with the following command:

```
$SCHRODINGER/canvas &
```

It is usually a good idea to change to the desired working directory before starting the Canvas interface. This directory then becomes the working directory.

Windows:

The primary way of running Schrödinger applications on a Windows platform is from a graphical interface. To start the Canvas interface, double-click on the Canvas icon, on a Canvas project, or on a structure file; or choose Start → All Programs → Schrodinger-2013-3 → Canvas. You do not need to make any settings before starting Canvas or running programs. The default working directory is the Schrodinger folder in your documents folder (Documents on Windows 7/Vista, My Documents on XP).

If you want to run applications from the command line, you can do so in one of the shells that are provided with the installation and have the Schrödinger environment set up:

- Schrödinger Command Prompt—DOS shell.
- Schrödinger Power Shell—Windows Power Shell (if available).

You can open these shells from Start → All Programs → Schrodinger-2013-3. You do not need to include the path to a program or utility when you type the command to run it. If you want access to Unix-style utilities (such as `awk`, `grep`, and `sed`), preface the commands with `sh`, or type `sh` in either of these shells to start a Unix-style shell.

Mac:

The primary way of running Schrödinger software on a Mac is from a graphical interface. To start the Canvas interface, click its icon on the dock. If there is no Canvas icon on the dock, you can put one there by dragging it from the SchrodingerSuite2013-3 folder in your Applications folder. This folder contains icons for all the available interfaces. The default working directory is the Schrodinger folder in your Documents folder (`$HOME/Documents/Schrodinger`).

Running software from the command line is similar to Linux—open a terminal window and run the program. You can also start Canvas from the command line in the same way as on Linux. The default working directory is then the directory from which you start Canvas. You do not need to set the `SCHRODINGER` environment variable, as this is set in your default environment on installation. To set other variables, on OS X 10.6 and 10.7 use the command

`defaults write ~/.MacOSX/environment` *variable* `"`*value*`"`

and on OS X 10.8 use the command

`launchctl setenv` *variable* `"`*value*`"`

1.2 Citing Canvas in Publications

The use of this product should be acknowledged in publications as:

Canvas, version 1.8, Schrödinger, LLC, New York, NY, 2013.

Please also cite the following reference:

Duan, J.; Dixon, S.L.; Lowrie, J.F; Sherman, W. Analysis and comparison of 2D fingerprints: Insights into database screening performance using eight fingerprint methods. *J. Molec. Graph. Model.* **2010**, *29*, 157-170.

The Canvas Graphical Interface

The Canvas graphical interface is designed for project-oriented work and provides chemical structure storage, data analysis and visualization, and access to a wide range of applications that are also available through the command line. The interface also provides links to Maestro that allow you to easily transfer structures and data between the two applications.

2.1 Starting and Quitting Canvas

Linux: To start the Canvas interface, enter the following command:

```
$SCHRODINGER/canvas [ -proj projName.cnv [ -mae maeFile|-sd sdFile
    [-append|-replace] [-autodelete]] | -zippedproj projName.cnv[.]zip]
```

The `-proj` option opens the specified project. If the project exists, it is opened. If the project does not exist, it is created, and you must provide a Maestro or SD structure file to add to the project with the `-mae` option or the `-sd` option. If you specify a structure file with an existing project, the structures and properties in the file replace those in the existing project unless you specify `-append`. The `-autodelete` option automatically deletes the structure file after it has been imported into the project. You can open a zipped project with the `-zippedproj` option.

Windows: To start the Canvas interface, double-click the Canvas icon on the desktop, or choose Start → Programs → Schrodinger-2013 → Canvas. You can also double-click on a Canvas project, to start Canvas and open the project.

Mac: To start the Canvas interface, click the Canvas icon on the dock. If there is no Canvas icon on the dock, drag it from the SchrodingerSuite2013 folder in your Applications folder to the dock. You can also start the interface from the command line in a terminal window, as for Linux.

To quit Canvas, choose File → Quit or type CTRL+Q (⌘Q) in the main window.

2.2 The Main Window

The Canvas main window has a menu bar, toolbars, a spreadsheet area, a Project View panel, a Messages View panel, and a status area. The two panels are docked into the main window by default but can be undocked. You can show or hide these views from the Project menu or from the context menu (right-click menu, shortcut menu). The status area reports statistics on the number of rows and columns: the total, the number visible, and the number selected.

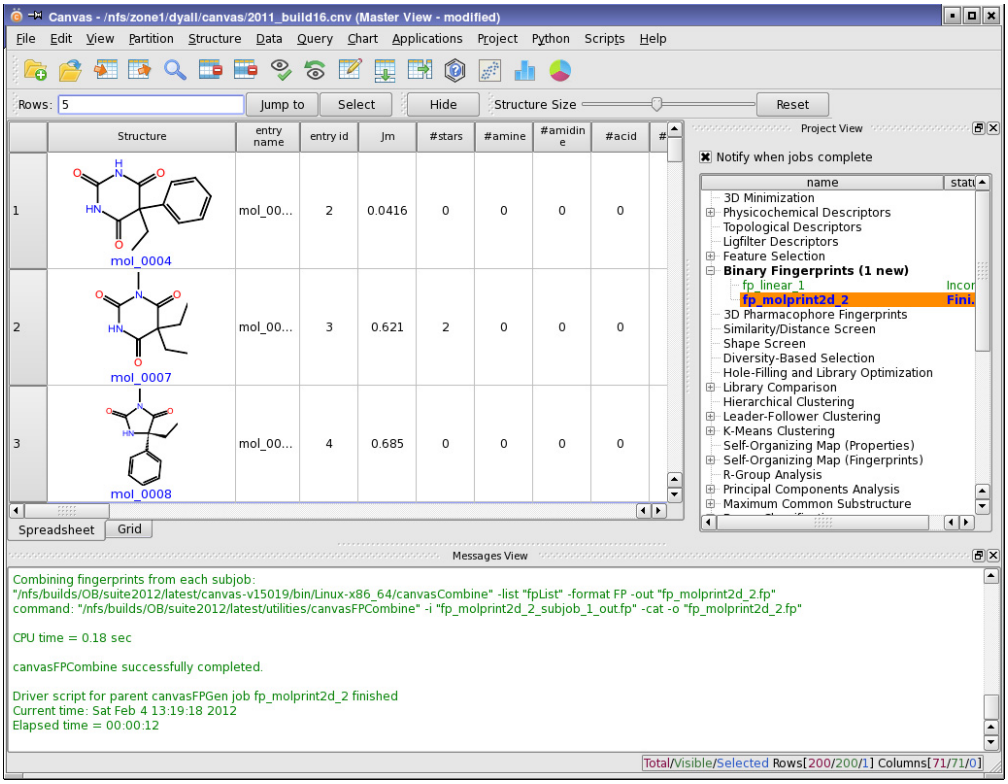


Figure 2.1. The Canvas main window

The spreadsheet contains the 2D structures in the first column (including the structure name if there is one), followed by columns containing the properties of the structures. It is described in detail in Section 2.4 on page 19.

2.2.1 The Menu Bar

The menu bar has the following menus:

- File—Open and close projects, import and export structures and data, set preferences, and quit Canvas. This menu is described in detail in Section 2.3 on page 10.
- Edit—Copy cells, clear cell content, add or delete rows or columns, select cells, rows, or columns, find and select text. This menu is described in detail in Section 2.4 on page 19.
- View—Create, delete, display, and perform operations on custom views. This menu is described in detail in Section 2.4.11 on page 29.

- Partition—Create, store, modify, apply, and delete partitions of the spreadsheet rows. This menu is described in detail in Section 2.5.2 on page 35.
- Structure—Configure the structure display and copy and paste structures. This menu is described in detail in Section 2.4.1 on page 19 and Section 2.4.5 on page 24.
- Data—Perform operations on the table data, such as sorting, filtering creating a heat map, calculating statistics and new properties, and editing data. This menu is described in detail in Section 2.5 on page 34 and Section 2.6 on page 41.
- Query—Query the database and return rows that match the query. This menu is described in detail in Section 2.6 on page 41.
- Chart—Create and display scatter plots, histograms, and pie charts. This menu is described in detail in Section 2.7 on page 47.
- Applications—Run Canvas applications. This menu is described in detail in Chapter 3.
- Project—Show or hide the Project View and the Message View.
- Python —Open a Python interpreter to run Python commands or scripts.
- Scripts—Install, manage, and run Python scripts. This menu is described in detail in Section 3.8 on page 114.
- Help—Display online help, the user manual, Canvas legal notices, and the Knowledge Base; open the Diagnostics panel. This menu is described in detail on page 191.

If you start Canvas without opening a project, only the File and Help menus are available.

2.2.2 The Toolbars

The Canvas interface has four toolbars: File, Edit, Select Rows, and Structure Size. You can show or hide the toolbars from the shortcut menu, and you can show or hide toolbar buttons in the Preferences panel (File → Preferences → Toolbar Buttons).

The File toolbar contains buttons for many of the common actions. A brief description of the File toolbar is given below, with cross-references to the sections where these actions are described in detail.

New Project
Create a new Canvas project. The current project is closed. See Section 2.3.1 on page 10.

Open Project
Open an existing Canvas project. Section 2.3.1 on page 10.

Import
Import structures into the project. See Section 2.3.2 on page 11.

Export
Export structures from the project to an external file. See Section 2.3.3 on page 15.

Find
Find text in the spreadsheet. Opens the Find panel. See Section 2.4.4 on page 23.

Collapse to Selected
Collapse the view to show only the selected structures. Same as View → Collapse to Selected.

Hide Selected
Hide the selected structures. Same as View → Hide Selected. See Section 2.4.11 on page 29.

Apply to Master
Apply the current custom view to the master view. Only available in custom views. See Section 2.4.11 on page 29.

Save View
Save the current view. In the master view, the view is saved as a new custom view, and the Save Custom View dialog box opens. See Section 2.4.11 on page 29.

Restore Default View / Revert All Changes
Restore the default view in the master view, or revert all changes to the view in a custom view. In either case, the view is restored to its initial (default) state. See Section 2.4.4 on page 23.

Edit Spreadsheet
Edit cells in the spreadsheet. When you click this button, the Edit toolbar is displayed. See Section Note: on page 28.

Add Row
Add a new empty row to the end of the spreadsheet.

Add Column
Add a new empty column to the right side of the spreadsheet.

Substructure Query
Run a substructure query. See Section 2.6.4 on page 45.

Scatter Plot
Create a scatter plot. See Section 2.7.1 on page 48.

Histogram
Create a histogram. See Section 2.7.2 on page 50.

Pie Chart
Create a pie chart. See Section 2.7.3 on page 51.

The Edit toolbar is only active when you are editing the spreadsheet. It is displayed in place of the File toolbar when you start editing, and has the following two buttons.

Save Spreadsheet Changes
Save the spreadsheet with the changes included. When you click this button, the spreadsheet is saved, and the File toolbar is redisplayed. See Section Note: on page 28.

Exit Without Saving Spreadsheet Changes
Discard all the changes to the spreadsheet. When you click this button, the spreadsheet is reverted to its state before you started editing, and the File toolbar is redisplayed.

The Select Rows toolbar is described in Section 2.4.3 on page 22, and the Structure Size toolbar is described in Section 2.4.1 on page 19.

2.2.3 Keyboard Shortcuts

Many of the menu items have keyboard shortcuts. These shortcuts are listed in Table 2.1.

Table 2.1. Keyboard shortcuts for menu items.

Menu Item	Shortcut
File → New Project	CTRL+N (⌘N)
File → Open Project	CTRL+O (⌘O) [letter O]
File → Close Project	CTRL+W (⌘W)
File → Quit	CTRL+Q (⌘Q)
Edit → Copy	CTRL+C (⌘C)
Edit → Select All	CTRL+A (⌘A)
Edit → Clear Selection	CTRL+U (⌘U)
Edit → Find	CTRL+F (⌘F)
View → Refresh	F5
Partition → New → From Selection	ALT+N (⌥N)
Partition → Open	ALT+O (⌥O) [letter O]
Partition → Add/Move to → Class *n*	ALT+*n* (⌥*n*)
Partition → Add/Move to → New Class	ALT+L (⌥L)
Partition → Unclassify	ALT+0 (⌥0) [number zero]
Partition → Close	ALT+W (⌥W)
Data → Undo Last Edit	CTRL+Z (⌘Z)
Data → Redo Last Edit	CTRL+SHIFT+Z (⇧⌘Z)

Table 2.1. Keyboard shortcuts for menu items. (Continued)

Menu Item	Shortcut
Python → Command Window	CTRL+P (⌘P)
Scripts → Manage	CTRL+M (⌘M)
Scripts → Import	CTRL+I (⌘I)

2.3 Canvas Projects

Canvas stores all of its data in a project, which is simply a directory that contains a number of subdirectories and files. A Canvas project directory has the extension `.cnv`. Canvas does not have scratch projects, so you must create a project before you can import structures.

2.3.1 Opening and Closing Projects

The basic operations of opening and closing projects can be performed from the File menu:

- New Project—Create a new Canvas project. You can also use CTRL+N (⌘N), or click the New Project toolbar button.

  Opens a file selector in which you can navigate to a location and name the project. The default location on Linux is the directory from which Canvas was started, and on Windows is the Schrodinger folder in your documents folder. If a project is already open, the open project is closed first.

- Open Project—Open an existing project. You can also use CTRL+O (⌘O), or click the Open Project toolbar button.

  Opens a file selector in which you can navigate to a location and select the project or a project archive. Project archives are unzipped in place.

  On Windows, you can open a project in a new Canvas session by double-clicking the project or project archive icon.

- Recent Projects—Open one of the most recently used projects. The submenu displays a list of up to 10 projects, by default. You can set the number of projects listed in the Preferences panel (File → Preferences → History).

- Close Project—Close the current project. You can also use CTRL+W (⌘W). The data in a project is automatically saved whenever you make a change.

Canvas projects are saved whenever a change is made, so there is no Save menu item.

You can create a zipped archive of a project by choosing Archive Project. The archive is given a `.cnvzip` extension. When you open an archived project, changes that are made in the opened project are not automatically re-archived: you must choose Archive Project and overwrite the previous archive.

You can make a project read-only by removing write permission from the project directory (`.cnv`). Users who do not have write permission will be able to open the project in Canvas and import structures from the project, but not perform any operations that would change the project.

2.3.2 Importing Structures and Data

There are several ways of importing structures and data into a Canvas project. From inside Canvas, you can import from a file or from a project. From Maestro, you can export selected entries to a Canvas project—see Section 3.5 of the *Maestro User Manual* for details.

To import structures and data from another Canvas project, choose File → Import Project. A file selector opens, in which you can navigate to a location and select the project. All structures and all data are imported.

To import structures and properties from a file into Canvas, choose File → Import, or click the Import button on the toolbar.

A file selector opens, in which you can navigate to a location, choose the format, and select one or more files. The supported file formats are:

- Maestro, compressed or uncompressed (`.mae`, `.mae.gz`, `.maegz`)
- SD file, compressed or uncompressed (`.sd`, `.sdf`, `.sd.gz`, `.sdf.gz`)
- CSV file with SMILES strings and properties (`.csv`)
- SMILES file with SMILES strings and optional titles (`.smi`).
- Canvas fingerprint file (`.fp`)

When you have selected the files, a dialog box opens (Figure 2.3 or Figure 2.2) for each file type, so that you can set options for importing the structures and matching the properties. Properties are imported in the order in which they are present in the input files or project. When you close this dialog box, the import progress is shown in the status bar, at the bottom of the main panel. There is also a button you can use to stop the import operation (and discard the results).

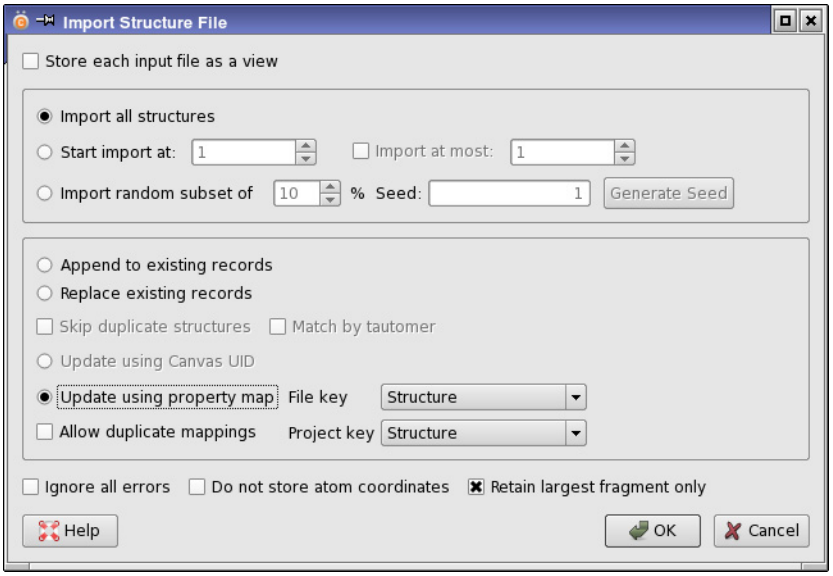


Figure 2.2. The Import Structure File dialog box

You can create a view for each file that is imported by selecting Store each input file as a view. The views are named after the input file. By default, no views are created on import.

All structures are imported by default. To restrict the range of structures, select Start import at, and enter the index of the first structure that you want to import in the box. The first structure in the file has the index 1. Then to import a given number of structures, select Import at most and enter the number of structures to import in the box. If this option is not selected, all structures from the first specified to the end of the file are imported. You can also import a random subset of structures, by selecting Import random subset of, specifying the percentage of structures, and generating a seed for the random number generator.

The structures that you import can be appended to the project (Append to existing records), used to replace the entire project (Replace existing records), or update the existing records where the structures are already in the project and append new structures (Update using Canvas ID or Update using property map).

You can also set options to skip structures that are already in the Canvas project, by selecting Skip duplicate structures. If any of the structures in the set that you selected for import are already in the project, they are not imported. If you want to treat tautomers as duplicates, select Match by tautomer.

Likewise, you can automatically skip problematic structures by selecting Ignore all errors. By default, you are prompted for an action when an unreadable structure is encountered. Problematic structures are written in the same format as the input file to the directory `failedimports`.

When you import from a structure file (Maestro or SD format), you can choose to import only the connectivity information, and discard the atom coordinates. To do so, select Do not store atom coordinates. By default, the coordinates are imported.

If there are structures in the input file that contain multiple fragments, you can import only the largest fragment by selecting Retain largest fragment (the default). Otherwise, all fragments are imported.

If you choose to update the project, you must choose an input file property from the File key option menu and a project property from the Project key option menu, so that the input structures with their data and project records can be matched, as follows:

- If the value of the file key property for a given structure matches a value of the project key property, the project record is updated with the properties from the file, and the structure is not changed.
- If multiple project records are matched for a single file key property value, and Allow duplicate mappings is selected, the properties are updated for all matching project records.
- If the file key property has multiple instances of the same value, then the last one in the file is used to update the project.
- If no match is found for the import property, a new record is added.

You might want to use a corporate ID, a unique SMILES string, or a title, as the property to match.

If you want to ensure that you do not add duplicate structures to the project, choose Structure for the Import property and the Project property, and ensure that Allow duplicate mappings is selected. The properties from the input file are still used to update those in the project if the input file structure matches a project structure.

When you import Canvas fingerprints, Update using property map is the only option available.

If you are importing structures that were previously exported from the same Canvas project, with the Canvas UID included, you can select Update using Canvas UID to update the properties in the project from the input file. The UID in the file is matched with the UID in the project, so using this option with another project is likely to corrupt the project data. An example of its use is to export binary fingerprints and reimport the bits as ordinary numeric properties with the value 0 or 1.

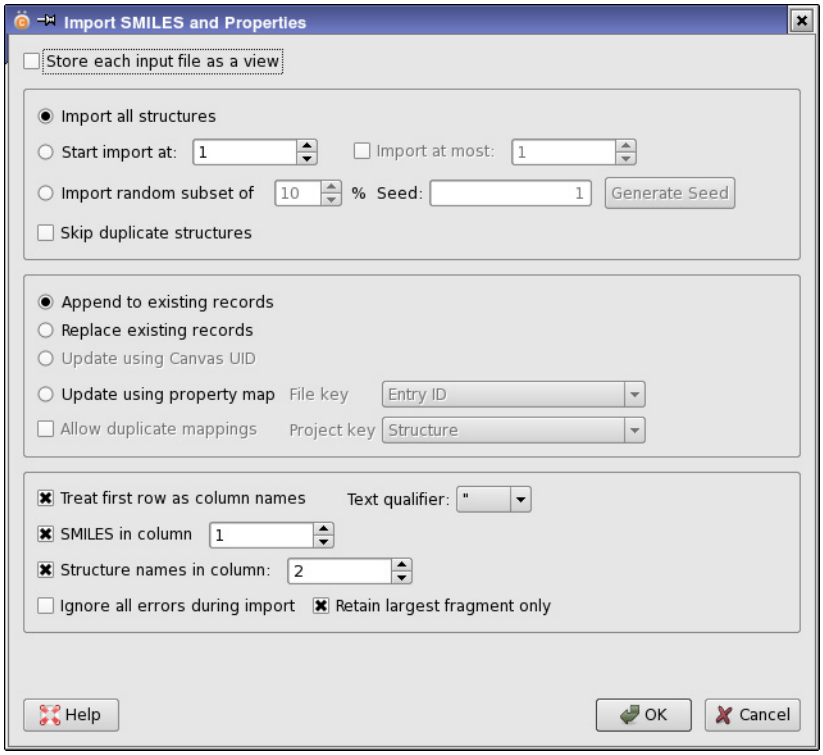


Figure 2.3. The Import SMILES and Properties dialog box

If you are importing structures in SMILES format with properties, from a `.csv` or `.txt` file, you have the choice of adding the structures with their properties, or just adding the properties. To add the structures and the properties, select SMILES in column and specify the column number that contains the SMILES strings. The default is 1. If you deselect this option, you can import properties without structures. If you use the Update using property map option, you can add properties from a CSV file to existing structures, provided you have a column in the CSV file to match a project property. Otherwise, the records in the input file are added to the project with an empty structure field.

If the first row of the file contains the names of the properties, select Treat first row as column names. If this option is deselected, the property names are set to Prop1, Prop2, and so on.

For string properties, you can choose the character that defines the string from the Text qualifier option menu. These characters are removed from the string when the string is added as a project property. The choices are a double quote, a single quote, or none. If you choose None, all quotes are preserved in the project. If you choose the single quote, then single quotes are

removed but double quotes are retained. Likewise, if you choose the double quote, then double quotes are removed but single quotes are retained.

To define which column the structure names (or titles) are in, select Structure names in column, and set the column number in the box. The default is column 2. If you deselect this option, the title is left blank. If you are importing from a SMILES file, you can select or deselect this option to indicate that the file contains a title, separated from the SMILES string by a space or a tab. The column number is then set to 2 and cannot be changed.

You can also import 2D structures on the basis of their common name or IUPAC name. To do this, create a new table row (Edit → Add Row or use the toolbar button), then double-click the Structure cell in this row to open the 2D structure editor. Choose Edit → Common or IUPAC Name Search, and enter the name in the text box. When you click OK in the 2D structure editor, the structure is added to the project and appears in the Structure cell.

2.3.3 Exporting Structures and Data

Canvas provides several ways of exporting structures and data. You can export the selected rows (rows in which all columns are selected) to a file or to Maestro. You can also copy and paste from Canvas into a spreadsheet. Properties are exported in the order in which they appear in the view, so you can rearrange the properties in the view into the order you want them to appear in the file before exporting.

To export structures and properties to a file, choose File → Export, or click the Export toolbar button.

A file selector opens, in which you can navigate to a location, choose the format, and name the file. The supported file formats are:

- Maestro, compressed or uncompressed (`.mae`, `.mae.gz`, `.maegz`)
- SD file, compressed or uncompressed (`.sd`, `.sdf`, `.sd.gz`, `.sdf.gz`)
- CSV file with SMILES strings and properties (`.csv`)
- SMILES file with SMILES strings and optional titles (`.smi`).
- Fingerprint file (`.fp`).

When you click Export, the Export Options dialog box opens. This dialog box allows you to select the properties and fingerprints to export, and set various options. When you dismiss this dialog box, the export starts, and the progress of the export is shown in the status bar, at the bottom of the main panel. There is also a button you can use to stop the export operation (and discard the results).

All properties are initially selected for export. If you want to export a random subset of the rows, select Export random subset, and specify the desired number of rows in the box. You can also specify the random number seed, by entering it in the Seed text box, or clicking Generate Seed.

If you are exporting to a structure file, you can export the coordinates as 2D coordinates by selecting Force 2D coordinates. When you do so, you can choose one of the three options for exporting hydrogens: Suppress hydrogens, Export polar hydrogens only, or Export all hydrogens. The default is to export only polar hydrogens. If the file is an SD file, you can choose to export in the newer version 3.0 format by selecting Use MDL version 3.0 format for SD files.

If you are exporting to a SMILES or CSV file, the default is to export unique (canonical) SMILES. To export a SMILES string that depends on the order of the atoms in the structure (first atom in the structure is first in the SMILES string), deselect Export unique SMILES.

The structures in a Canvas project have a unique identifier (UID), which can be exported by selecting Export Canvas UIDs. Exporting the UIDs allows you to add or change properties for the structures that you export, and then ensure that these properties are associated with the correct structures when you reimport them into the Canvas project.

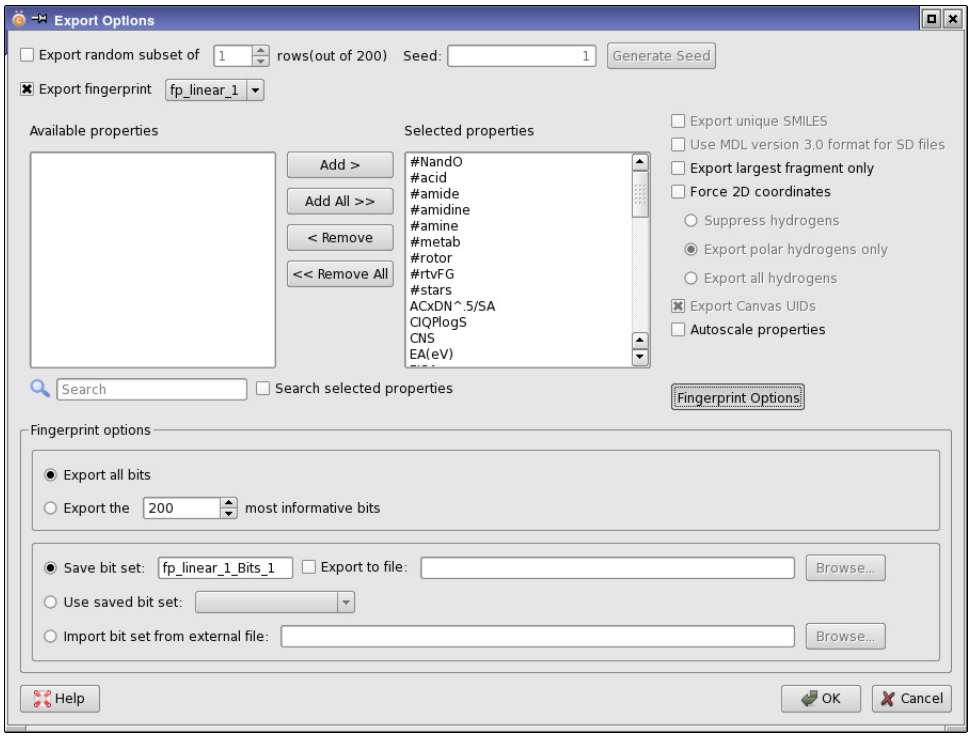


Figure 2.4. The Export Options dialog box showing fingerprint options

If you want to scale the properties so that they have a mean of 0 and a standard deviation of 1, you can do so by selecting Autoscale properties. This can be useful when exporting properties to other applications that require automatically scaled variables. If you want to scale only some of the properties rather than all properties exported, select them in the Selected properties list.

Fingerprints can be exported for two choices of file format: a CSV file (SMILES+Properties) or a binary file (Fingerprint, `.fp`). When you choose either of these file formats, the Export fingerprint option is available in the Export Options dialog box, and you can select this option to export fingerprints and choose the fingerprint to export. To display options for exporting the fingerprints, click Fingerprint Options.

You can decide whether to export all bits or the most significant *n* bits. By default, all bits are exported to a binary file, but only the most informative 200 bits are exported to a CSV file. This is because each bit in the fingerprint is stored as a separate property in the CSV file, with values of 0 or 1. The prefix for naming these properties can be entered in the Prefix for bit columns text box, which is absent if exporting to a binary file.

Another alternative is to export a set of bits that has previously been stored (a "bit set"). A bit set stores information on which bits were chosen for a given fingerprint to form the set, and thus indirectly stores information on the features that these bits represent. One reason for choosing a stored bit set is for consistency on export. The most informative 200 bits for one set of molecules might not be the most informative for another set, so if you export the most informative bits, the exported fingerprints will not have information about the same features in the molecules. You can avoid this discrepancy by exporting a saved bit set.

To use a bit set from the project, select Use saved bit set, and choose the bit set from the option menu. This menu lists saved bit sets for the fingerprint you are exporting. To use a bit set from a file, select Import bit set from external file, and click Browse to navigate to the file, or enter the file name in the text box. When you choose either of these options, the options to export all or the most significant bits are not available.

If you choose to export all or the most significant bits, the bit set that you use to export the fingerprint is saved in the project with the fingerprint, and can also be saved in an external file. Save bit set is automatically selected and you can enter a name for the bit set in the text box. The bit set is then saved in the project. To save it in an external file as well, select Export to file, and click Browse to navigate to the file, or enter the file name in the text box.

To export structures and properties from a Canvas project directly to Maestro, choose File → Export to Maestro. The structures and properties are exported from the Canvas project to a temporary Maestro file, then Maestro is started and the file contents are imported.

To copy rows or columns from Canvas to a spreadsheet program, such as Microsoft Excel or Open Office Calc, simply select the rows or columns and copy them to the clipboard with CTRL+C (⌘C) or Edit → Copy. You can then paste them into the chosen application.

2.3.4 The Project View Panel

This panel gives a summary of all the information that is available in the project. It lists the applications and the various classifications of the data (filters, views, partitions) as a tree. When information exists in any of these categories, you can expand the display by clicking the turner (the + in a square) to the left of the category name. To collapse the display, click the turner again (which now shows a – in the square). For applications, the display lists the names of the jobs that have been run. The status of the job is displayed in the status column. For views, partitions, and filters, the named entities for that category are listed. The description of a view is displayed in a tool tip when you pause the pointer over the view name in the tree.

Both the "branches" and the "leaves" in the tree have a shortcut menu that you can use to apply an action. For the branches, the shortcut menu has the following items, some of which are only available on particular branches:

- Open—Open an application panel or a filter panel; open the dialog box for selecting a view or a partition.
- New—Create a new partition from a property or a random assignment. Only on the Partitions branch.
- Import Queries—Import queries from another project. Opens a file selector so you can navigate to and select the project. Only on the Substructure Queries branch.
- Sort—Sort views in the default order (order of addition), ascending order or descending order. Only on the Views branch.
- Delete All—Delete all items in the branch and the data associated with them.

For the leaves, there is one common item, Delete, which deletes the item and its associated data from the project. For applications, the other available actions are:

- View—view the results of the job, by opening the relevant panel.
- Incorporate—incorporate the results of the application into the project.
- Clone Job—run a job with the same settings as the selected job.
- Kill Job—kill a running job.

Other actions that are available only for certain branches are:

- Apply to Master—Apply the view to the Master View. Only for Views.
- Rename View—change the name of the view and its description. Only for Views.
- Apply Filter—Apply the filter to the Master View. Only for filters.
- Run Query—Run the selected query. Only for Substructure Queries.
- Delete Query—Delete the selected query. Only for Substructure Queries.
- Open—Open a view or a partition. Only for Views and Partitions.

You can show and hide this panel by choosing Project → Project View. You can hide the panel by clicking the close button in the title bar. You can undock this panel and place it wherever you want, and you can redock it, using the docking button.

2.3.5 The Messages View Panel

The Messages View panel displays messages generated by actions on the project, including the running of applications. Log messages and job error messages are displayed in this panel.

You can show and hide this panel by choosing Project → Messages View. You can hide the panel by clicking the close button in the title bar. You can undock this panel and place it wherever you want, and you can redock it, using the docking button.

2.4 The Canvas Spreadsheet

The spreadsheet is the area where structures and properties are displayed. Although it does not have the full capabilities of a spreadsheet program such as Microsoft Excel or OpenOffice Calc, it does function in many respects in the same way.

2.4.1 Configuring and Navigating the Spreadsheet

The spreadsheet can be configured and navigated in much the same way as other spreadsheets, as described below.

You can **resize** individual rows and columns by dragging their borders. To resize multiple rows or columns simultaneously, select the rows or columns and drag one of the borders inside the selection. To resize a column to fit the data, double-click the right border of the column heading. If you have several columns selected, double-clicking the right border of the heading of one of these columns fits each column to the width of its data.

You can **move** a column by dragging it to the new location, or you can move it to either end of the table by right-clicking in the heading row and choosing Move to Start or Move to End. You can **sort** the columns by name by right-clicking in the heading row and choosing Sort columns by name. The sort is case-insensitive.

You can change the number of decimal places displayed for property values in the Preferences panel—see Section 2.11 on page 59.

You can **scroll** the spreadsheet with the mouse wheel or the scroll boxes. If you scroll rapidly in a vertical direction, structures are not drawn until you pause or release the scroll box, because Canvas retrieves structures and properties on demand from an SQLite database. This allows you to scroll through a project containing thousands of rows with no delay. You can also jump to a specific row by entering the row index in the Rows text box on the toolbar and clicking Jump to or pressing TAB, then ENTER.

You can **step** through cells in the table with the arrow keys or the tab key. Stepping through cells changes the selection. The current cell has a border drawn around it in reverse video (black on white, yellow on blue).

The first column in the spreadsheet displays the **structure**. This is a 2D image created from the connection table for the structure. You can hide the structure and just display its title in this column by choosing Structure → Hide. To show it again, choose Structure → Show. You can also use the toolbar button labeled Show (if the structures are hidden) or Hide (if they are displayed). When the structures in the spreadsheet are hidden, the tool tip for the table cell displays the structure as well as the title. You can also show or hide the hydrogen atoms in the structure that are not displayed by default by choosing Structure → Show All Hydrogens.

The structures are scaled automatically so that the bond lengths are a reasonable size and approximately equal for all structures. Structures that are too large for the cell with the scaling are scaled down in size. If you want the structures to always fill the cell, deselect Autoscale on the Structure menu. You can also change the structure size with the Structure Size slider. If you adjust the structure sizes, you can reset them to the default sizes by clicking Reset.

2.4.2 Shortcut Menus

The spreadsheet has shortcut menus for the selected rows, for the column headings, and for the selected structure cell. These are displayed when you right-click and hold.

The selected rows shortcut menu has the following items, given with their menu equivalents:

- Delete—delete the selected rows (Edit →Delete).
- Collapse to Selected—show only the selected rows (View → Collapse to Selected).
- Hide—hide the selected rows (View → Hide Selected).
- Export—export the selected rows to a file (File → Export).
- Export to Maestro—export the selected rows to a new Maestro session (File → Export to Maestro).

The column heading shortcut menu has the following items when you right-click on a single column:

- Sort by *property* (Ascending)—Sort the rows by the values of the property in this column, in ascending order.
- Sort by *property* (Descending)—Sort the rows by the values of the property in this column, in descending order.
- Sort Columns by Name—Rearrange all columns in alphabetical order.
- Restore Natural Column Order—restore the columns to the order in which they were added to the project.
- Rename Column—Change the name of the column. Opens the Edit Column Name dialog box, so you can change the name and description.
- Move to Start—Move the column to the first position (leftmost) in the spreadsheet.
- Move to End—Move the column to the last position (rightmost) in the spreadsheet.
- Hide—Hide the column from the current view. The data remains in the project.
- Delete—Remove the property in this column from the project, and from all views.
- Set Property Value—Set the value of the property in the selected rows to a specified value. A blank value clears the property values.
- Create Modal Fingerprint—Create a modal fingerprint row using the data in the column, which must be a fingerprint column. Opens a dialog box in which you can choose the selected or the visible rows to define the modal fingerprint, and name the fingerprint. A new row is added, with the name displayed in the Structure column. This item is only available if you right-click on a fingerprint column.

When multiple columns are selected, only the Hide and Delete items are present.

The selected structure cell shortcut menu has the following items:

- Edit Structure—Edit the structure in the 2D structure editor.
- Edit Structure Name—Change the name of the structure.
- Copy—Copy the structure to the clipboard in SD format.
- Copy SMILES—Copy the structure to the clipboard as a SMILES string
- Copy Image—Copy an image of the structure to the clipboard.
- Paste—Paste the SD-format structure from the clipboard into the current cell to replace the current structure.

- Paste SMILES—Paste a SMILES string from the clipboard into the current structure cell to replace the current structure.

2.4.3 Selecting and Copying Rows, Columns, and Cells

There are several ways of selecting cells in the spreadsheet. The selected cells are highlighted with a blue background.

Selection of cells with the mouse follows the usual rules for tables:

- To select a row, click the row index.
- To select multiple rows, use shift-click and control-click, or drag over the row indices.
- To select a column, click the column heading.
- To select multiple columns, use shift-click and control-click.
- To select a cell, click in the cell.
- To select multiple cells, use shift-click and control-click, or drag over the cells.

You can move the selection to a neighboring cell with the arrow keys. The arrow keys move the current cell, and the new cell is selected. This action clears the selection of any other cells.

You can make selections or change the selection from the Edit menu:

- Select All—Select all cells in the spreadsheet.
- Clear Selection—Clear the selection of cells.
- Invert Selection—Select the cells that are not selected, and deselect the cells that are selected.
- Expand Selection To Entire Rows—Expand the selection so that the entire row is selected for any row that has one or more cells selected. Entire rows must be selected to export data.
- Expand Selection To Entire Columns—Expand the selection so that the entire column is selected for any column that has one or more cells selected.

To select specific rows, enter the row indices in the Rows text box on the toolbar, and click Select (or press TAB twice, then ENTER). The row indices are given as a comma-separated list, and can include row ranges. Ranges can be specified either as $n:m$ or $n-m$. To specify a range that includes the first or the last row, use $:m$ or $n:$. Examples are given in the tool tip for the text box. The current cell is set to the first cell in the row selection.

To select cells whose contents match a text expression, use the Find panel, which you open from the Edit menu. When the text is found, click Select All Matches. The Find panel is described in detail in Section 2.4.4.

You can also make some selections from the View menu:

- Select Rows with Missing Values—Select the rows for which one or more columns do not have values.
- Select Columns with Missing Values—Select the columns for which one or more rows do not have values.
- Select Rows with Invalid Chemistry—Select the rows for which a structure is not available. This can happen on import if the structure block is badly formatted or the SMILES string is invalid; in this case a string noting the error is displayed in the structure cell.

Once you have a selection, you can copy it to the clipboard with CTRL+C (⌘C) or Edit → Copy. The selection that is copied is the rectangular region that contains the selected cells. You can then paste the selection into a spreadsheet application such as Microsoft Excel (Windows), OpenOffice Calc (Linux), or iWork Numbers (Mac).

2.4.4 Finding Text

If you want to search for text in the spreadsheet, you can use the Find panel. To open this panel, choose Edit → Find in the main window, or click the Find toolbar button.

To specify the search string, enter it in the Find what text box. You can qualify the search to match the entire contents of the cell and to match the case. If you want to find the exact string, select Literal string. To match a string using the wildcard characters * (zero or more characters of any type) and ? (one character of any type), select Allow wildcards. If you want to perform more complex pattern matching, select Interpret as regular expression, and enter a regular expression in the text box. The regular expression uses metacharacters to represent variable elements in the search, and follows the regular expression syntax for Perl. To match numeric values to the precision given in the table, select Interpret as floating point value.

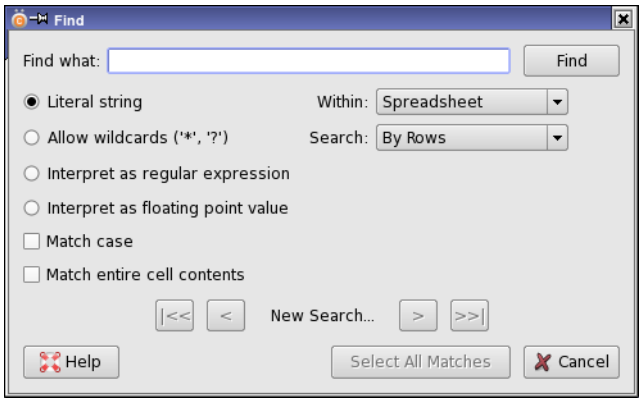


Figure 2.5. The Find panel

To restrict the range of the search in the spreadsheet, choose Spreadsheet or Selected cells from the Within option menu. To find columns rather than data values, choose Column Names.

To set the search order, choose By rows or By columns from the Search option menu.

Once you have set up the text, the range, and the search order, click the Find button to perform the search. All instances of the search string are located with a single click. You can select all the matching cells in the spreadsheet by clicking Select All Matches.

You can navigate through the matches with the navigation buttons. The double-arrow buttons |<< and >>| skip to the first and last matches; the single-arrow buttons step to the next and the previous match. The current match is indicated between the forward and backward navigation buttons. The current cell in the spreadsheet is set to the current match, and the spreadsheet scrolls to the current cell.

2.4.5 Coloring Structures

You can control some aspects of the structure coloring. To color the non-carbon atoms, choose Structure → Color Heteroatoms. If you do a substructure query, you can color the matches by choosing Structure → Highlight Matches. The color used can be set by choosing Structure → Choose Highlight Color.

2.4.6 Copying and Pasting Structures

Structures can be copied and pasted in MDL (SD) format within Canvas. To copy the current structure in MDL format, choose Structure → Copy Structure; to paste a structure into a cell (replacing the existing structure), select the cell and choose Structure → Paste. Pasting cannot be undone.

You can copy and paste structures as SMILES strings, both within Canvas and to other applications. To copy in SMILES format, choose Structure → Copy SMILES. You can then paste the structures into another application, or into a different cell in the Structure column. To paste a structure into a cell, select the cell and choose Structure → Paste SMILES. The structure that you paste can come from any external application that supports SMILES format, such as Maestro, ChemDraw, or ISIS/Draw. SMILES strings can also be pasted into a text box that requires a SMILES string. The structure is pasted as a 2D structure.

Copying and pasting structures can be done from the shortcut menu for the structure cell: right click in the cell and choose the menu item. These items are the same as on the Structure menu.

For creating presentations or other documents, you can copy an image of a structure, by selecting the structure and choosing Structure → Copy Image. The image is copied to the clipboard, and can then be pasted into any application that accepts an image.

2.4.7 Renaming Structures

You can change the structure name by selecting the structure and choosing Structure → Edit Name or right-clicking in the structure cell and choosing Edit Name. In the dialog box that opens, you can type in a new name. You can also change the names used for all the structures by choosing a property for the structure name. To do so, choose Structure → Change Structure Name Property. You can then choose a property from those in the spreadsheet, and store the current structure name in a new property. Properties with real number values are not available. The property you use for the structure name can be kept in the spreadsheet, or it can be removed from the spreadsheet.

2.4.8 Editing Structures

If you want to edit a structure, you can open the 2D structure editor by any of these methods:

- Double-click the cell containing the structure.
- Right-click on the structure and choose Edit Structure.
- Select the structure in the spreadsheet and choose Structure → Edit Structure.

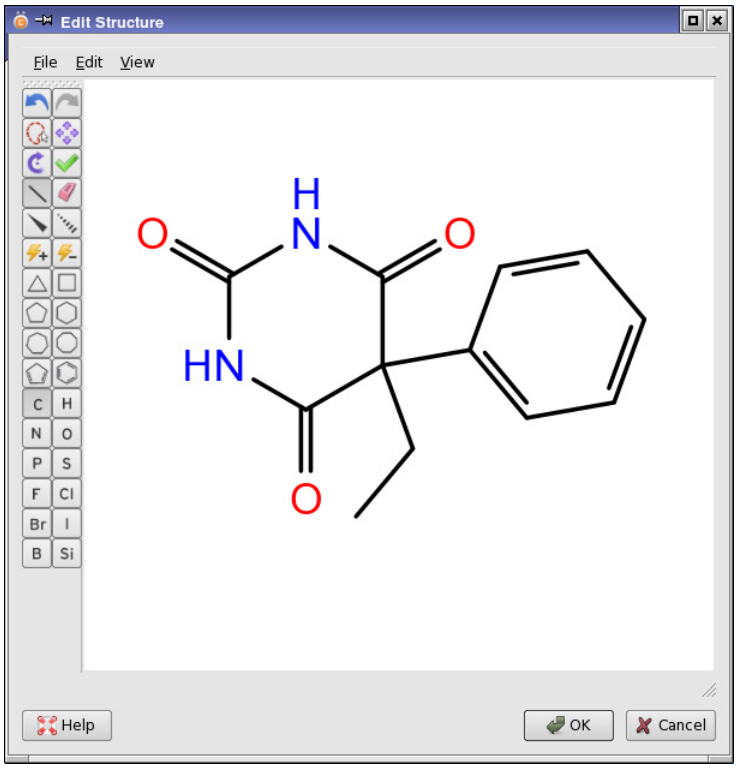


Figure 2.6. The Edit Structure dialog box

The 2D structure editor (Edit Structure dialog box) consists of a menu bar, a toolbar, and a display area.

The toolbar has three sets of buttons. The first set of buttons is for performing actions, which are described below. The second set is a collection of rings that you can add to the structure. The third set is a collection of elements, which you can use to change atoms to a chosen element. The selection of the element can also be done by typing the element symbol (case insensitive).

Undo
Undo the last action. Can be performed multiple times.

Redo
Redo the last action that was undone. Can be performed multiple times.

Lasso
Select atoms by drawing around them ("lassoing"). The selected atoms can be moved, rotated, deleted, copied.

Move
Move (translate) the selected atoms or the entire structure. Drag in the drawing area to move the structure. Drag the selection to move the selected atoms. (This is the default action for dragging a selection, so you do not need to click this button before dragging it.) You can also move the structure by dragging with the right mouse button.

Rotate
Rotate the selected atoms or the entire structure. Drag in the drawing area to rotate the structure. The angle through which the structure has been rotated is displayed near the center of rotation while you are rotating it. If rotation does not seem to be working properly, check whether you have other structures in the drawing area that are not visible. You can also rotate the structure by dragging with the middle mouse button.

Cleanup
Clean up the structure (2D coordinates) and normalize orientation and arrangement of groups. If you have atoms selected, only those atoms are cleaned up.

Draw
Draw a single bond. Click on an atom to draw a bond to a new (carbon) atom, click on a bond to add another bond between two atoms.

Erase
Delete atoms or bonds. Clicking on a carbon atom deletes it; clicking on a non-carbon atom changes it back to carbon, then clicking again deletes it. Clicking on a bond removes one bond.

Wedge Bond
Draw a wedge bond. Clicking on an atom draws a wedge bond to a new (carbon) atom; clicking on a single bond converts it to a wedge bond.

Dashed Bond
Draw a dashed bond. Clicking on an atom draws a dashed bond to a new (carbon) atom; clicking on a single bond converts it to a dashed bond.

Increase Charge
Increase the formal charge on an atom by 1.

Decrease Charge
Decrease the formal charge on an atom by 1.

Some common tasks are listed below:

- To add a singly bonded carbon, click the Draw button, then click on the atom that you want to attach it to. Use the Wedge Bond or Dashed Bond button instead to indicate the stereochemistry.
- To define the stereochemistry of a bond, click the Wedge Bond or Dashed Bond button, the click on the bond you want to change.
- To increase the order of a bond, click the Draw button, then click on the bond.
- To decrease the order of a bond, click the Erase button, then click on the bond.
- To delete an atom, click the Erase button, then click on the atom you want to delete. If it is not a carbon atom, it is first converted to carbon, then clicking again deletes the atom.
- To add a ring fragment, click the fragment button, then click on the atom or bond that you want to replace with the fragment.
- To add a functional group, pause the pointer over the atom you want to add it to, then type – followed by its name or SMARTS pattern, and press ENTER, e.g. `-et` to add an ethyl group. The named functional groups are listed in Table 2.2, with the corresponding SMARTS patterns; the names are case-insensitive. You can also replace an atom with a functional group, by omitting the initial –.
- To change the element, click the button for the desired element or type the element symbol, then click on the atom you want to change.
- To increase or decrease the charge on an atom, click the Increase Charge or Decrease Charge button, then click on the atom. You can also pause the pointer over the atom and type in the charge.
- To move the structure, click the Move button, then drag in the display area.
- To rotate the structure, click the Rotate button, then drag in the display area.

Table 2.2. Functional group names and SMARTS patterns for use in editing structures.

Name	SMARTS	Name	SMARTS	Name	SMARTS
me	C	ph	C(C=C1)=C-C=C1	coo-	C(=O)[O-]
et	CC	bz	CC(C=C1)=C-C=C1	no2	[N+](=O)[O-]
pr	CCC	chex	C(CC1)CCC1	so2	S(=O)=O
ipr	C(C)C	cp	C(C=C1)C=C1	so2n	S(=O)(=O)N
nbu	CCCC	ome	OC	so3	S(=O)(=O)[O-]
ibu	CC(C)C	oet	OCC	po3	P(=O)([O-])[O-]
sbu	C(C)CC	cooh	C(=O)O	con	C(=O)N
tbu	C(C)(C)C	coome	C(=O)OC	nco	NC=O
nhex	CCCCCC	cooet	C(=O)OCC	cn	C#N

Note: If you edit a 3D structure in the 2D structure editor, all 3D structural information is discarded, and the structure is stored as 2D. To edit a structure in 3D, you can export it to Maestro, edit it, and reimport it.

2.4.9 Editing Data Cells

You can edit the data in the spreadsheet cells directly by choosing Data → Edit Spreadsheet, or clicking the Edit Spreadsheet toolbar button. The spreadsheet is placed in editing mode, and you can click in any cell and change the data in the cell, with the exception of the structure cells (first column) and the cells of a partition. You can undo the last edit with CTRL+Z (⌘Z) or Data → Undo Last Edit, and you can redo an edit with CTRL+SHIFT+Z (⇧⌘Z) or Data → Redo Last Edit. When you have finished making changes, choose Data → Save Spreadsheet Changes to make the changes permanent and exit editing mode, or click the Save Spreadsheet Changes toolbar button. If you want to discard all the changes you have made and return the spreadsheet to the state it was in before editing, choose Data → Exit Without Saving Spreadsheet Changes, or click the Exit Without Saving Spreadsheet Changes toolbar button.

