LigPrep 2.0

User Manual

Schrödinger Press

Copyright © 2006 Schrödinger, LLC. All rights reserved. CombiGlide, Epik, Glide, Impact, Jaguar, Liaison, LigPrep, Maestro, Phase, Prime, QikProp, QikFit, QikSim, QSite, SiteMap, and Strike are trademarks of Schrödinger, LLC.
Schrödinger and MacroModel are registered trademarks of Schrödinger, LLC.

Python is a copyrighted work of the Python Software Foundation. All rights reserved.

To the maximum extent permitted by applicable law, this publication is provided "as is" without warranty of any kind. This publication may contain trademarks of other companies.

Please note that any third party programs ("Third Party Programs") or third party Web sites ("Linked Sites") referred to in this document may be subject to third party license agreements and fees. Schrödinger, LLC and its affiliates have no responsibility or liability, directly or indirectly, for the Third Party Programs or for the Linked Sites or for any damage or loss alleged to be caused by or in connection with use of or reliance thereon. Any warranties that we make regarding our own products and services do not apply to the Third Party Programs or Linked Sites, or to the interaction between, or interoperability of, our products and services and the Third Party Programs. Referrals and links to Third Party Programs and Linked Sites do not constitute an endorsement of such Third Party Programs or Linked Sites.

Revision A, April 2006

Contents

Document Conventions ix

Chapter 1: Introduction 1

1.1 The LigPrep Manual 1

1.2 The LigPrep Process 2

1.3 LigPrep Installation and Licensing 4

1.4 Other Utilities 4

1.5 Citing LigPrep in Publications 4

Chapter 2: Introduction to Maestro 5

2.1 General Interface Behavior 5

2.2 Starting Maestro 5

2.3 The Maestro Main Window 6

2.3.1 The Menu Bar 8

2.3.2 The Toolbar 9

2.3.3 Mouse Functions in the Workspace 12

2.3.4 Shortcut Key Combinations 13

2.4 Maestro Projects 13

2.4.1 The Project Table Toolbar 15

2.4.2 The Project Table Menus 16

2.4.3 Selecting Entries 17

2.4.4 Including Entries in the Workspace 17

2.4.5 Mouse Functions in the Project Table 18

2.4.6 Project Table Shortcut Keys 19

2.5 Building a Structure 20

2.5.1 Placing and Connecting Fragments 20

2.5.2 Adjusting Properties 22

2.5.3 The Build Panel Toolbar 22

2.6 Selecting Atoms 23
2.6.1 Toolbar Buttons 23
2.6.2 Picking Tools 24
2.6.3 The Atom Selection Dialog Box 25
2.7 Scripting in Maestro 25
2.7.1 Python Scripts 25
2.7.2 Command Scripts 26
2.7.3 Macros 27
2.8 Specifying a Maestro Working Directory 27
2.9 Undoing an Operation 28
2.10 Running and Monitoring Jobs 28
2.11 Getting Help 30
2.12 Ending a Maestro Session 30
Chapter 3: Running LigPrep **31**
3.1 Running LigPrep From Maestro 31
3.1.1 Job Input 32
3.1.2 Options for Structural Variation 32
3.1.3 Job Submission 34
3.2 The ligprep Command 35
3.3 Distributing LigPrep Jobs with para_ligprep 40
3.3.1 Load Balancing 41
3.3.2 Running para_ligprep 41
3.4 LigPrep Files 43
3.5 LigPrep Limitations 45
Chapter 4: Using LigPrep **47**
4.1 LigPrep Panel Examples 47
4.1.1 Preparing to Run the Examples 47
4.1.2 Generating Variations on 2D Structures 48

4.1.3 Generating Ring Conformations of a 3D Structure 51

4.2 Command-Line Examples 53

4.2.1 Default Operation 53

4.2.2 Producing One Output Structure for Each Input Structure 54

4.2.2.1 Adjusting Only the Geometry 54

4.2.2.2 Adjusting the Chemistry 55

4.2.3 Generating Variations 56

4.2.3.1 Generating a Small Set of Low-risk Variations 56

4.2.3.2 Generating Many Variations 56

4.2.3.3 Using Meta-Options for Convenient Selection of Run Characteristics 57

4.2.4 Preparing Ligands for QikProp 58

4.2.5 Filtering Structures 58

4.2.6 Running LigPrep on a Remote Host 59

4.3 Chirality Manipulations Using Maestro and LigPrep 59

Chapter 5: Tools Used by LigPrep **63**

5.1 sdconvert 63

5.2 smiles_to_mae 64

5.3 maesubset 65

5.4 applyhtreat 65

5.5 desalter 66

5.6 neutralizer 67

5.7 ionizer 68

5.8 tautomerizer 69

5.9 ligparse 69

5.10 stereoizer 70

5.10.1 Stereochemical Properties 71

5.10.2 Known Limitations 72

5.11 ring_conf 73

5.12 premin 75

5.13 MacroModel (bmin) 76
5.13.1 MacroModel Opcodes Enabled for LigPrep 77
5.13.2 The LigPrep Command File for MacroModel 79
5.13.3 MacroModel Limitations and Workarounds 80