To clear the contents of one or more cells, select the cells and choose Edit → Clear.

If you want to assign a single value to the selected rows for a particular property, right-click on the column heading and choose Set Property Value. A dialog box opens, in which you can set the value in a text box. When you click OK, the property values in the selected rows are set to the value you entered in the text box. If there was no value specified in the text box, the values in the selected rows are cleared.

2.4.10 Adding and Deleting Rows and Columns

To append a row to the spreadsheet, choose Edit → Add Row or click the toolbar button. A new empty row is added, and the Structure cell is selected, so that you can paste a structure into the cell, or build one in the Edit Structure dialog box.

To append a column to the spreadsheet, choose Edit → Add Column. The New Column dialog box opens, in which you can name the column. The name must be unique. When you click OK, a new empty column is added.

To delete rows or columns from the spreadsheet, select the rows or columns and choose Edit → Delete. The rows or columns are removed after confirmation. You can also right-click on a column heading and choose Delete from Project to delete the columns.

2.4.11 Displaying Different Views of the Spreadsheet

There are occasions on which you might want to see only a particular range of rows and columns, arranged in a particular way. Such an selection and arrangement is called a “view”. Changes to the view can be made manually, or as a result of some other process. When the view is changed, the title bar indicates that the view is modified. To revert the changes, choose View → Restore Default View, or click the Restore Default View toolbar button.

The View menu allows you to hide or show the selected rows and columns. To display a subset of columns and rows, select them and choose View → Collapse to Selected. This action displays all rows and columns that have any cells selected. If you want to hide particular rows or columns, select them and choose View → Hide Selected, or click the Hide Selected button.

To select the columns that you want to show, choose View → Manage Properties. The Manage Properties dialog box has two lists: Available properties, which are the hidden properties, and Visible properties, which are the shown properties. By default, all properties are on the Visible properties list. You can transfer properties between lists by selecting them and clicking Add or Remove. You can limit the list to those properties that match a text string by typing the string in

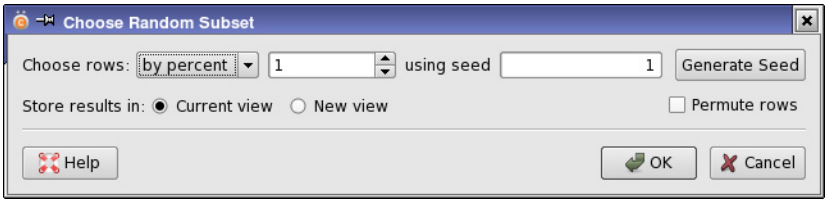


Figure 2.7. The Choose Random Subset dialog box.

the search text box, after setting the Search visible properties option to select the list to limit. When you click OK, the spreadsheet shows only those properties that were in the Visible properties list.

If you want to display a random subset of rows, choose View → Random Subset. You can select the number or percentage of rows in the subset, enter or generate a seed for the random number generator, randomize the row order, and decide whether to save the results in the current view or create a new custom view (see below).

When you run a query or filter the data, only the rows that match are displayed. These operations are described in Section 2.6 on page 41.

2.4.12 Creating and Using Custom Views

Changes made to the view are transient: as soon as another action is taken, the previous arrangement and visibility of the rows and columns is modified. If you want to preserve a particular view of the data, you can create a *custom view* that is saved in the project and can be displayed later. As well as providing a view of the data, custom views can be used to select data to run applications.

To **create** a custom view, choose View → Save As, or click the Save View toolbar button.

The Save Custom View dialog box opens, and you can name the view and provide a description. The view can be based on all rows that are visible in the parent view (Use all rows), or the selected rows (Use selected rows). When you create the view, you can choose to display it immediately by selecting Open view. The named view is stored in the project and is listed under the View node in the Project View panel, and on menus in application panels.

Once you have created a custom view, you can work in this view to run queries, filter the data, and create charts. However, you cannot change the data in a custom view: this must be done in the *master view*, which is the view of the data shown in the main window.

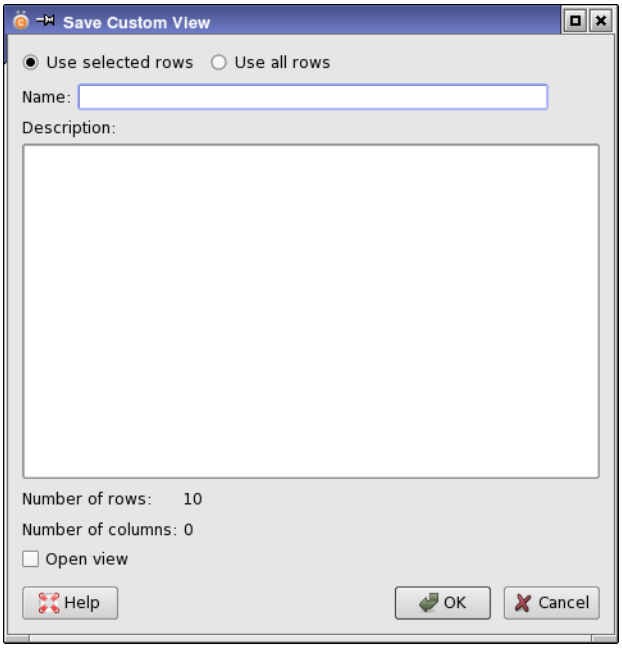


Figure 2.8. The Save Custom View dialog box

If you want to **rename** a custom view, you can do so in the Project View panel. Expand the View node, right-click on the view and choose Rename View. The Save Custom View dialog box opens, and you can enter a new name and change the description. If the view is open, it is closed before renaming.

If you want to **examine** or change the description of the view, choose View → Details. The Details panel for the view opens, and displays the name, description, row and column count, and information on the time it was created, last saved, and last modified. You can edit the description in this panel.

Custom views can be **opened** by choosing View → Open. This action opens a dialog box, in which you can select the view that you want to open. Another way to open a custom view is to expand the View node in the Project View panel, and double-click the desired view, or right-click it and choose Open. The tool tip for the view shows its description.

Custom views are always opened in a **separate window**. This window is a reduced version of the main window: the menu bar does not have the Applications and Project menus, the File menu only has export items, and the Project View and Messages View windows are absent. The absence of these components reflects the fact that the custom view is intended for viewing and rearranging the data, but not making changes in the data.

Changes that you make to a custom view are not automatically saved. To **save** a custom view, choose View → Save in the custom view window. You can also save a custom view as a new custom view, in the same way as from the master view.

You can **undo** changes to the view, by choosing View → Undo change or View → Undo All View Changes. The latter operation undoes changes back to the last time the view was saved.

When you have finished working in a view, you can **close** it by choosing View → Close. If you no longer want to keep the view, choose View → Delete to **delete** it, or right-click on its name in the Project View and choose Delete.

To **sort** the list of views that is displayed in the Project View panel and in application menus, right-click on the View node, choose Sort, then choose a direction, from Ascending, Descending, or Default. The first two items sort the views in alphabetical order; the last restores the default order, which is the order the views were created.

If you want to run **applications** from the command line with the rows in a custom view, you can **export** the row IDs for the view by choosing View → Export Row IDs. The binary file that is written can then be used in conjunction with the project to run an application on the rows in the view. If you run a command line job that generates new properties with `canvasJob` (see Section 5.6.3 on page 174), you should choose View → Refresh to view the new properties.

Custom views can also be created from the results of an **application**. These views have a File menu, for exporting the view; a View menu, from which you can apply the view to the master, save the view in the project and manage properties; and a Structure menu with items for the scaling and visibility of the structures. These views show only the structures by default, but properties can be added by choosing View → Manage Properties and selecting properties. Once you have selected a property, this property is added to the view whenever you open a view of the results of that application.

2.4.12.1 Applying a Custom View to the Master View

Custom views are independent views of the spreadsheet. However, you can apply a custom view to the master view by choosing View → Apply to Master. This action is also available from the shortcut menu for the view in the Project View panel.

Choosing View → Apply to Master allows you to apply the current row and column selection and ordering to the master view. The Apply to Master dialog box opens, offering choices for the rows, columns, and visibility of the rows in the master view.

To change the visible rows in the master view to match those in the current view, select Retain current for Rows. If there are rows in the current view that are not visible in the master view, and you want to make them visible, select Allow expansion of Master View row set. If you want

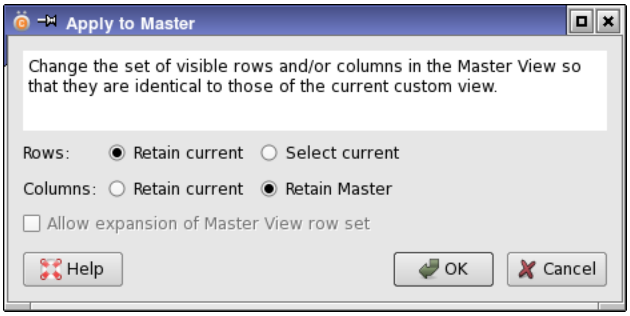


Figure 2.9. The Apply to Master dialog box.

to change the visible columns in the master view to match the current view, select Retain current for Columns; to keep the master view visible columns the same, select Retain Master.

If you only want to apply the selection of rows to the master view rather than change the visibility, select Select current for Rows and Retain Master for Columns.

2.4.12.2 Combining Views with Logical Operations

Custom views can be combined with logical operations to update the current view or create new custom views. This capability is useful, for example if you have created custom views with several different filters, and want to combine the results. To combine other custom views with the current view, choose View → Logic, which opens the Apply Logic to View dialog box.

Four logical operations are available:

- And—keep only the rows or columns that are in both views.
- Not—keep the rows or columns that are not in the other view.
- Or—keep the rows or columns that are in either view.
- Exclusive Or—keep the rows or columns that are in either view but not in both.

You can choose whether to retain the rows or the columns, by selecting the appropriate Retain option. These selections restrict the scope of the logical operations: they are not applied to the rows if you select Retain rows in current view, and likewise for columns.

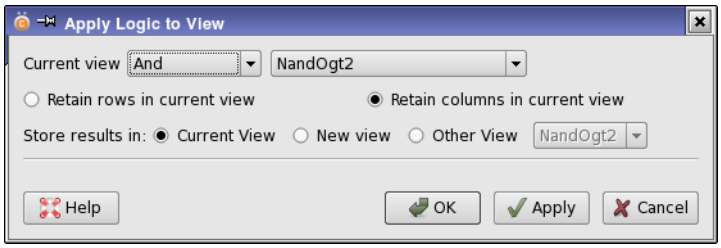


Figure 2.10. The Apply Logic to View dialog box.

If you want to create a new view, select New view. When you click Apply or OK, the Save Custom View dialog box opens, in which you can name, describe, and save the view.

If you want to store the results in an existing view (overwriting that view), select Other view and select the view from the option menu.

To apply multiple logical operations to a view, first select Current view. After selecting the operation, the other view, and the row and column retention options, click Apply. You can then repeat the selection and click Apply as many times as you like.

If you want to apply a sequence of logical operations on a view and create a new view with the result, but keep the original view, choose View → Save before you start the process. After you have created the new view, choose View → Undo All View Changes in the original view. This operation undoes the operations back to the last save.

2.5 Organizing the Data

Canvas provides various ways of organizing the rows in the spreadsheet. Some of them, such as clustering, involve running an application. These means of organizing data are treated in Chapter 3. The simplest ways of organizing the data are sorting and partitioning, which are described in the first two subsections below. Another way of highlighting the data values is by coloring the table cells using a heat map.

2.5.1 Sorting

One of the simplest ways of organizing the data is sorting. To sort the rows in any view of the spreadsheet, choose Data → Sort, which opens the Sort dialog box. This dialog box allows you to sort the data (New sort), to reverse the order of the previous sort (Reverse current order), or restore the order of the rows as they are stored in the database (Restore natural order).

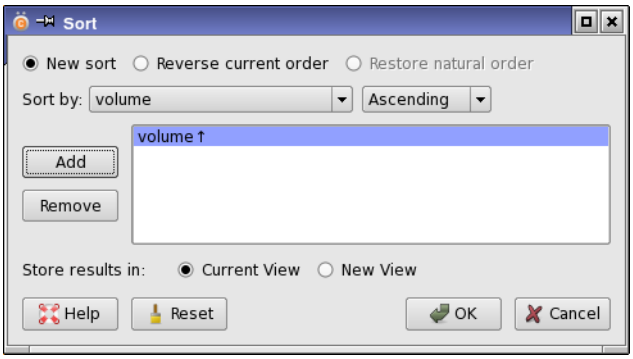


Figure 2.11. The Sort dialog box.

To start a new sort of the data, choose New sort, then start the selection of the sort keys. You can sort by multiple property values, in the order in which you add them to the sort list. To add a sort key to the sort list, choose the property from the Sort by option menu, choose the sort direction, from Ascending or Descending, and click Add. The property and the direction is added to the end of the sort list. You can then add another sort key.

Sorting is done in the order in which the keys are specified in the sort list. If you want to change the order, you can drag a sort key to a new position in the list.

You can apply the results of the sort to the current view or create a new view with the results, by choosing one of the Store results in options. If you create a new view, the Save Custom View dialog box opens, and you can name, describe, and save the sort results as a new view.

For a simple sort on the values of a single property, right-click in the column heading and choose Sort by *property* (Ascending) or Sort by *property* (Descending).

2.5.2 Partitioning Rows Into Classes

Another way of organizing the structures (or rows) is to partition them into classes. The classes can be based on the value of some property, or they can be assigned manually or randomly.

Partitioning the rows creates a new property, which is shown as a column in the spreadsheet whose cells are colored according to the class that the row belongs to. The column heading has Class:: prepended to the name, so it is distinguished from any actual property of the same name. The value of this property for any row is the class value. You can then sort the rows by the class value to display classes contiguously.

2.5.2.1 Creating Partitions

There are three ways of creating a partition from scratch: from the selected rows, from values of a property, and by random assignment. You can also save an existing partition with a new name (Partition → Save As) and then edit the partition (Partition → Edit).

To create a new partition from the selection, choose Partition → New → From Selection or press ALT+N. The New Partition from Selection dialog box opens. In this dialog box, you can name the partition and provide a description. When you click OK, a new partition is created, and the New Class dialog box opens with the selected rows as the members of the class. If you selected Create classes for selected and unselected rows, the unselected rows are assigned as members of a second class. You can edit the class values in the table, and assign a color by right-clicking in the cell in the Color column and choosing Change Color. A color selector opens, in which you can choose the desired color.

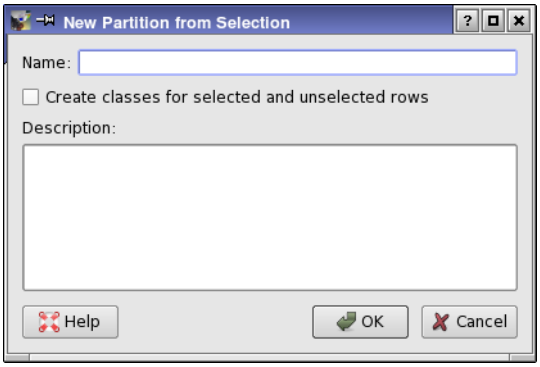


Figure 2.12. The New Partition from Selection dialog box.

To create a partition based on property values, choose Partition → New → From Property. The New Partition from Property dialog box opens. You can choose the property from the Property option menu. The property name is the default name for the partition, but you can edit it in the Name text box. You can enter text describing the partition in the Description text area.

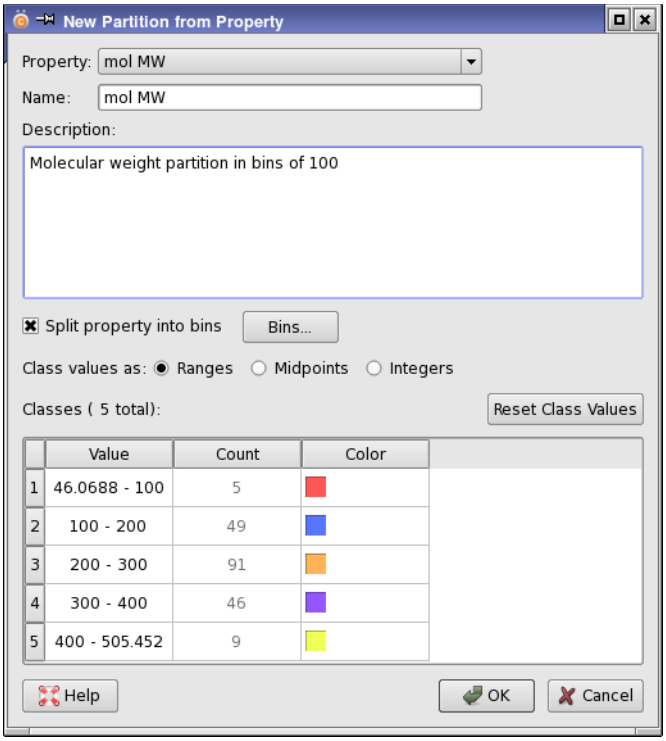


Figure 2.13. The New Partition from Property dialog box.

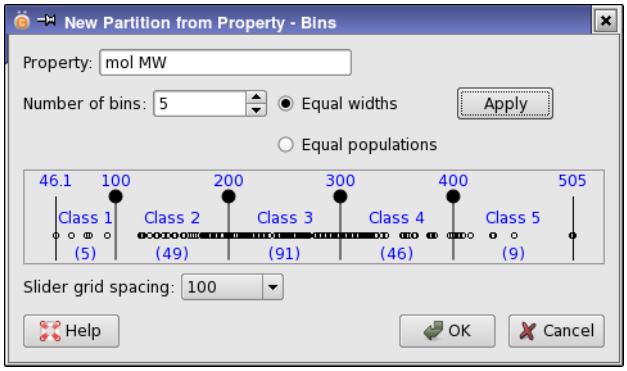


Figure 2.14. The Bins dialog box.

If you want to create a class for each property value, ensure that Split property into bins is not selected. If you want to create classes that cover a range of property values, select this option and click Bins, to specify the widths of the classes. Property values that are less than the minimum value are placed in the first class, and property values that are greater than the maximum value are placed in the last class. The lower boundary of each of these classes is included in the class; the upper boundary is in the next class.

You can choose how to set the class values by selecting one of the Class values as options:

- Ranges—Use a string *m–n* that indicates the range of property values in the class.
- Midpoints—Use the midpoint of the range.
- Integers—Use the bin index for the class value.

For classes based on individual property values, the class value is the property value, and these options are not available.

The Classes table shows the class values, the count, and the color used in the partition column in the spreadsheet. You can edit the class values if you wish by editing the table cells. For classes based on property value ranges, you can reset the class values to the defaults as described above by clicking Reset. You can change the color for a class by right-clicking in the cell in the Color column and choosing Change Color. A color selector opens, in which you can choose the desired color.

To create a random partition, choose Partition → New → Random Split. The New Partition from Random Split dialog box opens. A standard name is supplied in the Name text box, which you can edit. You can enter text describing the partition in the Description text area. You can specify the number of classes, and set the seed used to generate random numbers for assignment of the rows to classes. To assign the rows to classes after making a change, click Update.

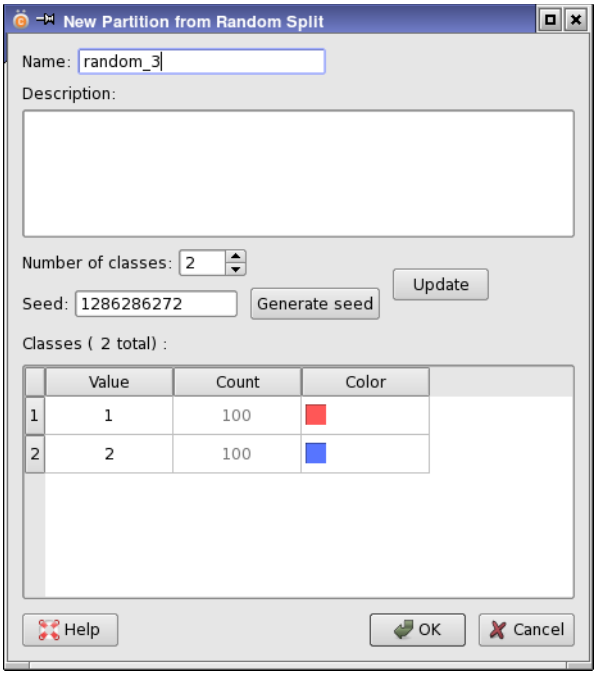


Figure 2.15. The New Partition from Random Split dialog box.

As for the New Partition from Property dialog box, the Classes table shows the class values, the count, and the color used in the partition column in the spreadsheet. The class values here are a sequence of integers, which you can change by editing the table cells. You can change the values to text strings if you want labels rather than numeric values. You can change the color for a class by right-clicking in the cell in the Color column and choosing Change Color, which opens a color selector.

2.5.2.2 Changing the Classes in a Partition

When a partition has been created, you can add classes, move rows to new classes or declassify them, edit the partition, or delete the partition.

To create a new class, first select the rows in the spreadsheet. The rows can be unclassified rows, or rows that are already in one or more classes. Next, choose Partition → Add/Move to → New Class, or press ALT+L (⌥L). The New Class dialog box opens, showing the class table. You can then edit the class value and change the color, as described above.

To move rows to an existing class, select the rows and choose Partition → Add/Move to → Class *n*, or type ALT+*n* (⌥*n*), where *n* is the class index. Classes are indexed sequentially from 1, and the class value is displayed on the Add/Move to submenu.

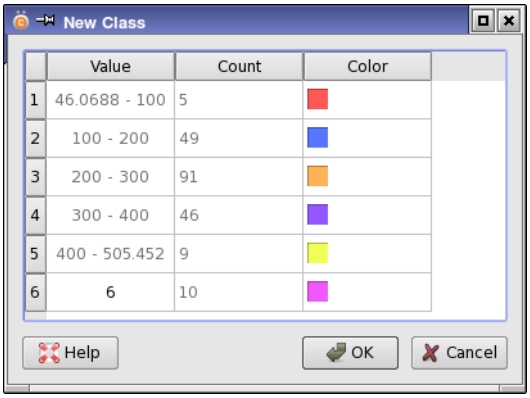


Figure 2.17. The New Class dialog box.

To remove rows from their classes, so that they are not in any class, select the rows and choose Partition → Unclassify, or press ALT+0 (⌥0). You do not need to select entire rows to add or remove them from a class: any row with a selected cell is included in operations on a class.

To make changes to the partitions, choose Partition → Edit. In the Edit Partition dialog box, you can change the name and description of the partition, change the class values and colors, and delete classes. To delete a class, right-click on it in the Class table and choose Delete.

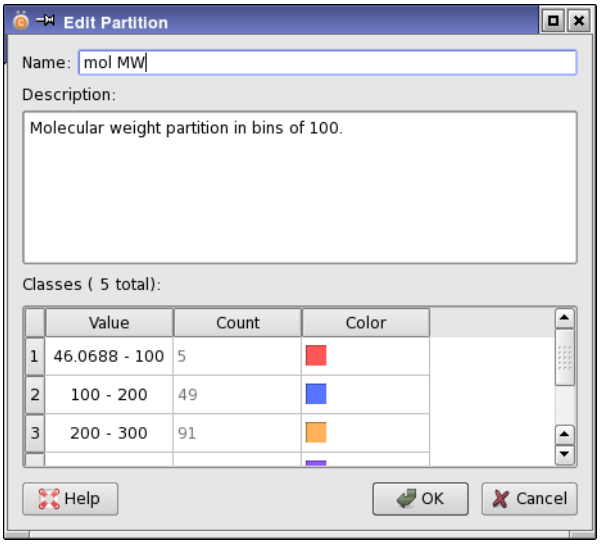


Figure 2.16. The Edit Partition dialog box.

2.5.3 Creating a Heat Map

To highlight the values of a property, you can apply a heat map for the property, which colors the cells in the spreadsheet according to the property value. The colors provide a visual organization of the property values. To set up the heat maps, choose Data → Heat Map, which opens the Heat Map dialog box.

You can choose the properties to which you want to apply a heat map, by selecting them in the Properties list and clicking Add. Properties to which you no longer want to apply a heat map can be removed by selecting them in the heat map table and clicking Remove.

Four color spaces are available:

- RGB—Color is interpolated linearly between the RGB colors for the maximum and minimum values.
- Rainbow (HSV)—Color follows the spectrum (rainbow) between the maximum and minimum values.
- Transparency—Transparency increases from the maximum value and the minimum value towards the mean value (which is white).
- Color Wheel (discrete)—A selection of colors from the spectrum is applied to individual integer values. For real-valued properties, the value is rounded down to the nearest integer and the color assigned based on the integer value.

Only one of these spaces can be used at a time, and is applied to all properties. You can set the colors for the minimum and maximum values of the property range, by right-clicking in the Min or Max column of the heat map table, and choosing Edit Color.

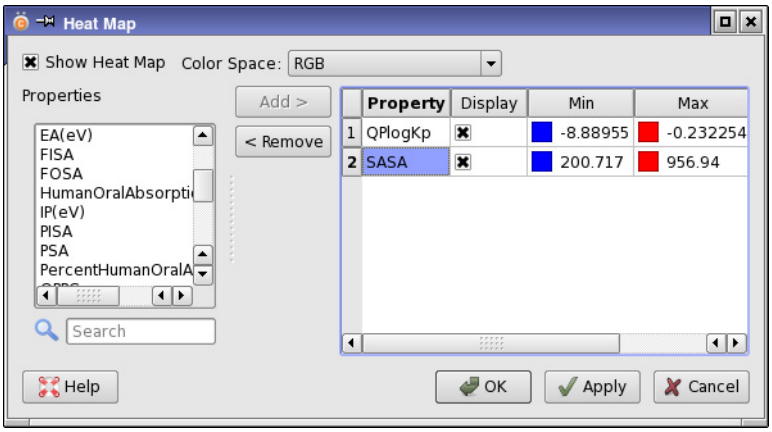


Figure 2.18. The Heat Map dialog box.

Overall display of the heat maps is controlled by selecting or deselecting Show Heat Map. Display of heat maps for individual properties is controlled by selecting or deselecting the check box in the Display column for the property.

The property ranges are divided evenly to create the heat map. If you want to customize the property ranges used for the colors, you can instead create a partition for the property and color the classes with the colors that you choose.

2.6 Filtering and Querying the Data

Canvas allows you to filter the data on the values of properties or on counts of functional groups, which are represented as SMARTS patterns. You can also run a query to find all molecules that have a particular substructure. Filters and queries can be stored and be reused. All types of filters and queries can be run from the Project View panel with default settings, using the shortcut menu.

2.6.1 Filtering by Property Values

You can filter the structures in the spreadsheet by a combination of property values. This task is performed in the Property Filter dialog box, which you can open by choosing Data → Property Filter or Query → Properties.

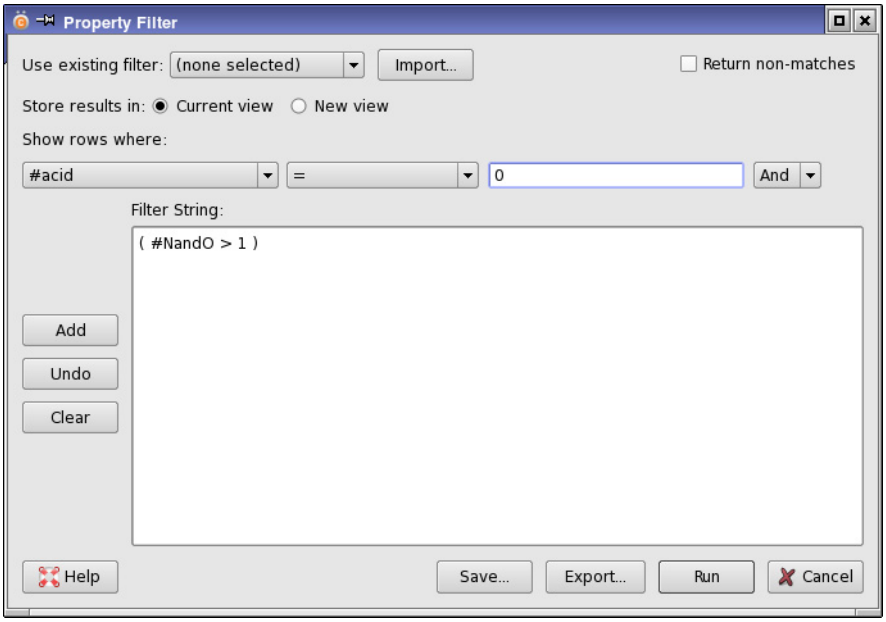


Figure 2.19. The Property Filter dialog box.

To construct a new filter, you first set up a condition on a property by selecting a property and a relational operator from the Show rows where option menus, and entering a value in the text box. The operator selection includes arithmetic operators, operators for matching strings, and operators for highest and lowest counts and percentages. Once the condition is defined, click Add to add it to the Filter String text area, where the entire filter string is displayed. To add more conditions, construct each condition in the same way, select And or Or to define its relationship to the previous conditions, then click Add. To remove the last condition added, click Undo. If you need to start again, click Clear, and the filter string is removed. The string can also be edited directly.

You can load, edit, and apply existing filters. To load a filter from the project, choose the filter from the Use existing filter option menu. The Filter String is replaced with the string for the selected filter. To load a filter from outside the project, click Import. A file selector opens, in which you can navigate to and choose the filter file (`.flt`).

When you have created a filter, you can save it in the project by clicking Save, and entering a name in the Save Property Filter dialog box. The filter is then listed on the Use existing filter option menu, and under Property Filters in the Project View window. You can also export the filter to a `.flt` file so that it can be used in other projects, by clicking Export.

To delete a filter, right click on it in the Project View window and choose Delete.

The filter can be applied to the current view, or the results can be stored as a new view. If you select New view, the Save Custom View dialog box opens when you apply the filter, so that you can name and describe the view.

To apply the filter, click Run. If you have not saved the filter, you are prompted to save it. If you click Yes, the Save Custom View dialog box opens; if you click No, the filter is applied without saving it.

To apply the inverse of the filter, select Return non-matches before applying the filter.

2.6.2 Filtering by Property Classes

In a similar way to filtering by property values, you can filter the data by the property classes stored in a partition. This feature allows you to perform more complex filtering on a single property. To do so, first open the partition, then choose Partition → Filter. In the Partition Filter panel, you can select the classes whose rows you want to show (or hide). To hide the selected classes instead of showing only the selected classes, select Hide rows for selected classes. You can also choose whether or not to hide rows that are not in any class with the Hide unclassified rows option. As for property filters, the filter is applied to the visible rows, so the number of rows never increases when you apply the filter. The filter can only be applied to the current view.

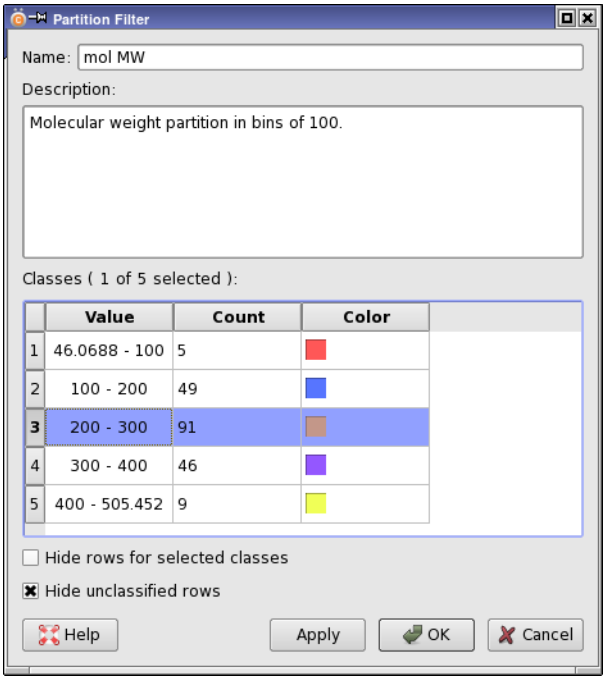


Figure 2.20. The Partition Filter dialog box.

If you click Apply, the panel remains open, and you can interact with the spreadsheet normally. This allows you to make different selections of classes and apply the filter. For example, if you want to display different classes in turn, you can select the first class and click Apply, then perform whatever actions you want to do in the spreadsheet. You can then select the next class and click Apply, and perform actions.

When you click OK, the most recent selection is applied and the panel closes.

2.6.3 Filtering by Structure

You can filter the structures based on their chemical content, which includes the usual functional groups. This task is performed in the Structure Filter dialog box, which you open by choosing Structure → Structure Filter.

Each condition on the occurrence of a chemical feature is specified by a row of the table. Complex filters can be set up by using multiple rows. The structures that pass all the filters are returned when you run the job. Thus, there is an implicit AND applied between the rows. You can invert the filter to return the structures that fail any of the filters, by selecting Invert logic (retain failures).

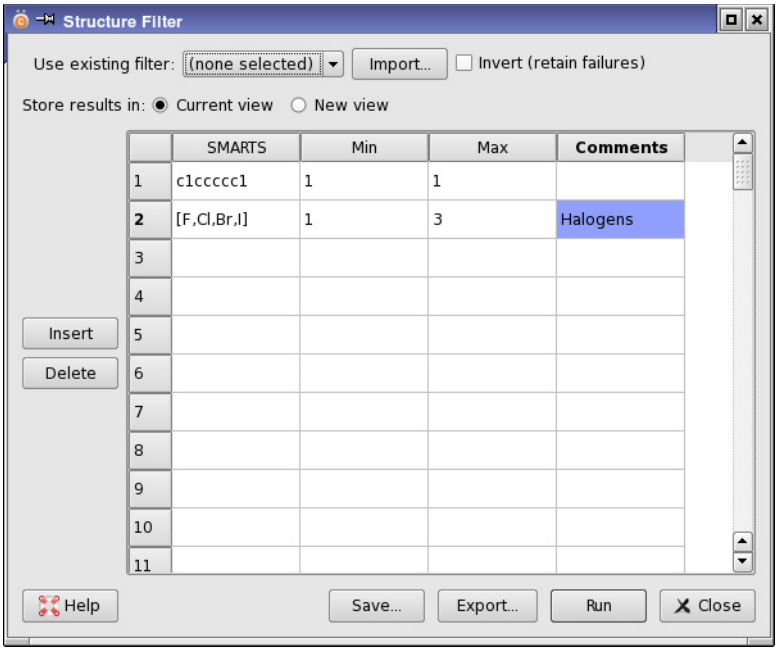


Figure 2.21. The Structure Filter dialog box.

The chemical features are defined in terms of SMARTS patterns. In the SMARTS column you can enter the SMARTS patterns to filter on, and in the Min and Max columns, you can enter the minimum and maximum number of occurrences of the SMARTS pattern in a structure that are required for the structure to pass through the filter. You must enter at least a minimum or a maximum, and you can enter both. In the Comments column you can annotate the SMARTS pattern—for example, to give names to functional groups. You can insert a row above the selected row by clicking Insert, and you can delete the selected rows by clicking Delete.

Structure files can be used to generate SMARTS patterns. To do this, import a file in Maestro, SD, or SMILES only format with the Import button. The structures in the file are converted into SMARTS patterns, and added to the rows of the table.

When you have created a filter, you can save it in the project for later use by clicking Save. A dialog box opens, in which you can name the filter. The filter is then listed on the Use option menu and under Structure Filters in the Project View window. If you want to store the filter outside the project, for example, to import into another project, click Export. A file selector opens, in which you can save the filter in a file with a `.flt` extension.

To delete a filter, right click on it in the Project View window and choose Delete.

You can also load, edit, and apply existing filters. To load a filter from the project, choose the filter from the Use option menu. The table is replaced with the data for the selected filter. To apply multiple filters, select Apply, then choose the filter from the option menu. The selected filter is appended to the patterns in the table. Four standard filters are available: REOS ("rapid elimination of swill"), PAINS1, PAINS2, and PAINS3 (pan assay interference filters [15] for different classes of assays, corresponding to filter C, filter B, and filter A, respectively). To load a filter from outside the project, click Import. A file selector opens, in which you can navigate to and choose the filter file (`.flt`).

The filter can be applied to the current view, or the results can be stored as a new view. If you select New view, the Save Custom View dialog box opens when you apply the filter, so that you can name and describe the view.

To apply the filter, click Run. If you have not saved the filter, you are prompted to save it. If you click Yes, the Save Custom View dialog box opens; if you click No, the filter is applied without saving it. If Highlight Matches is selected on the Structure menu, the substructures are highlighted on the structures that match.

2.6.4 Filtering by Substructure

You can filter the structures in the database by running a substructure query. This task is performed in the Substructure Query dialog box, which you open by choosing Query → Substructure Query or clicking the Substructure Query toolbar button.

The query can be run on the selected rows in the spreadsheet, on all rows, or on an external project. To make the choice, select the appropriate Screen option.

By default, the query returns all structures that contain the specified substructure. If you want to return structures that match the query exactly, select Require exact match. This choice will find all instances of a single molecule, and distinguish it from structures that contain other molecules, such as water. You can also return structures that do not match the query by selecting Invert (return non-matches). If you are querying an external project, the matching structures are appended to the current project.

The query can be specified either by providing a SMARTS string, or by building or importing a 2D structure.

- To use a SMARTS string, choose SMARTS from the Query Source option menu. You can enter or paste a SMARTS string into the Enter SMARTS query text box, or choose a previously used SMARTS string from the option menu.

- To use a 2D structure, choose Builder from the Query Source option menu. You can build a structure, by clicking New and drawing the structure in the Edit Structure panel; import a query from a MOL file or an SD file, by clicking Import and navigating to the file; or select a previously used query from the option menu. Only the first structure is used if an SD file is chosen.

The display area shows the query as a 2D structure. You can set the number of queries on the option menus in the Preferences panel, in the History category.

By default, the structures are prescreened with the 2D index for the structure (a special 512-bit fingerprint), to skip structures that cannot match the query, before checking for a match. This feature speeds up the search. It can be used for prescreening if the query is an exact SMARTS pattern; it cannot be used if the query contains SMARTS primitives that disqualify it as an exact substructure, such as [C,N], or [*]. To turn off the prescreening, deselect Use 2D index.

You can create or edit a query with the 2D builder or with ChemDraw (if ChemDraw is installed) by clicking the Build button in the Mol File tab. If you have already specified a file, it is loaded into the 2D editing application. When you close the application, the query is saved and imported into Canvas. You have the option of saving the query in a named file for future use.

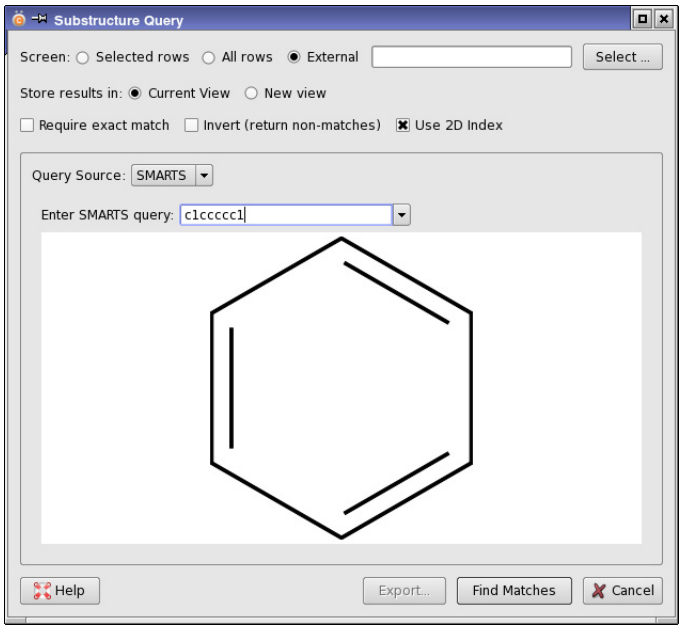


Figure 2.22. The Substructure Query dialog box.

The query results can be applied to the current view, or the results can be stored as a new view. If you select New view, the Save Custom View dialog box opens when you run the query, so that you can name and describe the view. The matches are highlighted on the structure if Structure → Highlight Matches is selected.

To run the query, click Find Matches.

When you have run the query, you can align the 2D structure of the matches, as shown in the spreadsheet, to a reference structure of your choice. The alignment is done with a least-squares method on the matching atoms. The reference structure is the first match by default, but you can choose any single matching structure as the reference. To perform the alignment, choose Structure → Align Matches. If you deselect this option, the matching structures are returned to their default orientation. The alignment is removed if you delete the reference structure, modify it so that it no longer matches the query, or hide it in a custom view and then save the view. or if you undo the query filter. If you want to align to a different structure, deselect the Align Matches option, then choose the new reference structure and select the option again.

2.6.5 Detecting Duplicate Structures

There is no restriction in Canvas on the structures that you can import, other than any filters you apply in the process of importing. This means that the project can contain duplicates. If you want to locate the duplicates, you can do so by choosing Structure → Detect Duplicates. A column is added to the spreadsheet, named Duplicate of. Values are assigned only for the second and subsequent occurrences of a structure. The value is the UID of the first occurrence of that structure (the one with the lowest UID).

You can use this column to select the unique structures or the duplicates with a property filter. In the Property Filter dialog box, choose the Duplicate of property under Show rows where, and choose has no value for the operator to select the unique structures. To select the duplicates, choose has a value for the operator.

2.7 Making Charts of the Data

Canvas provides several kinds of charts for visualizing the data: scatter plots, histograms, and frequency pie charts. To create a chart, choose the chart type from the Chart menu, or click the toolbar button for the chart type.

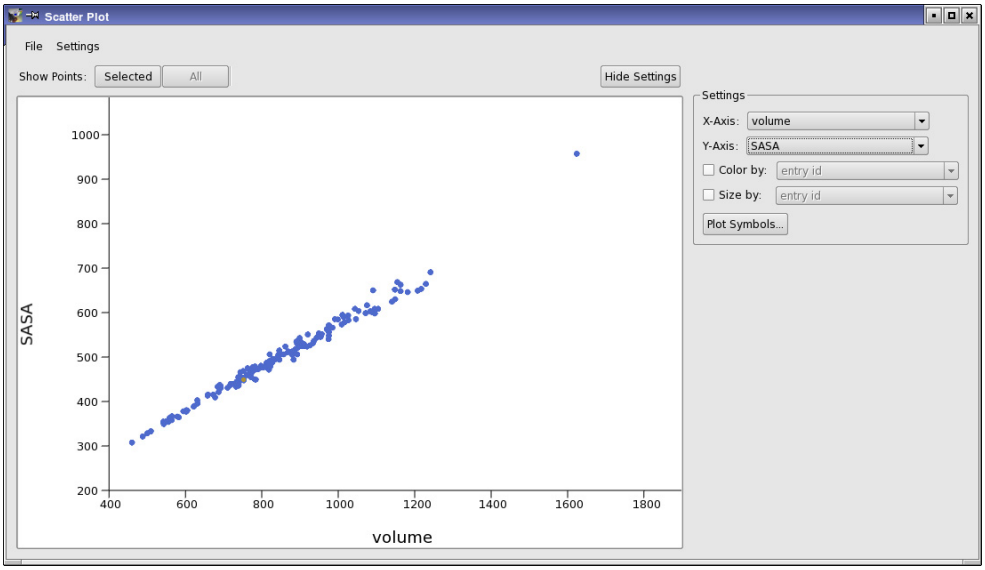


Figure 2.23. The Scatter Plot panel.

Each chart panel has a File menu, a Settings menu, and one or more option menus to choose the properties that are used for the chart. The properties listed on the option menus are the properties that are selected in the current view.

2.7.1 Scatter Plots

Scatter plots can be used to represent up to four different variables. The default plot shows two variables, on the x and y axes. The other two variables can be represented by the color and the size of the plot symbols. The property values for each point are displayed in a tool tip.

The properties that are displayed can be chosen from the X-Axis, Y-Axis, Color by, and Size by option menus in the Settings section of the panel. If this section is not displayed, click Show Settings. The color and size properties can also be chosen in the Plot Symbols dialog box, which you open by choosing Settings → Plot Symbols. To make your choice, select By property and choose the property from the option menu. You can then enter the limits of the range in the Limits text boxes. For coloring by property, you can set the colors for the ends of the property range with the Min and Max buttons. The colors between the end points are a linear interpolation in RGB color space. For size by property, you can set the minimum and maximum point size of the plot symbols. This dialog box also allows you to set the color and point size of the plot symbols if you do not want to map them to properties. To do this, choose Fixed for Color and Size, and set the color or the point size.

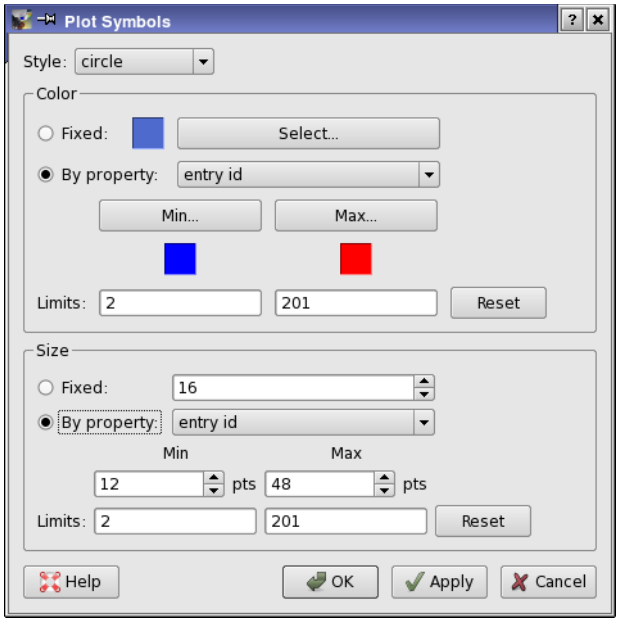


Figure 2.24. The Plot Symbols dialog box.

To configure the axes, choose Settings → Axes. The configuration options are described in Section 2.7.4 on page 52.

Scatter plots can be used to interact with the view. If you select points in the scatter plot, the corresponding rows in the view are selected. Likewise, if you select rows in the view, the corresponding points in the scatter plot are selected. You can select points by clicking or dragging. Shift-clicking or shift-dragging adds to the selection without removing any points; control-clicking or control-dragging changes (inverts) the selection of points without affecting the selection of other points.

Once you have selected some points, you can show only those points in the chart by clicking Selected. If you then change the selection, click Selected again to show only the new selection. To redisplay all points, click All.

You can also label the selected points, by right-clicking and choosing Show Labels from the shortcut menu. The labels include the structure and title, and the names and values of the properties that are represented by that point (x, y, color, size). The contents change when any of the properties plotted changes. The labels can be dragged to any point in the plot. They are connected with the plot point by a line. To hide individual labels, click the X in the label. To hide all the labels, choose Hide Labels from the shortcut menu.

2.7.2 Histograms

Histograms show the frequency of occurrence of values of a single variable, which you can choose from the Property option menu. The variable can be numeric, and the bars defined in terms of ranges of the variable; or it can be categorical, and the bars record the frequency of occurrence of the values of the variable. String variables and partitions are always treated as categorical, and real variables as numeric. Integer variables can be treated as either, by selecting Property is numeric or Property is categorical. If they are treated as categorical, the bars represent the frequency of occurrence of each integer value.

The bars can be shown as frequency counts or as percentages. To set up the histogram bars, choose Settings → Bars. To configure the vertical axis, choose Settings → Y Axis. Many of the settings are common to other plots, and are described in Section 2.7.4 on page 52.

Clicking on a bar selects the table rows that have the property values covered by the bar. The bar is highlighted with a pattern. The proportion of the bar that is patterned represents the fraction of rows in that bar that are selected in the spreadsheet. You can add a bar to the selection with shift-click, and you can change the selection of a bar with control-click. To clear the selection, click in an empty part of the chart.

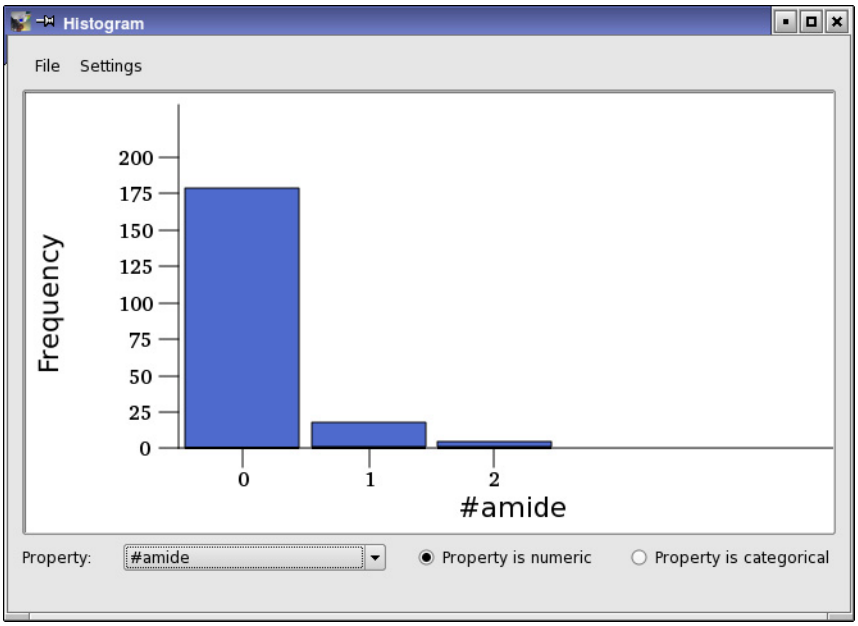


Figure 2.25. The Histogram panel.

2.7.3 Pie Charts

Pie charts also show the values of a single variable. You can choose the property that is represented from the Property menu. The variable can be numeric, and the sectors defined in terms of ranges of the variable; or it can be categorical, and the sectors record the frequency of occurrence of the values of the variable. String variables and partitions are always treated as categorical, and real variables as numeric. Integer variables can be treated as either, by selecting Property is numeric or Property is categorical. If they are treated as categorical, the sectors represent the frequency of occurrence of each integer value.

Clicking on a sector selects the table rows that have the property values covered by the sector. The sector is highlighted with a pattern. The proportion of the sector that is patterned represents the fraction of rows in that sector that are selected in the spreadsheet. You can add a sector to the selection with shift-click, and you can change the selection of a sector with control-click. To clear the selection, click in an empty part of the chart.

To set the ranges represented by the chart, the legend, and other options, choose Settings → Sectors. The settings are described in Section 2.7.4 on page 52.

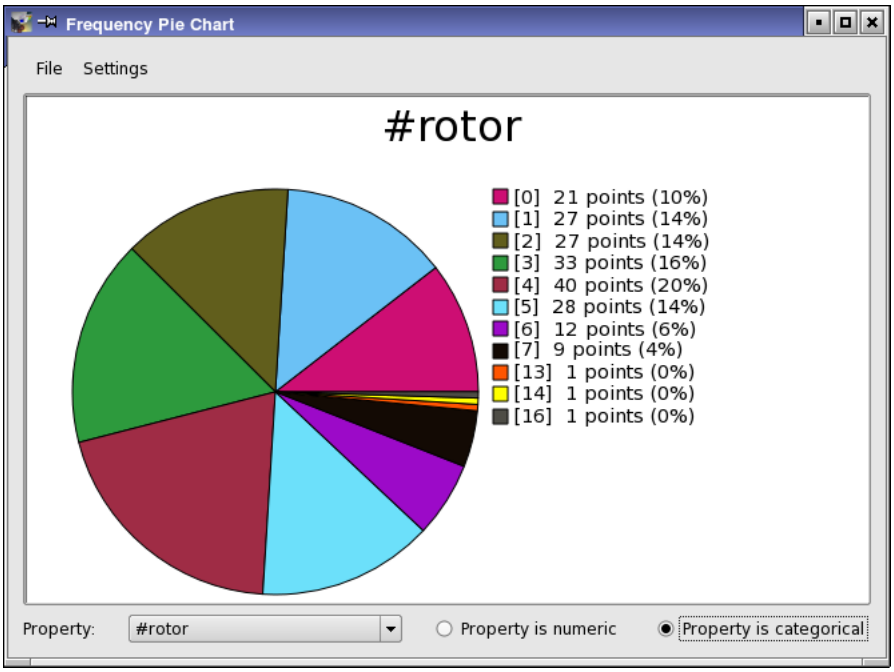


Figure 2.26. The Frequency Pie Chart panel.

2.7.4 Chart Settings

The common settings for each chart are made in the General dialog box, which you open by choosing Settings → General in the chart panel. You can set the title of the chart and change its font size and color, and set the background and foreground colors for the chart.

For scatter plots and histograms, you can change various attributes of the axes. To make these settings, choose Settings → Axes in a scatter plot panel or Settings → Y Axis in a histogram panel. Most of these settings are common to the two chart types:

- Minimum and maximum values
- Axis line color and weight
- Tick mark spacing, start value, color and weight
- Tick mark label spacing, start value, font size, color, and number of decimals
- Axis title, font size, and color.

For scatter plots, you can decide where the y axis line crosses the x axis, and display or hide grid lines. For histograms, you can choose whether to display raw counts or percentages.

For histograms, the horizontal (x) axis can be configured by choosing Settings → Bars. The dialog box depends on whether the property is numerical or categorical. You can set the following attributes for numeric properties:

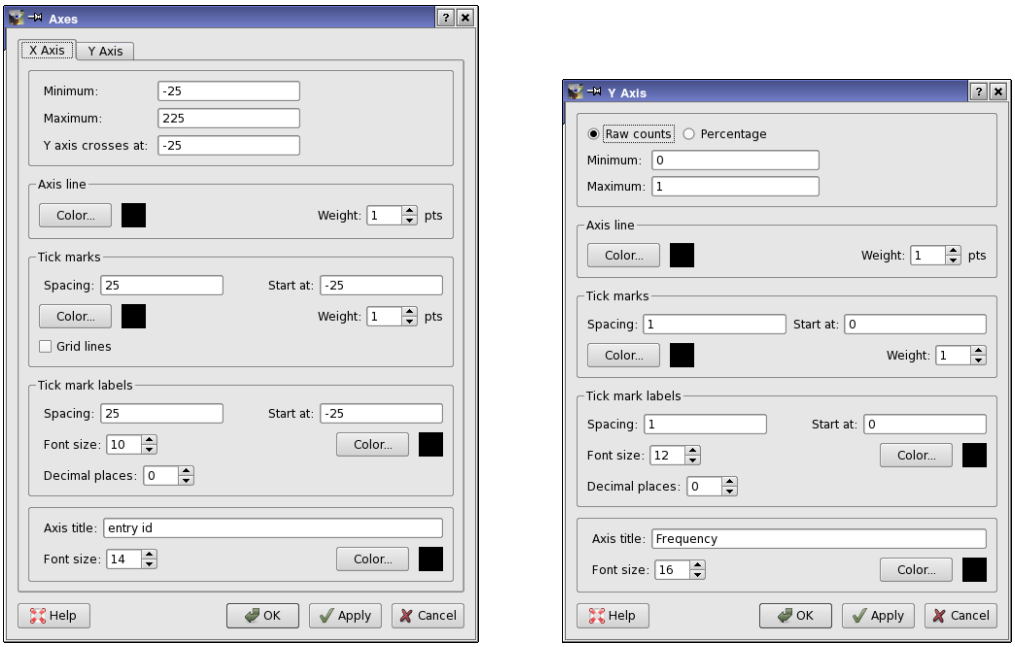


Figure 2.27. Axis settings dialog boxes.

- Axis line color and weight
- Tick mark color, weight, and centering
- Tick mark label font size, color and number of decimal places
- Axis title, font size, and color
- Bin width, lower and upper limits of the data
- Bar width relative to the data range, bar color

For categorical properties, you can set the following attributes:

- Axis line color and weight
- Tick mark color and weight
- Tick mark label font size and color
- Axis title, font size, and color
- Bar width and color

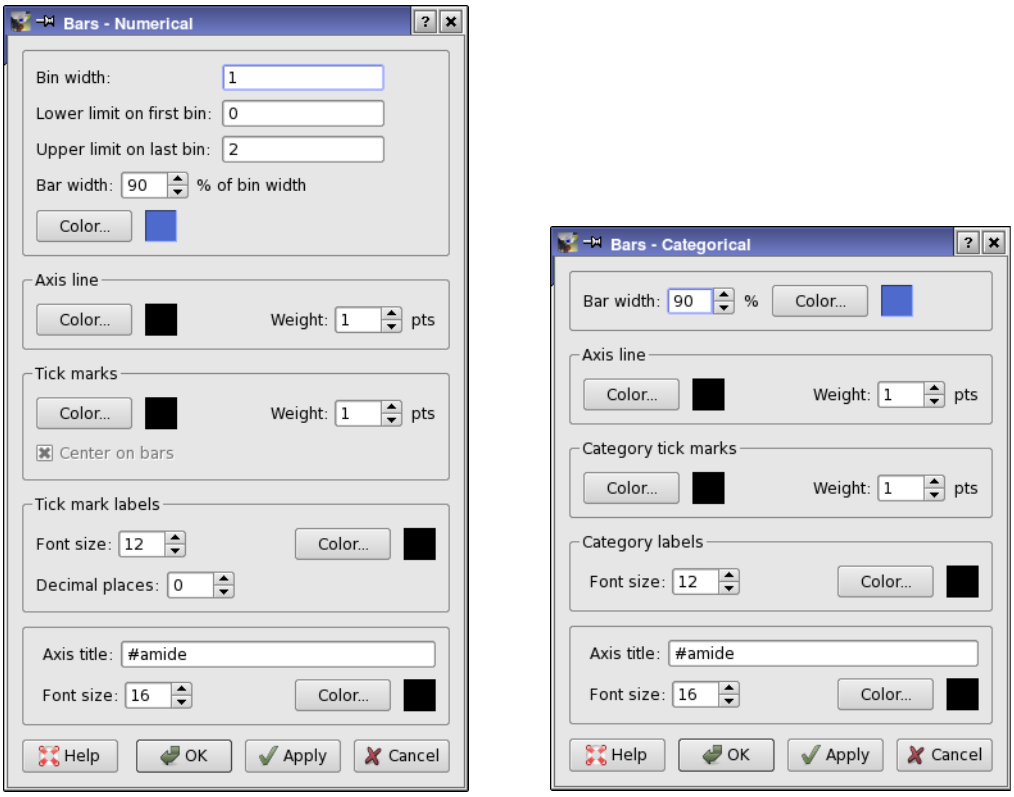


Figure 2.28. The Bars dialog box for numerical and categorical properties.

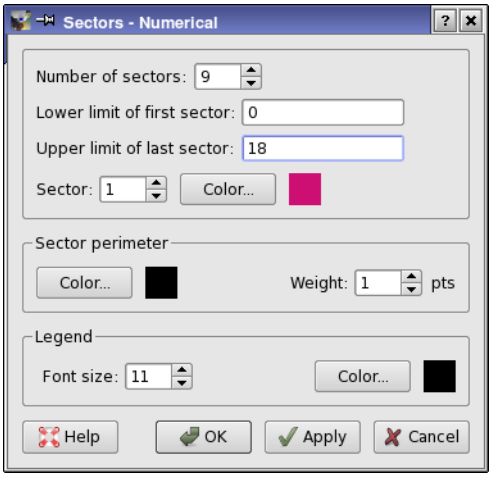


Figure 2.29. The Sectors dialog box.

For pie charts, you can make settings for the sectors and the legend, by choosing Settings → Sectors, which cover the following attributes:

- Number of sectors and lower and upper limits for the sectors (numeric only)
- Color of each sector and color and weight of the perimeter
- Legend font size and color.

2.7.5 Saving Images of Charts

You can save an image of any chart by choosing File → Save Image in the chart panel. A file selector opens, in which you can navigate to a location, name the image, and choose the image format. The available formats are PNG, TIFF, and JPEG.

2.8 Calculating Statistics

You can compute various univariate and bivariate statistics for properties in the Statistics dialog box, which you open by choosing Data → Statistics.

The statistics are calculated for the row and column selection of the current view if you select Selected Data, or for all rows and columns if you select All Data. When the statistics are calculated, you can select Ignore properties with missing values if you want to exclude such properties from the calculation (and the results). If you do not exclude them, a warning is posted and the results table contains question marks (?) for this property.

Univariate and bivariate statistics are computed and displayed separately. You can compute multiple univariate statistics by selecting them from the list in the Univariate tab, and clicking Compute. The results are displayed in a table. If you make a new selection, the statistics that were not previously computed are computed and added to the table.

In addition to the statistics, a Count column is included in the table that lists the number of values that are set for the property. This enables you to identify properties that do not have values for all rows in the spreadsheet.

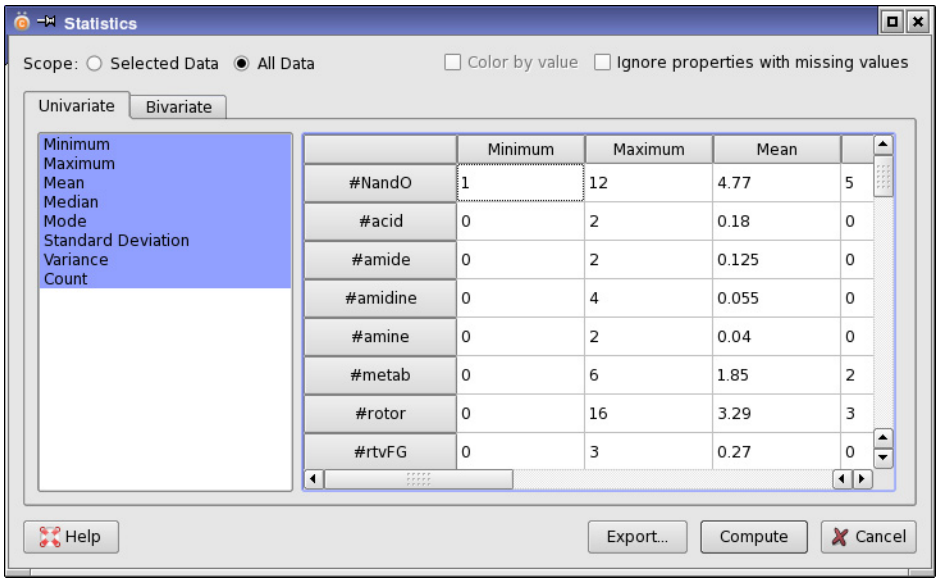


Figure 2.30. The Univariate tab of the Statistics dialog box.

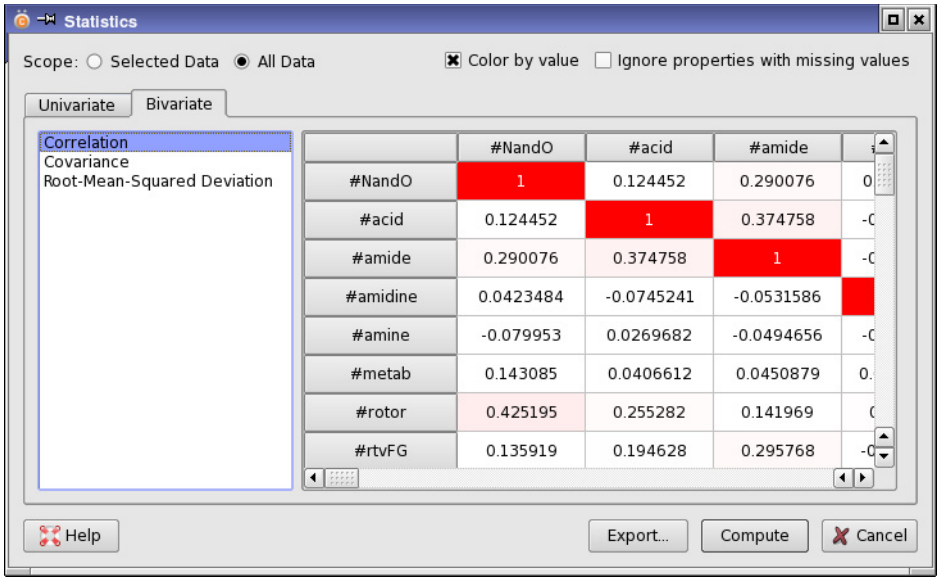


Figure 2.31. The Bivariate tab of the Statistics dialog box.

You can compute one bivariate statistic at a time, by selecting it in the list in the Bivariate tab and clicking Compute. The results are displayed in a table. If you select a statistic that has been previously computed in this session, the results are displayed without having to click Compute. For bivariate statistics, you can select Color by value to color the cells by value, using a blue-white-red color map. You can also display a scatter plot of the two variables (properties) used to compute the bivariate statistic by clicking the table cell.

To export the statistics in the current tab to a CSV file, click Export. A file selector opens, in which you can navigate to the location and name the file.

2.9 Calculating New Properties from the Data

Canvas provides a calculator that can be used to combine properties by constructing a formula. The calculator provides mathematical functions as well as basic arithmetical operations. It is designed to generate entire new properties from existing properties by operating on property columns, rather than to operate on individual cells, as in a spreadsheet program like Microsoft Excel or Open Office Calc. To open the calculator, choose Data → Calculator.

The calculation can be performed on all rows, or on the selected rows. If you want to use the selected rows, you must select them before opening the dialog box. The result is stored in a new property, for which you must provide a name in the Create column text box.

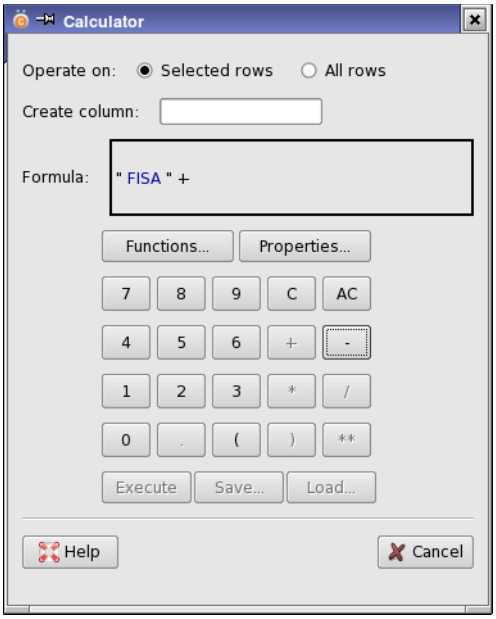


Figure 2.32. The Calculator dialog box.

To construct a formula, use the keys on the keypad, which are set out like a traditional calculator, and the Functions and Properties buttons. The Functions button opens a dialog box, in which you can choose a mathematical function. The Properties button opens a dialog box, in which you can choose an existing property from the spreadsheet.

When you have finished constructing the formula, click Execute to perform the calculation and add the new property to the spreadsheet. If you chose to operate on the selected rows, values for the unselected rows are not defined.

You can save a formula for later use by clicking Save. A dialog box opens, in which you can name the formula. To use a saved formula, click Load, and select the formula in the Formulas dialog box. This dialog box also allows you to delete saved formulas.

2.10 Viewing Structures in PyMOL

The Canvas spreadsheet only displays structures in 2D. To view them in 3D, you would normally have to export them and then import them into a 3D viewer (like Maestro). Canvas provides an interface to PyMOL that allows you to select structures in the Canvas spreadsheet and display them directly in PyMOL. To do so, the location of the PyMOL executable must be included in your `PATH` environment variable. For information on setting this environment variable on Windows, see Appendix A of the *Installation Guide*.

When you choose Structure → Show Structures in PyMOL, the PyMOL interface is opened, and check boxes are added in the lower left corner of the structure cells in the spreadsheet. Selecting a check box sends the structure to PyMOL for display. You can select multiple check boxes with the usual shift-click and control-click actions, and the structures are displayed in PyMOL.

Custom views have an independent set of checkboxes that can be selected and deselected. When you create a new view, the view inherits the structure selection for PyMOL from the parent view. When you filter a view (master or custom), any structures that are filtered out are not considered to be selected for viewing.

The structures that you see in PyMOL are a combination of the structures that are marked for display in all open views. That is, the displayed structures correspond to rows that are visible in any of the open views and are checked for viewing in PyMOL. So if a particular structure is selected and visible in two open views, deselecting it in one view does not remove it from PyMOL, because it is stile selected and visible in the other view.

To finish viewing structures in PyMOL, choose Structure → Close PyMOL.

If you want PyMOL to open automatically when you open a project, you can set a preference in the Preferences dialog box, which you open with File → Preferences. Click General in the list on the left, then select Start PyMOL.

2.11 Setting Preferences

Some customization of the interface features and other settings can be made in the Preferences dialog box, which you open with File → Preferences. The panel has a list of preference categories on the left. When you select an item from this list, the preferences in that category are displayed on the right. If you want to restore the default preferences, click Reset. The preferences in each category are described below.

General preferences

This category has two settings.

- Start PyMOL—Start a PyMOL session when a project is opened, so that you can view structures in 3D. If this option is not selected, you can start PyMOL at any time from the Structure menu.
- Canvas max memory—Set the maximum memory that can be used by Canvas, in bytes. You can use commas in the value, e.g. 1,000,000. You can use suffixes for kilobytes, megabytes, and gigabytes, in any form, case-insensitive, e.g. 2 Mb, 2 Mbytes, 2 megabytes all mean 2097152 bytes. This limit is used by any applications that are started from Canvas. Sets the `SCHRODINGER_CANVAS_MAX_MEM` environment variable for the Canvas session.
- Incorporate jobs automatically—When a job finishes, incorporate the results immediately into the Canvas project. The project is unavailable while the incorporation takes place. This preference sets the default incorporation option in the Start Application dialog box.
- Number of decimal digits to display—Set the number of digits past the decimal point to display in the spreadsheet. This value applies to all properties.

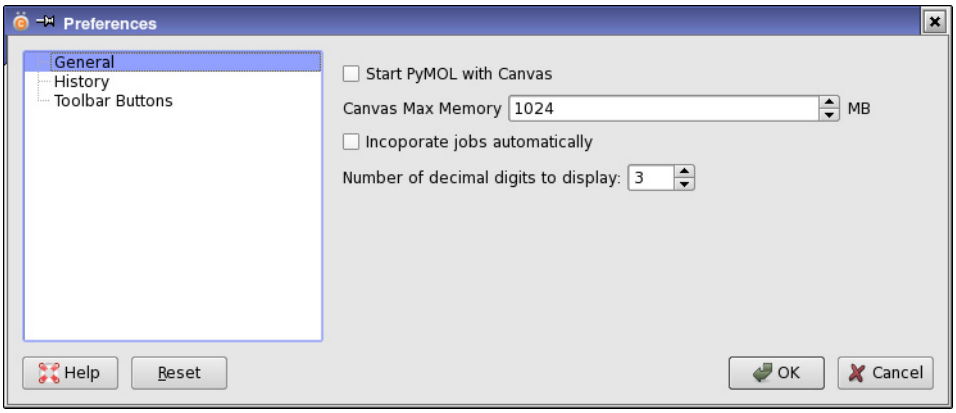


Figure 2.33. The Preferences dialog box.

History preferences

This category allows you to set preferences for the number of items stored in various history lists.

- Recent project list—Specify the number of projects to show on the Recent Projects submenu of the File menu. The default is 10. Click Clear to clear the project history list.
- Substructure query list—Specify the number of SMARTS queries or query files to show in the Substructure Query dialog box, in the lists that are displayed for the Query File text box or the SMARTS Query text box.

Toolbar buttons preferences

With these preferences, you can check or clear the boxes for the toolbar buttons that you want to have on the File toolbar, or click Select All to check all the boxes. The buttons are added or removed when you click OK in the dialog box. The buttons are described in Section 2.2.2 on page 7.

Running Applications from Canvas

The Canvas interface provides access to a range of applications for molecular descriptors, fingerprints, similarity and clustering, calculating statistics, and building regression and other predictive models based on properties of the structures. The applications can only be run from the master view.

3.1 General Features

Nearly all of the applications run on a data set that includes both structures and properties. The Data set option menu is used to choose the data set. You can choose the selected rows or the visible rows in the master view, which makes all properties in the spreadsheet available for selection. You can choose to use a saved view and select the view from an option menu; the rows or columns or both can be restricted to those of the saved view by choosing from the Domain option menu, while the other rows or columns are taken from the visible part of the master view. If you are examining the results of a job or cloning a job, you can choose the original data set from that job to start a new job.

Some applications require the selection of multiple properties. These panels contain property selection tools, which consist of two lists, Available properties, and Selected properties, a search box to restrict one or other list to properties that contain the text entered in the box, and buttons for transferring the properties that are selected in one list or all properties to the other list. The Selected properties list may have a different name in some panels.

For applications that are run under Job Control, when you click the action button (usually Compute) in the application panel, the Start Application dialog box opens. This dialog box allows you to choose the host on which to run the application, and the number of CPUs, for applications that can be distributed across CPUs. It also allows you to choose whether to incorporate the results immediately into the project or to wait until you explicitly incorporate the results. The default incorporation behavior can be set in the Preferences panel (File → Preferences). For information on setting up the host list and access to hosts, see Chapter 7 of the *Installation Guide*. The job is run under the Job Control facility, which is described in detail in the *Job Control Guide*. If you want to bypass Job Control, you can do so from the command line—see Chapter 5.

The default amount of memory used by the job is about 500 MB. You can change this value in the Preferences dialog box (File → Preferences), in the General category.

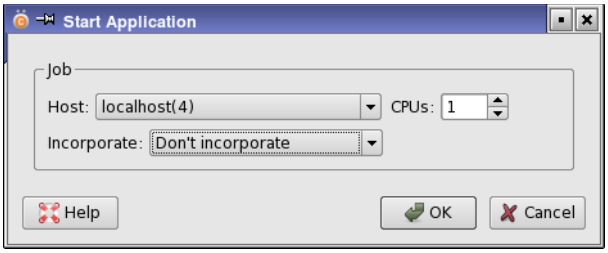


Figure 3.1. The Start Application dialog box.

When you run an application, messages about its progress are displayed in the Messages View panel, and a record is added under the application in the Project View panel. The record is color-coded by job status. When a job finishes, the application node is expanded. The results are not automatically incorporated into the spreadsheet unless you set the preference to do so or selected the option to do so in the Start Application dialog box. To add them, display the jobs for the application in the Project View panel, right-click on the record and choose Incorporate. If you want to be notified when a job finishes, select Notify when jobs complete in the Project View panel.

To run another job with the same options or using the job's options as a basis, right-click on the job in the Project View panel and choose Clone Job. The Clone Job dialog box opens, and offers an option to use the original data set. When you click OK, the application dialog box opens with the options set as they were for the job you cloned. You can then modify the options and start the job.

Some applications have panels for viewing and using the results of the job. For these applications, right-click on the job in the Project View panel and choose View.

3.2 Molecular Properties

Molecular properties are used as the basis of modeling applications such as QSAR or QSPR. Canvas provides a means of calculating a wide range of molecular properties and descriptors based on the 2D structure. 3D properties can be generated by other applications and imported into Canvas. With a large number of properties it is often necessary to reduce the number of properties to a relatively independent set. This section describes applications for molecular property generation and molecular property selection.

3.2.1 Calculating Molecular Properties

Canvas can calculate a range of molecular properties from the 2D structures that can be used as descriptors. Some of these are simple counts of features, others are numerical properties evaluated from a set of rules. The calculation of these properties can be done in the Molecular Properties dialog box, which you open by choosing Applications → Molecular Properties.

You can choose whether to calculate the properties for the selected rows or the visible rows, a saved view, or the data set from a previous run, and choose a name to use to save the results in the project.

There are four classes of properties, Physicochemical Descriptors, Topological Descriptors, LigFilter Descriptors, and QikProp Descriptors, which you can select by choosing the class from the tree on the left. The property selection tools are displayed on the right. For each class of properties, you can select a host and a number of CPUs for the job. The jobs for each class run independently and are incorporated independently. The job status is listed in the Project View under the class name.

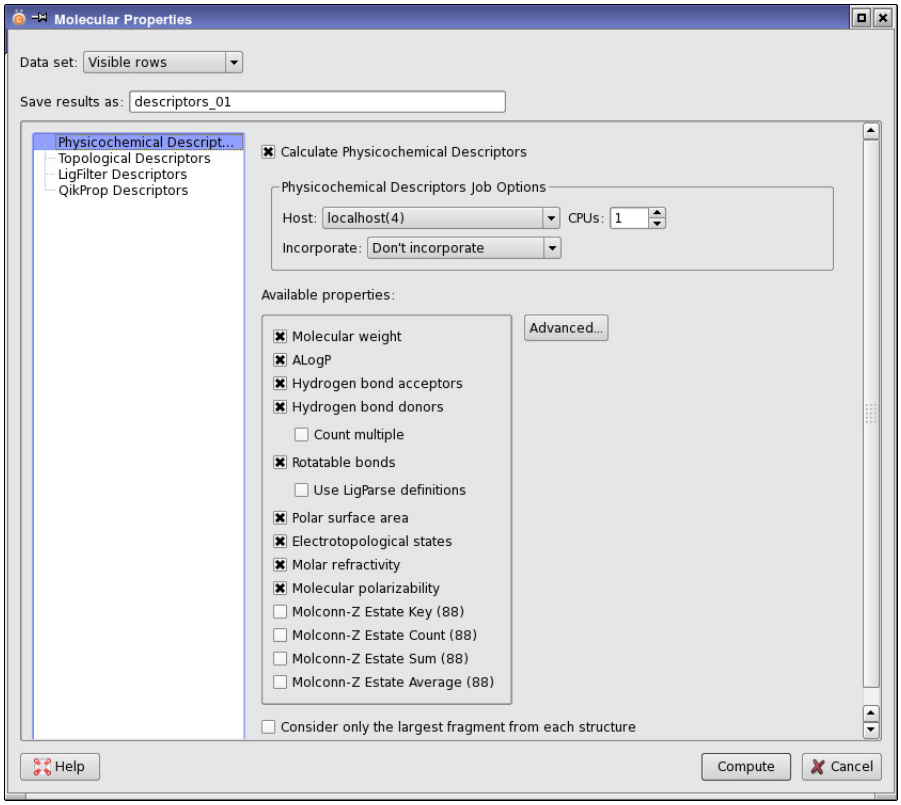


Figure 3.2. The Molecular Properties dialog box.

The physicochemical descriptors that can be calculated are:

Molecular weight
AlogP [9]
Hydrogen bond acceptors
Hydrogen bond donors
Rotatable bonds
Polar surface area [13]
Electrotopological states (Estates) [8]
Molar refractivity [10]
Molecular polarizability [11]
Molconn-Z Estate Key [12]
Molconn-Z Estate Count [12]
Molconn-Z Estate Sum [12]
Molconn-Z Estate Average [12]

References to methods are given in square brackets. Each of these corresponds to an option in the Available properties section. By default all but the Molconn-Z Estate properties are selected. The latter correspond to 88 properties for each of the options.

For the hydrogen bond acceptor and donor counts and the rotatable bond counts, you can specify alternative definitions. The definitions used by LigParse (in LigPrep) for rotatable bonds are available as an option, Use LigParse definitions. The default (Canvas) definitions differ from the LigParse definitions by excluding amide bonds and bonds that connect to a terminal heavy atom (one that otherwise only has H atoms attached, such as methyl, amino, or hydroxyl). You can also count groups that can act as donors of more than one H-bond as multiple donors, e.g. an NH_2 group would be counted as two donors. For all three counts, you can specify custom definitions in the Molecular Properties - Advanced Options dialog box, which you open by clicking Advanced. To use a custom definition, select the appropriate option and click its Browse button to load the definition file. The file must contain a list of SMARTS patterns for the feature. More information on creating custom definitions is given in Section 5.5.3 on page 156.

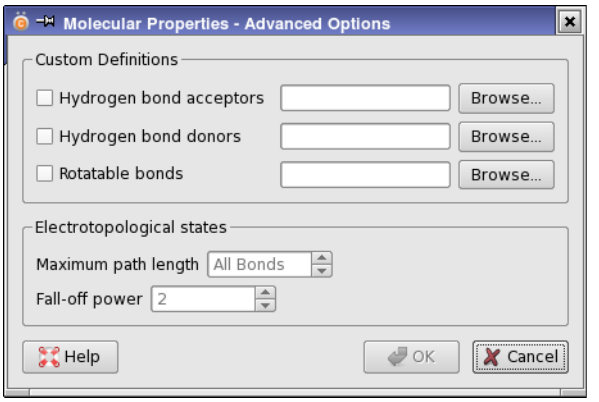


Figure 3.3. The Molecular Properties - Advanced Options dialog box.

When computing the sum and average for the Estates, you can set the maximum path length and the fall-off power in the Molecular Properties - Advanced Options dialog box.

If any of the structures contains more than one molecule (fragment), you can limit the property calculation to the largest fragment by selecting Consider only the largest fragment from each structure.

The topological descriptor list is long, so a property selection tool is provided in which you can search for properties in the list, and then transfer them from the Available properties list to the Properties to calculate list.

The LigFilter descriptor list is also long, but it is broken into sections with a tree tool. For more information on the LigFilter descriptors, see Section 2.4 of the *General Utilities* manual.

The QikProp descriptors are described in Chapter 1 of the *QikProp User Manual*. By default, "fast mode" is used, in which the dipole moment, ionization potential and electron affinity are not calculated, as they require a PM3 semiempirical quantum mechanical calculation. To include these properties, select Compute PM3 properties.

3.2.2 Selecting Representative Properties

Often, the set of molecular properties or descriptors that you obtain from various sources can have a high degree of correlation. For model-building, it is useful to remove the linear dependence in the properties, to speed up the calculation and to reduce numerical error. One way of doing this is to select a smaller set of properties that adequately represents the large set, in some measurable way.

Canvas provides the ability to select a representative set of properties. The set is chosen by clustering the properties with a hierarchical clustering method, based on the correlation matrix. The distance between two properties is calculated as one minus the absolute value of the correlation matrix element. A default number of clusters is chosen using Kelley criteria, and the property that is nearest the centroid of each cluster is chosen as the representative property. The job to perform the clustering is set up in the Feature Selection dialog box, which you open from the Applications menu.

To set up the job, choose a structure source from the Data set option menu, enter a name for the results, and choose the property set that you want to select from. You should check the list for properties that are not relevant, such as an entry ID or possibly results from other jobs. The list of properties that you choose can be reduced by removing those that have mostly identical values. To do this, you can select Exclude properties whose values are identical in more than *N* % of molecules, or Exclude properties with Shannon entropy lower than *X*. Click Compute to run the job.

When the job finishes, you can view the results in the Feature Selection Viewer panel, which you open by selecting View from the shortcut menu for the feature selection job, under Feature Selection in the Project View panel. This panel lists the original set of properties and the reduced (representative) set of properties, and provides tools to change the selection of representative properties.

To determine how well the selected properties (reduced list) represents the original properties, multiple linear regression is performed for each property to calculate an R^2 value for the fit of the property by the reduced list. First, a principal components analysis is done on the reduced list and principal components are selected to account for 95% of the variance. The X variables in the PLS model are taken from the projection of the principal components onto the property data from the reduced list, and the Y variable is the original property. These R^2 values are reported in the second column of the Original properties table.

If when setting up the job, you chose to remove properties for which a specified percentage of the values were identical, these properties are dimmed in the Original properties list, and the cells in the R^2 column are colored yellow and contain text describing the elimination criterion rather than a numeric value. Likewise, cells for properties that did not meet the Shannon entropy cutoff are colored magenta, with descriptive text. If you did not choose to remove such properties, the cells in the R^2 column in which all values of the property are identical are colored red and contain the text “Zero variance”.

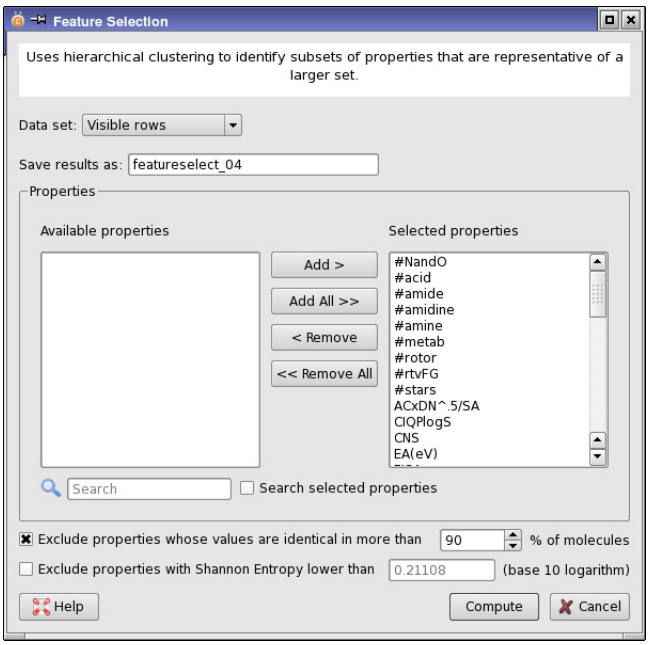


Figure 3.4. The Feature Selection dialog box.

Properties on the original list that are also on the reduced list are dimmed, and have a line of - characters instead of a numeric value in the R^2 column. Properties that are manually transferred between lists (as described below) are colored cyan.

You can change the number of properties in the reduced list in several ways:

- Select Number of clusters and change the number of clusters, then click Update Properties to update the lists and the regression results.
- Select Maximum correlation and change the value of the maximum allowed correlation between properties on the reduced list, then click Update Properties.
- Select a property in either of the lists and click Add or Remove, then click Update R^2.

You might want to change the property manually if you have a preference for a particular property. The property chosen from each cluster is the one that is closest to the centroid. You might want to replace it with another property from the same cluster. To do this, it is useful to sort the list of original properties by the clustering order (right-click and choose Sort by Cluster), so that properties in the same cluster are adjacent. It might also be useful to view the correlation matrix (click Correlation Matrix) for the original properties, ordered by cluster, so that you can identify the properties that are in the same cluster as the one you want to replace. With cluster ordering, the heat map of the correlation matrix should have red diagonal blocks that mark the clusters.

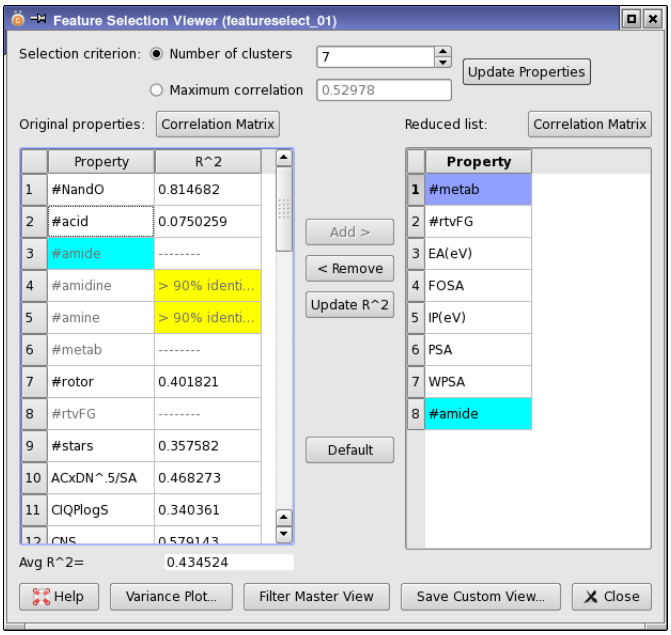


Figure 3.5. The Feature Selection Viewer panel.

3.3 Fingerprints

Canvas can calculate a variety of fingerprints from the 2D structure, both hashed fingerprints and structural keys, which are used by other Canvas applications. Canvas can also calculate fingerprints from 3D all-atom structures, based on 3-point or 4-point pharmacophore models. Once you have a fingerprint, you can create modal fingerprints [6] (averages over several structures) using the shortcut menu for the fingerprint property—see page 21 or page 117 for details.

3.3.1 2D Fingerprints

Binary fingerprints from the 2D structure are calculated using the Binary Fingerprints from Structures dialog box, which you open by choosing Applications → Binary Fingerprints from Structures.

You can choose whether to calculate the properties for the selected rows or the visible rows, a saved view, or the data set from a previous run, and choose a name for the new fingerprint column. If any of the structures contains more than one molecule (fragment), you can limit the property calculation to the largest fragment by selecting Consider only the largest fragment from each structure.

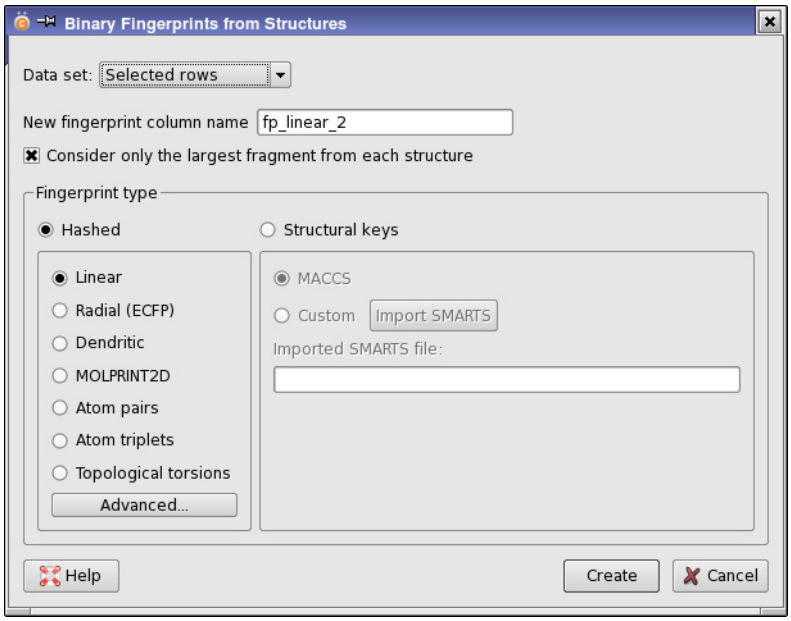


Figure 3.6. The Binary Fingerprints from Structures dialog box

To calculate a hashed fingerprint, select Hashed, then choose the particular fingerprint type, from Linear, Radial, Dendritic, MOLPRINT2D, Atom pairs, Atom triplets, or Topological torsions.

If you want to change any of the parameters used in the fingerprint calculations, click Advanced, and make your selections in the Hashed Fingerprints - Advanced Options dialog box. This dialog box has four main sections.

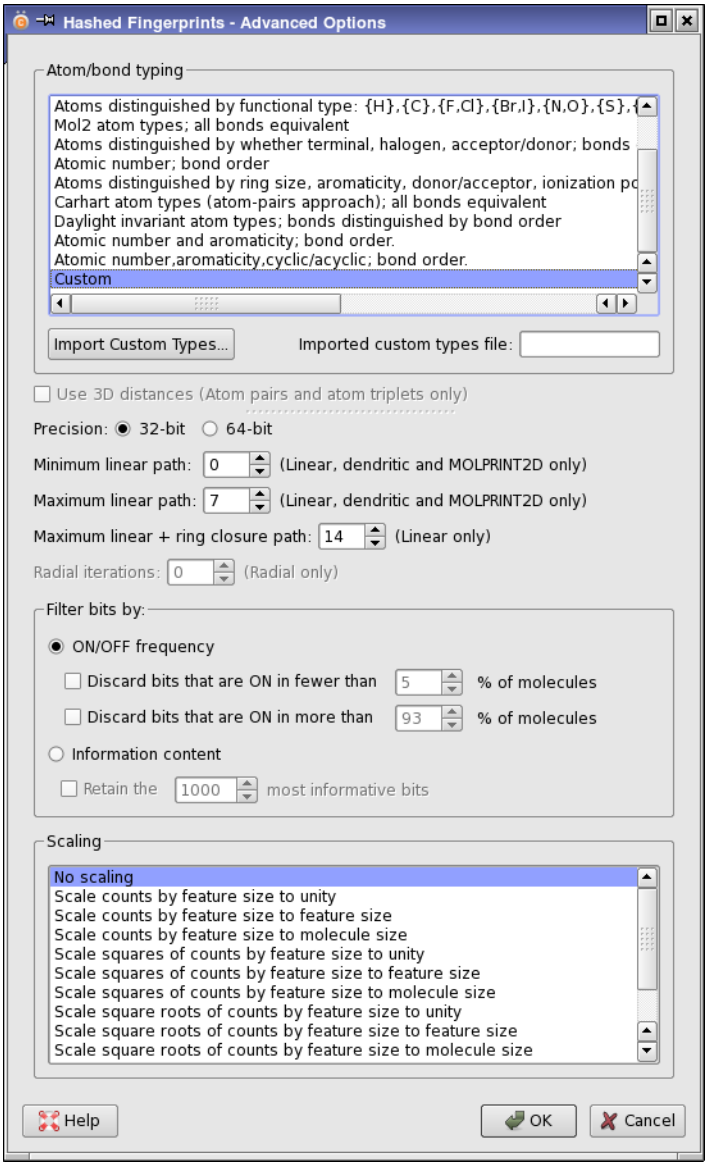


Figure 3.7. The Hashed Fingerprints - Advanced Options dialog box

- Atom/bond typing—Choose the atom and bond typing scheme to be used to classify atoms and bonds in the structures. If you choose Custom, click Import Custom Types to locate the file that contains the custom atom and bond types. The file must contain SMARTS patterns for the fingerprints, one per line.
- Various options—These include options for individual fingerprint types and precision options:
  - You can set the precision to 32 or 64 bits. Using 64 bits reduces collisions of ON bits, but doubles the space required to store each key.
  - For atom pairs and atom triplets, you can choose to use 3D distances if they are available.
  - For linear, dendritic, and MOLPRINT2D fingerprints, you can set the minimum and maximum linear path, and for linear fingerprints, you can also set the maximum linear+ring closure path.
  - For radial fingerprints, you can set the number of radial iterations.
- Filter bits by—Select an option for filtering out bits. If you select ON/OFF frequency, you can then choose to discard bits that are ON in fewer than or more than a specified percentage of molecules, and set the percentages. If you select Information content, you can choose to retain the *N* most informative bits, and set the value of *N*.
- Scaling—Select a method from this list to scale the binary fingerprint data to real numbers. The default is not to scale the fingerprint data.

As well as hashed fingerprints, you can calculate MACCS or custom structural keys. If you choose Custom, you must import a SMARTS definition file for the structural keys, by clicking Import SMARTS. Each line in the file must contain one SMARTS pattern, then the name of the key, separated by one or more spaces.

3.3.2 3D Fingerprints from Pharmacophores

Fingerprints based on pharmacophore models of the 3D structure are calculated using the 3D Pharmacophore Fingerprints dialog box, which you open by choosing Applications → 3D Pharmacophore Fingerprints. The fingerprint bits are assigned based on the feature type and the distances between the features.

You can choose whether to calculate the properties for the selected rows or the visible rows, a saved view, or the data set from a previous run, and choose a name for the new fingerprint column. A default name is supplied. If any of the structures contains more than one molecule (fragment), the largest fragment is used.

The fingerprints can be generated from 3-point pharmacophores (the default), 4-point pharmacophores, or both. 4-point pharmacophores usually set an order of magnitude more bits than 3-point pharmacophores, and do not provide much additional benefit.

Pharmacophores can be generated for a single conformer of each structure, or for a set of conformers that are generated temporarily for this purpose. To use the existing structure and not generate conformers, select Use existing 3D structures. To generate conformers, select Generate conformers in memory. The conformer set does not explicitly include the original structure, but is likely to contain a similar structure. The conformers are discarded after use.

When you start this job, you can choose to distribute it over multiple processors by setting the CPUs value in the Start Application dialog box.

If you want more control over the process, click Advanced, and make settings in the 3D Pharmacophore Fingerprints - Advanced Options dialog box.

- Custom feature definitions—Select Use custom feature definitions to supply your own definitions of the pharmacophore features in terms of SMARTS patterns. The definitions are in the format of a Phase feature definition file—see Appendix B.3 of the *Phase User Manual* for details. Click Import to import the definition file.
- Precision—Select 32-bit or 64-bit for the fingerprint precision. Using 32 bits is usually adequate and results in few collisions. Using 64 bits reduces collisions, but requires more space to store the fingerprints.
- Inter-feature distance bins—Set the smallest distance in the Lower limit text box, and the largest distance in the Upper limit text box. These distances set the range of the bins: distances smaller than the lower limit are placed in the first bin, and distances larger than the upper limit are placed in the last bin. Set the width of the bin in the Bin width text box, and

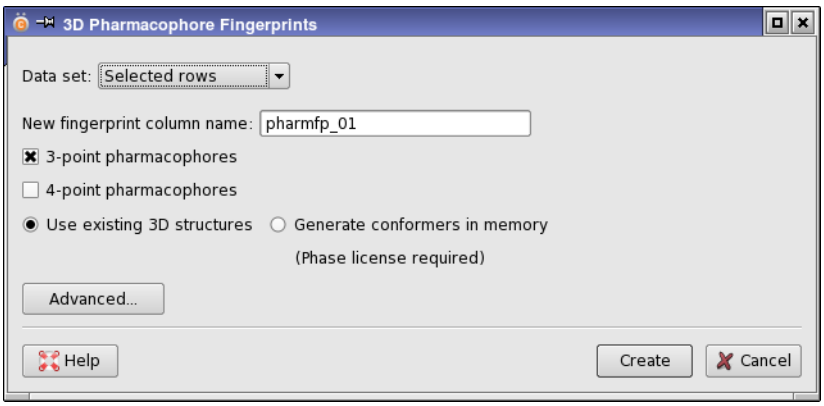


Figure 3.8. The 3D Pharmacophore Fingerprints dialog box

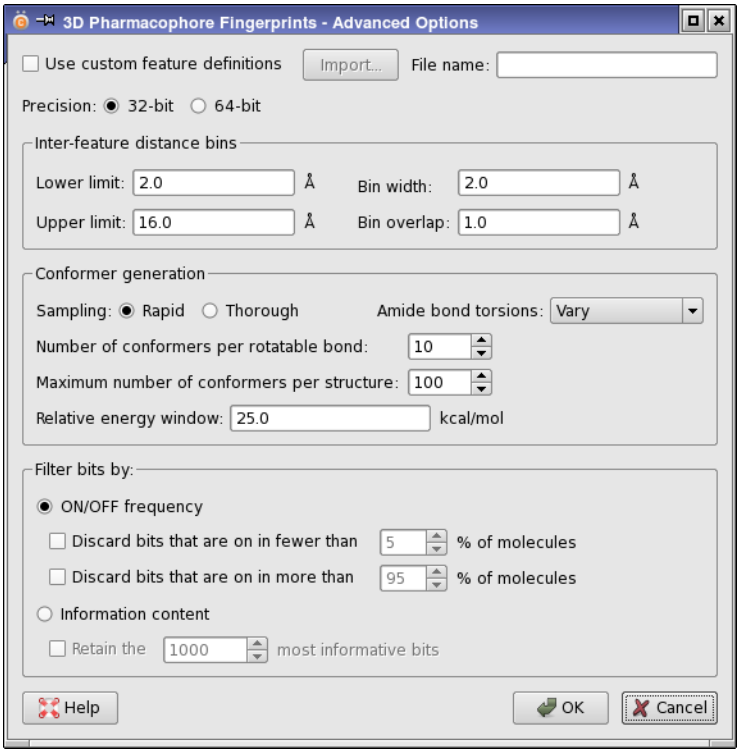


Figure 3.9. The 3D Pharmacophore Fingerprints - Advanced Options dialog box

the overlap in the Bin overlap text box. Distances are assigned to a neighboring bin as well if the distance is within the overlap distance of the bin boundary. This means that the effective bin width is the specified width plus twice the overlap width.

- Conformer generation—Set options and thresholds for generating conformers. This section is only available if you selected Generate conformers in memory in the main panel.
  - Select Rapid or Thorough for the sampling of conformational space. Rapid sampling varies rotatable groups independently; thorough sampling varies them together.
  - Choose how to treat amide bonds from the Amide bond torsions option menu. Vary allows full rotational freedom; the other two options fix the value at the input value or at the trans conformation.
  - Limit the number of conformers generated by setting values in the Maximum number of conformers per structure and Number of conformers per rotatable bond text boxes. Both limits are applied. You can also limit the number of conformers by filtering out conformers whose energy is higher than that of the lowest conformer by the amount specified in the Relative energy window text box.