Chapter 6: The ionizer Utility **81**

6.1 Running the ionizer 81
6.2 Input Structure Requirements 84
6.3 Restricting Ionization State Generation 84
6.4 Structural Output 85
6.5 Log File Output 85
6.6 Maestro Properties 87
6.7 Creating a Customized Patterns File 89
6.8 Ionizer Limitations 89

Chapter 7: The tautomerizer Utility **91**

7.1 Running the tautomerizer 92
7.2 The Tautomer Database 93
7.2.1 name Data Item 93
7.2.2 clear_standard Directive 93
7.2.3 group_def Data Structure 93
7.2.4 tautomer_set Data Structure 94
7.3 Tautomer Processing 97
7.3.1 Tautomer Pattern Matching 97
7.3.2 Structure Generation 98

Chapter 8: The ligparse Utility **101**

8.1 Descriptors and Composite Descriptors 101
8.2 Molecular Database Structural Analysis 102
8.3 Molecular Database Subset Selection: “Culling” 103

8.4 The ligparse Command 104

8.5 Input and Output Files 107

Chapter 9: Getting Help 109

Index 111

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.

Table 2.1.

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, and screen output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

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.

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

LigPrep is a robust collection of tools designed to prepare high quality, all-atom 3D structures for large numbers of drug-like molecules, starting with 2D or 3D structures in SD or Maestro format. The resulting structures can be saved in either SD or Maestro format. The simplest use of LigPrep produces a single, low-energy, 3D structure with correct chiralities for each successfully processed input structure. LigPrep can also produce a number of structures from each input structure with various ionization states, tautomers, stereochemistries, and ring conformations, and eliminate molecules using various criteria including molecular weight or specified numbers and types of functional groups present.

LigPrep can be run from Maestro or from the command line. The `ligprep` script provides an efficient way to use a set of tools for ligand preparation collectively and consistently. Some of the tools may also be used separately. The ligand preparation process typically takes about one second per output structure on a 2 GHz x86-compatible computer. Depending on the nature of the input structures and on the options selected, multiple output structures may be produced from each input structure. LigPrep calculations can be distributed across multiple processors to increase throughput using the `para_ligprep` script.

1.1 The LigPrep Manual

This manual explains how to use LigPrep™, a program suite designed to prepare high quality, all-atom 3D structures for large numbers of drug-like molecules, starting with 2D or 3D structures. The manual is organized as follows:

- Chapter 1 provides an overview of LigPrep and the process it uses to prepare ligands.
- Chapter 2 gives a general introduction to Maestro™, the graphical user interface for all Schrödinger's products.
- Chapter 3 explains how to run LigPrep from Maestro and from the command line.
- Chapter 4 provides instruction on the use of LigPrep in several scenarios.
- Chapter 5 describes the tools used by LigPrep.
- Chapter 6 describes the `ionizer` and its options, input, and output.
- Chapter 7 describes the `tautomerizer` and how it generates tautomers in detail.
- Chapter 8 describes using the `ligparse` utility to examine and select structures in a file.

1.2 The LigPrep Process

The LigPrep process consists of a series of steps that perform conversions, apply corrections to the structures, generate variations on the structures, eliminate unwanted structures, and optimize the structures. Many of the steps are optional and are controlled by selecting options in the LigPrep panel or by specifying command-line options. The steps are outlined below. Each step is performed by the script or program mentioned in the step.

While the generation of ionization states and tautomers (Step 6 and Step 7) can be carried out with tools that are part of LigPrep, a separate product, Epik, can be used instead to perform these structural adjustments during a LigPrep run. Epik more rigorously adjusts the tautomerization and ionization states than separate ionizer and tautomerizer treatments. However, it is slower, taking on average around 6 seconds per structure on a 2 GHz Pentium IV processor.

1. Convert the structure format.

   If the input structure file is in SD format, it is converted to Maestro format by `sdconvert`. Parities specified in the SD file are converted into chiralities, which are stored as properties in the Maestro file. If the input structure file is in SMILES format, it is converted to Maestro format by `smiles_to_mae`. The input file should have one SMILES pattern per line. The SMILES pattern can be followed by an optional title string.

2. Select the structures.

   A subset of the input structures can be selected for processing. The selection is done by `maesubset` for Maestro input files and by `sdconvert` for SD input files.

3. Add hydrogen atoms.

   Structures that have implicit hydrogen atoms may need to have hydrogen atoms added before the 3D structures can be minimized. Hydrogen atoms are added in a manner that is consistent with a particular force field. This step is performed by `applyhtreat`, the program used by the Hydrogen Treatment panel in Maestro.