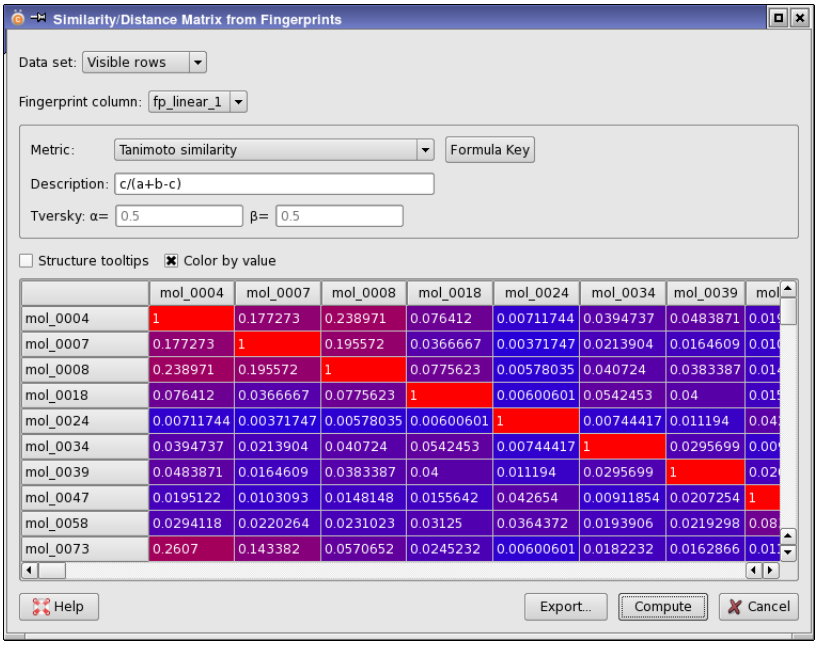
- Filter bits by—Choose a filtering method to reduce the number of bits. Select ON/OFF frequency to discard bits that are on in most molecules or only a few molecules, and select the relevant options and the percentage of molecules. Select Information content and set the number of informative bits to keep to filter by the amount of information provided by the bits.

3.4 Similarity, Dissimilarity, and Clustering

Molecular similarity or dissimilarity is measured by a distance between molecules in the space of a property set. The similarity or distance is defined by a metric. Similarity or distance can be used for clustering and screening of structures, or selection of diverse structures.

3.4.1 Computing a Similarity or Distance Matrix

Similarity and distance matrices can be computed in Canvas based on fingerprints or on molecular properties, and over a range of metrics. The computed matrices can be examined in table form, and they can be exported to a file, where they can be used by other applications. The structures for any table cell are shown in a tool tip if you choose Structure tooltips.



	mol_0004	mol_0007	mol_0008	mol_0018	mol_0024	mol_0034	mol_0039	mol
mol_0004	1	0.177273	0.238971	0.076412	0.00711744	0.0394737	0.0483871	0.01
mol_0007	0.177273	1	0.195572	0.0366667	0.00371747	0.0213904	0.0164609	0.01
mol_0008	0.238971	0.195572	1	0.0775623	0.00578035	0.040724	0.0383387	0.01
mol_0018	0.076412	0.0366667	0.0775623	1	0.00600601	0.0542453	0.04	0.01
mol_0024	0.00711744	0.00371747	0.00578035	0.00600601	1	0.00744417	0.011194	0.04
mol_0034	0.0394737	0.0213904	0.040724	0.0542453	0.00744417	1	0.0295699	0.00
mol_0039	0.0483871	0.0164609	0.0383387	0.04	0.011194	0.0295699	1	0.02
mol_0047	0.0195122	0.0103093	0.0148148	0.0155642	0.042654	0.00911854	0.0207254	1
mol_0058	0.0294118	0.0220264	0.0231023	0.03125	0.0364372	0.0193906	0.0219298	0.08
mol_0073	0.2607	0.143382	0.0570652	0.0245232	0.00600601	0.0182232	0.0162866	0.01

Figure 3.10. The Similarity/Distance Matrix from Fingerprints dialog box

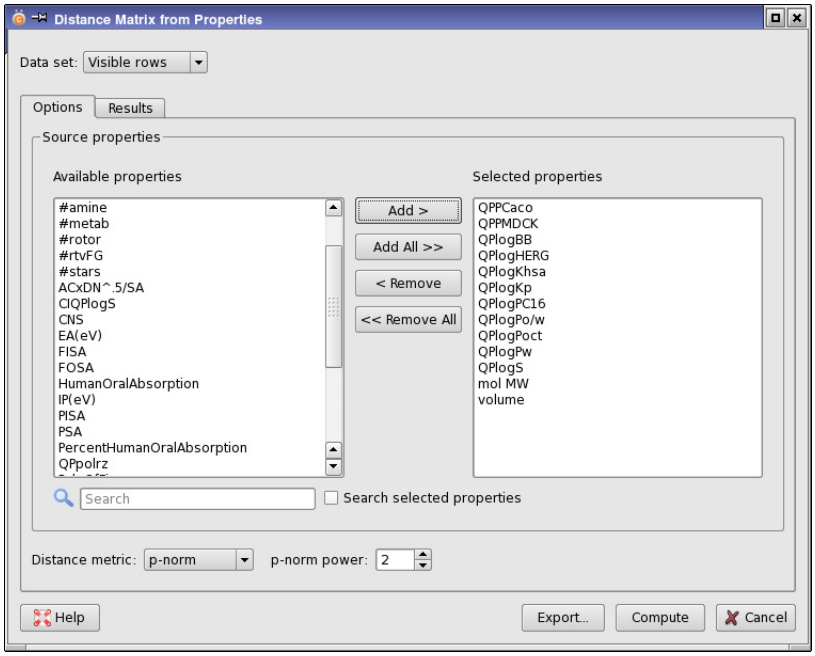
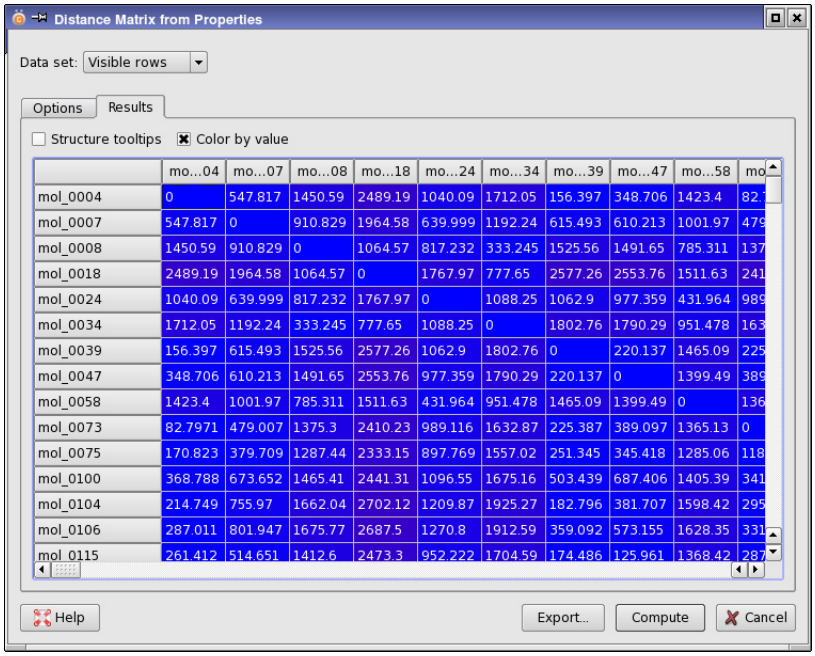


Figure 3.11. The Distance Matrix from Properties dialog box, Options tab.

To calculate similarity or distance matrices based on fingerprints, choose Applications → Similarity/Distance Matrix → Create from Fingerprints. In the Similarity/Distance Matrix from Fingerprints panel, you can choose whether to calculate the matrix for the selected rows or the visible rows, a saved view, or the data set from a previous run, choose the fingerprint column and the metric, and choose whether to color the results table cells by value. Twenty-four metrics are available—see Table 5.22 on page 142 for a list. When you click Compute, you are prompted to either to display the results in a table or export them to a file.

To calculate distance matrices based on properties, choose Applications → Similarity/Distance Matrix → Create from Properties. In the Distance Matrix from Properties panel, you can choose whether to calculate the matrix for the selected rows or the visible rows, a saved view, or the data set from a previous run, select the properties and the metric, and choose whether to color the results table cells by value. Two metric types are available: p-norm and Mahalanobis. The p-norm is defined by $(\Sigma x^p)^{1/p}$. The Mahalanobis distance matrix is calculated using the covariance matrix to correct non-orthogonality of the data. When you click Compute, you are prompted to choose whether to display the results in a table or export them to a file.



	mo...04	mo...07	mo...08	mo...18	mo...24	mo...34	mo...39	mo...47	mo...58	mo
mol_0004	0	547.817	1450.59	2489.19	1040.09	1712.05	156.397	348.706	1423.4	82.7
mol_0007	547.817	0	910.829	1964.58	639.999	1192.24	615.493	610.213	1001.97	479
mol_0008	1450.59	910.829	0	1064.57	817.232	333.245	1525.56	1491.65	785.311	137
mol_0018	2489.19	1964.58	1064.57	0	1767.97	777.65	2577.26	2553.76	1511.63	241
mol_0024	1040.09	639.999	817.232	1767.97	0	1088.25	1062.9	977.359	431.964	989
mol_0034	1712.05	1192.24	333.245	777.65	1088.25	0	1802.76	1790.29	951.478	163
mol_0039	156.397	615.493	1525.56	2577.26	1062.9	1802.76	0	220.137	1465.09	225
mol_0047	348.706	610.213	1491.65	2553.76	977.359	1790.29	220.137	0	1399.49	389
mol_0058	1423.4	1001.97	785.311	1511.63	431.964	951.478	1465.09	1399.49	0	136
mol_0073	82.7971	479.007	1375.3	2410.23	989.116	1632.87	225.387	389.097	1365.13	0
mol_0075	170.823	379.709	1287.44	2333.15	897.769	1557.02	251.345	345.418	1285.06	118
mol_0100	368.788	673.652	1465.41	2441.31	1096.55	1675.16	503.439	687.406	1405.39	341
mol_0104	214.749	755.97	1662.04	2702.12	1209.87	1925.27	182.796	381.707	1598.42	295
mol_0106	287.011	801.947	1675.77	2687.5	1270.8	1912.59	359.092	573.155	1628.35	331
mol_0115	261.412	514.651	1412.6	2473.3	952.222	1704.59	174.486	125.961	1368.42	287

Figure 3.12. The Distance Matrix from Properties dialog box, Results tab.

3.4.2 Screening Structures by Similarity or Distance

Similarity or distance matrices computed from fingerprints can be used to screen structures by similarity to one or more reference structures. The results are returned as properties that contain the similarity to the reference structure.

To set up a screen, choose Applications → Similarity/Distance Screen. The screen can be run on the selected rows or the visible rows, a saved view, or the data set from a previous run, a saved view, or the data set from a previous run, and you can choose the fingerprint from those that are already available in the project.

The names of the similarity properties are generated from a base name, which you can enter, with a suffix added that distinguishes the reference structures. The suffix can be taken from either a generated sequence of names, or the reference structure name. A default is supplied.

The reference structures can be taken from the selected rows, a saved view, or they can be selected by hand. If you choose Select by hand, the structures in the screen are shown in the reference structures table, and you can select the desired table rows. The column name is displayed for the selected structures when you select them.

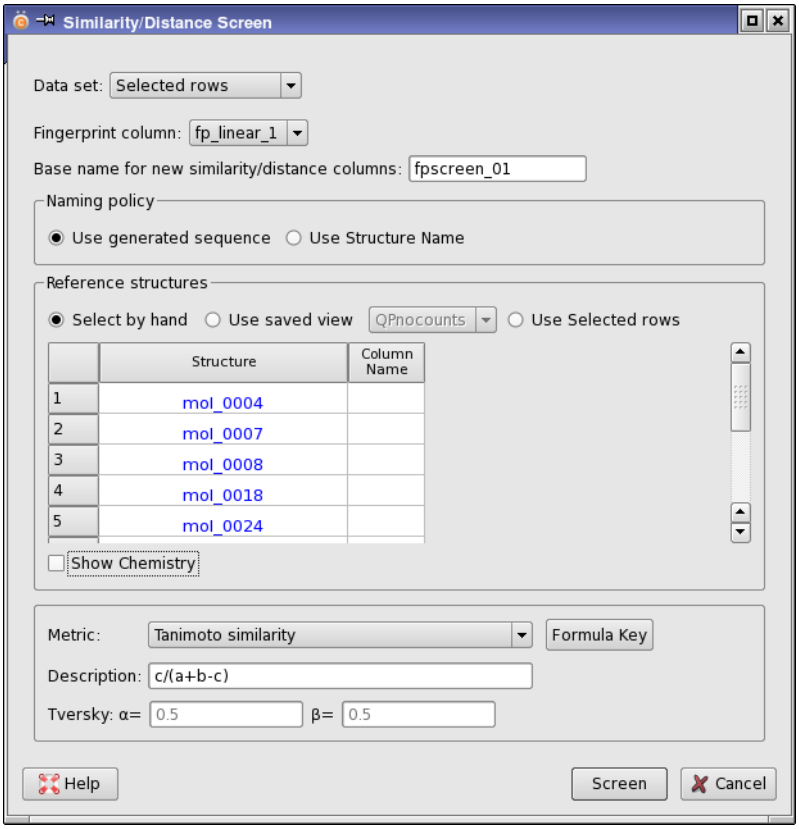


Figure 3.13. The Similarity/Distance Screen dialog box

The metric used to calculate similarity or distance can be chosen from the 24 available metrics. The description of the metric is given in the Description text box, and the definition of the symbols can be viewed by clicking Formula Key.

When you have selected all the options, click Screen to generate the results. After the results are incorporated into the spreadsheet, you can run a property query to select structures based on similarity to one or more of the references.

3.4.3 Screening Structures by Shape

If you have 3D structures, you can select structures that have a similar shape to a query molecule in the Shape Screen dialog box, which you open from the Applications menu. The screening is done with the `phase_shape` program, and requires a Phase license. For details on shape screening, see Chapter 14 of the *Phase User Manual*.

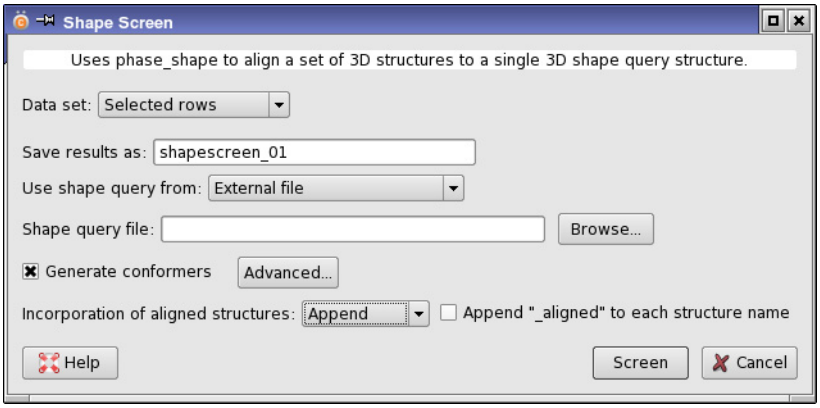


Figure 3.14. The Shape Screen dialog box

The screen can be run on the selected rows or the visible rows, a saved view, or the data set from a previous run, and the results are saved with the name you provide. The shape query must be a single molecule and can come from an external file (Maestro or SD), or the structure that is displayed in the PyMOL viewer.

If you want more control over the screening process, you can set options in the Shape Screen - Advanced Options dialog box, which you open by clicking Advanced.

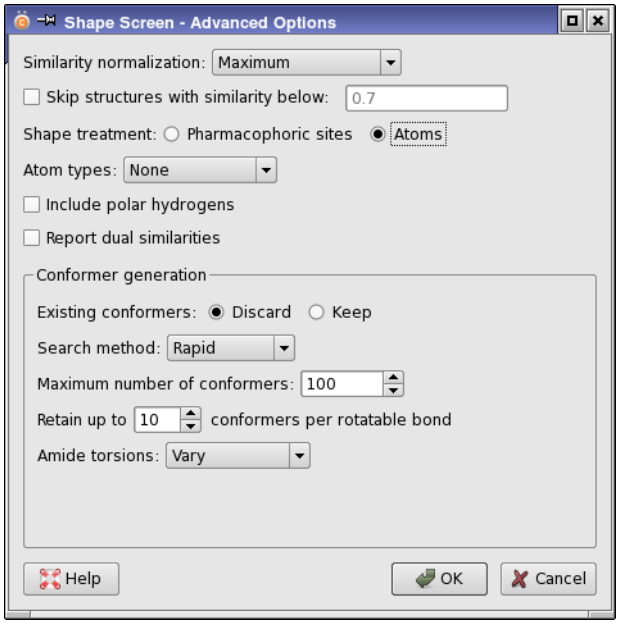


Figure 3.15. The Shape Screen - Advanced Options dialog box

- You can generate conformers for the molecules that are screened. The conformational search varies only the torsional angles of rotatable groups in the molecules.
- You can screen molecules based on pharmacophore types (the default) or atom types, and select different atom type schemes for determining which atoms are treated as similar. For a simple shape screen, choose Atoms with Atom type: None. When using atom types, you can choose whether to include polar hydrogens in the determination of similarity, and report shape similarities with and without atom typing.
- You can select a similarity normalization (default is Maximum), and skip structures with low similarity. The similarity to the query is returned as a property.

3.4.4 Selecting Diverse Structures

It is often useful to be able to select structures that are as different as possible from each other. You can do this in Canvas in the Diversity-Based Selection dialog box, which you open from the Applications menu.

The analysis can be run on the selected rows of the spreadsheet or on all rows, and the results are saved with a name which you can specify. The similarity or distance matrix is computed from fingerprints, which you can select from those already available in the project. You can also choose the metric used for the similarity or distance from the 24 available metrics.

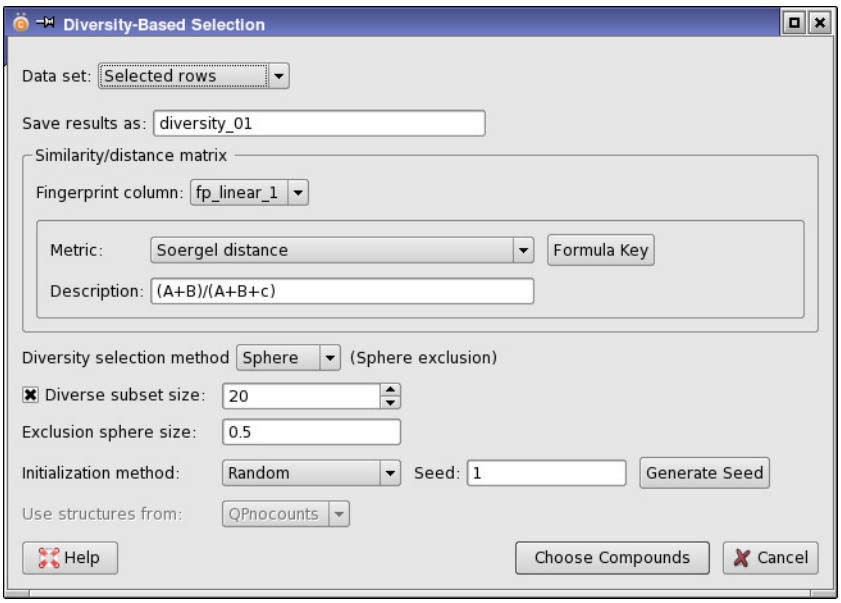


Figure 3.16. The Diversity-Based Selection dialog box

Four methods are available from the Diversity selection method option menu: Sphere (sphere exclusion), DISE (directed sphere exclusion [2]), MAXSUM (maximum sum of pairwise distances), and MAXMIN (Maximum nearest-neighbor distance). The size of the subset can be specified for all four methods, but is only applied for the Sphere and DISE methods if the exclusion sphere size produces a larger diverse subset.

The selection must be initialized with one or more structures. The initialization method depends on the selection method. The DISE method has its own initialization, so no method is available. For the other three methods, you can choose random initialization and specify the or generate the seed. For the Sphere method, you can choose existing structures, and specify the structures as all structures that are in the view of your choice. These structures serve as an excluded set: the diverse set does not include any structure within the exclusion sphere of the input structures. For the MAXSUM and MAXMIN initialization method, you can choose Representative (structure with the greatest similarity to the other structures) or Dissimilar (structure with the least similarity to the other structures).

When you have made your choice, click Choose Compounds. After the job has finished, you can open a restricted custom view of the selected structures, by right-clicking on the record in the Project View panel under Diversity and choosing View. You can save this view as a normal custom view, or apply it to the master view. When the job is incorporated, a new property is added to the spreadsheet, with 1 for the structures that were selected and 0 for those that were not selected. The property is named *jobname* set, with underscores replaced by blanks.

3.4.5 Hole-Filling and Library Optimization

If you want to choose a set of compounds based on their dissimilarity and optionally by optimizing the values of a set of properties, you can do so with in the Hole-Filling and Library Optimization dialog box, which you open from the Applications menu. With this application you can choose a set of dissimilar structures from a pool, and you can use them to fill holes in an existing library. Like Diversity-Based Selection, this application uses nearest-neighbor distances to choose the most diverse structures, but with a different approach. Here, the desired number of structures is selected at random, and then the similarity of the structures to each other and to the library is minimized. In addition, the process can be guided by selection of structures whose properties lie in specified ranges. This is done by minimizing the fraction of properties for each selected structure whose values lie outside the ranges.

The structure pool can be chosen from selected or visible rows in the master view, a saved view, or the data set from a previous run, and the results are saved with a name that you can specify. If you want to use the selected rows, you must select them before opening the dialog box.

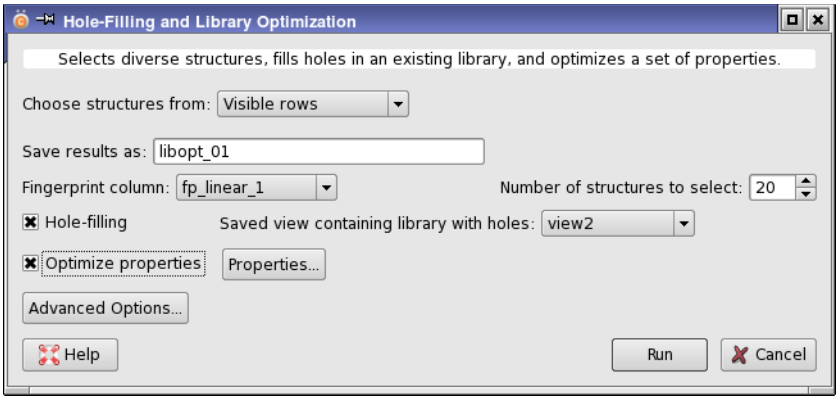


Figure 3.17. The Hole-Filling and Library Optimization dialog box

If you select Hole-filling, the library whose holes are to be filled is taken from a saved view, which you can choose. It is best to ensure that the structure pool and the library do not have structures in common. The similarity matrix is computed from fingerprints, which you can select from those already available in the project.

If you want to optimize the values of properties, select Optimize properties, and click Properties to choose the properties and set the desired ranges. To save the property set and their ranges as a property filter, click Export Filter. Filters can be imported (Import) or read from another job (Use filter from).

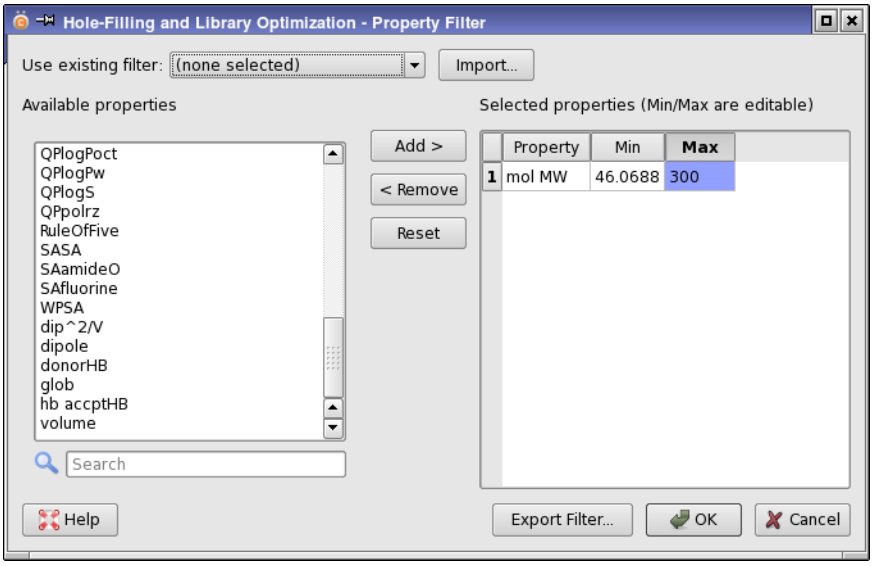


Figure 3.18. The Hole-Filling and Library Optimization - Property Filter dialog box

You can optimize properties without filling holes in a library, but the structures are still selected by minimizing their similarity to one another.

When you have specified the structures and set options, including the number of structures to select, click Run. The time taken scales as the square of the number of structures selected, so selecting a large number of structures could take some time. If you want to set parameters to control the optimization process, click Advanced Options. In the dialog box that opens, you can set the maximum number of optimization cycles, thresholds for stopping the optimization, a limit on the run time, the random seed for selecting structures and other aspects of the optimization, and a threshold for a simulated annealing process used when the score used in the optimization increases. See Section 5.4.4 on page 136 for more information on these parameters and the algorithm used for the optimization.

3.4.6 Comparing Libraries

A histogram of nearest neighbor similarities is a convenient way to assess the diversity of a single collection of compounds, or the similarity between two collections of compounds. You can create the data for this histogram in the Library Comparison dialog box, which you open from the Applications menu.

The library comparison works by finding the nearest neighbor in the reference library for each compound in the query library, using fingerprint similarity based on a chosen metric. The nearest-neighbor similarities are returned in a column named MaxSim. The results of the library comparison can be used to create a histogram from the nearest neighbor similarities.

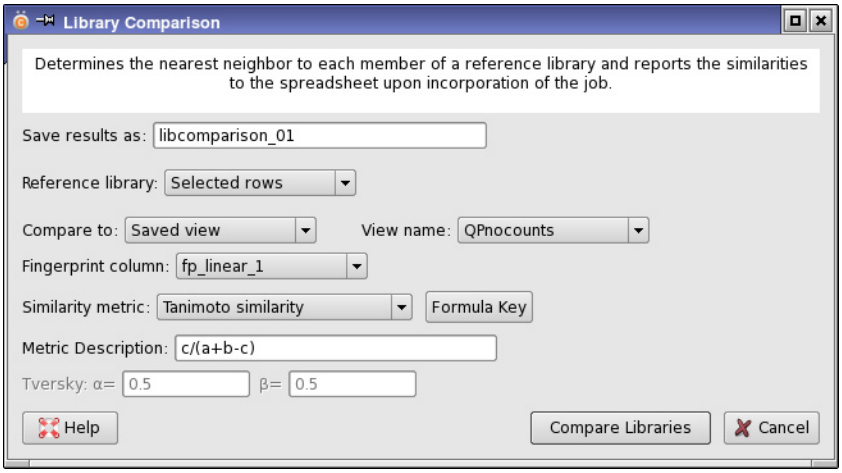


Figure 3.19. The Library Comparison dialog box

The reference library can be chosen from the selected rows or the visible rows of the spreadsheet, a saved view, or the data set from a previous run, and the results are saved with a name which you can specify. If you want to use the selected rows, you must select them before opening the dialog box. The library for comparison can be taken from a saved view, which you can choose, or it can be the reference library itself. The similarity matrix is computed from fingerprints, which you can select from those already available in the project. You can also choose the metric used for the similarity from the available similarity metrics.

When you have made all the desired settings, click Compare Libraries. After the job finishes, you can create a histogram based on the MaxSim property, by clicking the Histogram button.

The histogram for a library with a high degree of similarity to the reference library will have most of the library at the greatest similarity, whereas a library with a low similarity to the reference library will have a large part of the library at smaller similarities.

3.4.7 Clustering Structures by Similarity or Distance

Canvas provides three methods for clustering structures: hierarchical clustering, leader-follower clustering, and k-means clustering. Leader-follower and k-means clustering can be used on much larger data sets than hierarchical clustering. Clustering by these methods can be done in the Hierarchical Clustering dialog box, the Leader-Follower Clustering dialog box, and the K-Means Clustering dialog box, which you open from the Applications menu.

For all three methods, you can choose the dataset for the clustering as described in Section 3.1 on page 61, and the results are saved with a name that you can specify.

All three methods can use fingerprints for clustering, which you can select from those already available in the project. For hierarchical and leader-follower clustering, you can choose the metric used for the similarity or distance from the 24 available metrics. For leader-follower and k-means clustering, you can use properties for clustering, which you select from the properties available in the project. The properties can be automatically scaled to a common range, by selecting Autoscale.

The remaining options in the three dialog boxes are unique to the method. When you have finished making your selections, click Create Clusters to run the clustering job. A job record is added in the Project View panel under the clustering method, and the progress of the job is shown in the Messages View panel.

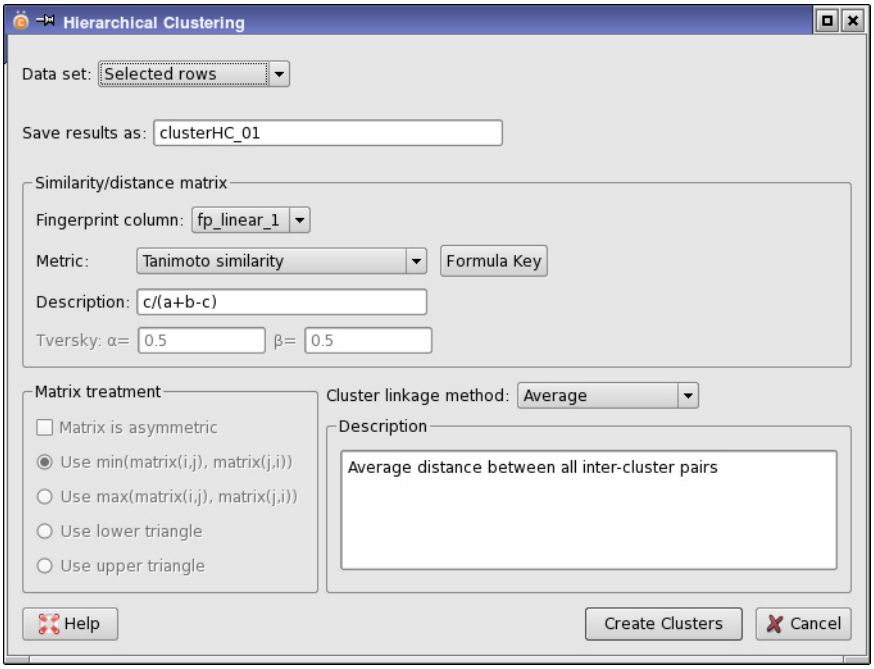


Figure 3.20. The Hierarchical Clustering dialog box

3.4.7.1 Hierarchical Clustering

The specific options for hierarchical clustering include the following:

- If the matrix is asymmetric, you can select Matrix is asymmetric and then choose how to treat the off-diagonal elements.
- You must choose a clustering linkage method, from the nine available methods. The linkage method describes how to calculate the distance between clusters. When you choose a linkage method, the description is displayed in the Description text area. The linkage methods are listed in Table 3.1.

The results of hierarchical clustering are not incorporated directly into the project. Instead, you can view a dendrogram of the clusters, and choose a set of clusters to export either to a file or to the spreadsheet.

To view the dendrogram, right-click on the job record in the Project View panel and choose View. In the Hierarchical Clustering Dendrogram panel, you can choose the cluster set by the number of clusters, by the Kelley criterion, or by the merging distance. The merging distance is displayed as a tool tip when you pause the pointer over a split point. For the last of these, you must click Update after changing the merging distance to view the new set of clusters. To

Table 3.1. Description of linkage methods for hierarchical clustering.

Method	Description
Single	Closest inter-cluster pair (nearest neighbor)
Complete	Farthest inter-cluster pair (farthest neighbor)
Average	Average distance between all inter-cluster pairs
Centroid	Euclidean distance between cluster centroids
McQuitty	Average distance to the two clusters merged in forming a given cluster
Ward	Sum of squared distances to merged cluster centroid (minimum variance)
Weighted centroid	Weighted center of mass distance, also known as median
Flexible beta	Weighted average intra-cluster and inter-cluster distances (Lance-Williams) with beta=0.25
Schrödinger	Closest distance between terminal (right-to-left) points in 1D cluster orderings.

change the scaling of the dendrogram, use the mouse wheel. You can export an image of the dendrogram in PNG, JPEG, or TIFF format by clicking Save Image.

By default, the dendrogram is colored by cluster membership. You can color the leaf nodes by the values of a property with a heat map, by selecting Color by property, and clicking Edit Heat Map to choose the properties and setting the color range for the property (see Section 2.5.3 on page 40). The properties that you select are added to the option menu in the Hierarchical Clustering Dendrogram panel, so you can switch between properties. To switch back to coloring by cluster membership, select Color by cluster.

When you have selected a set of clusters by any of the three methods, you can click Export to export the clusters or their centroids to a file or to the spreadsheet. The Export Selection dialog box opens, in which you can specify the destination and make other related settings. In addition to the structures, three properties are exported: Cluster, which is the cluster index, Centroid, which is 1 for the centroid of each cluster, and 0 otherwise, and Total, which is the number of structures in the cluster. If there is a tie for the centroid, the first structure is chosen.

If you choose To file, click Browse to navigate to the location for the file, select the file format from Maestro, SD, or CSV, and name the file. You can export the structures in each cluster to a separate file, labeled with the cluster index, by selecting Place results into separate files. CSV files include the structure name, a SMILES string for the structure, and the three properties. If you chose to export only the centroids, they are written to a single file.

If you choose To spreadsheet, the column names for the properties are derived from the property name by adding a prefix, which you can specify in the Name prefix text box. The resulting column names are *prefix*:Cluster, *prefix*:Centroid, and *prefix*:Total. The default prefix is

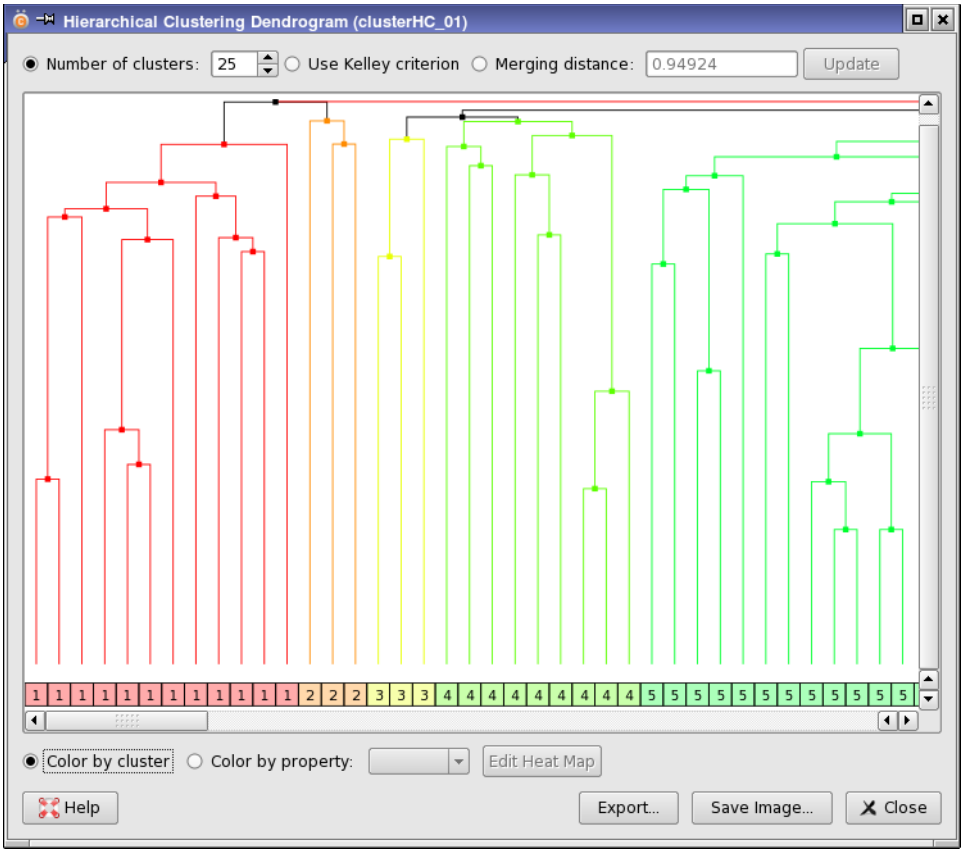


Figure 3.21. The Hierarchical Clustering Dendrogram panel

HClust::*jobname*. The clusters can be exported to separate views, named *prefix*::cluster N, by selecting Place clusters in separate views. If you chose to export only the centroids, they are written to a single view named *prefix*::centroids.

You can view the structures that belong to a cluster by clicking the "leaf" (the square at the bottom of the dendrogram) for one of the cluster members. A limited custom view is displayed that contains only the structures by default, with the background color of the cluster from the dendrogram. You can add properties to the view by choosing View → Manage Properties and selecting the properties. When you do, these properties are added by default to any cluster view opened by clicking its leaf. To view the structures for more than one cluster, click in the dendrogram panel, then control-click the leaf.

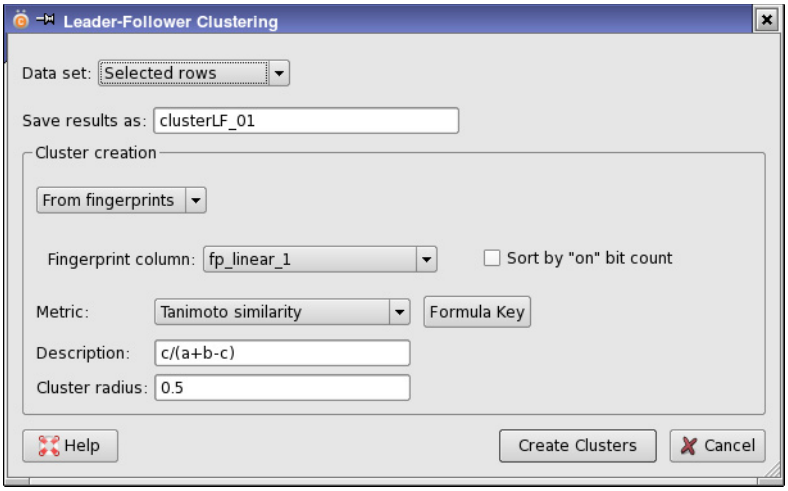


Figure 3.22. The Leader-Follower Clustering dialog box

3.4.7.2 Leader-Follower and K-Means Clustering

For leader-follow clustering, you can sort the fingerprints by the count of "on" bits, in descending order. This option is only available with some of the metrics. Sorting can speed up the clustering calculation considerably.

For k-means clustering, the parameters of the method can be set. You must specify the number of clusters, but the remaining parameters can be left at their defaults.

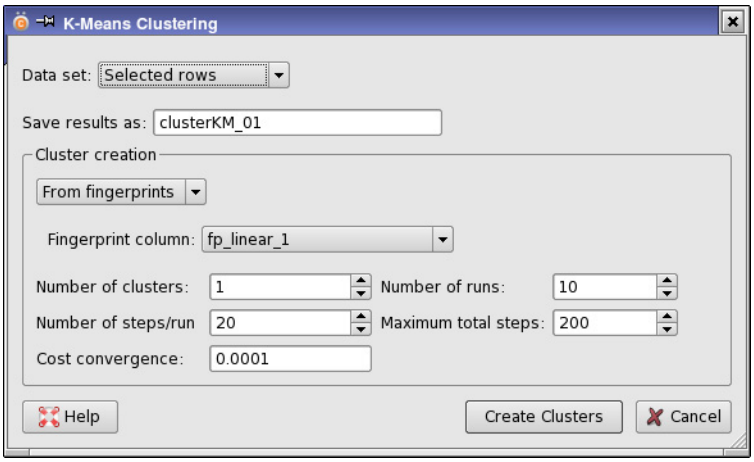


Figure 3.23. The K-Means Clustering dialog box

Information on the algorithms for k-means and leader-follower clustering is given in Section 5.4.11 on page 149 and Section 5.4.12 on page 150.

Incorporating the results of leader-follower and k-means clustering adds a new column to the spreadsheet, named *prefix*:*jobname*:Cluster, where prefix is LFClust or KMClust. This column contains the index of the cluster to which each structure belongs. For leader-follower clustering, another column is added that marks the leader in each cluster, named LFClust:*jobname*:Leader. The column contains 1 for the leader in each cluster and 0 otherwise. Similarly, for k-means clustering a column is added that marks the centroid of each cluster, named KMClust:*jobname*:Centroid.

You can view the clusters by right-clicking the job record in the Project View panel, and choosing View. A restricted view of the spreadsheet opens, named Leader-Follower Clustering Results, or K-Means Clustering Results. This view has only the structure column and the File, View and Chemistry menus, but you can add properties from the View menu. You can export the clusters, select the cluster to view, save the view as a custom view, or apply the view to the master view. You can open more than one of these panels to display multiple clusters. For leader-follower clustering, the cluster leader is marked by a gray background.

3.5 Building a Predictive Model

Canvas provides a range of methods for building a model that can be used to predict properties. These models include the common regression models—multiple linear regression, partial least-squares regression, and principal components regression—and also Bayes classification, recursive partitioning, and neural network models. The models can be built, applied, exported and plotted from their dialog boxes, which you open from the Applications menu.

The dialog boxes have a common structure, as they are performing similar tasks. Each has an Options tab for setting up the input and options, and a tab for selecting the training set and displaying the results. This tab is labeled Training Set initially, and is relabeled Results after the model is built. The common features are described below. See also the general descriptions in Section 3.1 on page 61.

- The models can be built or applied for the selected rows or the visible rows of the spreadsheet, for a saved view, or for the original data set for a run, by selecting from the Data set option menu.
- The results are saved with the name that you specify in the Save results as text box. This name is also the job name and appears in the Project View panel.
- Existing models can be applied to the data set. Models from within the project can be loaded by selecting Apply saved model and choosing the model from the option menu. External models can be loaded by selecting Apply imported model, clicking Import, and navigating to the file that contains the model. To apply the selected model, click Apply Model. The results are displayed as columns in the structures table, and include the predicted value, and the statistics are displayed in the tabs next to it (Training and Test).
- A new model can be built when you select Build new model. You can select the independent variables with the X variable property selection tools or fingerprint option menu (if available), and the dependent variable (or variables, for neural networks) from the Y variable option menu or selection tools. The values of a Y variable are displayed in the data set structures table when you select it.
- The training set and the test set can be assigned in three ways:
    - Randomly: select Assign randomly, specify a percentage in the text box, specify a seed by entering it the Seed text box or clicking Generate Seed, then click Apply. The training and the test set are not completely random: the random selection process ensures that they have comparable distributions of activities.
    - From a partition: select Assign from partition, choose the partition name from the option menu, select partition values from the Training set value and Test set value option menus, then click Apply. The percentage of the values in the partition is displayed in text boxes to the right of the option menus.

  - Manually: set the values in the Model Set column of the structures table. To change the value for a single row, double-click in the table cell and choose a value from the option menu that appears. To change the value for multiple rows, select the rows, right-click on any of the selected cells and choose the value from the Change Selected submenu of the shortcut menu.

- The model is built when you click Build Model. For most of the models, statistics on the training set and the test set are displayed in the tabs to the right of the structures table.
- You can export the model to a file, by clicking Export Model, and navigating to the desired location and naming the file in the file selector that opens.
- You can display a scatter plot of the results by clicking Scatter Plot (if the panel has this button). A dialog box opens, in which you can specify the data set to be plotted, choose colors for the training and test set points, and choose whether to draw the 45-degree line of perfect fit. Clicking Plot then opens a standard scatter plot panel, as described in Section 2.7.1 on page 48.
- You can export the data from selected rows in the structures table, by right-clicking and choosing Export to File or Export to Spreadsheet. For export to a file, data in all columns for the selected rows are written to a CSV file with column headings. Exporting to the spreadsheet adds the predicted values to the spreadsheet.

The features that differ between the dialog boxes are for setting up the parameters of the method, and are described in the following sections.

The final section describes Kohonen self-organizing maps. Although not quite the same as the other predictive models, it can also be built and applied.

3.5.1 Multiple Linear Regression

For multiple linear regression, you can build a model from the selected X variables, or you can choose the best subsets. If you select Choose best subsets, click Options to set parameters for finding the best subsets in the Multiple Linear Regression - Best Subsets Options dialog box.

The best subsets are determined by a simulated annealing process, in which you specify a number of Monte Carlo steps, an initial temperature and a final temperature. The temperatures are expressed in terms of the standard deviation in the dependent variable. The number of X variables in the subsets must be specified. To average the best models, select Use average of and specify the number of models in the box. The average can be weighted by the R^2 value.

You can force the regression line to pass through the origin by selecting Suppress y intercept.

When the model is built, the coefficients for each independent variable and the standard deviation for the variable is given in a table below the statistics tabs.

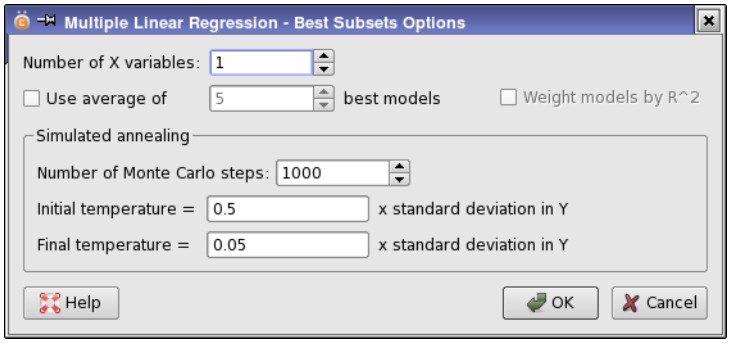


Figure 3.25. The Multiple Linear Regression - Best Subsets Options dialog box

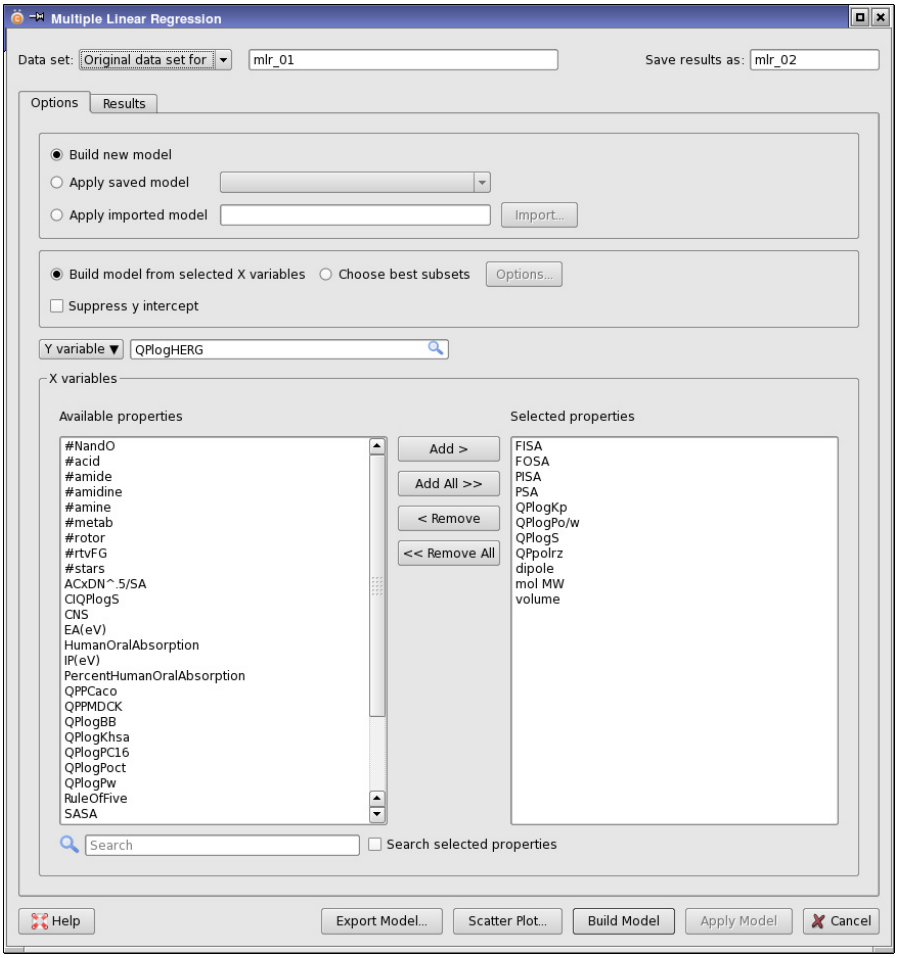


Figure 3.24. The Multiple Linear Regression dialog box

3.5.2 Partial Least-Squares Regression

For partial least-squares (PLS) regression, you can set the maximum number of PLS factors, and you can stop adding PLS factors when the standard deviation drops below a limit that you set. The dependent variables can be scaled automatically so that they lie on the same range. You can also set a minimum t-value for selecting significant independent variables.

When you build the model, it includes models for each number of PLS factors from 1 to the maximum, or the point at which the standard deviation falls below the threshold.

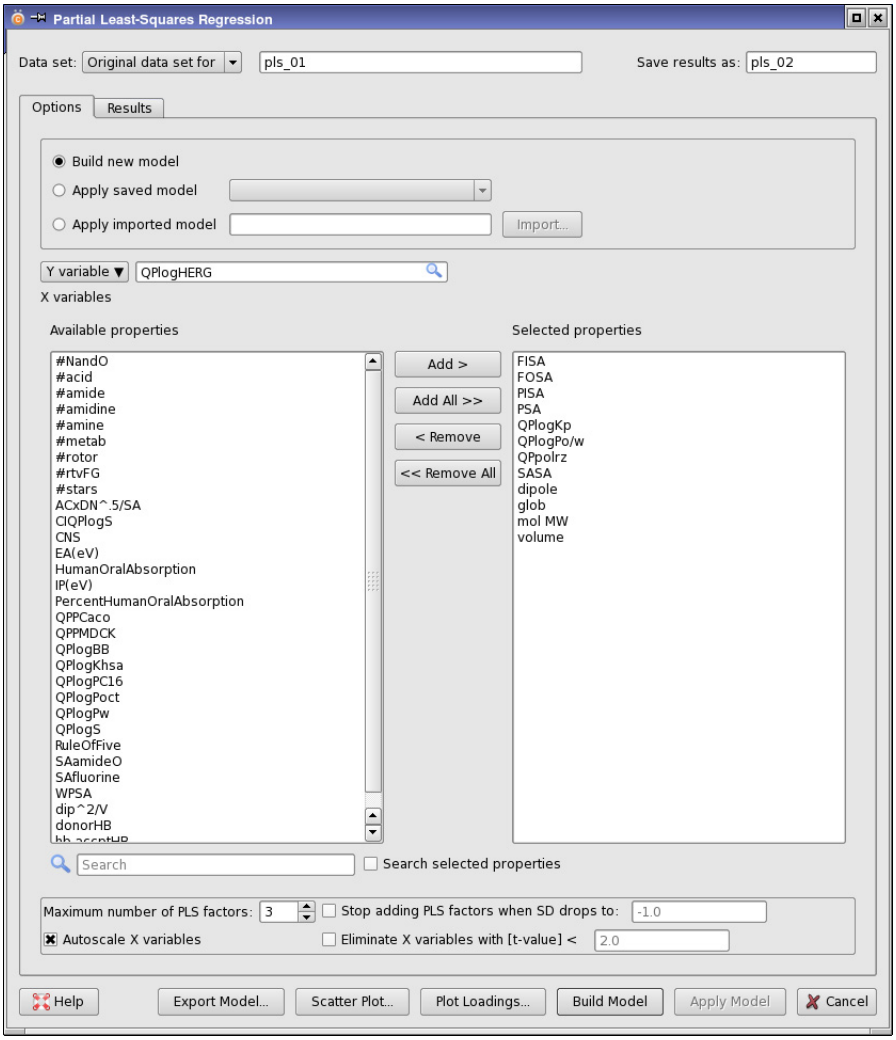


Figure 3.26. The Partial Least-Squares Regression dialog box

After the model is built, the loadings for each PLS factor for each X variable are displayed in the Factor loadings tab. You can plot the loadings by clicking Plot Loadings. A standard scatter plot panel opens (see Section 2.7.1 on page 48), and you can choose the factors that you want to plot on the *x* axis, the *y* axis, by color, and by symbol size.

3.5.3 Kernel-Based Partial Least-Squares Regression

Kernel-based partial least-squares regression is an extension of partial least-squares regression that introduces some nonlinearity into the scalar products of X variables used in the regression via a "kernel", which is some nonlinear function of these scalar products [7]. In Canvas, the kernel is a Gaussian function,

$$K(i,j) = \exp(-d^2{}_{ij}/\sigma^2{}_{ij})$$

where d_{ij} is the Euclidean distance between X variables *i* and *j*, and σ is the non-linearity parameter. This kernel replaces the simple scalar products of the X variables in the regression. In Canvas, no automatic tuning of σ is done.

You can set the maximum number of KPLS factors, and you can stop adding KPLS factors when the standard deviation drops below a limit that you set. You can manually adjust the kernel nonlinearity using the slider or the box. The value set by this slider is 1/σ, so values close to zero mean almost linear, and large values mean very nonlinear. Higher nonlinearity typically leads to tighter fitting, but it also tends to give poorer predictions on new compounds.

The uncertainty in the predictions for the test set can be estimated using bootstrapping. Select Calculate uncertainty on test set predictions, and set the number of cycles used in the bootstrapping method. Bootstrapping is done by sampling the training set randomly with replacement to generate a new test set of the same size (that may include duplicates), building a model and making predictions of the test set, then repeating the procedure a specified number of times. The standard deviation from the original test set is then calculated as the uncertainty.

When you build the model, it includes models for each number of KPLS factors from 1 to the maximum, or the point at which the standard deviation falls below the threshold. The error and the predicted uncertainty (if calculated) is displayed in the structures table for the number of KPLS factors chosen in the Test or Training results tab.

Models can be built on properties or on fingerprints. If you build a model based on fingerprints, you can visualize the contributions of each atom to the model. When the job finishes, right-click on the job name in the Project View and choose View to open the Kernel-Based Partial Least-Squares Regression dialog box, then click Visualize Model.

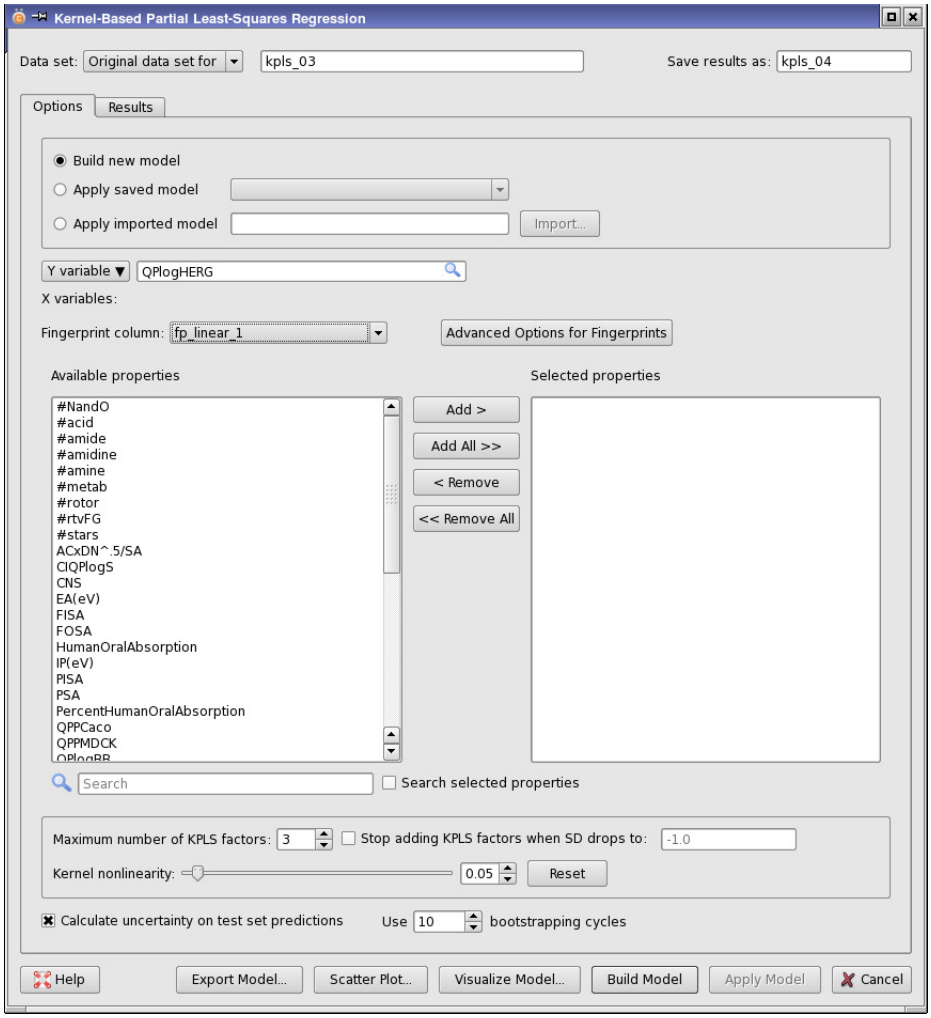


Figure 3.27. The Kernel-Based Partial Least-Squares Regression dialog box

The Model Visualization panel displays the structures in a spreadsheet with the observed and predicted values of the property, the uncertainty in the predicted value for test set molecules (if calculated) and the set to which the were assigned in the building of the model. You can sort the spreadsheet rows in ascending order of property values.

Each atom that contributed to a fingerprint used in building the model is marked with a colored disk that represents the value of the contribution to the property due to that atom. Two colors are used: one for negative values and one for positive values. The color saturation indicates the

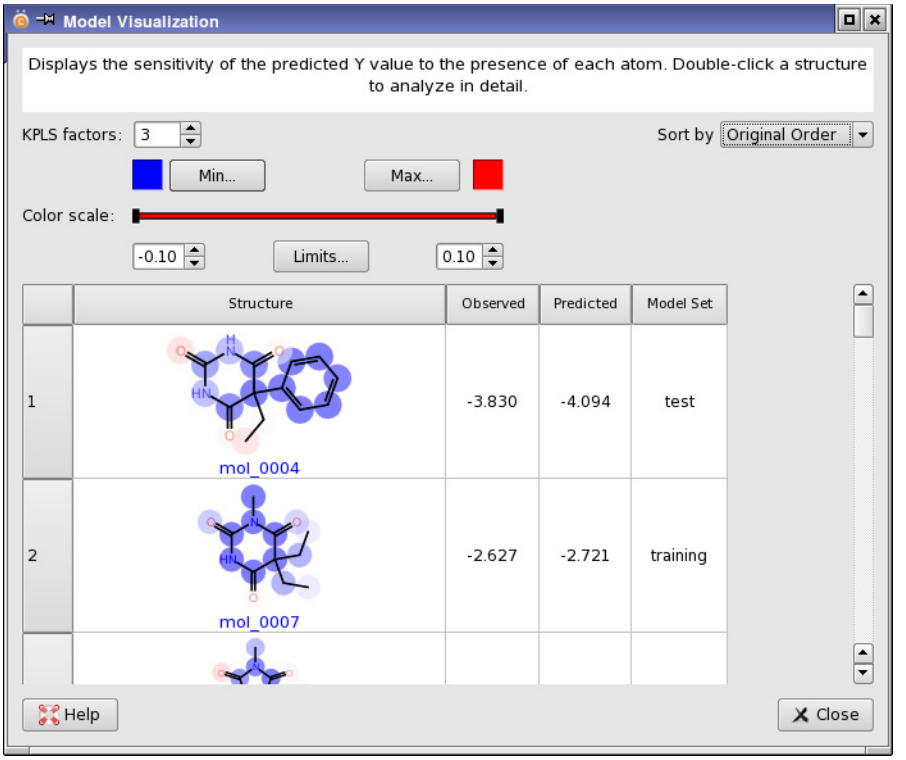


Figure 3.28. The KPLS Model Visualization panel.

magnitude of the contribution. Atoms that did not appear in any fingerprint are not marked with a disk.

You can change the number of KPLS factors for the display, to examine the effect of adding KPLS factors. You can change the color scale, either with the slider or the text boxes, to represent the range of values of interest, and you can adjust the limits on the range of values in the Color Scale Limits dialog box, which you open by clicking Limits. The Min and Max buttons open color selectors, so you can change the colors used in the display.

For more detailed information on the results for a particular structure, double-click it, to open the Analyze Atomic Contributions dialog box. In this panel, tool tips display the numerical values of the contributions for each atom. Double-clicking on an atom opens another dialog box that shows copies of the 2D structure molecule with each fragment in the fingerprint that contains the atom marked on the structure, and the contributions from the fragment and from the atom in the fragment.

As well as examining contributions, you can edit the structure to create a new structure, by clicking Edit Structure. When you click OK, the new structure is displayed, its fingerprint is calculated, and the contributions are reanalyzed and displayed. The observed value is still that of the original, but the predicted value is the value for the new molecule. You can then save the new molecule to the project by clicking Save to Project. It is added as a new row, with the predicted property value. This allows you to interactively optimize the property value.

3.5.4 Principal Components Regression

For principal components (PCA) regression, you can set the maximum number of principal components, and the dependent variables can be scaled automatically so that they lie on the same range.

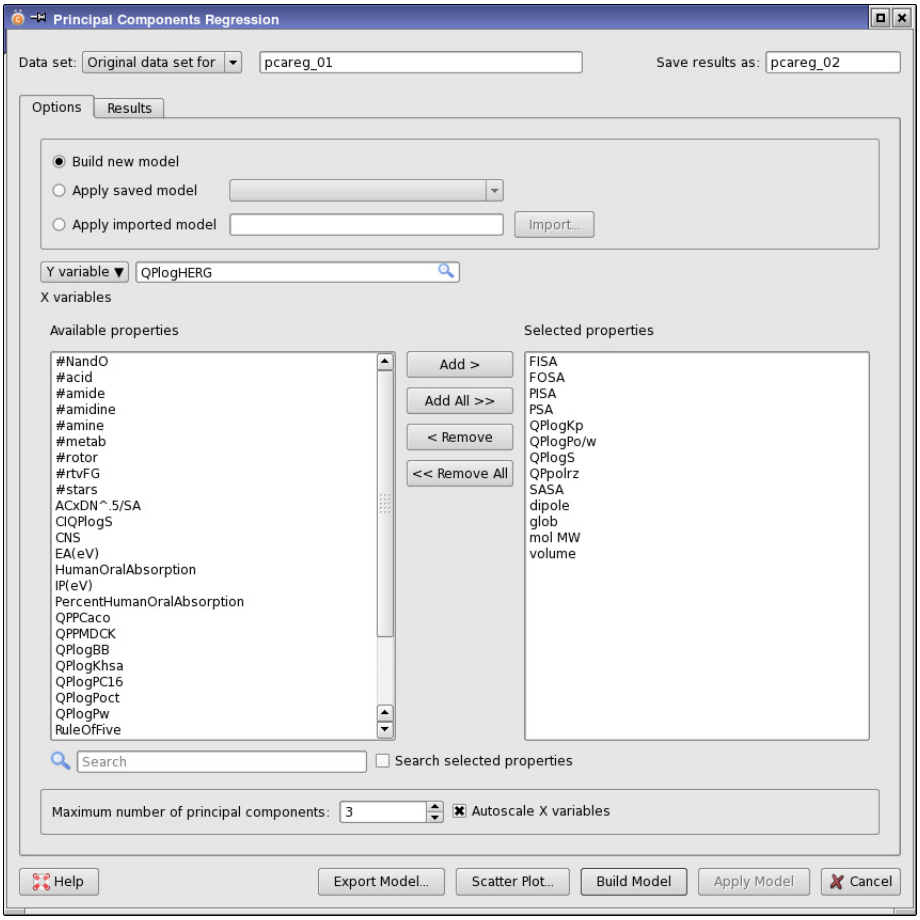


Figure 3.29. The Principal Components Regression dialog box

3.5.5 Bayes Classification

For Bayes classification, you can choose a Canvas fingerprint for the independent variables instead of selecting properties, and set options for partitioning the dependent variable. When a model is built, the tabs to the right of the structures table show the classes and the number of members of each class that were correctly classified and incorrectly classified.

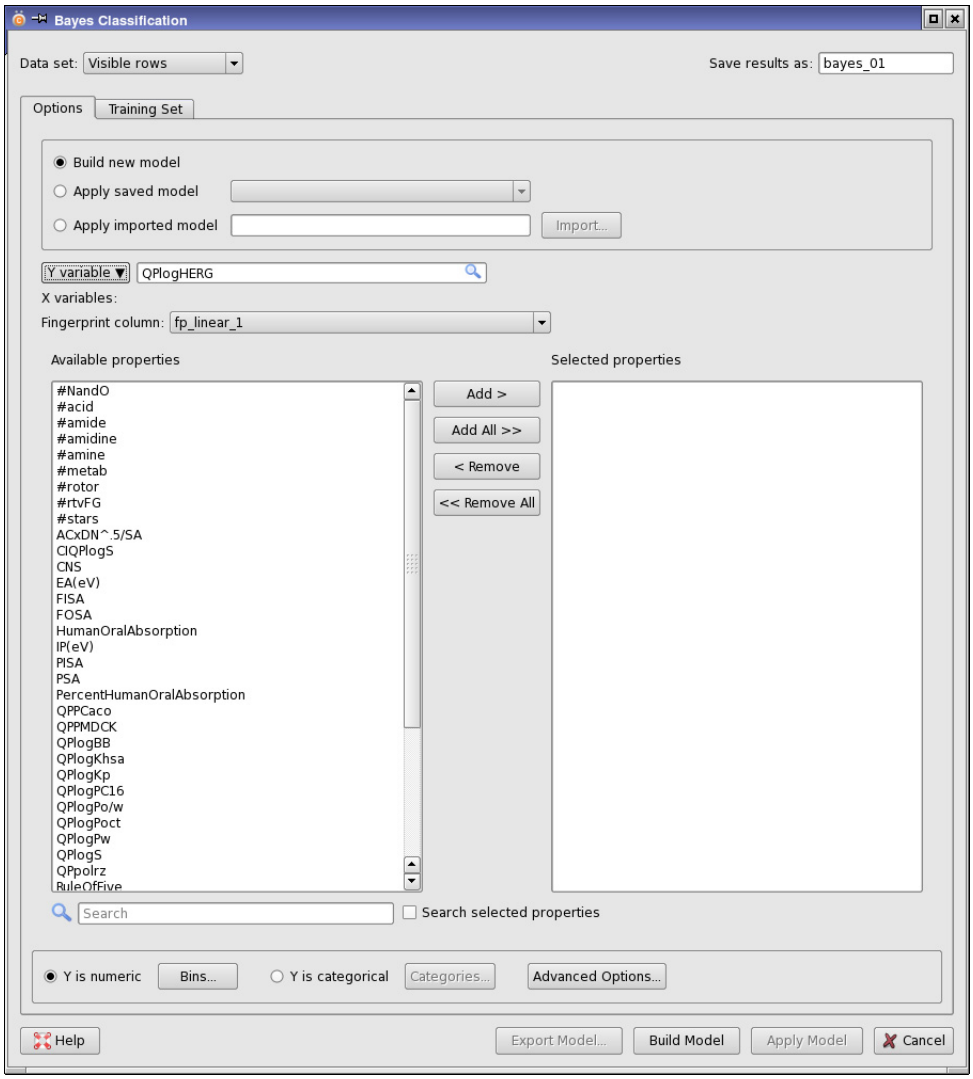


Figure 3.30. The Bayes Classification dialog box

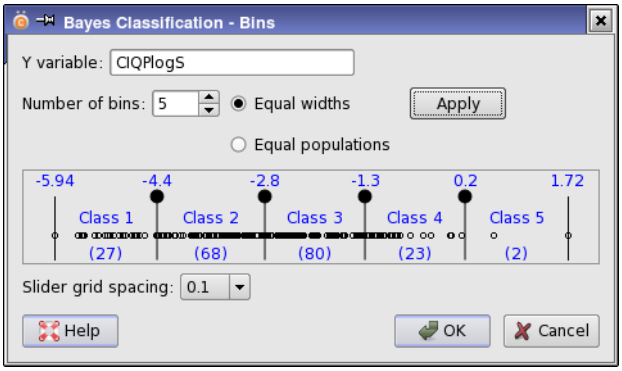


Figure 3.31. The Bayes Classification - Bins dialog box

If the dependent variable is numerical, you can choose whether to divide it into ranges that may cover more than one value by selecting Y is numeric, or you can treat each variable as a category by selecting Y is categorical. To assign the values to bins, click Bins, and make settings in the Bayes Classification - Bins dialog box. You can set the number of bins, and partition the variable so that the bins are of equal width or of approximately equal population. After making changes, click Apply. The display area shows information on the bins. The data points are plotted horizontally. The bin boundaries are represented by vertical lines, with a dot on the top. The value of the Y variable at the bin boundary is displayed above the boundary line. Labels for each bin, or class, are displayed above the data points and populations of each bin are displayed below the data points. You can add boundaries by clicking between the dots, and you can move boundaries by dragging the dots. The amount by which the boundary changes when you drag the dots is determined by the choice made from the Slider grid spacing option menu.

Parameters of the method can be set in the Bayes Classification - Advanced Options dialog box, which you open by clicking Advanced Options. You can set the smoothing coefficient, and for fingerprint data you can choose whether to apply distance cutoffs, or to keep bits in the top *N* percent. The methods are described in ref. 14.

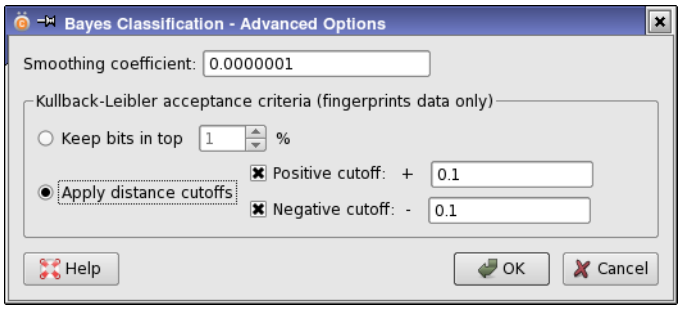


Figure 3.32. The Bayes Classification - Advanced Options dialog box

Note that running a Bayes classification with more than about 10 categories is unlikely to produce meaningful predictions.

3.5.6 Neural Networks

For neural networks, you can specify the number of networks to train, the number to keep, the number of training cycles, and the fraction of the training set to use for cross-validation, selected randomly.

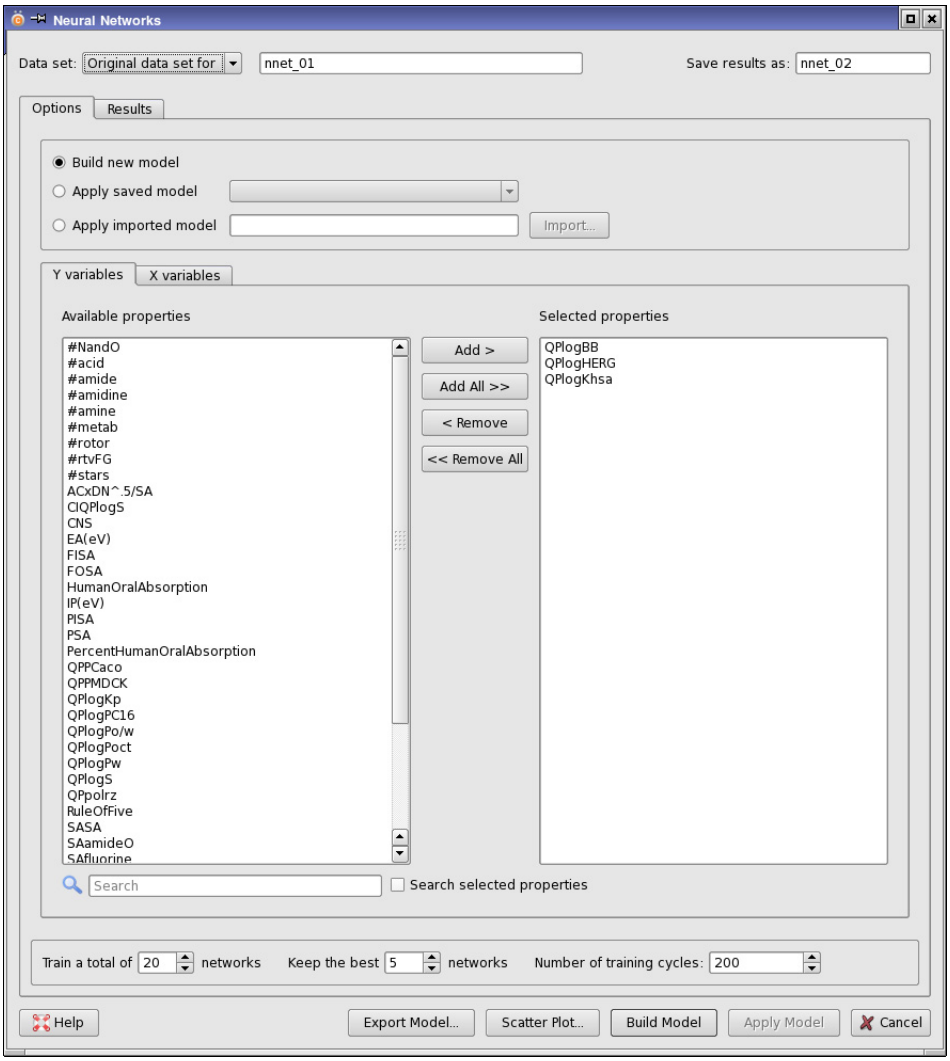


Figure 3.33. The Neural Networks dialog box

3.5.7 Recursive Partitioning

Recursive partitioning is a method like Bayes classification that can be used to build a model for the prediction of property ranges or categories, using a decision tree.

You can set options for partitioning the dependent variable and for building a single tree or an ensemble model, which generates multiple trees. Building an ensemble model filters out noise and corrects the biases in a single tree. When a model is built, the tabs to the right of the structures table show the classes and the number of members of each class that were correctly classified and incorrectly classified.

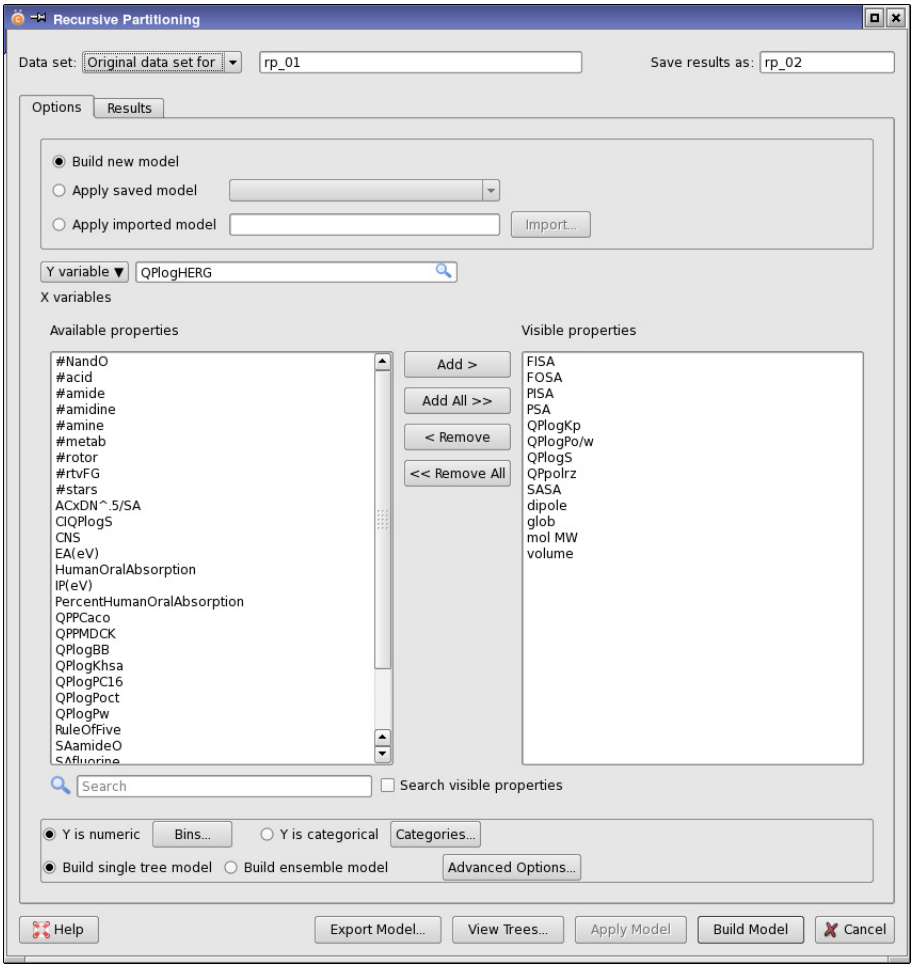


Figure 3.34. The Recursive Partitioning dialog box

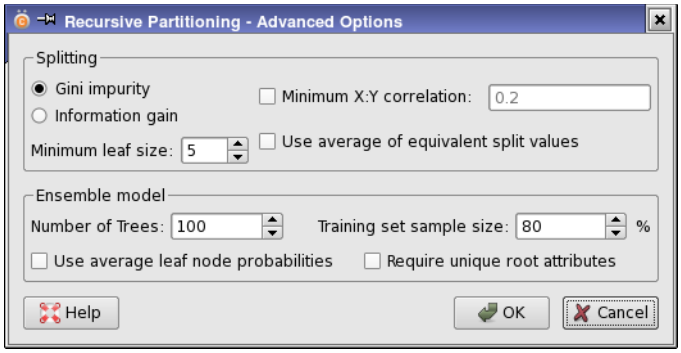


Figure 3.35. The Recursive Partitioning - Advanced Options dialog box

If the dependent variable is numerical, you can choose whether to divide it into ranges that may cover more than one value by selecting Y is numeric, or you can treat each variable as a category by selecting Y is categorical. To assign the values to bins, click Bins, and make settings in the Recursive Partitioning - Bins dialog box (This is the same as for Bayes classification—see Figure 3.31.) You can set the number of bins, and partition the variable so that the bins are of equal width or of approximately equal population. After making changes, click Apply. The display area shows information on the bins. The data points are plotted horizontally. The bin boundaries are represented by vertical lines, with a dot on the top. The value of the Y variable at the bin boundary is displayed above the boundary line. Labels for each bin, or class, are displayed above the data points and populations of each bin are displayed below the data points. You can add boundaries by clicking between the dots, and you can move boundaries by dragging the dots. The amount by which the boundary changes when you drag the dots is determined by the choice made from the Slider grid spacing option menu.

Parameters of the method can be set in the Recursive Partitioning - Advanced Options dialog box, which you open by clicking Advanced Options. You can choose the splitting method, filter out variables that have little correlation with the dependent variable, and set parameters for generating an ensemble model. The ensemble in Canvas can be compared to the random forest method, with the difference being that Canvas builds each tree from a different random sample of the training set, whereas random forest uses the full training set, but chooses a different random pool of X variables at each decision node.

3.5.8 Self-Organizing Maps

Kohonen self-organizing maps can be built and applied in the Self-Organizing Map panel, which you open from the Applications menu. The panel has some features in common with those of the regression and other models.

- The models can be built or applied for the selected rows or the visible rows, a saved view, or the data set from a previous run.
- The results are saved with the name that you specify in the Save results as text box. This name is also the job name and appears in the Project View panel.
- Existing maps can be applied to the data set. Maps from within the project can be loaded by selecting Apply saved map and choosing the map from the option menu. Only maps that match the X variables choice are listed on the option menu. External maps can be

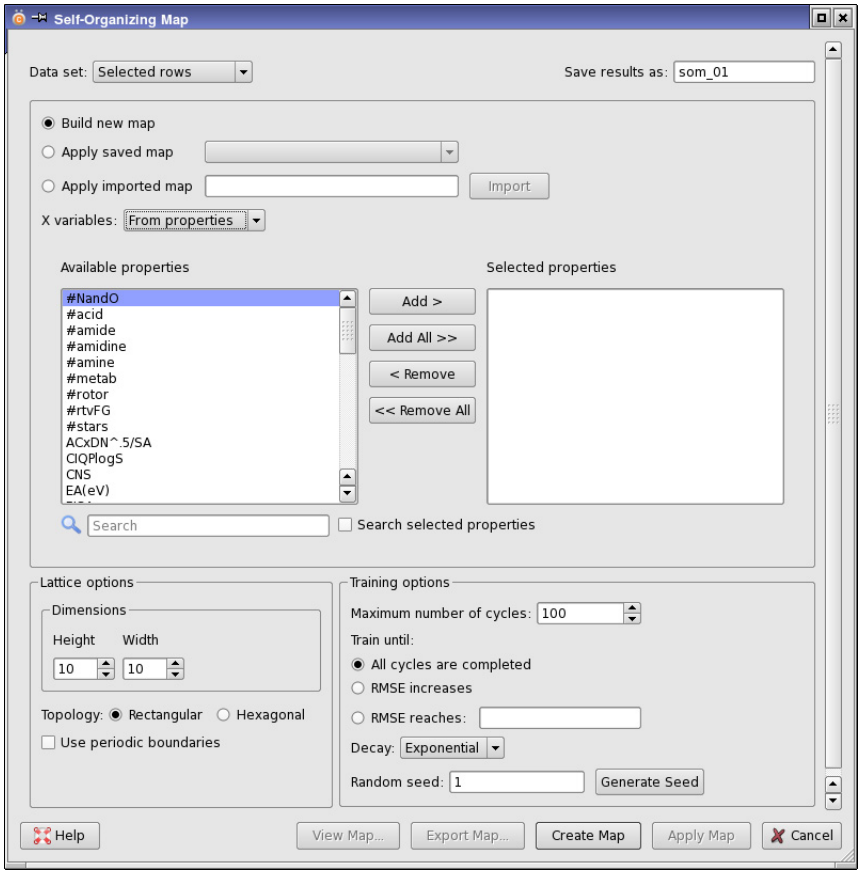


Figure 3.37. The Self-Organizing Map dialog box

loaded by selecting Apply imported map, clicking Import, and navigating to the file that contains the map. To apply the selected map, click Apply Map. The Self-Organizing Map Viewer panel opens and displays the results of applying the map to the current data set. When a map is loaded, you can export the map to a file by clicking Export Map.

To build a map, first select Build new map, then choose the type of X variable you want to use, from Numeric or Fingerprints. The panel is updated with the appropriate properties or fingerprints. You can choose a single fingerprint, or multiple properties.

In the Lattice options section, you can set the dimensions of the lattice and select the topology. You can also choose to use periodic boundary conditions.

In the Training options section, you can set the number of cycles, and specify a condition on the RMSE for stopping the training: either stop if the RMSE increases, or stop when the RMSE reaches a specified value. You can select the type of decay used to decrease the gain term as the training proceeds, and specify or generate a random seed for the initialization of the map.

Click Create Map to create the map. The dialog box closes, and the progress is logged in the Messages View panel.

To display the map, right click on the record in the Project View panel and choose View. The Self-Organizing Map Viewer panel opens with the map displayed. Alternatively, open the Self-Organizing Map panel again, select Apply saved map and choose the map from the option menu, and click View Map. You can have multiple viewer panels open at the same time.

The Self-Organizing Map Viewer panel provides several choices for coloring the map cells:

- Distance to selected cell—Color the map by the distance to the selected cell, which is outlined in red. Click on a cell in the map to select it.
- Cell population—Color the map by the number of structures in each cell.
- Cell property value—Color the cell according to the property chosen from the option menu.
- Average property value—Color the cell according to the average value of the property chosen from the option menu. This menu is populated by choosing Numeric from the Properties menu, and selecting the properties you want to use from the dialog box that opens.
- Category—Color the cell according to the number of structures in the cell with the selected value of the selected categorical property. The menu of categorical properties is populated by choosing Categorical from the Properties menu, and selecting the properties you want to use from the dialog box that opens.

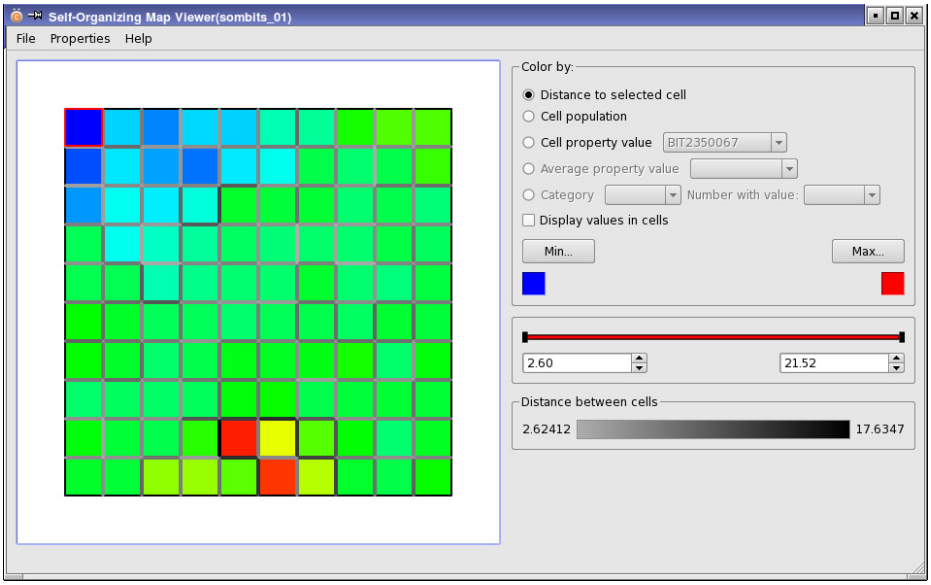


Figure 3.38. The Self-Organizing Map Viewer panel

The value that was used to determine the color for the cell can be displayed in the cell by selecting Display values in cells.

You can choose the colors for the minimum and maximum values of any quantity used to color the map by clicking the Min and Max buttons. Each button opens a color chooser. The color is displayed below the button. Intermediate values are interpolated on a spectrum.

The boxes below the bar display the minimum and maximum values of the property used to color the map. You can adjust these values so that the cells with values outside the range specified in the boxes are colored by the colors for the minimum and maximum. The bar above these boxes indicates the range of possible values for which a color ramp is applied. You can drag the sliders on the bar to adjust the minimum and maximum property values used.

The shade of gray in the cell borders indicates the distance between adjacent cells: the darker the border, the larger the distance. A legend of the gray shades is displayed in the Distance between cells section.

When viewing properties other than the distance to a cell, you can right-click a cell to display a custom view of the structures that are members of that cell. If you want to add members of other cells to the view, control-right-click the cell you want to add. Cells for the current view are outlined in yellow. You can add properties to the view by choosing View → Manage Properties, and selecting the desired properties.

Cell memberships can be exported to the spreadsheet or a file with File → Export. The cells are numbered from left to right, top to bottom, starting with 1. The Cell property contains the cell number for each structure, and the Total property contains the number of structures in the cell.

An image of the map can be saved in PNG, JPEG, or TIFF format by choosing File → Save Image. A dialog box opens that allows you to name the file and set the dimensions of the image, in pixels.

3.6 Other Applications

The remaining applications on the Applications menu are described in this section.

3.6.1 Principal Components Analysis

Analyzing the principal components of a set of independent variables can help in the selection of a subset of variables for use in other applications. The analysis can be done in the Principal Components Analysis dialog box, which you open from the Applications menu.

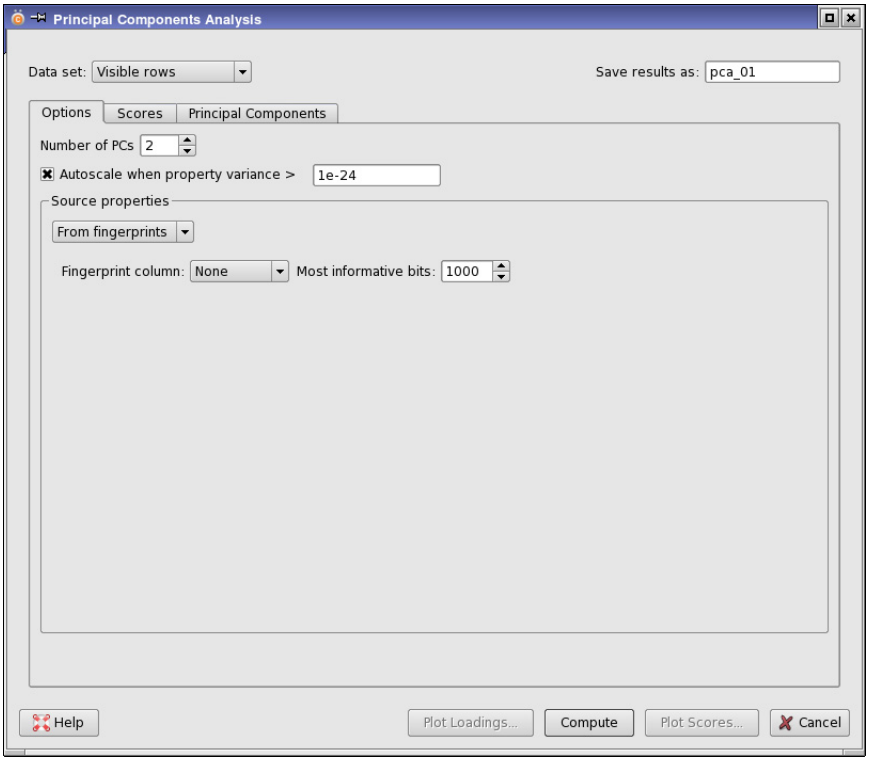


Figure 3.39. The Principal Components Analysis dialog box

To set up the analysis, choose the Data set option for the rows you want to use in the analysis, provide a name in the Save results as text box, specify the number of principal components (PCs), and choose a set of numeric properties or a fingerprint in the Source properties section.

You can select the Autoscale option to scale the property values by dividing by the standard deviation if the variance is greater than the specified value.

For fingerprint analysis, you can specify the number of most informative bits to keep. This reduces the time taken for the analysis.

The job to generate the principal components and compute the scores and loadings is run when you click Compute.

When the job finishes, the components are displayed in the Principal Components tab, showing the eigenvalue, cumulative variance, and coefficients for each property. The principal component scores for each of the structures are shown in the Scores tab. If you pause the pointer over the structure name, the 2D structure is displayed in a tool tip. To export the scores to a file or the spreadsheet, select the columns that you want to export and right-click in the table. The Export Selection dialog box opens, in which you can choose the destination, and for export to the spreadsheet, the property prefix.

You can plot the scores or the loadings by clicking Plot Scores or Plot Loadings. In both cases, a Scatter Plot panel opens, displaying the scores or loadings for two principal components. You can represent values for two more principal components by the color and size of the plot symbols. See Section 2.7.1 on page 48 for more information on Scatter Plot panels. The name and structure of the molecule is displayed in a tool tip for each point in the scores plot, and the name of the property is displayed in a tool tip for each point in the loadings plot.

3.6.2 Finding the Maximum Common Substructure

The task of finding the maximum common substructure (MCS) of a number of structures can be run from the Maximum Common Substructure dialog box, which you open from the Applications menu. As for other applications, the structures that are processed can be the selected or the visible rows in the spreadsheet, a saved view, or structures from a previous run. If you choose Selected rows or Visible rows, you must ensure that the rows are selected or visible before you open this dialog box. You can name the job in the Save results as text box.

The search does not need to encompass all the structures in the set. You can specify a minimum number of structures that must match the substructure and a maximum number that must match. If the minimum is greater than the number of structures in the set, all structures must match. If the minimum is less than the maximum, a series of substructures (MCS groups) is found for each number of structures from the minimum to the maximum. You can restrict the groups so that any structure is only in one group, which is chosen to be that of the largest MCS.

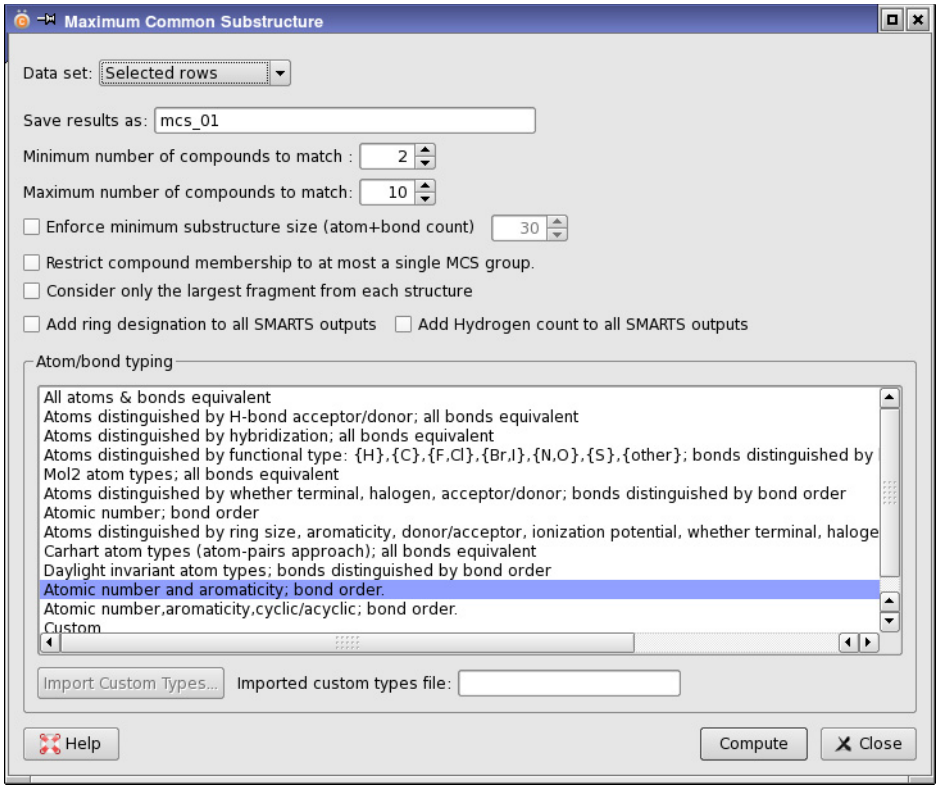


Figure 3.40. The Maximum Common Substructure panel

You can set a lower limit on the size of the MCS, in terms of the combined number of atoms and bonds. If the structure contains more than one molecule (fragment), you can restrict the search to the largest fragment.

The matching is done using a set of definitions for the atom types and the bond types. Twelve definitions are provided in the Atom/bond typing list, and you can add your own scheme by choosing Custom, then clicking Import Custom Types to load the definition file, which must consist of SMARTS strings that define the atom or bond type. The schemes are described in Table 5.7 on page 125; all except E can be used for MCS searches.

The results include the SMARTS strings for the substructures, the number of substructures in each group, and the membership of the groups. You can add ring and hydrogen atom designations to the SMARTS strings by selecting the appropriate options. By default they are not included.

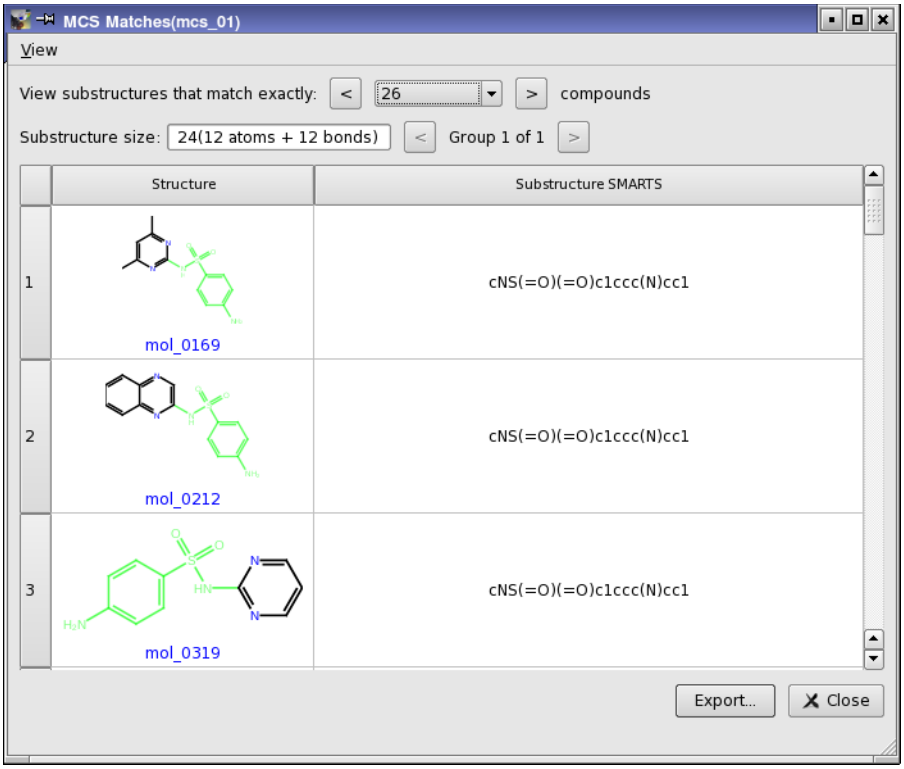


Figure 3.41. The MCS Matches view

To view the results, right-click on the job record under MCS in the Project View panel, and select View. The MCS Matches panel opens, displaying the structures and the substructure SMARTS patterns. The substructures are marked in green on the structures. You can view the substructures that match different numbers of structures, and if there is more than one group for a given number of molecules, you can display each group.

If you want to add properties to the view, choose View → Manage Properties, and select the desired properties in the dialog box that opens. The property is added to all subsequent views of MCS matches.

The set of structures displayed can be applied to the master view by choosing View → Apply to Master. The master view displays only the structures in the MCS Matches panel, and these structures are selected.

You can export the structures to a Maestro, SD, or CSV (SMILES + Properties) file by clicking Export. A file selector opens, in which you can choose the file format, navigate to a location, and name the file.

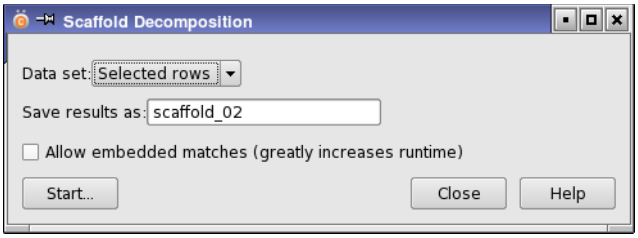


Figure 3.42. The Scaffold Decomposition dialog box.

3.6.3 Scaffold Decomposition

Another way of finding common structural features in a set of structures is to decompose each molecule into a set of scaffolds, then make a list for each scaffold of the structures that contain that scaffold. You can perform this scaffold decomposition in the Scaffold Decomposition panel, which you open from the Applications menu. In this panel you can choose the structure source from the Data set option menu, and give a name to the results.

The decomposition is done by first removing side chains from all ring systems, to define the largest scaffold in the structure. This structure is then split into all possible subscaffolds by breaking bonds and removing linkers between rings, other than those in fused rings. The list of scaffolds is collated, and then for each scaffold, the structures that contain that scaffold are located and added to a list.

When finding the structures that contain a scaffold, you can choose to match scaffolds to a fused ring system in a structure that contains the scaffold. This is called an “embedded match”, and you can allow them by selecting Allow embedded matches. For example, if the list of scaffolds contains imidazole and benzimidazole, a structure that contains benzimidazole is included in the list of structures that contain the imidazole scaffold, if embedded matches are allowed. Identifying embedded matches takes substantially more time than not doing so.

For more information on the process, see Section 5.6.8 on page 184.

After the job finishes, you can view the results in the Scaffold Decomposition of *jobname* panel, by right-clicking on the job and choosing View.

The Scaffold Decomposition of *jobname* panel shows the 2D structures of the scaffolds in a table. The scaffolds can be classified by the number of rings in each scaffold, or by the number of ring systems. The scaffolds for a given classification are shown on the same table row. You can select scaffolds in the table, and limit the display to show only those scaffolds. Each time you limit the display (or remove the limits), that view of the table is stored, and you can use the arrow buttons to step through these views. If you want to see all the subscaffolds or superscaffolds of a particular scaffold, right-click in its table cell and choose the appropriate command.

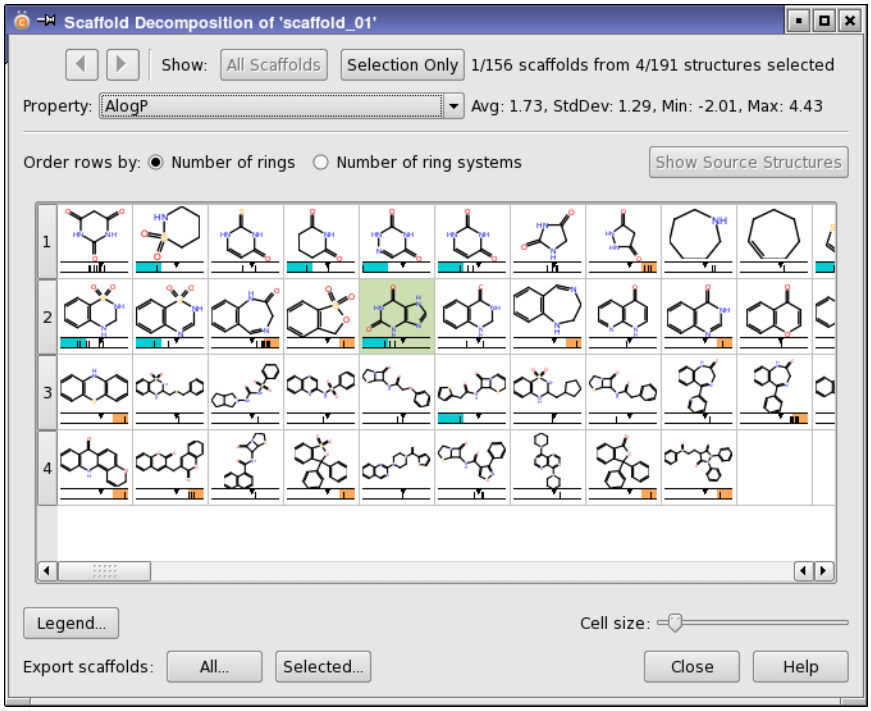


Figure 3.43. The Scaffold Decomposition of jobname panel.

The table also shows the property distribution of the structures that contain each scaffold, for a property that you can choose from the Property option menu. The average, standard deviation, minimum, and maximum values of the property are displayed to the right of the option menu. The property distribution is displayed for each scaffold as a "stick spectrum" in a bar that represents the full range of the property. For each structure there is a vertical line at the property value for that structure. The average property value is marked, and the region outside one standard deviation from the average is colored if it contains vertical lines for property values.

You can export the scaffolds to a structure file. The scaffolds are stored internally as canonical SMILES strings, so if you choose Maestro or SD format for the file, they are exported as 2D structures without hydrogens, except where the hydrogens are part of the SMILES string.

If you want to see the source structures for a set of scaffolds, select them in the table and click Show Source Structures. The Source Structures panel opens, and displays the structures for those scaffolds, arranged in rows with the same classification as for the scaffolds, and including the property distribution display. You can export selected structures or all structures to a file. You can run an R-group analysis on the selected structures or on all structures in the

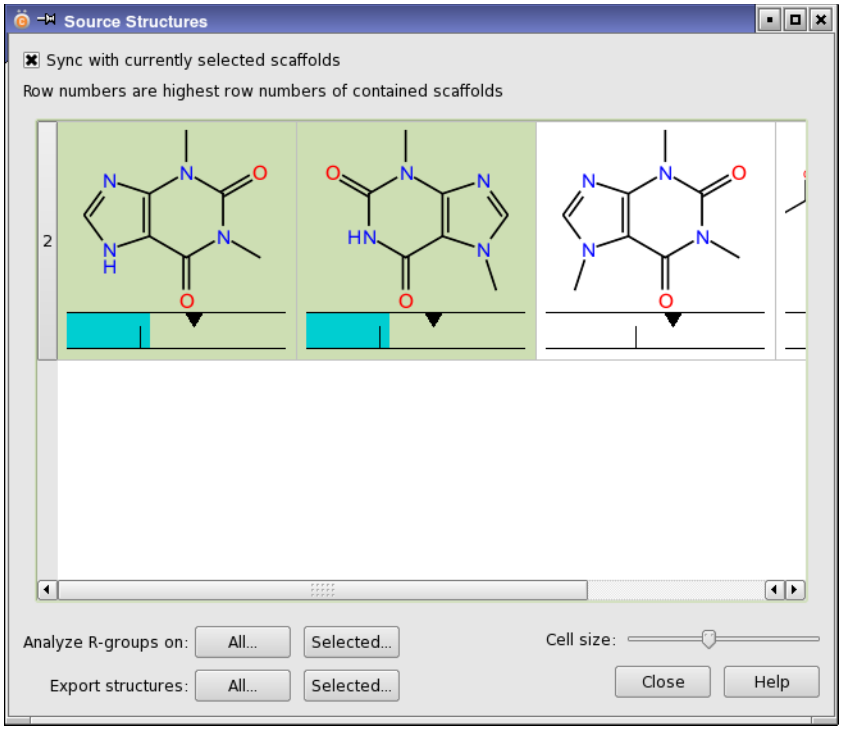


Figure 3.44. The Source Structures panel.

panel. If the structures have more than one scaffold in common, you are prompted to choose the scaffold you want to use for R-group analysis.

3.6.4 R-Group Analysis

R-group analysis identifies the R groups that are attached to a scaffold for a set of structures, and presents analyses of properties as a function of the attachment points and the R groups at each point. This application is described in a separate document, *R-Group Analysis*.

3.6.5 Running a 3D Minimization

The structures that are imported into Canvas or created in Canvas do not have to be 3D structures. You can convert them into 3D structures and minimize them with the 3D Minimization dialog box, which you open from the Applications menu.

You can minimize the structures in the selected rows or the visible rows, a saved view, or the data set from a previous run. When the minimized structures are returned, you can replace the existing structures or append the minimized structures to the project, with an optional suffix to

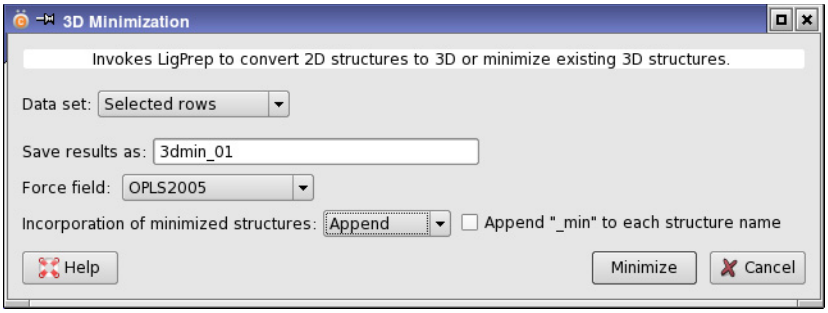


Figure 3.45. The 3D Minimization dialog box.

the name. You can export them to an external file in Maestro or SD format. The extension you give to the file name determines whether it is compressed or not. If no extension is given, the files are compressed, with extension `.maegz` for Maestro format and `.sdf.gz` for SD format.

The minimization process uses LigPrep, for which a license is required. LigPrep can generate stereoisomers, tautomers, and protonation states, and search for ring conformations, as part of the 2D-3D conversion process. In this application, it retains the tautomeric and protonation state, and searches for the lowest-energy ring conformation. If stereochemical information is provided with the 2D structure, it uses that information, but if the information is not available, it generates a stereoisomer derived from the stereochemistry observed in a database of natural products.

Two force fields are available for the minimization of the final structure: MMFFs and OPLS_2005. OPLS_2005 supports a wider range of elements than MMFFs.

3.7 Running External Applications

If you want to run an external application and incorporate the results into Canvas, you can do so with the External Application dialog box, which you open from the Applications menu.

In common with other applications panels, you can use the selected rows or the visible rows, a saved view, or the data set from a previous run for the structure input to the application. If the application requires certain properties as input, you can select the properties to use with the property selection tools in the Application input section. The structures and properties are written to a file, which you can name in the Application input section. The file is written to a special directory in the Canvas project, and the full path to the file is used when the command for the application is executed. You can choose a file format from the Format option menu. The application must be able to read one of the formats on this option menu.

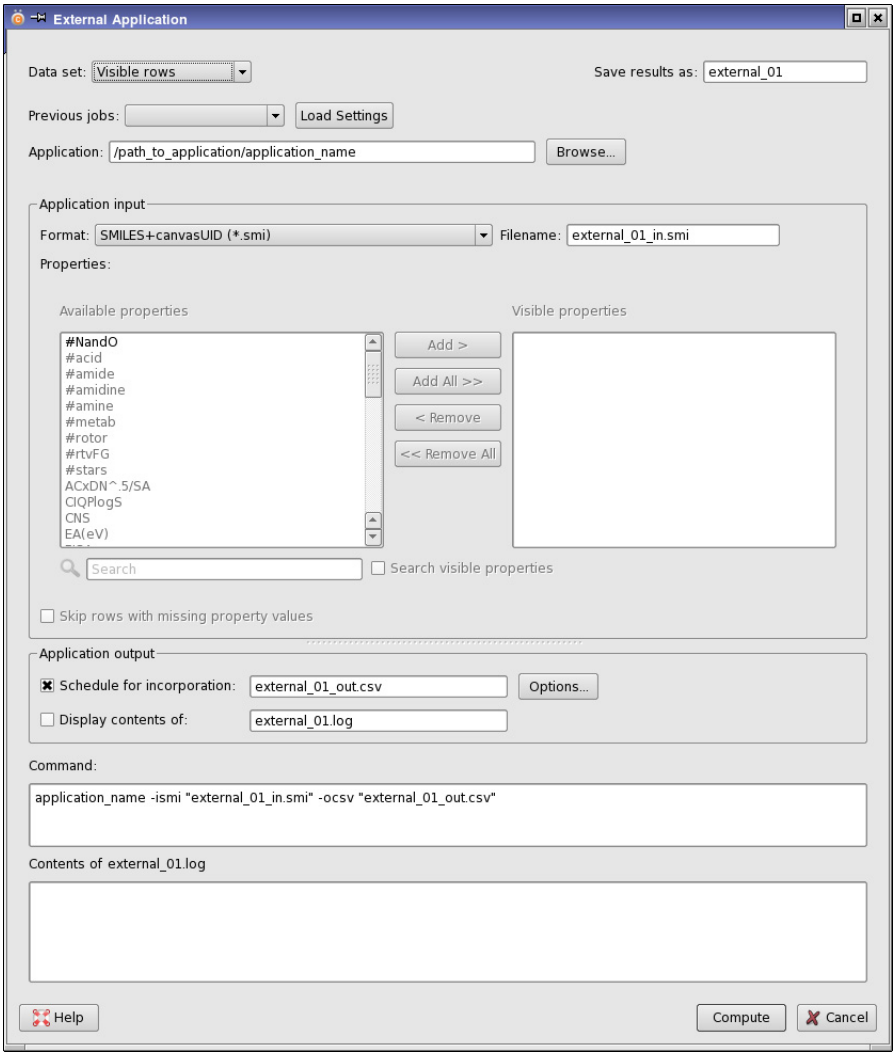


Figure 3.46. The External Application dialog box

The application can be specified in the Application text box as a full path, or you can click Browse and navigate to the application. If the application makes use of Schrödinger software, libraries, or APIs, you can select Invoke with $SCHRODINGER/run so that the Schrödinger environment is set up automatically.

If you want the results to be incorporated back into the Canvas project, select Incorporate in the Application output section and specify the name of a CSV file in the text box. In order to incor-

porate results into Canvas, the application must generate a CSV file that has a header row, the SMILES string for the structure in the first column, and the structure name in the second column. The following columns can contain the properties generated by the application. If the application cannot process one of the structures, it must still write a row for that structure in the CSV file, with the SMILES string, the structure name, and the correct number of commas. You can also display a text file in the Contents of text area when the job finishes, by selecting Display contents of and specifying the file name in the corresponding text box. As for the input file, the output file and the displayed file are written to a directory in the Canvas project, and the full path is used when the command is run.

The default file names are constructed from the label given in the Save results as text box. This label is used to identify the calculation in the Project View, under External Application.

The command for running the application is specified in the Command text area. You do not need to specify the path to the application, because it is specified in the Application text box, and the path is prepended to the application name when the command is executed. Likewise, you do not need to specify the path to the input file or the output file, because the path is prepended to the file names. A sample command is given in the Command text area, which you can edit to provide the correct syntax for running the command, including the method of specifying the input and output file. The names of these two files are automatically supplied and updated from the relevant text boxes in the Application input and Application output sections.

All files must be specified as command arguments, because the command is not run in a shell. The application must therefore process all the arguments you give for the command, and not use shell redirection for input or output.

If you choose to display the contents of a file in the panel, that file must appear in the command. The command can include other arguments used by the application, and it can specify other files needed by the application. Any files other than those specified in the panel must include the full path.

When you are ready to run the application, click Compute. The controls in the panel are made unavailable, except for the Stop button. The panel must remain open until the run has finished. If you chose incorporation, the results are automatically incorporated into the Canvas project.

If you want to run a similar job, or correct the settings, you can select Use settings from and choose a run from the option menu. The panel is set up with the settings from the selected run, which you can change before clicking Compute. This option does the same as cloning a job.

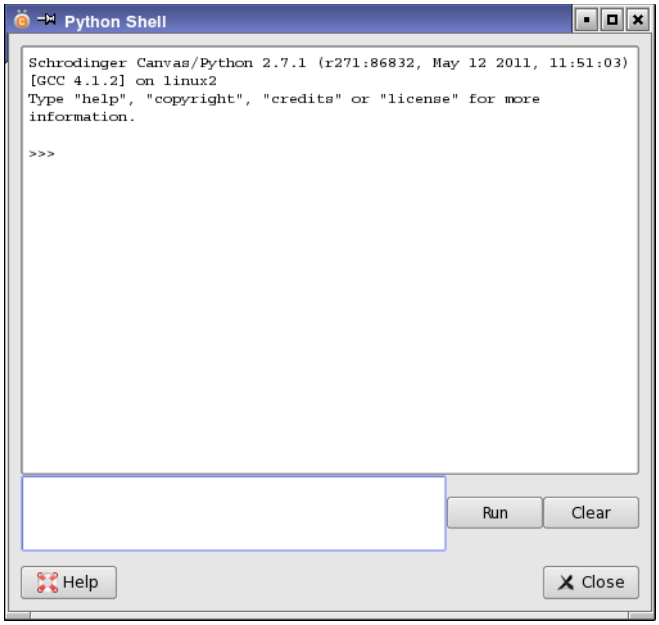


Figure 3.47. The Python Shell panel.

3.8 Running Python Scripts

Canvas provides two ways of running Python scripts from within the GUI.

You can install your own Python scripts on the Scripts menu with Scripts → Install and manage them (delete, reload) with Scripts → Manage. Once installed, you can run them by choosing the item on the Scripts menu. Scripts can be downloaded from the Script Center at http://www.schrodinger.com/scriptcenter, or you can create your own. In general, the scripts should run without the need to supply arguments. The scripts are run in the Schrödinger environment, so they can use any of the *Canvas Python API* modules. The mechanism for installing and managing the scripts is the same as used in Maestro—see Section 15.1 of the *Maestro User Manual* for more information. However, scripts that run in Maestro will not in general run in Canvas, because they do not have access to the Maestro libraries.

You can run Python scripts and Python commands in a Python interpreter, which you open by choosing Python → Open Shell or typing CTRL+P. This interpreter uses the version of Python that is in the Schrödinger software distribution, and various Schrödinger and Canvas modules are imported by default.

An overview of the general Schrödinger Python modules and a detailed description of those modules is available from the Help menu.

Using Canvas

Canvas has many possible uses. In this chapter, instructions are given for some of the scenarios in which Canvas can be used to solve a particular problem.

4.1 Selecting Compounds Not Represented in a Library

For purchasing new compounds, it can be useful to determine which compounds in a given set (available for purchase, for example) are not represented in your existing library. You can use diversity-based selection to find these compounds and export the structures to a file as follows. It is assumed that the two libraries are in separate files.

1. Import the existing library and the pool of compounds to choose from into a new Canvas project.

   When you import the libraries, ensure that you create a view from each library (select Store each input file as a view).

2. Create a fingerprint for all compounds, with Applications → Binary Fingerprints. The fingerprint is used for assessing the diversity.

3. Select all the compounds in the pool that you want to choose from.

   You can do this by opening the view for the pool of compounds, choosing View → Apply to Master, and selecting the Rows option Select current.

4. Open the Diversity-Based Selection dialog box from the Applications menu.

5. For Data set, choose Selected rows.

6. Choose the fingerprint you generated from the Fingerprint column option menu.

7. For the Diversity selection method, choose Sphere.

8. Enter the number of compounds you want in the Diverse subset size text box.

9. For the Initialization method, choose Existing structures.

10. For Use structures from, choose the custom view that contains the existing library.

11. Click Choose Compounds.

    A diverse collection is selected that avoids the structures in the existing library, i.e., all diverse compounds will lie outside the spheres of the existing library structures. The collection is displayed in a new custom view.

12. Choose File → Export, or click the Export button in the new custom view, to export the diverse collection to a file.

If the existing library is already in a project, you can open the project, create a view from all the compounds in the project, then import the pool of compounds you want to choose from, ensuring that you create a view when you import it. Then you can proceed from step 2 above. At the end, you can delete the compound pool if you want to keep your library project. Alternatively, you could import the existing library project into a new Canvas project, create a view, and then import the compound pool.

4.2 Removing Duplicate Structures from a Project

1. Choose Structure → Detect Duplicates.

   A new column, labeled Duplicate of is added to the spreadsheet.

2. Choose Data → Property Filter.

   The Property Filter dialog box opens.

3. Ensure that Current view is selected for Store results in.

4. Choose Duplicate of for the property under Show rows where.

5. Choose has a value for the operator.

6. Click Add.

   The filter string `Duplicate of has a value` is added to the list.

7. Click Run.

   The dialog box closes and the rows for duplicate structures are now selected in the Master View.

8. Choose Edit → Delete.

   The duplicates are deleted from the project.

4.3 Choosing a Fingerprint

Choosing which fingerprint to use and the options to set can be difficult, as no single setting is best for all targets. Some large scale screening studies [5,6] reported the following findings:

- MOLPRINT2D, Dendritic, Radial, Triplet, and Pairwise fingerprints performed better than others.
- More specific atom typing schemes, such as Daylight, Mol2, Carhart, were better.
- Collisions in the fingerprints significantly degraded performance, so using more bits is better.
- Combining complementary fingerprints can minimize the risk of choosing the wrong fragment.
- Combining actives into a modal fingerprint almost always increases the retrieval rate.
- Not much sensitivity was found to bit scaling or the similarity or distance metric.

4.4 Creating Modal Fingerprints

Modal fingerprints [6] are fingerprints that are averaged over several structures, thus combining the information from several query molecules into a single fingerprint.

To create a modal fingerprint:

1. Create a binary fingerprint with the type of your choice.
2. Select the rows in the spreadsheet for the structures whose fingerprints you want to average to create a modal fingerprint.
3. Right-click on the fingerprint column heading, and choose Create Modal Fingerprint.
4. Choose Selected rows from the Domain option menu.
5. Enter a name for the row in which the modal fingerprint will be stored
6. Click OK.

A new table row is created, with the name you chose displayed in the Structure column, and the modal fingerprint in the fingerprint column you used.

4.5 Tips

This section contains some tips for common operations.

To define training and test sets before building a model:

- Create a partition from a property with classes for the training set and the test set, and select Assign from partition to assign the sets in the model-building panel.

To copy model predictions to the spreadsheet:

- Right-click in the structures table and select To spreadsheet.

To run an application using a custom view:

- Choose View → Apply to Master in the custom view and select Select current, then run the application on the selected rows.

To add properties to an application view:

- Choose View → Manage Properties, and select the properties you want to add.

Running Applications from the Command Line

All the Canvas command-line programs are available in `$SCHRODINGER/utilities`. The syntax statements below are therefore given with respect to this directory.

Canvas programs generally run as standalone processes in the foreground. Some programs have Job Control options, which allow you to specify a job name and run the program under Job Control. This gives you access to all the Job Control features, including running the job on a remote host.

Canvas supplies two utilities, `canvasJob` and `canvas_app`, that allow you to run jobs that read from and write to a Canvas project. The list of programs you can run in this way is given in Section 5.6.3 on page 174 with the description of `canvasJob`.

The default amount of memory used by Canvas applications is approximately 500 MB. To change this value, set the environment variable `SCHRODINGER_CANVAS_MAX_MEM` to the desired number of bytes (MB and GB units are not permitted). For information on setting environment variables on Windows, see Appendix A of the *Installation Guide*.

5.1 Common Syntax Descriptions

Many of the programs share common syntax, especially those that perform similar tasks. This section describes some elements of common usage.

A range specification is a comma-separated list of indices or ranges, with no spaces. A range is defined by the endpoints of the index range, separated by a colon. Examples of valid range specifications are:

1,4	structures 1 and 4
1:10,14	structures 1 through 10 and 14
2:	structures 2 through the end of file
:5,13:18	structures 1 through 5 and 13 through 18

In the option syntax, ranges are often used for selecting structures from a file, and are denoted *structureRange*.

Structure files can often be specified in either compressed or uncompressed format. These files are specified on the command line with `-i`*fmt* for input or `-o`*fmt* for output. The file types that are generally accepted by Canvas, listed with the value of *fmt* and the allowed extensions, are:

`mae`	Maestro file (`.mae`, `.maegz`, `.mae.gz`)
`sd`	SD file (`.sdf`, `.sd`, `.sdf.gz`, `.sd.gz`)
`csv`	Comma-separated value file including the structure in SMILES format, molecule name and properties (`.csv`, `.csv.gz`); can use comma, tab or space for the delimiter
`smi`	Space or tab-separated SMILES and molecule name (`.smi`, `.smi.gz`)
`proj`	Canvas project (`.cnv`).

When CSV files are used by a program, it is usually necessary to specify the delimiter that separates the fields on each row. You should use `' '` for space, and `'\t'` for tab. The single quotes are required for these two specifications. If the delimiter is a space, consecutive spaces are ignored.

Programs that take a range of structure files as input have a common input file argument syntax, which is described in Table 5.1. The output file syntax varies more, according to the purpose of the program. However, there are some common options associated with particular file types, which are listed in Table 5.2.

Table 5.1. Common input structure file arguments.

Argument	Description
`-i`*fmt inFile*	Structure file containing the input molecules. Required. The supported formats *fmt* are: `smi` SMILES string and name, separated by tab or space `sd` SD file `mae` Maestro file `csv` CSV file containing SMILES, name, and properties Valid file extensions are listed on page 119.
`-fieldAsName` *field*	Property in the SD or Maestro input file to use as the molecule name. For Maestro files, *field* must start with `s_` or `i_`. Cannot be used with `-ismi`, or `-icsv`. Default: `s_m_title` property for Maestro files, first line of CT block for SD files.
`-noHeader`	CSV input file does not have a header line.
`-d` *delimiter*	Delimiter used in CSV input file. Valid delimiters are comma (`','`), space (`' '`), and tab(`'\t'`). Default: comma.
`-smi` *SMILESCol*	Field in CSV input file that contains SMILES strings, either by name if there is a header row, or by column index starting at 1. Default: column 1.
`-name` *nameCol*	Field in CSV file to use as the molecule name, either by column name or by column index. Default: column 2 if SMILES is present, column 1 otherwise.

Table 5.2. Common output structure file arguments

Argument	Description
`-o`*fmt outFile*	Structure file containing the output molecules. The supported formats *fmt* are: `smi` SMILES string and name, separated by tab or space `sd` SD file `mae` Maestro file `csv` CSV file containing SMILES, name, and properties Valid file extensions are listed on page 119.
`-od` *delimiter*	Delimiter used in CSV output file. Valid delimiters are comma (`','`), space (`' '`), and tab(`'\t'`). Default: comma.
`-v3`	Use MDL version 3 format for output SD files.

Many of the programs can run under Job Control. For these programs, the syntax descriptions include *job-options*. Where the options are supported, the common set of supported options are listed in Table 5.3. With the exception of `-JOB` and `-MINREC`, these options are described in full in Table 2.1 and Table 2.2 of the *Job Control Guide*.

Most programs run on a single processor. Those that can be distributed over multiple processors are indicated in the description of the program.

Table 5.3. Common Job Control options.

Option	Description
`-JOB` *jobName*	Job name. If present, run the job under Job Control. If omitted, no other job control options are permitted.
`-HOST` *host*	Run job on *host*. The format *host*`:`*n* requests use of *n* CPUs on *host*.
`-LOCAL`	Store temporary job files in current directory.
`-MINREC` *nrec*	Minimum number of records per CPU. Prevents submission of a large number of subjobs that each contains only a small number of records. Only available for programs that support use of multiple CPUs Default: 100.
`-TMPDIR` *dir*	Store temporary job files in *dir*.
`-WAIT`	Do not return control to the shell until the job finishes.
`-NICE`	Run job at reduced priority.

5.2 2D Fingerprints

The programs available for 2D fingerprints are listed in Table 5.4, and are described in detail in the following sections.

Table 5.4. Tools for 2D fingerprints

Tool	Description
`canvasFPGen`	Generate fingerprints for a set of molecules.
`canvasFPCombine`	Combine fingerprints for distinct or overlapping sets of molecules.
`canvasFPBinary2CSV`	Extract fingerprints to a CSV file.
`canvasCSV2FPBinary`	Convert a CSV file to canvas binary fingerprint file.

5.2.1 canvasFPGen

This program generates fingerprints for molecules in a structure file. The syntax of the command is as follows.

`canvasFPGen` *inputFileArgs outputFileArgs* [*job-options*] [*options*]

The input file arguments are the common arguments listed in Table 5.1. The alternatives for specifying the output file are listed in Table 5.5. The job options are standard Job Control options, listed in Table 5.3. This program can be distributed over multiple processors. The options are listed in Table 5.6.

Table 5.5. Output file arguments for the canvasFPGen command.

Argument	Description
`-o` *fpFile*	Binary output file for generated fingerprints that stores only molecule name and fingerprint bits.
`-odata` *fpFile*	Binary output file for generated fingerprints that also includes additional data fields. If the input file is in SMILES format, there are no such additional fields.
`-ocsv` *fpFile*	Write MACCS or custom fingerprints directly to a CSV file. Column headers consist of either the SMARTS patterns in the fingerprint definition file or the optional label that follows each SMARTS pattern in the fingerprint definition file.
`-fieldOnly` *fields*	Space-separated list of property names (fields) from a Maestro or SD input file to write to the output file. Only valid with `-odata`.
`-uniform`	Each structure block in a Maestro or SD input file contains the same set of properties.

Table 5.6. Options for the canvasFPGen command.

Option	Description
`-fptype` *fpType* [*customfile*]	Fingerprint type. Allowed values are: `linear`, `maccs`, `radial`, `molprint2D`, `torsion`, `pairwise`, `triplet`, `dendritic`, and `custom`. `custom` must be followed by the name of the file that contains the SMARTS patterns for the fingerprints (1 per line). Default: `linear`.
`-xp`	Represent fingerprints using 64 bit precision. This reduces collisions of "on" bits, but doubles the space required to store each key.
`-fill`	Fill in the line for a molecule that fails to generate fingerprints in the output files. No data values are included on this line, apart from the molecule name. This option is useful to create placeholders that preserve positional alignment to external data. Default: skip failed molecule in the output.
`-compress`	Use frequency-based compression to reduce required storage by approximately tenfold.
`-3D`	Use actual 3D coordinate distances instead of topological distances. Applicable only to `pairwise` and `triplet` fingerprint types. Requires a Maestro or SD input file with 3D coordinates.
`-atomtype` *scheme* [*customfile*]	Atom typing scheme. Must be an integer value between 1 and 12 or C or E. See Table 5.7 for details. For C, *customfile* must be given. Defaults for each fingerprint are `linear`:10, `radial`:4, `torsion`:10, `pairwise`:9, `triplet`:10, `molprint2D`:5, `dendritic`:10.
`-path` *pathLength*	Maximum path length for `linear`, `dendritic`, and `molprint2D` fingerprint types. Defaults: 7 (`linear`), 5(`dendritic`), 2 (`molprint2D`).
`-ring` *ringSize*	Maximum linear ring path. Valid only with `linear` fingerprint type. 0 means no ring closure. Default: 14.
`-minpath` *pathLength*	Minimum path length for `linear`, `dendritic`, and `molprint2D` fingerprint types. Defaults: 0 (`linear`, `dendritic`), 2 (`molprint2D`).
`-halfstep`	When growing linear fragments, additionally include growth by bonds alone. Valid only for `linear` fingerprint type.
`-binwidth` *value*	Bin width for distances (`pairwise`, `triplet` fingerprint and `E` atom typing scheme). Default: 1.
`-binoverlap` *value*	Distance threshold for assignment of item to multiple bins. Useful for "fuzzy" binning. Default: 0.
`-minbin` *value*	Minimum distance for binning. Default: 0.
`-maxbin` *value*	Maximum distance for binning. Default: no limit.
`-iter` *iterations*	Number of iterations for radial fingerprints. Valid only with `radial` fingerprint type. Usually between 2 and 6. Default: 4.

Table 5.6. Options for the canvasFPGen command. (Continued)

Option	Description
`-estatePath` *length*	Path length for Estate atom typing. Default 2.
`-estateWidth` *width*	Binning step for Estate atom typing. Default: 0.25.
`-miniter` *iterations*	Set minimum for radial iterations below which features are discarded. Valid only with `radial` fingerprint type. Default: 0.
`-startH`	Calculate molprint2D codes for hydrogens. Valid only with `molprint2D` fingerprint type. Default: do not calculate.
`-endH`	Include terminal hydrogens in molprint2D codes. Valid only with `molprint2D` fingerprint type. Default: do not include.
`-min` *fracMin*	Omit bits that are set by less than the specified fraction of molecules. Default: use all bits.
`-max` *fracMax*	Omit bits that are set by more than the specified fraction of molecules. Default: use all bits.
`-noone`	Omit bits only set in a single molecule. Overrides `-min`.
`-noall`	Omit bits set in all molecules. Overrides `-max`.
`-reduce` *bits*	Reduce the precision of fingerprints by the specified number of bits. May be applied after min/max filters. This option increases the chance of feature collisions. For example, a value of 22 will reduce each single precision key (32 bits) into a range of 1024 (10 bits).
`-mostSig` *nbits*	Keep only the *nbits* most informative bits across the chosen input set. Default: use all bits.
`-n` *structureRange*	The set of input structures to process. *structureRange* is a range specification, as defined on page 119. Default: process all structures.
`-obad` *badMolFile*	Save the molecules that failed to generate a fingerprint to a file.This option is only available for SMILES, SD, or CSV input. Default: write information on failures to standard output.
`-scaling` *option*	Options to rescale binary fingerprint data to reals. Must be an integer—see Table 5.8 for details. Default: 0.
`-strip`	Process only the largest fragment by atom count. Default: process all fragments.
`-stamp` *name*	Override molecular title with the supplied string for all molecules. Useful in conjunction with `canvasFPCombine` for creating an ensemble or modal fingerprint.

Table 5.7. Atom typing schemes for the -atomtype option of the canvasFPGen command.

Scheme	Description
1	All atoms equivalent; all bonds equivalent.
2	Atoms distinguished as hydrogen bond acceptors or donors; all bonds equivalent.
3	Atoms distinguished by hybridization state; all bonds equivalent.
4	Atoms distinguished by functional type: {H}, {C}, {F,Cl}, {Br,I}, {N,O}, {S}, {other}; bonds by hybridization.
5	Mol2 atom types; all bonds equivalent.
6	Atoms distinguished by whether they are terminal, halogens, hydrogen bond acceptors or donors; bonds distinguished by bond order.
7	Atomic number and bond order.
8	Atoms distinguished by ring size, aromaticity, hydrogen bond acceptor or donor, ionization potential, whether terminal or halogen; bonds distinguished by bond order.
9	Carhart atom types (atom-pairs approach); all bonds equivalent.
10	Daylight invariant atom types; bonds distinguished by bond order.
11	Atoms distinguished by atomic number and bond order, aromaticity.
12	Atoms distinguished by atomic number and bond order, aromaticity, if aliphatic whether cyclic or acyclic.
C	Custom. Must be followed by location of a type definitions file. See the definition files (`.typ`) in `$SCHRODINGER/mmshare-vversion/data/canvas` for examples. Cannot be used with fingerprint types `maccs` and `custom`.
E	Estate atom types. Cannot be used with fingerprint types `maccs` and `custom`.

Table 5.8. Fingerprint scaling options.

Option	Description
0	No scaling (default)
1	Scale counts by feature size to unity
2	Scale counts by feature size to feature size
3	Scale counts by feature size to molecule size
4	Scale squares of counts by feature size to unity
5	Scale squares of counts by feature size to feature size
6	Scale squares of counts by feature size to molecule size
7	Scale square root of counts by feature size to unity
8	Scale square root of counts by feature size to feature size
9	Scale square root of counts by feature size to molecule size
10	Use raw feature counts
11	Use square of raw feature counts
12	Use square root of raw feature counts
13	Use constant value of one

5.2.2 canvasFPCombine

This program combines fingerprints for distinct or overlapping sets of molecules. When a molecule appears more than once, the bit sets are combined using a logical OR. If fingerprints were created using the `-scaling` option of `canvasFPGen`, real-valued data for duplicate molecules are averaged. The syntax of the command is as follows:

`canvasFPCombine -i` *file-list* `-o` *fpFile* `[`*options*`]`

The command arguments are described in Table 5.9.

Table 5.9. Arguments for the canvasFPCombine command.

Argument	Description
`-i` *file-list*	Space-delimited list of input fingerprint files. If a single file is specified, fingerprints are combined for consecutive structures with the same molecule ID.
`-o` *fpFile*	Output fingerprint file.
`-strict`	Input files must have the same molecule IDs in the same order.
`-cat`	Makes no attempt to merge records, Concatenates files in the order specified in *file-list*. Within each input file, the natural order is preserved.
`-compress`	Compress the output file.
`-d`	Delete input files after successful completion.
`-min` *fracMin*	Omit bits that are set by less than the specified fraction of molecules. Default: use all bits.
`-max` *fracMax*	Omit bits that are set by more than the specified fraction of molecules. Default: use all bits.
`-noone`	Omit bits only set in a single molecule. Overrides `-min`.
`-noall`	Omit bits set in all molecules. Overrides `-max`.
`-reduce` *bits*	Reduce the precision of fingerprints by the specified number of bits. May be applied after min/max filters. This option increases the chance of feature collisions. For example, a value of 22 will reduce each single precision key (32 bits) into a range of 1024 (10 bits).
`-mostSig` *nbits*	Keep only the *nbits* most informative bits across the chosen input set. Default: use all bits.

5.2.3 canvasFPBinary2CSV

This program extracts fingerprints to a CSV file. The syntax of the command is as follows:

`canvasFPBinary2CSV -i` *fpFile* {`-o`|`-odata`} *csvFile* [*options*]

The input file must be a binary fingerprint file generated by `canvasFPGen`. The output file is in CSV format, and contains the bit data if specified with -o; if `-odata` is specified, additional descriptor data is written out. By default, each row represents a molecule, and each column represents a bit or a descriptor. The options are described in Table 5.10.

Table 5.10. Options for the canvasFPBinary2CSV command.

Option	Description
`-dense`	Include bits that are off in all molecules. Ignored unless input fingerprint file has been prepared with `-reduce`. Warning: using this option may require large amounts of disk space.
`-dtab`	Use tab as delimiter.
`-force`	If binary fingerprint has been rescaled to continuous values, force binary (0,1) representation. Default: print rescaled values.
`-max` *fracMax*	Omit bits that are set by more than the specified fraction of molecules. Default: use all bits.
`-min` *fracMin*	Omit bits that are set by less than the specified fraction of molecules. Default: use all bits.
`-noall`	Omit bits set in all molecules. Overrides `-max`.
`-noone`	Omit bits only set in a single molecule. Overrides `-min`.
`-notot`	Suppress printing of total set bit counts to *csvFile*. Useful for generating raw data for input into other programs. Default: print total set bit counts.
`-off` *value*	Use *value* instead of a blank when a bit is not set by a molecule.
`-on` *value*	Use *value* instead of a 1 when a bit is set by a molecule.
`-p`	Pivot output: bits as rows, molecules as columns. Rescaled values are not used in this mode (implies `-force`).
`-pf` *prefix*	Prefix to the bit column names in the header. Default is `BIT`. Cannot be used with `-sparse`.
`-sparse`	Emit comma separated list of keys for each molecule. Preceded by molecule name, without headers or footers. Incompatible with `-odata`, `-p`, `-pf`, `-off`, `-on` and implies `-force` and `-notot`.
`-totonly`	Suppress printing of individual rows. Useful for viewing only summary data. Default: print individual rows.

5.2.4 canvasCSV2FPBinary

This program converts a CSV file into a Canvas fingerprint binary file. If binary columns share a common prefix such as `BIT`, then non-binary data can be embedded alongside. If binary-specified columns contain real-valued data, then a Canvas scaled fingerprint will be generated. The command syntax is as follows:

`canvasCSV2FPBinary` {`-icsv` *csvFile*|`-isparse` *file*} `-o` *binaryFile* [*options*]

The input CSV file must include a header, and may contain both binary and nonbinary data. If `-isparse` is used for the input file, the file must contain the molecule ID in the first column, followed by the list of "on" bit indices. The options are described in Table 5.11.

Table 5.11. Options for the canvasCSV2FPBinary command.

Option	Description
`-prefix` *string*	Prefix for binary columns in the header. The default prefix of a binary column is `BIT`. A blank string (`-prefix " "`) means to treat all columns as binary.
`-fieldAsName` *field*	Field in the CSV file to use as the molecule name. Default: first column.
`-noheader`	No header is supplied. The first column is presumed to be the molecule name and the rest binary data.
`-d` *delimiter*	Delimiter used in the CSV or sparse input file. Default: comma.
`-n` *rows*	Convert the specified rows in the CSV file. The first line in *csvFile* is the header; the second line in *csvFile* is the first data row, and is row 1. *rows* is a range specification, as described on page 119. Default: convert all rows.
`-type` *string*	Supply custom type information for Canvas. If you are using this utility for more than one type of CSV file, where the same bits have different meanings, it is recommended you supply a distinct string for each type using this option. This utility is useful when multiple sets of fingerprints are generated independently from Canvas and you want to distinguish between them in the Canvas GUI. Fingerprints generated by Canvas and KNIME are tracked automatically, and do not need to be differentiated. Default: `custom`.
`-offset` *N*	Add this value to each binary key found. Useful for concatenation of multiple fingerprints.
`-bit` *on*,*off*	Identifiers to use for ON and OFF states delimited with a comma. Example: `ON,OFF`

5.3 3D Pharmacophore Fingerprints

The program `canvasPharmFP` generates fingerprints from 3D pharmacophores for the input structures. The fingerprint bits are set on the basis of the feature types and inter-feature distances.

The command syntax is as follows:

`canvasPharmFP` [*job-options*] *program-options*

The job options are described in Table 5.3 on page 121. This program can be distributed over multiple processors. The program options are described in Table 5.12.

Table 5.12. Options for the canvasPharmFP command.

Option	Description
`-iproj`|`-imae`| `-isd` *source*	Source of input structures. `-iproj`: Canvas project (`.cnv`) `-imae`: Maestro file (`.mae`, `.maegz`, `.mae.gz`) `-isd`: SD file (`.sdf`, `.sd`, `.sdf.gz`, `.sd.gz`) If *source* is a project, it must include the absolute path, and it must be accessible to all processors on which the job is run.
`-fieldAsName` *prop*	Use an alternate property as the source of structure names. For a list of project properties, run `canvasPharmFP -helpColumns`.
`-n` *list*	Process the specified subset of rows or structures. The format of the list is given in Section 5.1 on page 119.
`-file`	Interpret *list* as the name of a binary file containing row numbers exported from the Canvas GUI. Valid only when *source* is a Canvas project.
`-o`|`-odata` *fpFile*	Binary output file for fingerprints. If `-o` is used, the file contains fingerprints and structure names; if `-odata` is used the file contains fingerprints and other properties from the input source, as specified by `-fieldOnly`.
`-fieldOnly` *fields*	Write out only the specified properties to *fpFile*. The list is a space-separated list of property names. Use `-helpColumns` to get a list of project properties. Valid only with `-odata`.
`-uniform`	All structures in the input Maestro or SD file contain the same properties. Valid only with `-odata`. Not valid with `-iproj`.
`-fill`	Insert a blank fingerprint as a placeholder when a given structure cannot be processed.
`-3pt`	Generate 3-point pharmacophore fingerprints. This is the default.
`-4pt`	Generate 4-point pharmacophore fingerprints. To include 3-point and 4-point pharmacophores in each fingerprint, use `-3pt` and `-4pt`.

Table 5.12. Options for the canvasPharmFP command. (Continued)

Option	Description
`-dmin` *dmin*	Minimum distance between pharmacophore features in angstroms. Distances smaller than this value are placed in the first bin. Default: 2.0 Å.
`-dmax` *dmax*	Maximum distance between pharmacophore features in angstroms. Distances larger than this value are placed in the last bin. Default: 16.0 Å.
`-width` *width*	Distance bin width in angstroms. Default: 2.0 Å.
`-overlap` *overlap*	Set the bit for a neighboring bin if a distance is within *overlap* of that bin. Default: *width*/2.
`-fd` *fdFile*	Use pharmacophore feature definitions in *fdFile*. This file must be a feature definition file in Phase format (see Appendix B.3 of the *Phase User Manual*). If omitted, default Phase feature definitions are used.
`-flex`	Generate conformers and combine bits from different conformers using logical OR. Conformers are discarded after use. Requires a Phase license and uses one PHASE_DBCREATE token per subjob.
`-sample {rapid|thorough}`	Sampling method for conformational space. Rapid sampling varies the rotatable groups independently; thorough sampling varies them together. Only valid with `-flex`. Default: `rapid`.
`-cmax` *maxConfs*	Maximum number of conformers. Only valid with `-flex`. Default: 100.
`-bf` *numPerBond*	Number of conformers to retain per rotatable bond in the structure. The total number of conformers is bounded by the product of *numPerBond* and the number of rotatable bonds, so if *maxConfs* is increased, it may be necessary to increase *numPerBond* in order to retain additional conformers for more rigid structures. Only valid with `-flex`. Default: 10.
`-ewin` *deltaE*	Energy window for keeping conformers in kJ/mol. Conformers whose energy is higher than this amount above the minimum energy conformer are discarded. Only valid with `-flex`. Default: 104.6 kJ/mol (25 kcal/mol).
`-amide {vary|orig|trans}`	Specify the method for amide torsion sampling. The allowed values are: `vary`—vary the torsion freely `orig`—use the input conformation `trans`—set the conformation to trans Only valid with `-flex`. Default: `vary`.
`-xp`	Store fingerprints using 64-bit precision. Default: 32-bit.
`-compress`	Use frequency-based compression to reduce required storage by approximately tenfold.
`-fmin` *fmin*	Omit bits that are set by less than the specified fraction of input structures. Not valid with `-mostSig`. Default: 0.0.

Table 5.12. Options for the canvasPharmFP command. (Continued)

Option	Description
`-fmax` *fmax*	Omits bits that are set by more than the specified fraction of input structures. Not valid with `-mostSig`. Default: 1.0.
`-mostSig` *n*	Keep only the *n* most informative bits over the set of input structures. Not valid with `-fmin` or `-fmax`.
`-reduce` *bits*	Reduce fingerprint precision by the specified number of bits. May be applied after the bit filters.
`-h[elp]`	Print this message and exit.
`-helpColumns`	Print list of Canvas project column names and exit. Valid only with `-iproj`. No other options are required.

5.4 Similarity, Dissimilarity, and Clustering

The programs for generating similarity information, dissimilarity-based selections of molecules, and for clustering are listed in Table 5.13, and are described in the following sections.

Table 5.13. Tools for similarity, dissimilarity, and clustering.

Tool	Description
`canvasCSV2PW`	Generate a binary pairwise similarity or distance matrix using a CSV input
`canvasCSVMatrix`	Generate a distance matrix based on CSV input. The result is a symmetric row-by-row matrix.
`canvasDBCS`	Dissimilarity-based compound selection using pairwise matrix or Canvas fingerprint files as input
`canvasFPHist`	Generate a histogram of nearest neighbor similarities from one or two fingerprint files
`canvasFPMatrix`	Generate a pairwise similarity or distance matrix using fingerprints from one or two sets of molecules.
`canvasHC`	Hierarchical clustering and report details for specific levels.
`canvasHCBuild`	Hierarchical clustering.
`canvasHCSelect`	Report details for specific levels of clustering.
`canvasKMeans`	K-means clustering.
`canvasLC`	Leader-follower clustering.
`canvasPW2CSV`	Convert binary pairwise matrix to CSV.
`canvasTreeDraw`	Draw dendrogram from a tree file.

5.4.1 canvasCSV2PW

This program generates a binary pairwise similarity or distance matrix from a CSV input file.

`canvasCSV2PW -i` *csvFile* `-o` *binaryFile* `[-rowNames ] [-colNames ]`

The arguments are described in Table 5.14.

Table 5.14. Arguments for the canvasCSV2PW command.

Argument	Description
`-i` *csvFile*	File containing CSV data to be converted.
`-o` *binaryFile*	Output file for matrix in binary format.
`-rowNames`	Input CSV matrix has molecule name to begin each line. May be omitted if CSV file was created by Canvas.
`-colNames`	Input CSV matrix lists molecule names as first line. May be omitted if CSV file was created by Canvas.

5.4.2 canvasCSVMatrix

This program generates a distance matrix based on CSV input data, resulting in a symmetric row-by-row matrix.

`canvasCSVMatrix -icsv` *csvFile* {`-o` *binaryFile*|`-ocsv` *csvFile*} `[-order` *n*`]`
`[-cov [-scale [-t` *threshold*`]]]`

The input CSV file, specified by `-icsv`, must contain the desired real-valued data. The output file can be a binary file, specified with `-o`, or a CSV file, specified by `-ocsv`. Both formats can be used in subsequent steps. The options are described in Table 5.15.

Table 5.15. Options for the canvasCSVMatrix command.

Option	Description
`-order` *n*	Order of distance matrix as defined in $(\Sigma x^n)^{1/n}$. *n* must be a positive integer. The default value of *n* is 2, which corresponds to Euclidean distance.
`-cov`	Generate Mahalanobis distance matrix using the covariance matrix to correct non-orthogonality of the data.
`-scale`	Scale to unit variance, if it is within threshold.
`-t` *threshold*	Threshold for scaling. Default value is 1E-3.

5.4.3 canvasDBCS

This program performs dissimilarity-based compound selection based on a pairwise matrix or a Canvas fingerprint file. The command syntax is as follows:

`canvasDBCS` {`-im` *matrixFile*|`-ifp` *fpFile* [`-ifp2` *fpFile2* [`-ifp3` *fpFile3*]]}
[*options*]

The arguments for this command are described in Table 5.16. The metrics are listed in Table 5.17; a description of the metrics is given in Table 5.22 for `canvasFPMatrix`.

Table 5.16. Arguments for the canvasDBCS command.

Argument	Description
`-im` *matrixFile*	Pairwise matrix file generated by `canvasFPMatrix` or another program. Must be symmetric and contain either similarities or distances. Can be used with all methods except `dise`.
`-ifp` *fpFile*	Canvas fingerprint file generated by `canvasFPGen`. Requires `sphere` or `dise` method.
`-ifp2` *fpFile2*	Canvas fingerprint file representing the chemical space to avoid when choosing diverse compounds. A compound is not selected if it lies within the exclusion sphere of any compound in this file. Requires `sphere` method.
`-ifp3` *fpFile3*	Canvas fingerprint file representing the chemical space to cover when choosing diverse compounds. A compound is selected only if it lies within the exclusion sphere of a compound in this file. Requires `sphere` method.
`-o` *outputFile*	Output file for results. Default is stdout.
`-n` *size*	Diverse subset size. Required for methods `maxsum` and `maxmin`. Allowed for methods `sphere` and `dise` where it is applied as a secondary filtering step if the exclusion distance produces a diverse subset larger than *size*.
`-d` *distance*	Exclusion distance for methods `sphere` and `dise`. The meaning of this parameter is the same whether a similarity or a distance metric is used. For example, if using Tanimoto similarity, `-d 0.7` selects compounds that exhibit pairwise distances of least 0.7, which means the pairwise similarities will be no higher than 0.3. Default: 0.5.
`-method` *type*	Choice of selection method. The allowed values are: `sphere` Sphere exclusion `dise` Directed sphere exclusion [2]. `maxsum` Maximum sum of distances. Each round adds the compound with the largest total distance from the current subset. `maxmin` Largest minimum distance. Each round adds the compound with the largest minimum distance from the current subset.

Table 5.16. Arguments for the canvasDBCS command. (Continued)

Argument	Description
`-metric` *name*	Metric type. *name* can be either the name or the index. Allowed values are listed in Table 5.17 by index and name. Applies only if a fingerprint file is specified as input. Default: `soergel` (20).
`-noselfcheck`	When using `-ifp` and `-ifp2` to choose structures from the first set (`-ifp`) that are dissimilar to those in a second set (`-ifp2`), do not check whether the structures chosen from the first set are dissimilar to each other. Requires `sphere` method, `-ifp` and `-ifp2`. The default is to exclude structures from the first set that are similar to each other.
`-helpMetric`	Prints the definition of each metric then exits.
`-forceBinary`	Ignore any scaled fingerprint values in the input file. Binary values are always used. Only relevant to an input fingerprint file.
`-rowNames`	Input CSV matrix has molecule name to begin each line. Only applies to input CSV file.
`-colNames`	Input CSV matrix lists molecule names as first line. Only applies to input CSV file.
`-init` *type*	Initialization method. Allowed method types are: `random`, `first`, `representative`, `dissimilar`. `representative` and `dissimilar` can only be used with the `maxsum` and `maxmin` methods. Default: `random`.
`-seed` *int*	Seed for random number generation if the random initialization method was specified. Defaults: use the computer clock.
`-xpdise`	Run DISE method in a mode that produces even better coverage at the expense of speed.
`-guiprop`	Add a property column to the output file that contains 1 for selected structures and 0 for unselected structures.

Table 5.17. Available metrics for the canvasDBCS command.

Index	Name	Index	Name
1	`buser`	13	`patternDifference`
2	`cosine`	14	`pearson`
3	`dice`	15	`petke`
4	`dixon`	16	`rogersTanimoto`
5	`euclidean`	17	`shape`
6	`hamann`	18	`simpson`
7	`hamming`	19	`size`
8	`kulczynski`	20	`soergel`
9	`matching`	21	`tanimoto`
10	`mcConnaughey`	22	`variance`
11	`minmax`	23	`yule`
12	`modifiedTanimoto`		

5.4.4 canvasLibOpt

This program selects a subset of compounds from a pool by minimizing their similarity, and optionally by optimizing the ranges of specified properties. It can be used to fill holes in an existing library of compounds. The command syntax is as follows:

`canvasLibOpt` *fpIn nstr csvOut* [*options*]

where *fpIn* specifies the Canvas fingerprint file containing the pool of structures from which selections will be made, *nstr* specifies the number of structures to select, and *csvOut* specifies the output CSV file containing the selected structures. The options are described in Table 5.18.The columns in the CSV file are described in Table 5.19. The algorithm used is described below.

Given a pool of structures in a Canvas fingerprint file, `canvasLibOpt` uses a greedy, stochastic algorithm to select a diverse subset, with the goal of minimizing the sum of the average nearest neighbor Tanimoto similarity within the subset and the average property filter score, depending on which task is chosen.

If hole-filling was chosen, the nearest neighbor to a given structure is determined by considering both the structures in the subset and the structures in the library (fingerprint file *fpHoles*).If property filtering was chosen, the optimization attempts to minimize the average number of property filters that failed within the subset. A property filter is simply a range of

Table 5.18. Options for the canvasLibOpt command.

Option	Description
`-range` *rows*	Comma-delimited series of colon-separated ranges that define the subset of rows to select structures from in *fpIn*. By default, all rows are considered.
`-fill` *fpHoles*	Fill holes in the library represented by *fpHoles*. The fingerprints in this file must have been created using the same settings as *fpIn*. The range of structures considered to be in the library can be specified with `-range2`. *fpHoles* can be the same file as *fpIn* if you use `-range2` to define the library. Computational effort increases as the product of *nstr* and the size of this library, and memory increases linearly with the size of this library.
`-range2` *rows2*	Consider only the specified subset of rows in *fpHoles* to be in the library. *fpHoles* and *fpIn* may be the same file if `-range` and `-range2` are used. Only valid with `-fill`.

`-filter` *file*	File containing a set of property filters, with one filter per line. Structures that fail fewer filters are preferred. Each filter consists of a property name, followed by minimum and maximum allowed values for that property. For example,
	`AlogP -1.5 4.5`
	`HBA 0 5`
	`MW 200 500`
	If the property name contains spaces, it must be surrounded by quotes.
	Use `-help_props` to see a list of the properties stored in *fpIn*.

`-cycles` *ncyc*	Number of optimization cycles. A given cycle consists of making *nstr* passes through the selected structures, with an attempt to replace the poorest scoring structure at each pass. A candidate structure is chosen randomly from the pool and the poorest scoring structure is replaced by that candidate if doing so improves the average Sim_NN and Filter values among all structures in the subset. If improvement does not occur, the candidate structure is accepted with a probability determined by a Monte Carlo test. The default number of cycles is 10.
`-stop` *dx*`,`*n*	Stop optimization when the total score changes (up or down) by less than *dx* for *n* consecutive cycles. Default: `0.001,3`.
`-tmax` *sec*	Run optimization for no longer than the specified number of CPU seconds. There is no time limit by default.
`-mc` *tol*	Monte Carlo acceptance criterion (`etest`). A total score increase of *tol* is accepted with a 50% probability in the first cycle and a 1% probability in the last cycle. Default: `0.001`.
`-rand` *seed*	Integer random seed. Controls the choice of initial subset and all random operations throughout the course of the optimization. Default: `1`.
`-col` *prefix*	Add the supplied column name prefix to the CSV output file columns Row, Sim_NN, Row_NN, Hole_NN, and Filter. For example, `-col "job1::"` would yield the column names job1::Row, job1::Sim_NN, etc. See Table 5.19 for a description of the columns.

Table 5.18. Options for the canvasLibOpt command. (Continued)

Option	Description
`-help`	Print usage message and exit.
`-help_props`	Print the names of the properties stored in *fpIn*.

Table 5.19. Description of columns in the CSV output file for canvasLibOpt.

Column	Description
SMILES	SMILES string for the structure. If *fpIn* does not contains a SMILES property, a dummy SMILES "C" is written to each row.
Name	Name or ID of the structure.
Row	Row number of the structure in *fpIn*.
Sim_NN	Tanimoto similarity to nearest neighbor in the optimized library.
Row_NN	Nearest neighbor row number.
Hole_NN	Indicator or whether the nearest neighbor is in the original library (with holes). The value is 1 if it is in the original library, 0 otherwise.
Filter	Number of property filters failed by this structure divided by the total number of property filters. Set to 0 if no filters are used.

allowed values for the property. Failure means that the property value lies outside the specified range. The filter score for a structure is computed as the number of failures divided by the total number of filters, i.e. the fraction of failures. The average filter score is the average of the filter scores for all structures in the subset.

The total score is the average nearest neighbor similarity plus the average filter score. Both of these quantities lie on the interval [0, 1], so the total score lies on the interval [0, 2].

Optimization is done by choosing the requested number of structures (*nstr*) at random, then performing a series of optimization cycles that attempt to improve both the average nearest neighbor similarity and the average filter score.

Each optimization cycle consists of making *nstr* passes through the subset, with an attempt to replace the poorest scoring structure at each pass. The poorest scoring structure is the one that has the highest nearest neighbor similarity. If there are ties in the nearest neighbor similarity, the filter score is used to break the tie. A candidate structure is chosen at random as a possible replacement, and the replacement is done if doing so improves the total score for the subset. The total score is considered to improve if one of the following happens:

- Average nearest neighbor similarity and average filter score decrease.
- Average nearest neighbor similarity decreases and average filter score remains the same.
- Average nearest neighbor similarity remains the same and average filter score decreases.

If the total score does not improve, a Monte Carlo test is used:

```
etest = deltaSim + deltaFilter
```

where `deltaSim` is the increase in average nearest neighbor similarity, and `deltaFilter` is the increase in average filter score, both of which are greater than or equal to 0 if the score does not improve. The replacement is done if the probability, defined by

```
prob = exp(-etest/temperature)
```

is greater than a random number between 0 and 1.

The temperature decreases linearly over the course of the optimization, and the cooling schedule is chosen such that an `etest` value of 0.001 will yield `prob` = 0.5 in the first cycle, and `prob` = 0.01 in the last cycle. The default `etest` tolerance of 0.001 can be overridden using the option `-mc` *tol*.

When the optimization has completed the requested number of cycles, or when other criteria are met, a CSV file is written with the optimized subset of structures. A SMILES column is included, so the CSV file can be imported into the Canvas GUI. The SMILES corresponds to the actual structure if the fingerprint file contains a SMILES property, which is always the case if the fingerprint file is exported from the GUI. And in that case, the Canvas UID is also included in the CSV file, so it is possible to read it back into the same project and update on Canvas UID. If there is no SMILES in the fingerprint file a dummy SMILES `"C"` is written for each structure.

5.4.5 canvasFPHist

This utility generates a histogram of nearest-neighbor similarities, by locating the largest off-diagonal value in each row of a similarity matrix for a single fingerprint file, or the largest value in each row of the similarity between two fingerprint files. The command syntax is as follows:

`canvasFPHist [`*options*`]` *fpFile histFile*

The input file *fpFile* must be a Canvas fingerprint file. If the matrix between two sets of fingerprints is required, specify the second input file with `-ifp2`. The output file *histFile* is a CSV file containing the similarity bin boundaries and the frequencies, normalized to sum to 1.

The options are described in Table 5.20.

Table 5.20. Options for the canvasFPHist command.

Option	Description
`-range` *range*	Specify the range of molecules to use from the input fingerprint file *fpFile*. *range* is a range specification, as described on page 119. Default: use all.
`-metric` *name*	Metric type. The allowed values of name are `buser`, `cosine`, `dice`, `hamann`, `kulczynski`, `matching`, `mcConnaughey`, `minmax`, `modifiedTanimoto`, `pearson`, `petke`, `rogersTanimoto`, `simpson`, `tanimoto`, `tversky`, `yule`. Default: `tanimoto`.
`-osim` *simFile*	Write histogram similarities to the CSV file *simFile*. The format is SMILES,Name,Sim, where SMILES is a dummy column with the string "C" in each row, Name is the molecule name, and Sim is the maximum similarity. The similarity column name can be set with the `-col` option.
`-col` *simCol*	Similarity column name in *simFile*. Default: `Sim`.
`-bin` *dbin*	Histogram bin spacing. The default spacing is determined by the number of rows and ranges from 0.5 to 0.05, with smaller spacing for larger numbers of rows.
`-ifp2` *fpFile2*	Specify a second fingerprint input file, for the columns of the similarity matrix. If specified, only similarities between *fpFile* and *fpFile2* are calculated.
`-range2` *range*	Specify the range of molecules to use from the second input file, *fpFile2*. *range* is a range specification, as described on page 119. Default: use all.
`-block` *value*	Maximum number of row fingerprints to load in memory at a time. Default: 1000.
`-alpha` *value*	Tversky alpha parameter. Default: 0.5.
`-beta` *value*	Tversky beta parameter. Default: 0.5.

5.4.6 canvasFPMatrix

This utility generates a pairwise similarity or distance matrix using binary or scaled fingerprints from one or two sets of molecules. The command syntax is as follows:

`canvasFPMatrix -ifp` *fpFile* `[-ifp2` *fpFile2*`]` {`-o` *binaryFile*|`-ocsv` *csvFile*}
    [*options*]

The input file specified with `-ifp` must be a Canvas fingerprint file. If the matrix between two sets of fingerprints is required, specify the second input file with `-ifp2`. The output file can be in binary format, if specified with `-o`, or in CSV format, if specified with `-ocsv`. Both formats can be used in subsequent operations. The options are described in Table 5.21. The metrics are listed in Table 5.22, along with their type and formula. The quantities in the formula are defined in Table 5.23, except for the Tversky α and β parameters.

Table 5.21. Options for the canvasFPMatrix command.

Option	Description
`-filter` *minSim* `[-all]`	Report only rows with at least one similarity greater than or equal to *minSim*. Valid only with `-ocsv` and when a similarity metric is used. If `-all` is used, report only rows where all similarities are greater than or equal to *minSim*.
`-sort avg|best`	Sort the rows of the output matrix in order of the largest similarity or smallest distance in each row (`best`) or the largest average similarity or smallest average distance of the row (`avg`). When using a square matrix (`-ifp2` not used), the diagonal elements are ignored.
`-caprow` *N*	Write out only the top *N* rows, after sorting. Requires `-sort`.
`-capcol` *N*	For each row, write out only the top *N* columns. The output consists of the row name, followed by a list of *N* column names and the corresponding *N* metric values.
`-metric` *name*	Metric type. *name* can be either the name or the index. Allowed values are listed in Table 5.22 by index and name. Applies only if a fingerprint file is specified as input. Default: `soergel` (20).
`-helpMetric`	Print the definition of each metric then exit.
`-forceBinary`	Use binary values and Ignore any scaled fingerprint values in the input files.
`-range` *first*`:`*last*	Specify range of molecules in the input file. *first* and *last* are the indices of the first and last molecules to be included.
`-ifp2` *fpFile*	Specify a second input file. If specified, only pairs between the first and the second files are calculated.
`-range2` *range*	Specify range of molecules in the second input file. *first* and *last* are the indices of the first and last molecules to be included.
`-blocksize` *value*	Maximum number of fingerprints to load in memory at a time. Default: 5000.
`-alpha` *value*	Tversky alpha parameter. Default: 0.5.
`-beta` *value*	Tversky beta parameter. Default: 0.5.
`-flatten` *value*	Gaussian parameter to make output matrix sparse. Default is no flattening.
`-limitOffBits`	Limit the set of possible off bits. By default, the number of off bits is limited only by the fingerprint address space size (2^{32} or 2^{64}), which may yield undesirable behavior for metrics that incorporate off bits. If this option is used, the set of possible off bits is limited to those bits that are set by at least one compound in the fingerprints provided. Applies only to `buser`, `hamann`, `matching`, `modifiedTanimoto`, `patternDifference`, `pearson`, `rogersTanimoto`, `size`, `shape`, `variance`, and `yule` metrics.

Table 5.22. Available metrics for the canvasFPMatrix command.

Index	Name	Type	Formula
1	`buser`	similarity	(sqrt(cd)+c)/(sqrt(cd)+a+b-c)
2	`cosine`	similarity	c/sqrt(ab)
3	`dice`	similarity	2c/(a+b)
4	`dixon`	distance	(A+B)^2/(a+b-c)
5	`euclidean`	distance	sqrt(A+B)
6	`hamann`	similarity	(c+d-A-B)/N
7	`hamming`	distance	A+B
8	`kulczynski`	similarity	0.5(c/a + c/b)
9	`matching`	similarity	(c+d)/L
10	`mcConnaughey`	similarity	(c^2-(a-c)(b-c))/(ab)
11	`minmax`	similarity	sum{min(a,b)/max(a,b)}
12	`modifiedTanimoto`	similarity	α *T1 + (1.0-α)*T0
13	`patternDifference`	distance	AB/N^2
14	`pearson`	similarity	(cd-AB)/sqrt(ab(A+d)(B+d))
15	`petke`	similarity	c/max(a,b)
16	`rogersTanimoto`	similarity	(c+d)/(2(a+b)-3c+d)
17	`shape`	distance	(A+B)/N - ((A-B)/N)^2
18	`simpson`	similarity	c/min(a,b)
19	`size`	distance	((A-B)/N)^2
20	`soergel`	distance	((A-B)/N)^2
21	`tanimoto`	similarity	c/(a+b-c)
22	`tversky`	similarity	c/(α(a-c)+β(b-c)+c)
23	`variance`	distance	(A+B)/(4N)
24	`yule`	similarity	(c*d - A*B)/(c*d + A*B)

Table 5.23. Variables used in metric formulae

Variable	Description
a	Number of bits that are on in structure 1
b	Number of bits that are on in structure 2
c	Number of bits that are on in both structure 1 and structure 2
d	Number of bits that are off in both structure 1 and structure 2
A	Number of bits that are on in structure 1 but not in structure 2. A = a - c
B	Number of bits that are on in structure 2 but not in structure 1. B = b - c
L	Total number of bits. L = a + b - c + d
N	Restricted total number of bits. N = a + b - c + min(d,10000)

5.4.7 canvasHC

This program performs full hierarchical clustering and reports details for a specific number of clusters. The command syntax is as follows:

```
canvasHC -im matrixFile [-n clusters|-ns clusters|-d distance|-kelley]
    [-rowNames] [-colNames] [-og groupFile [-noties]] -or runFile]
    [job-options] [common-options]
canvasHC -ifp FPFile [-saveMatrix matrixFile] [-metric metric]
    [-ob baseFile] [-os statFile [-noties]] [-alpha alpha] [-beta beta]
    [job-options] [common-options]
```

where the common options are:

```
[-usemin|usemax|-useupper|-uselower] [-linkage {name|number}]
    [-helpLinkage] [-ot treeFile]
```

The input file specified by `-im` must be a similarity or distance matrix file in either binary or CSV format. The input matrix is assumed to be symmetric. If it is not symmetric, you should use one of the four `–use` options to specify how to handle the asymmetry. Alternatively, you can specify a fingerprint file with `-ifp` and calculate the similarity or distance matrix. The job options are listed in Table 5.3. The program options are described in Table 5.24.

Table 5.24. Options for the canvasHC command.

Option	Description
Common options:	
`-helpLinkage`	Print details about linkage methods and exit.
`-linkage` *name*	Method of calculating distance between clusters. The method can be referred to by name or by index. The allowed values are listed by index and name in Table 5.25. Default: `average` (3).
`-ot` *treeFile*	Intermediate file required by `canvasTreeDraw` to generate a dendrogram.
`-uselower`	Asymmetric distance/similarity matrix. Use the lower half for clustering.
`-usemax`	Asymmetric distance/similarity matrix. Use the minimum of D_{ij} and D_{ji} for clustering.
`-usemin`	Asymmetric distance/similarity matrix. Use the minimum of D_{ij} and D_{ji} for clustering.
`-useupper`	Asymmetric distance/similarity matrix. Use the upper half for clustering.
Options for matrix file input:	
`-colNames`	Input CSV matrix lists all molecule names on the first line.
`-rowNames`	Input CSV matrix has the molecule name at the beginning of each line.
`-d` *distance*	Print details for the set of clusters formed at or below the specified merging distance. The number of clusters decreases as *distance* increases.
`-kelley`	Print details for the set of clusters formed at the number of clusters specified by a minimum in the Kelley cost function [4].
`-n` *clusters*	Print details for the formation of the specified number of clusters. Default: 1.
`-ns` *clusters*	Print details for the formation of the specified number of non-singleton clusters. Singletons are not counted. Default: 1.
`-og` *groupFile*	Print cluster memberships in CSV format to *groupFile*. Default: print to standard output.
`-or` *runFile*	Print final embedding stress value to *runFile*. Default: print to standard output.
Options for fingerprint file input:	
`-alpha` *alpha*	Tversky alpha parameter. Only valid with `-metric`. Default 0.5.
`-beta` *beta*	Tversky beta parameter. Only valid with `-metric`. Default 0.5.
`-metric` *type*	Metric used to calculate similarity or distance matrix from a fingerprint file. Valid values of type are listed in Table 5.22. The value can be the name or the index. Default: `tanimoto` (21).

Table 5.24. Options for the canvasHC command. (Continued)

Option	Description
`-noties`	Disregard ties when assigning items nearest and farthest from the cluster centroid.
`-os` *statsFile*	Report file designed to help select an appropriate number of clusters.
`-saveMatrix` *file*	Save the similarity or distance matrix generated from fingerprints to the specified binary file. Default: do not save.

Table 5.25. Description of linkage methods for hierarchical clustering.

Method	Description
`1: single`	Closest inter-cluster pair
`2: complete`	Farthest inter-cluster pair
`3: average`	Average distance between all inter-cluster pairs
`4: centroid`	Euclidean distance between cluster centroids
`5: mcquitty`	Average distance to the two clusters merged in forming a given cluster
`6: ward`	Sum of squared distances to merged cluster centroid (minimum variance)
`7: weightedcentroid`	Weighted center of mass distance, also known as median
`8: flexiblebeta`	Weighted average intra-cluster and inter-cluster distances (Lance-Williams) with beta=0.25
`9: schrodinger`	Closest distance between terminal (right-to-left) points in 1D cluster orderings.

5.4.8 canvasHCBuild

This program performs full hierarchical clustering, but provides no details. You can use `canvasHCSelect` to obtain details for a specific level of clustering.The command syntax is:

`canvasHC -im` *matrixFile* [*options*]

The input file, specified by `-im`, must be a similarity or distance matrix file, and can be in either binary or CSV format. The input matrix is assumed to be symmetric. If it is not symmetric, you should use one of the four `-use` options to specify how to handle the asymmetry. The options are described in Table 5.26.

Table 5.26. Options for the canvasHCBuild command.

Option	Description
`-rowNames`	Input CSV matrix has the molecule name at the beginning of each line.
`-colNames`	Input CSV matrix lists all molecule names on the first line.
`-linkage` *name*	Method of calculating distance between clusters. The method can be referred to by name or by index. The allowed values are listed by index and name in Table 5.25. Default: `average` (3).
`-helpLinkage`	Print details about linkage methods and exit.
`-helpStats`	Print definitions of statistics reported in *statsFile* and exit.
`-n` *clusters*	Print details for the formation of the specified number of clusters. Default: 1.
`-noties`	Disregard ties when assigning items nearest and farthest from the cluster centroid.
`-ns` *clusters*	Print details for the formation of the specified number of non-singleton clusters. Singletons are not counted. Default: 1.
`-d` *distance*	Print details for the set of clusters formed at or below the specified merging distance. The number of clusters decreases as *distance* increases.
`-ob` *baseFile*	Intermediate file required by `canvasHCSelect` to generate cluster memberships.
`-os` *statsFile*	Report designed to help select an appropriate number of clusters.
`-or` *runFile*	Print final embedding stress value to *runFile*. Default: print to standard output.
`-ot` *treeFile*	Intermediate file required by `canvasTreeDraw` to generate a dendrogram.
`-usemin`	Asymmetric input matrix. Use the minimum of D_{ij} and D_{ji} for clustering.
`-usemax`	Asymmetric input matrix. Use the minimum of D_{ij} and D_{ji} for clustering.
`-useupper`	Asymmetric input matrix. Use the upper half for clustering.
`-uselower`	Asymmetric input matrix. Use the lower half for clustering.

5.4.9 canvasHCSelect

This program reports details for a specific level of clustering for a prior run of `canvasHCBuild`. The command syntax is as follows:

`canvasHCSelect -ib` *baseFile* `-im` *matrixFile* [*options*]

The input base file is the file specified with the `-ob` option to `canvasHCBuild`, and the input matrix file is likewise the matrix file specified with the `-im` option to `canvasHCBuild`. The options are described in Table 5.27.

Table 5.27. Options for the canvasHCSelect command.

Option	Description
`-rowNames`	Input CSV matrix has the molecule name at the beginning of each line.
`-colNames`	Input CSV matrix lists all molecule names on the first line.
`-n` *clusters*	Print details for the formation of the specified number of clusters. Default: 1.
`-ns` *clusters*	Print details for the formation of the specified number of non-singleton clusters. Singletons are not counted. Default: 1.
`-d` *distance*	Print details for the set of clusters formed at or below the specified merging distance. The number of clusters decreases as *distance* increases.
`-og` *groupFile*	Print cluster memberships in CSV format to *groupFile*. Default: print to standard output.

5.4.10 canvasTreeDraw

This program draws dendrograms from a clustering output file. The command syntax is:

`canvasTreeDraw -i` *treeFile* `-o` *psFile* [*options*]

The input tree file, *treeFile*, must be generated by running `canvasHC` or `canvasHCBuild` with the `-ot` option. The output file of the dendrogram, specified by `-o`, is a postscript file.

Table 5.28. Options for the canvasTreeDraw program.

Option	Description
`-c`	Color links (with number of groups from 2 to 19).
`-C`	Color labels (with number of groups from 2 to 19).
`-h`	Use rainbow colors, light and dark, allowing unlimited number of groups.
`-H`	Use rainbow colors, light only, allowing unlimited number of groups.
`-s` *lines*	Number of lines to skip within groups (can be non-integer).
`-S` *lines*	Number of lines to skip between groups (can be non-integer).
`-2`	Use PostScript level 2. Default: level 1.
`-L`	Suppress printing of labels.
`-n` *groups*	Number of groups to color by. Invalid with `-d`. Default: 1.
`-d` *distance*	Distance used to separate groups. Invalid with `-n`.
`-f` *fontSize*	Font size, in points. Default: 8.

5.4.11 canvasKMeans

This program performs k-means clustering using Lloyd's algorithm [3]. The algorithm consists of a set of runs, each of which starts with the random selection of k data points (structures) as the initial centroids of the clusters. For each run, a series of iterations is performed in which each of the points in the data set is assigned to a cluster based on the centroid closest to that point, and each centroid is updated by setting it equal to the average position of the points assigned to that cluster. The cost function that is minimized is the sum of intra-cluster variances. A run is terminated when the maximum number of steps is reached, or a convergence condition on the cost is reached. The clusters that are returned are taken from the run with the smallest cost.

The command syntax is as follows:

`canvasKMeans` [*job-options*] {`-ifp` *fpFile*|`-icsv` *csvFile* [*csv-options*]} `-o` *outFile* `-k` *clusters* [*options*]

The job options are listed in Table 5.3. The *csv-options* are the input file options for CSV files listed in Table 5.1 (except `-smi`). The remaining arguments are described in Table 5.29.

Table 5.29. Arguments for the canvasKMeans command.

Argument	Description
`-ifp` *fpFile*	Input fingerprint file.
`-icsv` *csvFile*	Input CSV file.
`-autoScale`	Apply autoscaling to the data in the CSV file. Only valid with `-icsv`.
`-o` *outFile*	Output file. Required.
`-k` *clusters*	Number of clusters.
`-runs` *maxRuns*	Maximum number of runs. Default: 10.
`-runSteps` *maxRunSteps*	Maximum number of steps per run. Default: 20.
`-totSteps` *maxTotSteps*	Maximum number of steps over all runs. Default: 200.
`-conv` *dCost*	Stop iterating when the relative change in the cost function drops below *dCost*. Default: 0.001.

5.4.12 canvasLC

This program performs leader-follower clustering. In the leader-follower clustering method, data points are added to a cluster based on a cutoff on their distance from the cluster "leader". The first leader is chosen by some method, then for each data point, the distance from the leader is evaluated. If it is within the cutoff, it is added to the cluster as a "follower". If it is not within the cutoff, a new cluster is created, with this data point as a new leader. Subsequent points are tested against each leader, in order.

This method depends on the choice of the first leader, and on the order of the data points (structures). To eliminate the order dependence, the list can be sorted by the fingerprint bit count. The leaders are then chosen in order of the number of bits set. In the Canvas implementation, the first leader is the first structure in the set that you select for clustering or the set after sorting.

`canvasLC` [*job-options*] {`-ifp` *fpFile*|`-icsv` *csvFile* [*csv-options*]} `-o` *outFile* [*options*]

The job options are listed in Table 5.3. The *csv-options* are the input file options for CSV files listed in Table 5.1 (except `-smi`). The arguments for this program are described in Table 5.30.

Table 5.30. Arguments for the canvasLC command.

Argument	Description
`-autoScale`	Apply autoscaling to the data in the CSV file. Only valid with `-icsv`.
`-dist` *cutoff*	Distance cutoff for clustering. Default: 0.5.
`-group`	Group items in the same cluster together. The first item within a cluster is the leader.
`-helpMetric`	Print types of metrics available with different options and exit.
`-icsv` *csvFile*	Input CSV file.
`-ifp` *fpFile*	Input fingerprint file.
`-metric` *metric*	Metric used to calculate similarity/distance between two molecules. Default is `tanimoto` if `-ifp` is used or `euclidean` if `-icsv` is used. Allowed values are listed in Table 5.22 by index and name, either of which may be used for *metric*.
`-ocsv` *outFile*	Output CSV file. By default, it contains 3 columns: ID, Cluster, and Leader. ID is the name or row index; the latter is used when the input CSV file has no name column (see `-name`). Cluster is the index of the cluster into which an observation (input row) falls. Leader is a 0/1 value that marks the leader of each cluster.

Table 5.30. Arguments for the canvasLC command. (Continued)

Argument	Description
`-odata`	Include property data in the output CSV file. For fingerprint input, the non-fingerprint data from *fpFile* is copied; for CSV input, all fields from *csvFile* are copied.
`-rep {sim|leader}`	Print out only the representatives in each cluster. If `sim` is used, the representative has the minimum average distance from other members in the cluster. If leader is used, the representative is the leader in each cluster.
-s	Sort the fingerprints in descending order of the number of "on" bits. Supported metric types with sorted fingerprints are `tanimoto`, `cosine`, `dice`, `kulczynski`, `mcConnaughey`, `petke`, and `soergel`.

5.4.13 canvasPW2CSV

This program generates a CSV matrix file from a binary pairwise similarity or distance matrix. The command syntax is as follows:

`canvasPW2CSV -i` *binaryFile* `-o` *csvFile* `[-norow] [-nocol]`

The arguments are described in Table 5.31.

Table 5.31. Arguments for the canvasPW2CSV command.

Argument	Description
`-i` *binaryFile*	Input matrix file in binary format.
`-o` *csvFile*	Output destination file containing CSV data.
`-norow`	Omit header row that lists all molecule names in output.
`-nocol`	Omit molecule name as first item of each line in output.

5.5 Model Building and Related Applications

The programs available for statistics, 2D QSAR, neural networks and machine learning are listed in Table 5.32, and are described in the following sections.

Table 5.32. Programs for statistics and 2D QSAR.

Program	Description
`canvasMDS`	Multidimensional scaling.
`canvasMLR`	Build and test multiple linear regression models.
`canvasMolDescriptors`	Calculate molecular descriptors
`canvasPCA`	Direct principal component generation without intermediate analysis
`canvasPCAGen`	Principal component generation
`canvasPCAProj`	Project data along one or more principal components generated by `canvasPCAGen`
`canvasPCAReg`	Build and test principal component analysis regression models.
`canvasPLS`	Build and test partial least square regression models.
`canvasKPLS`	Build and test kernel-based partial least square regression models.
`canvasBayes`	Build and test a Bayes model from binary or continuous training data
`canvasNnet`	Build and test an ensemble model of neural networks
`canvasRP`	Build and test a recursive partitioning model
`canvasSOM` `canvasSOMBits`	Generate Kohonen self organizing map

The programs that build models have many options in common. The common input and output arguments are listed in syntax statements as [*io-args*], and have the following syntax:

`-in` *inFile* `[-d` *delim*`]` `[-out` *outFile*`]` `[-plot` *plotFile*`]`

These arguments are described in Table 5.33.

The common build options specify the dependent and independent variable columns, the training and test sets, and the output file containing the model. They are listed in the syntax statements as [*build-options*], and have the following syntax:

`-y` *yvar* `[-omod` *modelFile*`]`
`[-lt` *trainList* | `-pt` *trainFraction* `[-rand` *seed*`]]`
`[-lx` *xvarList* | `-fieldIn` *list* | `-fieldOut` *list*`]`

Table 5.33. Common input and output arguments for model-building programs.

Argument	Description
`-in` *inFile*	Required. CSV file (delimited text file) containing independent (*x*) variables and, if building a model, a dependent (*y*) variable. The first line must contain a unique name for each column, and each name must start with a letter of the alphabet (A-Z or a-z).
`-d` *delim*	Optional. Delimiter used to separate values in *inFile*. Default: comma.
`-out` *outfile*	Optional. File for program output. Default: standard output.
`-plot` *plotFile*	Optional. CSV file to which observed and calculated *y* values are written. If using `-build` with both the training and test sets, the training set rows are written first.

These options are described in Table 5.34.

Table 5.34. Common build options.

Option	Description
`-y` *yvar*	Required. Name or index of the dependent variable column. For `canvasNnet`, multiple dependent variables can be specified for *yvar*, delimited by spaces. The first column in *inFile* has an index of 1.
`-lt` *trainList*	Training set observations. The second line in the input file is observation 1. *trainList* is a range specification, as described on page 119. Observations not in *trainList* are assigned to the test set. By default, all observations are included in training set.
`-pt` *trainFract*	Randomly assign the specified fraction of observations to the training set, with the remainder assigned to the test set. *trainFract* is a real number between 0 and 1.
`-rand` *seed*	Random seed integer for selecting the training set. If omitted, the seed is assigned from the current local time.
`-omod` *modelFile*	Write model to a file. This file must be created if the model is to be used later.
`-lx` *xvarList*	Independent variable column indices. *xvarList* is a range specification, as described on page 119. The first column in the input file is 1. Not available with `canvasBayes`.
`-fieldIn` *list*	Space-delimited list of field names (columns) to include in the model.
`-fieldOut` *list*	Space-delimited list of field names to exclude from the model. By default, all fields except the *y* variable are included when building the model. However, if input data includes non-numerical columns, use one of the above 3 options to specify the numerical columns to include.

The test options are listed in the syntax statements as [*test-options*], and have the following syntax:

[-y *yvar*|-unknown] -imod *modelFile*

These options are described in Table 5.35.

Table 5.35. Test options for regression models

Option	Description
-y *yvar*	Name or index of the dependent variable column. For canvasNnet, multiple dependent variables can be specified for *yvar*, delimited by spaces.
-unknown	Dependent variable data are not known.
-imod *modelFile*	File containing previously created model.

5.5.1 canvasMDS

This program preforms multi-dimensional scaling (metric scaling based on principal coordinate analysis) on the output from canvasCSVMatrix or canvasFPMatrix, or an external distance matrix in CSV format.

canvasMDS {-i *dmFile*|-icsv *dmFile*} -o *outputFile* [*options*]

The input file can be in binary format (-i) or CSV format (-icsv); the output file (-o) is in CSV format.

Table 5.36. Options for the canvasMDS command.

Option	Description
-n *dim*	Number of dimensions to use. Default: 2.
-eigval *neig*	Print out *neig* eigenvalues from the principal coordinate analysis.
-d *outputDelim*	Output delimiter. Default: comma.
-di *inputDelim*	Delimiter used in the input file if it is in CSV format. Default: comma.
-nameCol *col*	Column that contains the molecule names in a CSV input file. If the names are in the last column, use last for *col*. Default:1.
-noHeader	The input CSV file does not have a header row.

5.5.2 canvasMLR

This program builds and tests multiple linear regression models using data supplied in a CSV (delimited) file. In the build mode, a model is developed on a training set and optionally applied to a test set contained in the same file as the training set. In the test mode, an existing model is applied to a test set contained in its own file. The command syntax is as follows:

```
canvasMLR [job-options] input-args {-build build-options model-options|
    -test test-options}
```

The job options are listed in Table 5.3. The input arguments are described in Table 5.33, the common build options are described in Table 5.34, and the test options are described in Table 5.35. The syntax of the model options is as follows:

```
    -single [-noIntercept] |
    -best -nx numXvar [-steps n] [-start t1] [-stop t2]
        [-ensemble [-keep m] [-weight]]
```

The model options are described in Table 5.37.

Table 5.37. Build options for the canvasMLR command.

Option	Description
`-single`	Build a single MLR model from a list of independent variables. One of the options `-lx`, `-fieldIn` and `-fieldOut` is required with this model.
`-noIntercept`	Suppress *y* intercept term in the regression. Forces the regression lines to pass through the origin.
`-best`	Attempt to identify the best model containing a given number of independent variables, using a simulated annealing Monte Carlo technique.
`-nx` *numXvar*	Number of independent variables to include in model. Required with `-best`.
`-steps` *n*	Number of Monte Carlo steps. Default: 1000.
`-start` *t1*	Initial temperature factor. The actual temperature is the product of *t1* and the standard deviation in *y*. Default: 0.5.
`-stop` *t2*	Final temperature factor. Default: 0.05.
`-ensemble`	Build an algebraically averaged ensemble model from among those with the lowest standard deviation of regression.
`-keep` *m*	Number of models to include in ensemble. Default: 5.
`-weight`	Weight each model by its R-squared value. Default: do not weight.

5.5.3 canvasMolDescriptors

This program calculates molecular descriptors from a set of structures.

`canvasMolDescriptors -i`*fmt infile* `-o`*fmt outFile inputFileArgs* [*job-options*] [*options*]

The input file is specified with the arguments listed in Table 5.1. The job options are standard Job Control options, listed in Table 5.3. This program can be distributed over multiple processors. The options are described in Table 5.38, and include the output file specifications; the `-v3` option in Table 5.2 for SD files is also supported. You can specify as many of the descriptors as you wish, or use `-All` to calculate all descriptors.

Table 5.38. Options for the canvasMolDescriptors command.

Option	Description
`-AlogP`	Calculate atomic logP [9].
`-method last|sum|mean`	AlogP assignment method. Default is `last`.
`-Custom` *customFile*	Calculate a custom property according to the atom types and values supplied in *customFile*. The property is the sum of atom values for all atoms in a molecule.
`-CustomAvg`	Calculate the average atom value for each custom atom type in a molecule.
`-CustomCnt`	Calculate the count of each custom atom type in a molecule.
`-CustomKey`	Assign a value for the presence (1) or absence (0) of each custom atom type in a molecule.
`-CustomName` *customPropName*	Specify the name of the custom property.
`-CustomSum`	Calculate the sum of atom values for each custom atom type in a molecule.
`-Estate`	Calculate electrotopological states [8].
`-EstateAvg`	Calculate the average of each Estate atom type (sum/count) based on `-maxPath` and `-pow` values.
`-EstateCnt`	Calculate the count of each Estate atom type in a molecule.
`-EstateKey`	Indicate the presence (1) or absence (0) of each Estate atom type in a molecule.
`-EstateSum`	Calculate the sum of values for each Estate atom type in a molecule according to the `-maxPath` and `-pow` values.

Table 5.38. Options for the canvasMolDescriptors command.

Option	Description
`-maxPath` *n*	Consider only neighboring atoms within *n* bonds when calculating electrotopological states. Valid only with `-Estate` or `-All`. Default: all neighboring atoms are taken into account.
`-pow` *fallOffPower*	Set how fast the perturbation falls off for electrotopological state calculation. Valid only with `-Estate` or `-All`. Default: 2.
`-HBA`	Count hydrogen bond acceptors.
`-HBAfile` *filename*	File of SMARTS patterns to overwrite Canvas default definition of hydrogen bond acceptors.
`-HBD`	Count hydrogen bond donors.
`-HBDfile` *filename*	File of SMARTS patterns to overwrite Canvas default definition of hydrogen bond donors.
`-distinct`	Count each hydrogen as a distinct donor. By default a given atom that has one or more donatable hydrogens is counted as one donor.
`-MR`	Calculate molar refractivity [10].
`-MW`	Calculate molecular weight.
`-Polar`	Calculate Miller polarizability [11].
`-PSA`	Calculate polar surface area.
`-RB`	Calculate the number of rotatable bonds.
`-RBfile` *filename*	File of SMARTS patterns to modify or overwrite Canvas default definition of rotatable bonds. See the text below for examples of SMARTS patterns.
`-use_ligparse_def`	Use the same set of rules for rotatable bonds as in `ligparse`.
`-All`	Calculate all descriptors (default).
`-helpHBD`	Print out Canvas default definition of hydrogen bond donors. You can use this output as a template for your own definitions.
`-helpHBA`	Print out Canvas default definition of hydrogen bond acceptors. You can use this output as a template for your own definitions.
`-helpRB`	Print out a rotatable bond template file (with no SMARTS patterns).
`-smiles`	Include SMILES string as the first column for each molecule in the CSV output file.
`-fill`	Fill in the line for a molecule that fails to generate descriptors. The line contains no values except the molecule name. This option is useful to create placeholders that preserve positional alignment to external data. Default: skip failed molecules in the output.

Table 5.38. Options for the canvasMolDescriptors command.

Option	Description
`-n` *structureRange*	Input structures to process. *structureRange* is a range specification, as defined on page 119. Default: process all structures.
`-file`	Interpret *structureRange* as a file name. Each line in this file should contain a valid row range specification. Not available with `-iproj`.
`-o`*fmt* *outputFile*	Write results to *outputFile* in the specified format. Valid formats are `mae`, `sd`, and `csv`. Default: CSV format. If *outputFile* is omitted with `-ocsv`, standard output is used. Ignored if `-JOB` is used: the filenames are assigned as *jobname*`_desc.`*ext*, where ext is `csv`, `maegz`, or `sdf.gz`.
`-odata`	Copy all data fields in the input SD or Maestro files to the output file. Only available with `-isd` or `-imae`.
`-uniform`	All the molecules in the Maestro or SD input file have the same data fields.
`-fieldOnly` *fields*	Only the specified fields (space-separated list of property names) in the SD or Maestro input file are saved to the output file.
`-obad` *badMolFile*	Save the molecules that failed to generate descriptors to a file. By default, the list of failed molecules is written to standard output. This option is only available for SMILES, SD or CSV input.

You can create your own SMARTS patterns to define hydrogen-bond donors, acceptors, and rotatable bonds, and supply them in template files to `canvasMolDescriptors`. The format of the template files can be obtained with the `-helpHBD`, `-helpHBA`, and `-helpRB` options. For example, to create a template file for rotatable bonds, use the following command:

```
canvasMolDescriptors -helpRB > myRBtemplate.txt
```

The default definitions are included in the template file for hydrogen-bond donors and acceptors, but not for rotatable bonds. Examples for rotatable bonds are described below.

By default, a rotatable bond in Canvas is defined as a single, non-ring bond, bonded to a non-terminal heavy atom (non-hydrogen). Amide bonds (C–N) and a bond next to a triple bond are excluded because of the high energy barrier.

To include or exclude a user-defined bond type, you must specify both atoms connected to the bond with valid SMARTS. A positive integer value (>0) must follow each SMARTS pattern, separated by a space. Within each file, this integer value must be unique. Only heavy atoms may be specified, and only one SMARTS may be given per line. A line staring with `;` is treated as comment.

For example, the following two lines specifies a secondary amide bond:

```
C(=O)N[!#1]  1
N([!#1])C=O  2
```

If, in addition, you wants to include bonds between OH and SP3 carbon, you can include the following two lines in the “include” section of the template file:

```
C(*)(*)(*)-[OH]  3
[OH]-C(*)(*)*  4
```

If you want to exclude single bonds connecting a carbon attached to a halogen, you can copy the following two lines to the “exclude” section of the file:

```
*-C[I,Br,Cl,F]  5
C([I,Br,Cl,F])-*  6
```

Other SMARTS patterns may be considered in counting rotatable bonds such as acid groups:

```
[OH]-C=O  100
C(=O)-[OH]  200
```

Custom atom types (`-Custom`) are defined in a file that contain rules for assigning atom types. A property assignment rule has the following format:

SMARTS `?` *prop1 prop2* `...`

This rule assigns the property *prop1* to the first atom in the SMARTS pattern, *prop2* to the second atom, and so on. *prop1* is a name that can consist of alphanumeric characters (upper and lower case) and underscores, and must start with a letter. The property can be used instead of SMARTS patterns in a rule, as `$`*propN*. A type assignment rule has the format:

SMARTS `>` *type* `;` {*name*} *value*

This rule assigns the atom type *type* to the first atom in the SMARTS pattern. *type* is a positive integer, which indexes the atom types. The name of the atom type can be set by adding the optional {*name*}. The name is used when reporting the per-atom-type properties (Key, Cnt, Sum, Avg). The optional *value* is used to calculate the custom property.

Rules are applied in the order in which they are encountered in the file, so any property assignments must precede their use. The default atom type is 0. Atom types set by one rule can be changed by a subsequent rule.

5.5.4 canvasPCA

This program performs direct principal component generation without intermediate analysis. The command syntax is as follows:

`canvasPCA` [*job-options*] {`-icsv` *csvFile* [*csv-options*] | `-ifp` *binaryFile*}
`-o` *outputFile* `-ostat` *statFile*

The job options are listed in Table 5.3. The input file can be in CSV format (`-icsv`) or binary fingerprint format (`-ifp`). The output file (`-o`) is in CSV format, and contains the input data projected along all principal components generated. The options are described in Table 5.39.

5.5.5 canvasPCAGen

This program generates and writes out principal components. The command syntax is as follows:

`canvasPCAGen` {`-icsv` *csvFile* | `-ifp` *binaryFPFile*} `-o` *outputFile* [*options*]

The input file can be in CSV format (`-icsv`) or binary fingerprint format (`-ifp`). The output file (`-o`) is in comma-delimited CSV format. The options are the same as for canvasPCA, described in Table 5.39, except that `-ostat` is not recognized and `-d` applies only to an input CSV file.

Table 5.39. Options for the canvasPCA and canvasPCAGen commands.

Option	Description
`-ostat` *statFile*	Output file with variances and loadings. Not valid for `canvasPCAGen`.
`-n` *components*	Number of principal components to use. Default: 2.
`-d` *delimiter*	Input file delimiter if `-icsv` is used; also output file delimiter for `canvasPCA`. Default: comma.
`-noHeader`	The input data has no header row.
`-nameCol` *col*	Column index of the names. If names are in the last column, `last` can be used as the value for *col*. Default: 1.
`-scale`	Scales each dimension to unit variance if it is within the threshold.
`-t` *threshold*	Threshold for scaling. Only valid with `-scale`. Default: 0.001.
`-mostSig` *nbits*	Keep only the *nbits* most informative bits across the chosen input set. Default: use all bits. Only valid with `-ifp`.

5.5.6 canvasPCAProj

This program projects data along one or more principal components generated by `canvasPCAGen`. The command syntax is as follows:

`canvasPCAProj -ipc` *pcFile* `-idata` *dataFile* `-o` *outputFile* [*options*]

The arguments are described in Table 5.40.

Table 5.40. Arguments for the canvasPCAProj command.

Argument	Description
`-ipc` *pcFile*	Input data containing principal components generated by `canvasPCAGen`.
`-idata` *dataFile*	Original data file (input to `canvasPCAGen`) in CSV format.
`-o` *outputFile*	Output file for projections, in CSV format.
`-load`	Project by columns instead of rows to produce PCA loadings.
`-n` *components*	Number of principal components (dimensions) desired. Default: 2.
`-d` *delimiter*	Delimiter to use in the output file. Default: comma.
`-ddata` *delimiter*	Delimiter used in the original data file. Default: comma.
`-noHeader`	No header row is in the original data file.
`-nameCol` *col*	Column index of the names in the original data file. Default: 1.

5.5.7 canvasPCAReg

This program builds and tests principal component analysis regression models using data supplied in a CSV file. In build mode, a model is developed on a training set and optionally applied to a test set contained in the same file as the training set. In test mode, an existing model is applied to a test set contained in its own file. The command syntax is as follows:

`canvasPCAReg` [*job-options*] *io-args* {`-build` *build-options* *model-options* | `-test` *test-options*}

The job options are listed in Table 5.3. The input arguments are described in Table 5.33, the common build options are described in Table 5.34, and the test options are described in Table 5.35. The syntax of the model options is as follows:

`-maxf` *maxFactors* [`-autoScaleOff`]

The model options are described in Table 5.41.

Table 5.41. Model options for the canvasPCAReg command.

Option	Description
`-maxf` *maxFactors*	Maximum number of PCA factors to use in building the model. The model contains all factors from 1 to *maxFactors*.
`-autoScaleOff`	Do not use auto-scaling when creating the model. Default: use autoscaling.

5.5.8 canvasPLS

This program builds and tests partial least squares regression models using data supplied in a CSV (delimited) file. In build mode, a model is developed on a training set and optionally applied to a test set contained in the same file as the training set. In test mode, an existing model is applied to a test set contained in its own file.

`canvasPLS` [*job-options*] *io-args* {`-build` *build-options* *model-options* | `-test` *test-options*}

The job options are listed in Table 5.3. The input arguments are described in Table 5.33, the common build options are described in Table 5.34, and the test options are described in Table 5.35. The syntax of the model options is as follows:

`-maxf` *maxFactors* [`-autoScaleOff`] [`-sd` *sdLimit*] [`-tmin` *minTvalue*]

The model options are described in Table 5.42.

Table 5.42. Model options for the canvasPLS command.

Option	Description
`-maxf` *maxFactors*	Maximum number of PLS factors to use in building the model. The model contains all factors from 1 to *maxFactors*.
`-autoScaleOff`	Do not use auto-scaling when creating the model. Default: use autoscaling.
`-sd` *sdLimit*	If the standard deviation of the regression is less than or equal to *sdLimit*, stop adding PLS factors during model building. Overrides *maxFactors*. Default: -1.0.
`-tmin` *minTvalue*	Minimum T-value for selecting significant *x* variables. Must be positive.

5.5.9 canvasKPLS

This program builds and tests kernel-based partial least squares regression models using data supplied in a CSV (delimited) file. In build mode, a model is developed on a training set and optionally applied to a test set contained in the same file as the training set. In test mode, an existing model is applied to a test set contained in its own file.

`canvasPLS` [*job-options*] *io-args* {`-build` *build-options* *model-options* | `-test` *test-options*}

The job options are listed in Table 5.3. The input arguments are described in Table 5.33, the common build options are described in Table 5.34, and the test options are described in Table 5.35. The syntax of the model options is as follows:

`-maxf` *maxFactors* [`-sigma` *sigma*]

The model options are described in Table 5.43.

Table 5.43. Model options for the canvasKPLS command.

Option	Description
`-maxf` *maxFactors*	Maximum number of KPLS factors to use in building the model. The model contains all factors from 1 to *maxFactors*.
`-sigma` *sigma*	Set the linearity parameter. Larger values mean more linear. Allowed range is from 1 to 100. Default: 30.0.

5.5.10 canvasBayes

This program builds and tests a Bayes model from binary or continuous training data to predict the probability of a molecule at each activity level [14]. The syntax of the command is as follows:

```
canvasBayes [job-options] {-in CSVFile [-d delim] [-prefix string]
    [-name nameCol] | -infp FPBinaryFile}
    [-out outFile] [-plot plotFile [-bin]]
    {-build build-options model-options|-test test-options}
```

The job options are listed in Table 5.3. The input arguments are described in Table 5.44, the common build options are described in Table 5.34, and the test options are described in Table 5.35. The syntax of the model options is:

```
    [-category] [-c cutoffsList] [-ebin n] [-s coefficient]
    [-KL cutoff|-KLPos cutoff|-KLNeg cutoff|-F cutoff ] [-binary|-nobinary]
    [-noise value]
```

The model options are described in Table 5.45.

Table 5.44. Input and output arguments for the canvasBayes command.

Argument	Description
`-in` *CSVFile*	CSV file containing independent (x) variables and, if building a model, a dependent (y) variable. The first line must contain a unique name for each column, and each name must start with a letter of the alphabet (A-Z or a-z).
`-d` *delim*	Optional. Delimiter used to separate values in *inFile*. Default: comma.
`-prefix` *string*	Prefix for binary columns in the header of the CSV input file. The default prefix of a binary column is `BIT`.
`-name` *nameCol*	Name column in the CSV file. Default: first column.
`-infp` *FPFile*	Binary file generated by `canvasFPGen`, containing fingerprint bits and optionally other molecular property data as independent variables and, if building a model, a dependent variable.
`-out` *outfile*	File for program output. Default: standard output.
`-plot` *plotFile*	CSV file to which the observed and calculated categories that the dependent variable belongs to are written. If using `-build` with both the training and test sets, the training set rows are written first.
`-bin`	If specified, only the indexes, instead of the full range, of the observed and calculated categories are written to *plotFile*.

Table 5.45. Model options for the canvasBayes command.

Option	Description
`-category`	Chosen activity field is categorical, with no implicit scale or order.
`-c` *cutoffsList*	Cutoffs for the numerical activities. Default: use all distinct values.
`-ebin` *n*	Divide the training data into *n* equal sized bins based on the values of dependent variable. *n* must be at least 2.
`-s` *coefficient*	Smoothing coefficient, typically (0-1). Default: 1e-07.
`-KL` *cutoff*	Kullback-Leibler distance cutoff used during training. Binary features with a significance less than or equal to *cutoff* are excluded from the model. Applied to both positively and negatively correlated features. Default: 0.1.
`-KLPos` *cutoff*	Kullback-Leibler distance cutoff used during training. Applies to positively correlated features only. Can be used with `-KLNeg`.
`-KLNeg` *cutoff*	Kullback-Leibler distance cutoff used during training. Applies to negatively correlated features only. Can be used with `-KLpos`.
`-F` *cutoff*	Fraction of bits to keep during training, as ranked by Kullback-Leibler significance. Remaining bits are excluded. Default: 1.
`-binary`	Use only binary data to build a model.
`-nobinary`	Do not use binary data to build a model.
`-noise` *value*	Add random noise with standard deviation of *value* to all activities. Categorical fields are scrambled with a probability provided by *value*. New values are chosen by their relative occurrences.

5.5.11 canvasNnet

This program builds and tests an ensemble model of neural networks with data supplied in a CSV (delimited) file. The network has three layers, one input layer, one output layer, and one hidden layer. Networks are trained using a BFGS algorithm. In build mode, a model is developed from a training set, with built-in cross-validation. Optionally, this model can be applied to a test set in the same file as the training set. In test mode an existing model is applied to a test set in its own file. The command syntax is as follows:

`canvasNnet` [*job-options*] *io-args* {`-build` *build-options* *model-options*|`-test` *test-options*}

The job options are listed in Table 5.3. The input and output arguments are described in Table 5.33, the common build options are described in Table 5.34, the model options are described in Table 5.46, and the test options are described in Table 5.35.

Table 5.46. Model options for the canvasNnet command.

Option	Description
`-cvp` *cvPercent*	Percentage of the training set used for cross-validation (randomly selected). Default value is 0.1 (10%).
`-cycle` *nCycle*	Training cycles for each network. Default is 200 cycles.
`-nnet` *nNetwork*	The number of networks to train. Default is 20.
`-ensemble` *n*	The number of best networks to form the ensemble model. Default is 5.

5.5.12 canvasRP

This program builds and tests recursive partitioning trees with data supplied in a CSV file. The command syntax is as follows:

`canvasRP` *inFile* `[-d` *delimiter*`]` `[`*outFile*`]` `[`*job-options*`]` `{-build` *build-options* *model-options*`|-test` *test-options*`}`

The input file is a CSV file that contains x and y variables for the training and test sets. The default delimiter is a comma, but you can specify it with the `-d` option. Use `-d " "` for space and `-d "\\t"` for tab. Consecutive spaces are treated as a single delimiter. The output file name is optional: the default is standard output.

The common build options are described in Table 5.34, the model options are described in Table 5.47, and the test options are described in Table 5.48. The job options are described in Table 5.3.

The syntax of the model options is:

```
[-ensemble n [-sample fraction] [-pavg] [-hist]] [-ut] [-r minCorr]
[[-randattr perc] [-uniqattr] ] | [-diffattr]
[-split gini|gain] [-avgsplit ] [-leaf minObs] [-tree]
[-category] | [-c <cutoffsList>] [-nocheck]
```

Table 5.47. Model options for the canvasRP command.

Option	Description
`-exclude`	Treat *list* in the `-lx` parameter as the variables to exclude.
`-ensemble` *n*	Build an ensemble model of *n* trees, where each tree is created from a random sample of the training set.
`-sample` *fraction*	Training set sample fraction. Default: 0.8.
`-pavg`	Assign the class of a given observation based on the average probabilities from all leaf nodes to which it is assigned. By default, the probabilities are assigned from the number of votes for each class.
`-hist`	Show histogram of multiple recursive partitioning tree votes, including the breakdown for each category (TruePositives, FalsePositives, TrueNegatives, FalseNegatives)
`-ut`	Build each tree with a different root attribute.
`-r` *minCorr*	Minimum x:y Pearson correlation coefficient evaluated over the training set. Any x variable not meeting this threshold is ignored.
`-randattr` *perc*	For each tree, select attributes from a random subset with the specified percentage of the total attributes.

Table 5.47. Model options for the canvasRP command. (Continued)

Option	Description
`-uniqattr`	If the attributes are selected randomly (`-randattr`), this argument assures that trees do not share possible attributes. If *perc* is larger than 1/(number of trees), then this parameter defaults to `-diffattr`.
`-diffattr`	Specify that an equal number of attributes (1/(number of trees)) should be selected for each tree.
`-split gini|gain`	Splitting quality measure: `gini` Gini impurity (default) `gain` Information gain
`-avgsplit`	If a range of values yields equivalent splitting quality, use the average.
`-leaf` *minObs*	Minimum number of observations in any leaf node. Default: 5.
`-tree`	Write out trees in human-readable format.
`-category`	Flag to indicate chosen activity field is categorical, with no implicit scale or order. (Default)
`-c` *cutoffsList*	Cutoffs for the numerical activities. Default: use all distinct values.
`-nocheck`	Omit checks on reasonable assortments among categories.

The syntax of the test options is:

`-imod` *modelFile* `[-unknown] [-hist] [-accdetail]`

Table 5.48. Test options for the canvasRP command.

Option	Description
`-imod` *modelFile*	Input model file. The variable names in this file must be present in *inFile*.
`-unknown`	Dependent y variable is not present in *inFile*.
`-hist`	Show histogram of multiple recursive partitioning tree votes, including the breakdown for each category (TruePositives, FalsePositives, TrueNegatives, FalseNegatives)
`-accdetail`	Show accuracy of each tree on test set.

5.5.13 canvasSOM and canvasSOMBits

These two program creates a Kohonen self-organizing map (SOM) from data in a CSV file or a binary fingerprint file. `canvasSOM` creates the map from scaled (real) values, whereas `canvasSOMBits` creates the map from binary fingerprint data. Both programs have the same syntax, which is as follows:

`canvasSOM`|`canvasSOMBits` [*job-options*] {`-icsv` *csvFile*|`-ifp` *fpFile*}
`-outmap` *outputMap* [*options*]

The job options are listed in Table 5.3. The input can come from a CSV file (`-icsv`) or a binary fingerprint file (`-ifp`). For `canvasSOM`, the data should be generated with `canvasFPGen` using the `-scaling` option. For both programs, it is recommended to use `-min` and `-max` with `canvasFPGen` to limit the number of bits, though `canvasSOMBits` can handle much large numbers of bits than `canvasSOM`. The output map is written to the file specified with `-outmap`. The arguments are described in Table 5.49.

Table 5.49. Arguments for the canvasSOM command.

Option	Description
`-inmap` *inputMap*	Existing SOM map on which to place new data. Overrides `-topology`, `-xdim`, and `-ydim`. can optionally lock or tether any/all cells.
`-outdata` *file*	Name of file where cell membership info is stored. Default: stdout.
`-outrun` *file*	Name of file where run information is written. Default: stdout.
`-outstats` *file*	name of file where class statistics is stored.
`-cycles` *N*	Maximum number of training cycles. Default: 100.
`-update` *N*	Status update interval, in cycles. Default: 1.
`-topology` *type*	Topology. Allowed values are `rect` (rectangular) and `hexa` (hexagonal). Overridden if an existing map is read with `-inmap`. Default: `rect`.
`-xdim` *N* `-ydim` *N*	Lattice 2D dimensions. Default: 10 x 10. Only valid with `-topology`.
`-hcube` *bits*	Use hypercube lattice with 2^{bits} cells. Not valid with `-topology`, and implies `-wrap`.
`-kmeans` *k*	Use kmeans lattice with k cells. *k* should be an integer no smaller than 2. Not valid with `-topology`.
`-wrap`	Make lattice topology periodic. Default: nonperiodic.
`-noshuffle`	Do not randomize compound order. Default: randomize order.
`-stop`	Terminate training whenever the rms quantization error rises. Default: continue training.
`-stopWhen` *rmsqe*	Terminate training if the specified rms quantization error is reached.
`-decay` *type*	Training decay. Allowed values: `linear`, `power`, `inverse`, `expo` (exponential). Default: `expo`.
`-seed` *int*	Use the specified value to seed random number generator.
`-idcol` *N*	Zero-based index of identifier column. Default: 0.
`-catcol` *N*	Zero-based index of category column. Default: none.
`-blocksize` *N*	Number of records to load into memory at a time. Default: 1000.
`-fast`	Run faster by updating weights selectively. Recommended for large data sets.
`-round`	Force SOM weights to 0 or 1 after each training cycle. Valid only for `canvasSOMBits`.
`-maxdist` *float*	Assign no cell for items whose nearest cell exceeds this distance. Default: no limit.

5.6 Utilities

In addition to the programs described above, various utilities are available for job management, format conversion, and various other tasks. These utilities are listed in Table 5.50.

Table 5.50. Canvas utilities.

Utility	Description
`canvas_app`	Run jobs that are set up by `canvasJob`.
`canvasConvert`	Convert between molecular file formats.
`canvasJob`	Set up and clean up jobs associated with a Canvas project.
`canvasProjectDB`	Create and update a Canvas project database.
`canvasSDMerge`	Merge CSV data with an existing SD file.
`canvasSearch`	Search a list of target molecules against a set of queries.
`canvasMCS`	Find the maximum common substructure (MCS) in a set of molecules.
`canvasScaffold`	Decompose structures into ring-containing scaffolds and sort the scaffolds.

5.6.1 canvas_app

This utility runs a job according to the type and parameters specified in the input file. The syntax is as follows:

`canvas_app [-exec|-inc]` *jobName*

jobName is the name of the job to be run. The input file *jobName*`.inp` must exist; this file is created by `canvasJob`. The `-inc` option can be used to incorporate results into the Canvas project from a run that has completed already. The `-exec` option can be used to run a job without incorporating results into the Canvas project.

5.6.2 canvasConvert

This utility converts between SD, Maestro, Canvas compact, CSV and SMILES molecular structure file formats.

`canvasConvert -i`*fmt inFile* [*input-options*]
{`-o`*fmt outFile*|`-div N [-prefix` *basename*`]`} [*output-options*]

The available *fmt* format options for both input and output are:

`smi`	SMILES format
`sd`	SD format
`mae`	Maestro format
`csv`	CSV format

The input file arguments (including options) are the common arguments listed in Table 5.1; the output file arguments (including options) are the common arguments listed in Table 5.2. The remaining output options are described in Table 5.51.

Table 5.51. Output file options for the canvasConvert command.

Option	Description
`-div` *N*	Divide the original file into *N* parts, each containing approximately the same number of structures. The file is of the same type as the input file. The header line in an input CSV file, if it exists, is written to each of the divided files. Not valid with `-o`*fmt* options.
`-prefix` *basename*	When dividing a file with `-div`, name the files *basename*`_pt`*K*`.`*ext*, where *K* is an integer from 1 to *N*. If not specified, *basename* is taken from the input file name (without the path).
`-n` *structureRange* `[-file]`	List of input structures to write to the output file. *structureRange* is a range specification, as described on page 119. If `-file` is used, *structureRange* is a text file containing the row specifications. Default: process all structures.
`-uniform`	Each structure block in a Maestro or SD input file contains the same set of properties.
`-u`	Output to SMILES or CSV file as canonical (unique) SMILES.
`-k`	Output to SMILES or CSV as kekulized SMILES of aromatic atoms and bonds.
`-id`	Append molecule name after the SMILES string in the SMILES output file, separated by a space.
`-obad` *badMolFile*	Save the molecules that failed to convert in the specified file in the original format. Only available with `-ismi`, `-isd` and `-icsv`. Default: print to stdout.
`-compress`	Compress output files, including the file of failures *badMolFile*. The extension `.gz` is added automatically. Compression can also be requested by specifying the file with the appropriate extension.
`-2D`	Convert output coordinates to 2D.
`-allH`	Force output of all hydrogens.
`-noH`	Do not include hydrogens in output structures.
`-silent`	Suppress console output.

5.6.3 canvasJob

This program does setup and cleanup for jobs associated with a Canvas project. The command syntax is as follows:

```
canvasJob -setup jobName -run program -proj projName [-helpColumns]
    -arg "arguments" [options]
canvasJob -cleanup jobName
canvasJob -help | -helpSetup
```

The arguments (and options) are described in Table 5.52.

Table 5.52. Arguments and options for the canvasJob command.

Argument	Description
`-arg` "*arguments*"	Command line options for *program* enclosed in double quotes. Omit options that refer to an input or output file for *program* (e.g., `-isd` *sdFile*), or specific rows and columns to used, since those options are deduced from information provided to `canvasJob`.
`-cleanup` *jobName*	Clean up after `canvas_app` has run the specified job. May not be used in combination with any other options.
`-cols` *list*	The indices of the property columns to be used by *program*. *list* is a range specification, as defined on page 119. Use `-helpColumns` to see the mapping of column indices to column names. Do not include the structure column, since `canvasJob` determines whether it is needed based on the specified Canvas program. This option is required whenever *program* operates on properties (such as `canvasBayes`, `canvasMLR`), but it should not be used for programs that operate only on structure (such as `canvasFPGen`, `canvasMolDescriptors`).
`-help`	Print usage message and exit.
`-helpColumns`	Print the correspondence between column indices and column names in the Canvas project *projName* and exit. No other options are required.
`-helpSetup`	Print information on setting up a job, including the list of programs that can be used and how to set up the `-arg` option.

`-model` *prevJob* | `-model` *modelFile*	Use the model created during a previous run with the same project, or specify the file name of a previously built model. The file extension must be one of:
	`canvasPLS` `.pls`
	`canvasMLR` `.mlr`
	`canvasPCAReg` `.pca`
	`canvasNnet` `.nnet`
	`canvasBayes` `.bayes`
	`canvasSOM` `.som`
	`canvasSOMBits` `.som`

Table 5.52. Arguments and options for the canvasJob command. (Continued)

Argument	Description
`-proj` *projName*	Canvas project name, including absolute path.
`-rows` *list* `[-file]`	The project rows to process. *list* is a range specification, as defined on page 119. All rows are processed by default. If `-file` is used, *list* is the name of a binary file, written by the Canvas GUI with View > Export Row IDs, that contains the row selection.
`-rows2` *list* `[-file2]`	Project rows to process for the second input file when setting up a job for `canvasDBCS` or `canvasFPMatrix` (the file specified by `-ifp2`). If omitted, no second input file is used. *list* is a range specification, as defined on page 119. If `-file2` is used, *list* is the name of a binary file, written by the Canvas GUI with View > Export Row IDs, that contains the row selection.

`-run` *program*	Canvas program to run. Valid programs are:
	`canvasBayes` `canvasPCA`
	`canvasFPGen` `canvasPCAReg`
	`canvasMLR` `canvasPLS`
	`canvasMolDescriptors` `canvasSOM`
	`canvasNnet` `canvasSOMBits`

`-setup` *jobName*	Set up a Canvas job. The input file *jobName*`.inp` is created with options for `canvas_app`, which actually runs the job.
`-y` *list*	Dependent variable columns. *list* is a range specification, as defined on page 119. Use only for programs that build models, such as `canvasBayes`, `canvasMLR`.

5.6.4 canvasProjectDB

This program can be used to create and update a Canvas project database. It can be run as a regular foreground process, or as a single-CPU job on any host that has access to the project directory.

`canvasProjectDB` [*job-options*] `-proj` *projName*`.cnv`
[*import-options*|*export-options*]

The project name, specified by `-proj`, must include the full path, and end in `.cnv`. The Job Control options are the standard options listed in Table 5.3. The import options include the common input file options described in Table 5.1, except that `-i`*fmt* and `-fieldAsName` are extended as described below. These and the other import options have the following syntax:

`-i`*fmt* *source* {`-new`|`-append`|`-replace`|`-update` *prop1*,*prop2* [`-dup`]}
[`-helpColumns`] [`-skipDupStruct`] [`-merge`] [`-uniform`]
[`-fieldAsName` *field*] [`-rows` *rowRange* [`-file`]] [`-noIndex`]

These options are described in Table 5.53.

Table 5.53. Import options for the canvasProjectDB command

Option	Description
`-i`*fmt* *source*	Source of structures or properties to be imported. The supported formats are: `-imae` Maestro file `-isd` SD file `-icsv` SMILES, name and properties `-ismi` Space or tab-separated SMILES and name `-ifp` Binary fingerprints file generated by Canvas (`.fp`) `-iproj` Another Canvas project (`.cnv`). Must be given as absolute path. Include absolute path in *source* to avoid file copy when running as a job. Valid extensions are listed on page 119.
`-helpColumns`	Print the correspondence between column indices and column names in the Canvas project and exit. Not valid with `-new`.
`-skipDupStruct`	Skip duplicate structures in *source* file during import. Only the first of the duplicates is imported.
`-merge`	Specify if all fields in a CSV file are to be merged into a project. No name or SMILES field is required in the CSV file. Only valid with `-update`.
`-uniform`	Each structure in *source* contains the same set of properties. Valid only with `-isd` or `-imae`.
`-new`	Create a new Canvas project, then import.
`-append`	Append to an existing project.

Table 5.53. Import options for the canvasProjectDB command (Continued)

Option	Description
`-replace`	Delete all project records, then import.
`-update` *prop1*,*prop2*	Use the specified property mapping to add new records or update properties for existing records. *prop1* is the name (`-imae`, `-isd`, `-ifp`, `-iproj`) or column index (`-icsv`, `-ismi`, `-iproj`) of a string or integer property in *source*. *prop1* must be 2 with `-ismi`, must be at least 2 with `-iproj`. Omit *prop1* if you wish to use the default name of the molecule or fingerprint in *source*. If a column name starts with a number, use escape plus single quote around the name. e.g. use `'6'` for a column named `6`. *prop2* is the name of a string or integer property in the existing Canvas project database. If a column name starts with a number, use escape and single quote—see above. Omit *prop2* if you wish to use the ID that appears in the Structure column of the Canvas spreadsheet. You must use "`_uid_`" for *prop2* if you wish to use the unique row id in the Canvas project. If the value of *prop1* is unique with respect to all *prop2* values, a new database record is created to hold the imported structure and its properties. If, however, the value of *prop1* matches the value of *prop2* for an existing database record, the imported properties are merged into the existing database record, and the structure is not changed.
`-dup`	Perform the update even if the project contains records with duplicate *prop2* values. If this flag is not used, canvasProjectDB terminates when duplicate *prop2* values exist. Valid only with `-update`.
`-fieldAsName` *field*	Get structure names from *field*. This is the ID that appears in the Structure column of the Canvas spreadsheet. *field* must be the name (`-imae`, `-isd`, `-iproj`) or column index (`-icsv`, `-ismi`) of a string or integer property in *source*. *field* is ignored with `-ismi` since structure names always come from the second column when it is present. If this option is not used, structure names are assigned as follows: `-imae` `s_m_title` property `-isd` First line of each CT block `-icsv` Sequential integers starting with 1 `-ismi` Second column, or sequential integers starting with 1 `-iproj` ID that appears in the Structure column
`-rows` *rowRange*	Rows to import. *rowRange* is a range specification, as defined on page 119. All rows are imported by default.
`-file`	Interpret *rowRange* as a file name. Each line in the specified file should contain a valid range specification.
`-noIndex`	Do not generate an index for substructure matching when adding records to the project.

The export options include the `-v3` output file option given in Table 5.2, and the remaining arguments have the following syntax:

```
-ofmt outputFile [-compress] [-u] [-helpColumns] [-uidAsName]
[-rows rowRange [-file]] [-cols colRange] [-view viewName]
```

These options are described in Table 5.54.

Table 5.54. Export options for the canvasProjectDB command.

Option	Description
`-o`*fmt outputFile*	Destination of exported structures or properties. Supported formats: `-omae` Maestro file `-osd` SD file `-ocsv` Comma-separated SMILES, name and properties (`.csv`) `-osmi` Space-separated SMILES and name (`.smi`) `-ofp` Canvas fingerprint file (`.fp`). Must choose a single fingerprint column and any number of other non-fingerprint columns.
`-compress`	Compress output file. If specified, `.gz` is automatically appended to *outputFile*. Compression can also be requested by using the appropriate extension on *outputFile*. Not valid with `-ofp`.
`-u`	Write out canonical SMILES strings. Valid only with `-osmi` and `-ocsv`.
`-uidAsName`	Use the unique row ID in the Canvas project as the molecule name.
`-rows` *rowRange*	Rows to export. *rowRange* is a range specification, as defined on page 119. Default: export all rows.
`-file`	Interpret *rowRange* as a file name. Each line in the specified file should contain a valid range specification.
`-cols` *colRange*	Columns to export. *colRange* is a range specification, as defined on page 119. Column 1 (the structure) is always exported with `-omae`, `-osd`, and `-osmi`. This option must be used when exporting a fingerprint column. Default: export all columns.
`-helpColumns`	Print the correspondence between column indices and column names in the Canvas project and exit. No other options are required.
`-view` *viewName*	Export rows and columns from the saved view *viewName*.

5.6.5 canvasSDMerge

This utility merges CSV data with an existing SD file into a new SD file, preserving the original SD file order. The command syntax is

```
canvasSDMerge -isd sdfile -icsv csvfile [-key1 sdKey] [-key2 csvKey]
    [-not1 excludeList|-only1 includeList] [-not2 excludeList|-only2 includeList]
    [-o outputSDFile] [-keep1|-keep2|-keepboth]
```

The CSV input file must be comma-delimited. The options are described in Table 5.55.

Table 5.55. Options for the canvasSDMerge command.

Option	Description
`-key1` *sdKey*	Name of the field in the SD file to use as the key for merging. Default: molecule title.
`-key2` *csvKey*	Name of the field in the CSV file to use as the key for merging. Default: first column.
`-not1` *excludeList*	Fields in the SD file to exclude from the merge, space-delimited. Default: do not exclude any fields.
`-only1` *includeList*	Fields in the SD file to include in the merge, space-delimited. Default: include all fields.
`-not2` *excludeList*	Fields in the CSV file to exclude from the merge, space-delimited. Default: do not exclude any fields.
`-only2` *includeList*	Fields in the SD file to include in the merge, space-delimited. Default: include all fields.
`-o` *outputSDFile*	Output file for results after merging. Default: standard output.
`-keep1`	Retain value from SD file when field names collide.
`-keepboth`	Retain value from both files when field names collide. This is the default.
`-keep2`	Retain value from CSV file when field names collide.

5.6.6 canvasSearch

This program searches a list of target molecules against a set of queries, composed of either molecules or partial structures. This program can also be used to filter a target list based on either standard REOS rules or a user-defined file containing SMILES queries and minimum and maximum number of times a query should be matched. Filtering and searching can be performed separately or sequentially (filtering first). The command syntax is as follows:

```
canvasSearch -ifmt structureFile [job-options] [input-options]
    [-ofmt matchFile] [-ofmt2 failFile] [-no2DCoord] [-v3]
    [-n selection] [-qfmt queryFile [-require n|-qmol mdlFile]] [-exact]
    [-filter [-reos] [-file ruleFile [-d delimiter]] [-maxVio n] ]
    [-index indexFile|-newIndex indexFile] [-noIndex] [-helpREOS]
    [-matchCount countFile -qmap queryMapFile -prefix queryPrefix
        [-showAll] ]
```

The job options are standard Job Control options, listed in Table 5.3. This program can be distributed over multiple processors. The input file options are listed in Table 5.1. The remaining arguments are described in Table 5.56

Table 5.56. Arguments for the canvasSearch command.

Argument	Description
`-ifmt` *targetFile*	Structure file containing the target molecules. The supported formats are: `-imae` Maestro file `-isd` SD file `-ismi` Space or tab-separated SMILES and name `-iproj` Canvas project. Must be given as absolute path. Valid extensions are listed on page 119.
`-n` *selection*	Selected molecules in *targetFile* to search. *selection* is a range specification, as defined on page 119. Default: search all molecules.
`-index` *indexFile*	Use the specified previously-generated index file of all the molecules in *targetFile* in the search. A matching fingerprint is generated for each query. Not valid with `-iproj`.
`-newIndex` *indexFile*	Generate index file of both the target molecules and the queries before search. Not valid with `-iproj`. The saved index file of the target molecules can be used later with the `-index` option.
`-noIndex`	Do not use any index for search, even if present. Supersedes `-index` and `-newIndex`.
`-filter`	Filter the target file based on the maximum and optionally the minimum number of counts of a given set of patterns.

Table 5.56. Arguments for the canvasSearch command. (Continued)

Argument	Description
`-reos`	Use Rapid Elimination Of Swill, a set of rules to identify lead-like molecules. Only valid with `-filter`.
`-file` *ruleFile*	Rule file to use. Each line must contains one SMARTS/SMILES string, followed by the minimum and maximum number of allowed counts, and an optional comment surrounded by double quotes. Only valid with `-filter`.
`-d` *delimiter*	Delimiter used to separate each field in *ruleFile*. Default: tab character `'\t'`.
`-maxVio` *n*	Maximum number of violations allowed for the rules. Only valid with `-filter`. Default: 0.
`-helpREOS`	Print out REOS patterns, each followed by the minimum and maximum number of counts. Tab is used as the delimiter in each line.
`-q`*fmt queryFile*	Query file containing list of queries, which can be whole molecules or fragments. The supported formats are: `-qmae` Maestro file `-qmol` MDL Mol file (single query only) `-qsd` SD file `-qsmi` SMILES or SMARTS patterns, one per line.
`-require` *n*	Require the target molecules to match at least *n* queries. This option should only be used with `-q`*fmt queryFile*, except for MDL Mol query files. Default: match all queries.
`-exact`	Require an exact match for each query. Default is to match by substructure.
`-o`*fmt matchFile*	Target molecules that passed the filter (if `-filter` is used) and matched all or the required number of queries.The supported formats are: `smi` SMILES file. Input SMILES strings are written for SMILES input, Canvas-generated SMILES strings are written otherwise. `mae` Maestro file. `sd` SD file. Valid extensions are listed on page 119.
`-o`*fmt*`2` *matchFile*	Target molecules that failed to match.The supported formats are: `smi` SMILES file. Input SMILES strings are written for SMILES input, Canvas-generated SMILES strings are written otherwise. `mae` Maestro file. Input must be in Maestro format. `sd` SD file. Valid extensions are listed on page 119.
`-no2DCoord`	Do not generate coordinates in the output SD or Maestro file if SMILES was used as input. Default: generate 2D coordinates
`-v3`	Write output SD files in MDL version 3 format.

Table 5.56. Arguments for the canvasSearch command. (Continued)

Argument	Description
`-matchCount` *countFile*	Calculate the number of matches to each query or filtering pattern, 0 for no match. Counts are saved in *countFile*, which is in CSV format. If `-filter` and `-qf`*fmt* are both used, only queries in the latter are listed. By default, only targets that passed the filter (if `-filter` is used) and matched all or the required number of queries are printed out to *countFile*. Use `-showAll` to include all targets.
`-prefix` *queryPrefix*	Specify the heading prefix in the match count CSV file. Each query in *countFile* is represented by the following format: *queryPrefix*`::query`*n*. *queryPrefix* can be a search name, such as `my_search1`. Only valid with `-matchCount`.
`-qmap` *queryMapFile*	This file provides the mapping between the *queryPrefix*`::query`*n* headings in the *countFile* and the actual SMARTS/SMILES query patterns. Only valid with `-matchCount`.
`-showAll`	Write counts for all targets to *countFile*. Only valid with `-matchCount`.

5.6.7 canvasMCS

This utility finds the maximum common substructure (MCS) among a given set of molecules. The command syntax is as follows:

`canvasMCS -i`*fmt inputFile* [*input-options*] `-o`*fmt outputFile* [*output-options*] [*options*]

The input file arguments (and options) are the common arguments listed in Table 5.1, and in addition input from Canvas projects is supported with `-iproj`. The valid output formats are `mae`, `sd`, and `csv`, as described in Section 5.1 on page 119. The common output arguments listed in Table 5.2 are accepted. Other options are described in Table 5.57.

Table 5.57. Options for the canvasMCS command.

Option	Description
`-min` *minMatch*	Minimum number of molecules that must match the MCS. If *minMatch* exceeds the number of input molecules it is interpreted as requiring all to match. Default: match all molecules.
`-max` *maxMatch*	Maximum number of molecules that must match the MCS. If *maxMatch* exceeds the number of input molecules it is interpreted as requiring all to match. When *maxMatch* is different from *minMatch*, a series of solutions spanning this range is produced. Default is all.
`-stop` *size*	Stop processing when the MCS atom and bond count falls below this threshold.

Table 5.57. Options for the canvasMCS command. (Continued)

Option	Description
`-nodetail`	Omit output of atom and bond number lists to Maestro and SD files. These are never added to CSV files. Default: include lists.
`-n` *structRange* `[-file]`	Input structures to process. *structRange* is a range specification, as defined on page 119. If `-file` is specified, *structRange* is a file name that contains the range specification. If `-file` is used with `-iproj`, the file must be the binary set file written from the Canvas panel. Default: process all structures.
`-rs` *numMols*	Process only a random subset of the input molecules. If *numMols* is omitted, it is set to the square root of the total.
`-limit` *numMols*	Maximum number of molecules to process. Submitting larger sets may exceed available memory. Default: 2000.
`-timeout` *time*	Stop the job if the total time taken exceeds *time* seconds.
`-showall`	Output all equivalents for each MCS.
`-exclusive`	If an input molecule matches more than one MCS, report only the match to the largest MCS. Cannot be used with `-showall`.
`-sortname`	Sort output on molecule name. Default: keep input order.
`-opw` *filename*	Output MCS for all pairs of structures in a text file, one pair per line in the format: *title1*, *SMILES1*, *title2*, *SMILES2*, *MCSof1to2*, *MCSof2to1*
`-addring`	In output MCS SMARTS patterns, mark each atom as cyclic (R) or acyclic (R0). Default: do not mark atoms.
`-addh`	In output MCS SMARTS patterns, include hydrogen counts for each atom. Default: do not include hydrogen counts.
`-nox`	In output MCS SMARTS patterns, suppress addition of a connectivity qualification [nX3] for pyrolic nitrogens. The qualification is added by default.
`-nobreakring`	Do not consider partial rings as part of MCS.
`-nobreakaring`	Do not consider partial aromatic rings as part of MCS.
`-allH`	Consider hydrogens as explicit atoms. Sets `-addh`.
`-atomtype` *scheme*	Atom typing scheme. Must be an integer value between 1 and 12 or C. The schemes are described in Table 5.7 on page 125. Default: 7.

5.6.8 canvasScaffold

This utility decomposes structures into a set of scaffolds (scaffold detection), and searches a set of structures to determine which structures contain each scaffold in a set of scaffolds (scaffold matching).

Scaffold detection consists of identifying all unique scaffolds within a set of structures. The largest scaffold in a structure is obtained by stripping off all terminal side-chains with the exception of exocyclic and exolinker double bonds such as >C=O. For example, quinone contains two exocyclic double bonds and benzophenone contains a single exolinker double bond. These C=O bonds would be retained as part of the quinone ring system and part of the benzophenone linker, respectively.

A given scaffold is broken into smaller subscaffolds by removing linkers between ring systems, but not by splitting fused ring systems. So, for example, benzophenone would be split into two benzene rings, but naphthalene would not.

Generation of subscaffolds is exhaustive, so if R1, R2, and R3 are ring systems, the scaffold R1–R2—R3 would produce the subscaffolds R1–R2, R2–R3, R1, R2, and R3. Scaffold detection finds all such scaffolds, and stores them in a list that is sorted by decreasing scaffold size. The size of a scaffold is based on the number of ring systems, followed by the number of linkers, followed by the number of heavy atoms, followed by molecular weight, followed by canonical SMILES. Sorting in this manner guarantees that the subscaffolds of a given scaffold will always appear after the scaffold itself.

Scaffold matching consists of taking a set of structures and a set of scaffolds, and determining which scaffolds are contained in each structure. The scaffolds may or may not come from the structures being matched against them. Since matching produces a list of scaffolds for each structure, it is possible to assign the structures into clusters based on the scaffolds they contain.

The command syntax is:

`canvasScaffold` *inFile* {`-detect`|`-match`|`-full`} [*options*]

where *inFile* is the input structure file, in Maestro, SD, CSV (SMILES and properties), or SMILES format. Maestro and SD files can be compressed. CSV files must contain a column header line and SMILES must appear in the first column. SMILES files have no header, and each SMILES string may be followed by a tab and a structure name. The three run modes are

`-detect` Detect scaffolds among the input structures.

`-match` Match the scaffolds in j*obname*_`scaffolds.dat` against the input structures.

`-full` Perform scaffold detection and matching on the input structures.

The options are given in Table 5.58.

Table 5.58. Options for the canvasScaffold command.

Option	Description
`-exocyc` *file*	Extend exocyclic definitions to include the SMARTS patterns in *file*. For example, use `[CH3]a` to retain methyls attached to aromatic rings. Not valid with `-match`.
`-exolink` *file*	Extend exolinker definitions to include the SMARTS patterns in *file*. For example, use `O=C([O-,OH])([CD3;CR0])` to retain carboxylates attached to a tertiary carbon linker atom. Not valid with `-match`.
`-iscaff` *file*	Use scaffolds in *file* rather than *jobname*`_scaffolds.dat`. This file must have been created by a previous `canvasScaffold` job run with `-detect` or `-full`. Valid only with `-match`.
`-embed`	Allow embedded ring system and linker matches when detecting or matching. See text below for a detailed description with an example.
`-title` *prop*	Use an alternate property for structure names. Must be the name of a string or integer property in *inFile*. Not valid with `-detect` or with SMILES (`.smi`) input.
`-props` *list*	Comma-separated list of properties from *inFile* to include in the output file *jobname*`_match.csv`. By default, only the structure name is included. Not valid with `-detect` or with SMILES input.
`-include` *list*	List of scaffold IDs to consider when matching. See Section 5.1 on page 119 for list syntax. These are the only IDs reported in *jobname*`_match.csv`. If *list* is the name of a file, it is assumed that the file contains the desired IDs, one per line. Valid only with `-match`.
`-exclude` *list*	List of scaffold IDs that should be excluded when matching. See Section 5.1 on page 119 for list syntax. Excluded scaffold IDs never appear in *jobname*`_match.csv`. If a structure contains an excluded scaffold, that structure is not reported as a match to any subscaffolds of the excluded scaffold. If *list* is the name of a file, it is assumed that the file contains the list of IDs, one per line. Valid only with `-match`.
`-fuzzy`	Use fuzzy scaffold matching. A given structure is considered to match a scaffold if it contains a substructure with the same generic framework as the scaffold, treating all atoms and bonds as equivalent. Not valid with `-detect`.
`-arom`	When doing fuzzy matching, distinguish aromatic and non-aromatic atoms. Only valid with `-fuzzy`.
`-bonds`	When doing fuzzy matching, distinguish single, double, triple, and aromatic bonds. Only valid with `-fuzzy`.

Results of scaffold detection are written to j*obname*_detect.csv, which contains a canonical SMILES string for each scaffold, a unique scaffold ID, and quoted, space-delimited lists of the source input structure numbers, superscaffold IDs, and subscaffold IDs. For example,

```
SMILES,ID,Source,Super,Sub
n1ccccc1Cc2ccccc2,1,"1","","3 4"
c1ccccc1Cc2ccccc2,2,"2","","4"
c1ccncc1,3,"1","1",""
c1ccccc1,4,"1 2","1 2",""
```

A separate file, *jobname*_scaffolds.dat, is created with all the information required to reconstruct the original scaffolds and the relationships among them.

Results of scaffold matching are written to *jobname*_match.csv. Each structure that matches at least one scaffold is written to *jobname*_match.csv, with canonical SMILES, structure name, an index that corresponds to the position of the structure in the original input file, and the IDs of the scaffolds matched by the structure. For example,

```
SMILES,Name,Index,IDs
O=C(O)c1ccc(cc1)Cc2ccccn2,"CHEM0192587",1,"1 3 4"
O=C(O)c1ccc(cc1)Cc2ccccc2,"CHEM0204591",2,"2 4"
```

Here, CHEM0192587 matches scaffolds 1, 3 and 4, while CHEM0204591 matches scaffolds 2 and 4.

The use of -embed needs some explanation. If a mixture of imidazoles and benzimidazoles is being analyzed and -embed is used with -detect or -full, embedded benzimidazole-imidazole relationships would be reported in *jobname*_detect.csv. When an embedded relationship is reported, the letter e is appended to the applicable source structure numbers and scaffold IDs. For example,

```
SMILES,ID,Source,Super,Sub
c1cccc(c12)nc[nH]2,1,"1","","2e"
c1[nH]cnc1,2,"1e","1e",""
```

Here, the Source value 1e on line 3 indicates that scaffold 2 is an embedded scaffold of input structure 1. The Sub value 2e on line 2 indicates that scaffold 2 is an embedded subscaffold of scaffold 1, and the Super value 1e on line 3 indicates the reciprocal relationship.

If -embed is used with -match or -full, imidazole would be listed in *jobname*_match.csv as an embedded match to any benzimidazole-containing structure. For example,

```
SMILES,Name,Index,IDs
c1cccc(c12)nc([nH]2)C,"2-methyl-benzimidazole",1,"1 2e"
```

Here, `"1  2e"` indicates an ordinary match to benzimidazole (scaffold ID = 1), and an embedded match to imidazole (scaffold ID = 2).

By default, no relationship is reported between imidazole and benzimidazole. Note also that if scaffold detection is performed on structures that contain only benzimidazole ring systems, imidazole would not be found as a scaffold, even with `-embed`.

Scaffold detection scales linearly with the number of structures analyzed, and quadratically with the number of unique scaffolds contained in those structures. The quadratic part comes from the identification of superscaffold-subscaffold relationships. The worst case scenario is when there are a large number of unrelated structures, because many of the structures will have a unique backbone, and you will get a new scaffold for each of those backbones.

5.7 Scripting with Canvas

In addition to the command-line utilities, Canvas has a Python API that can be used to write Python scripts. The API is described in the document *Canvas Python API*. Canvas utilities are also available for use with KNIME, along with many other Schrödinger products and utilities.

References

1. Leach, A. R.; Gillet V. J. *An Introduction to Cheminformatics.* Springer: Dordrecht, 2007.

2. Gobbi, A.; Lee, M.-L. *J. Chem. Inf. Comput. Sci.* **2003,** *43*, 317.

3. Lloyd, S. P. *IEEE Transactions on Information Theory* **1982**, *28*, 129.

4. Kelley, L. A.; Gardner, S. P.; Sutcliffe, M. J. An automated approach for clustering an ensemble of NMR-derived protein structures into conformationally-related subfamilies. *Protein Eng.* **1996**, *9*, 1063.

5. Sastry, M.; Lowrie, J. F.; Dixon, S. L.; Sherman, W. Large-Scale Systematic Analysis of 2D Fingerprint Methods and Parameters to Improve Virtual Screening Enrichments. *J. Chem. Inf. Model.* **2010**, *50*, 771.

6. Duan, J.; Dixon, S.L.; Lowrie, J.F; Sherman, W. Analysis and comparison of 2D fingerprints: Insights into database screening performance using eight fingerprint methods. *J. Molec. Graph. Model.* **2010**, *29*, 157.

7. Rosipal R.; Trejo L.J. Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. *J. Machine Learning Res.* **2001**, *2*, 97.

8. Kier, L. B.; Hall, L. H. *Molecular Structure Description: The Electrotopological State*; Academic Press: San Diego, 1999.

9. Ghose, A. K.; Viswanadhan, V. N.; Wendoloski, J. J. Prediction of Hydrophobic (Lipophilic) Properties of Small Organic Molecules Using Fragmental Methods: An Analysis of ALOGP and CLOGP Methods. *J. Phys. Chem A* **1998**, *102*, 3762.

10. Viswanadhan, V. N.; Ghose, A. K.; Revankar, G. R.; Robins, R. K. Atomic Physicochemical Parameters for Three Dimensional Structure Directed Quantitative Structure-Activity Relationships. 4. Additional Parameters for Hydrophobic and Dispersive Interactions and Their Application for an Automated Superposition of Certain Naturally Occurring Nucleoside Antibiotics. *J. Chem. Inf. Comput. Sci.*, **1989**, *29*, 163.

11. Miller K. J. *J. Am. Chem. Soc.* **1990**, *112*, 8533.

12. Hall, L. H.; Kellogg, G. E.; Haney D. N. Molconn-Z Software Package for Molecular Topology Analysis. User's Guide. Version 4.12, 2008. http://www.edusoft-lc.com/molconn/manuals/400/

13. Ertl, P.; Rohde, B.; Selzer, P. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties. *J. Med. Chem*. **2000**, *43*, 3714.

14. Klon, A. E.; Lowrie, J. F.; Diller, D. J. Improved Naive Bayesian Modeling of Numerical Data for Absorption, Distribution, Metabolism and Excretion (ADME) Property Prediction. *J. Chem. Inf. Model.* **2006**, *46*, 1945.

15. Baell, J. R.; Holloway, G. A. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.* **2010**, *53*, 2719.

Getting Help

Schrödinger software is distributed with documentation in PDF format. The manuals are also available from the Schrödinger Support Center. If the documentation is not installed in `$SCHRODINGER/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*.

Canvas has an online help system that contains information about the Canvas interface.

- For information about a panel, click the Help button in the panel. The help topic is displayed in your browser.
- For other information in the online help, open the default help topic by choosing Help → Online Help in the main window. This topic is displayed in your browser. You can navigate to topics in the navigation bar.
- To search the help, click the Search tab in the navigation bar. You can then search on various combinations of words. The results are displayed in the lower pane of the search tab, and you can click links in this pane to display the help topics with the search terms highlighted.

The browser can be changed on Linux by setting the environment variable `SCHRODINGER_HELP_BROWSER` to the full path to the browser. On Windows, you can change the default browser from inside the browser.

The Help menu provides access to the knowledge base on the external web site, at http://www.schrodinger.com/kb, which you can search for information about Canvas. It also provides access to information about known issues with Canvas, on the Support Center. To diagnose problems with licensing or remote hosts, you can open the Diagnostics panel from the Help menu.

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail: help@schrodinger.com
USPS: Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone: (503) 299-1150
Fax: (503) 299-4532

WWW: http://www.schrodinger.com
FTP: ftp://ftp.schrodinger.com

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

- All relevant user input and machine output
- Canvas purchaser (company, research institution, or individual)
- Primary Canvas user
- Installation, licensing, and machine information as described below.

To gather the required machine, licensing, and installation information to send to technical support:

1. Open the Diagnostics panel.
   - **Canvas:** Help → Diagnostics
   - **Windows:** Start → All Programs → Schrodinger-2013 → Diagnostics
   - **Mac:** Applications → Schrodinger2013 → Diagnostics
   - **Linux/Command line:** `$SCHRODINGER/diagnostics`
2. When the diagnostics have run, click Technical Support.

   A dialog box opens, with instructions. You can highlight and copy the name of the file.
3. Attach the file specified in the dialog box to your e-mail message.

Index

Numerics

2D structures
- converting to 3D 110
- editing 25
- importing by name 15

3D Minimization dialog box 111
3D Pharmacophore Fingerprints - Advanced Options dialog box 72
3D Pharmacophore Fingerprints dialog box 71

A

Apply Logic to View dialog box 33
Apply to Master dialog box 33
archiving projects 11
atom typing schemes 125
Axis settings dialog boxes 52

B

Bars dialog box 53
Bayes Classification - Bins dialog box 97
Bayes Classification dialog box 96
Binary Fingerprints from Structures dialog box 68
bins
- assigning for Bayes classification 97, 100, 166
- inter-feature distance 71
- overlap 123

bit set, exporting and importing 17
bits
- fraction kept in Bayes classification 166
- limiting for self-organizing maps 170
- number in fingerprint 70, 124
- omitting from fingerprint 70, 73, 124
- selecting for fingerprint export 17
- sorting by count 86, 151

bivariate statistics 55

C

Calculator dialog box 57
Canvas main window 6
cells
- clearing 28
- editing 28
- selecting 22

Chemistry Filter dialog box 44
Choose Random Subset dialog box 30
class
- assigning values 37
- changing color 37
- creating 38
- editing values 38
- moving rows to 38
- removing rows from 39

clustering
- by properties 82
- linkage method 83, 146
- methods 82

clusters
- dendrogram 83
- exporting 84
- leader 87
- membership property 87
- viewing 87

color
- chart title, background, foreground 52
- distance matrix 74
- for class 35, 37, 38, 39
- histogram axis 53
- job record 62
- pie charts 54
- property values in spreadsheet 40
- scatter plot axes 52
- scatter plot from model 89
- scatter plot symbols 48
- self-organizing map 102
- similarity matrix 74
- SOM cell borders 103
- statistics 56

columns
- finding 24
- hiding 29
- moving 19
- resizing 19
- sorting 19

conformers for 3D pharmacophore fingerprints 71, 72
conventions, document vii
coordinates, discarding on import 13
custom view 30
- *See also* views

D

data
- clearing from cells ... 28
- exporting ... 16
- importing ... 14

decision tree ... 99
dendrogram, viewing ... 83
directed sphere exclusion (DISE) ... 79
directory
- installation ... 1
- Maestro working ... 2
- project ... 10

distance matrix
- from fingerprints ... 74
- from properties ... 74
- metric ... 74, 142

Distance Matrix from Properties dialog box ... 74, 75
Diversity-Based Selection dialog box ... 78
duplicate structures
- detecting in project ... 47
- removing from project ... 116
- removing on import ... 12, 13

E

Edit Partition dialog box ... 39
Edit Structure dialog box ... 25
electrotopological states—*see* Estates
environment variable
- `PATH` ... 58
- `SCHRODINGER` ... 1
- `SCHRODINGER_CANVAS_MAX_MEM` ... 59, 119

Estates
- atom typing ... 124
- calculating ... 64, 156

Export Options dialog box ... 16
External Application dialog box ... 112

F

Feature Selection dialog box ... 66
Feature Selection Viewer panel ... 67
file formats
- for import ... 11

Find panel ... 23
fingerprints
- 3D from pharmacophores ... 70
- calculating ... 68, 122
- exporting ... 17
- importing Canvas ... 11, 13
- omitting bits ... 70, 124
- precision ... 70, 71, 123
- reducing number of bits ... 124
- sorting by bit count ... 86, 151
- types ... 123

Frequency Pie Chart panel ... 51

H

Hashed Fingerprints - Advanced Options dialog box ... 69
heat map
- distance matrix ... 74
- similarity matrix ... 74
- spreadsheet ... 40
- statistics ... 56

Heat Map dialog box ... 40
hiding rows and columns ... 29
Hierarchical Clustering Dendrogram panel ... 85
Hierarchical Clustering dialog box ... 83
Histogram panel ... 50
histograms ... 50
- axis and bar settings ... 52

Hole-Filling and Library Optimization - Property Filter dialog box ... 80
Hole-Filling and Library Optimization dialog box ... 80
hydrogens, showing and hiding ... 20

I

Import SMILES and Properties dialog box ... 14
Import Structure File dialog box ... 12

K

Kernel-Based Partial Least-Squares Regression dialog box ... 93
K-Means Clustering dialog box ... 86
KPLS Model Visualization panel ... 94
KPLS Regression dialog box ... 93

L

Leader-Follower Clustering dialog box 86
Library Comparison dialog box 81
LigPrep..... 111
linkage method..... 83, 146
loadings, plotting 105

M

Maestro
  exporting structures to 17
  exporting structures to Canvas..... 11
master view 30
Maximum Common Substructure panel 106
MCS groups 105
MCS Matches view..... 107
memory, for applications 59, 61, 119
metric
  description 76
  for diverse compound selection 136
  for property distances 74
  for similarity/distance matrices 142
model
  building..... 88
  exporting..... 89
Molecular Properties - Advanced Options dialog box 64
Molecular Properties dialog box..... 63
multiple linear regression
  choosing subset of variables 89, 155
Multiple Linear Regression - Best Subsets Options dialog box 90
Multiple Linear Regression dialog box..... 90

N

Neural Networks dialog box 98
New Class dialog box 39
New Partition from Property dialog box..... 36
New Partition from Random Split dialog box .. 38
New Partition from Selection dialog box..... 35

P

partial least-squares regression
  number of factors..... 91, 92, 163, 164
  t-value 91, 163
Partial Least-Squares Regression dialog box.... 91
partition
  adding and rearranging classes..... 38
  assigning training and test set from..... 88
  creating from property..... 36
  creating from selection 36
  creating random..... 37
Partition Filter dialog box 43
pharmacophore feature definitions 71
pie charts..... 51
  settings..... 54
Plot Symbols dialog box..... 49
Preferences dialog box..... 59
Principal Components Analysis dialog box.... 104
Principal Components Regression dialog box .. 95
product installation 191
projects
  archiving 11
  creating 10, 176
  importing from 11
  incorporating results into..... 62
  information available in..... 18
  opening on startup 5
  recently used..... 10
  updating 176
properties
  adding to application results view 32
  calculating 63, 156
  cluster membership 87
  clustering by 82
  coloring plot symbols by 48
  displaying heat maps 40
  distance matrix 74
  diverse structure set 79
  exporting to Maestro 17
  filtering by 41
  for heat map..... 40
  from command-line jobs 32
  import order..... 11
  importing 14
  missing values 55
  selecting for export..... 16
  similarity, naming..... 75
  sizing plot symbols by..... 48
  sorting by value of..... 34
  statistics for 55
  updating on import 13
Property Filter dialog box 41

Python Shell panel 114

R

Recursive Partitioning - Advanced Options dialog box 100
Recursive Partitioning dialog box 99
REOS rules 180
rotatable bonds
counts of 64
default Canvas definition 158
sampling for conformer generation 72
rows
exporting random subset..... 16
hiding..... 29
random subset..... 30
resizing 19
selecting by row number 22
sorting by property values 34

S

Save Custom View dialog box 31
Scaffold Decomposition dialog box..... 108
Scaffold Decomposition for jobname panel ... 109
Scatter Plot panel 48
scatter plots 48
axis settings 52
from model predictions 89
of bivariate statistics variables..... 56
Schrödinger contact information 191
scores, principal component 105
Sectors dialog box..... 54
selection
applying from custom to master view 32
by matched text..... 24
clearing 22
expanding to entire columns..... 22
expanding to entire rows..... 22
from histograms..... 50
from pie charts..... 51
from scatter plots 49
inverting..... 22
shortcut menu 20
with arrow/tab keys 20
with mouse 22
Self-Organizing Map dialog box 101
Self-Organizing Map Viewer panel 103
self-organizing maps
cell border color 103
lattice parameters..... 102, 171
limiting number of bits..... 170
Shape Screen - Advanced Options dialog box.. 77
Shape Screen dialog box..... 77
similarity matrix
from fingerprints..... 74
metric..... 142
Similarity/Distance Matrix from Fingerprints dialog box 73
Similarity/Distance Screen dialog box 76
Sort dialog box..... 34
sort keys, reordering 35
Source Structures panel 110
sphere exclusion..... 79
spreadsheet, copying to..... 17
Start Application dialog box..... 62
Statistics dialog box..... 55, 56
statistics, univariate and bivariate 55
structural keys 70
structures
coloring..... 24
copying 24
diverse, selecting 78
drawing of..... 20
duplicates on import 12, 13
editing..... 25
excluding from diverse set..... 79
exporting to Maestro 17
importing a random subset 12
MCS membership..... 106
pasting 24
renaming..... 25
shortcut menu 21
showing and hiding 20
similarity to reference..... 75
size..... 20
using for chemistry filters..... 44
Substructure Query dialog box 46

T

training set
manual selection..... 89, 153
predefined..... 117
random..... 88, 153

U

undoing view changes 32
univariate statistics 55

V

views
- application results 32
- applying custom to master 32
- closing 32
- clustering results 87
- combining 33
- custom, definition 30
- deleting 32
- editing description 31
- limitations 31
- master, definition 30
- opening 31
- renaming 31
- saving 32
- sorting list of 32
- undoing changes 32

120 West 45th Street
17th Floor
New York, NY 10036

101 SW Main Street
Suite 1300
Portland, OR 97204

245 First Street
Riverview II, 18th Floor
Cambridge, MA 02142

8910 University Center Lane
Suite 270
San Diego, CA 92122

155 Gibbs St
Suite 430
Rockville, MD 20850-0353

Dynamostraße 13
D-68165 Mannheim
Germany

Zeppelinstraße 73
D-81669 München
Germany

Potsdamer Platz 11
D-10785 Berlin
Germany

Quatro House
Frimley Road
Camberley GU16 7ER
United Kingdom

8F Pacific Century Place
1-11-1 Marunouchi
Chiyoda-ku, Tokyo 100-6208
Japan

No. 102, 4th Block
3rd Main Road, 3rd Stage
Sharada Colony
Basaveshwaranagar
Bangalore 560079, India

SCHRÖDINGER®