4. Remove unwanted molecules.[1]

   If structures have additional molecules included, such as counter ions in salts and water molecules, these may need to be removed. The `desalter` removes all but the molecule containing the most atoms from each structure.

5. Neutralize charged groups.[1]

   Charged groups must be neutralized before ionization states can be generated. The neutralization is performed by the `neutralizer`, which adds or removes hydrogen ions.

1. Executed by default, but optional in the LigPrep process.

6. Generate ionization states.[2]

   For some applications, it is important that all species that exist in a given pH range are available. In this step, the `ionizer` generates various ionization states for each structure. This step should be preceded by a neutralization step.

7. Generate tautomers.[1]

   As with ionization, the tautomeric states with a significant population may be important for some types of calculations. The `tautomerizer` generates various tautomers for each structure.

8. Filter the structures.[1]

   In this step, structures that match specified conditions can be removed. The condition can be on a property, such as Molecular weight > 1000, or on the structure, such as the presence or absence of a specific functional group. This step is performed by `ligparse`.

9. Generate alternative chiralities.[1]

   Since 2D structures do not always have complete chirality information, it can be useful to vary the chiralities of the atoms to find all the low-energy structures or to provide a range of possible structures for investigation. This step identifies additional chiral atoms in the structures and generates additional structures with the same molecular formula but different chiral properties. The step is performed by `stereoizer`.

10. Generate low-energy ring conformations.[1]

    When ring conformation information is not available, it can be important to specifically generate low-energy ring conformations. Ring confirmations are generated for each structure by `ring_conf`.

11. Remove problematic structures.

    Structures that could cause subsequent processing failures either in the energy minimization of the structures or in other applications are removed by `premin`.

12. Optimize the geometries.

    The geometries of the generated structures are optimized using a restricted version of the MacroModel™ computational program, `bmin`, or a short conformational search is performed to relax the structure into 3 dimensions while strongly encouraging chiral centers to adopt the proper chiralities (if the structure is highly strained).

13. Convert the output file.

    If output in SD format was requested, `sdconvert` is run to perform the conversion.

2. Not executed by default, but can be enabled in the LigPrep process by setting the appropriate options.

1.3 LigPrep Installation and Licensing

LigPrep is licensed separately from other Schrödinger products. The MacroModel commands `premin` and `bmin` require LigPrep licenses when run in a LigPrep context, and are limited to a restricted set of commands when run using a LigPrep license. For details on the available commands, see Section 5.13 on page 76.

Instructions for installing LigPrep and obtaining a license are given in the *Installation Guide*. After you install LigPrep, set the `SCHRODINGER` environment variable to the installation directory. See Section 2.2 on page 5.

1.4 Other Utilities

A number of utilities that are provided with LigPrep, but are not part of LigPrep itself, might be useful in conjunction with LigPrep. These utilities are available in `$SCHRODINGER/utilities`, and include:

- `proplister`—extracts requested properties from Maestro-formatted files
- `propfilter`—selects structures from Maestro-formatted files using the properties stored in the files
- `maemmod` and `mmodmae`—converts files between Maestro and MacroModel formats
- `mmodmol` and `molmmod`—converts files between MacroModel and Sybyl Mol2 formats
- `pdbconvert`—converts files between Maestro or MacroModel and PDB formats
- `maesubset`—selects a subset of the structures in a Maestro-formatted file
- `sdsubset`—selects a subset of the structures in an SD-formatted file

More information on these utilities is available in the Appendix D of the *Maestro User Manual*.

1.5 Citing LigPrep in Publications

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

LigPrep, version 2.0, Schrödinger, LLC, New York, NY, 2005.

Introduction to Maestro

Maestro is the graphical user interface for all of Schrödinger's products: CombiGlide™, Epik™, Glide™, Impact™, Jaguar™, Liaison™, LigPrep™, MacroModel®, Phase™, Prime™, QikProp™, QSite™, SiteMap™, and Strike™. It contains tools for building, displaying, and manipulating chemical structures; for organizing, loading, and storing these structures and associated data; and for setting up, monitoring, and visualizing the results of calculations on these structures. This chapter provides a brief introduction to Maestro and some of its capabilities. For more information on any of the topics in this chapter, see the *Maestro User Manual*.

2.1 General Interface Behavior

Most Maestro panels are amodal: more than one panel can be open at a time, and a panel need not be closed for an action to be carried out. Each Maestro panel has a Close button so you can hide the panel from view.

Maestro supports the mouse functions common to many graphical user interfaces. The left button is used for choosing menu items, clicking buttons, and selecting objects by clicking or dragging. This button is also used for resizing and moving panels. The right button displays a shortcut menu. Other common mouse functions are supported, such as using the mouse in combination with the SHIFT or CTRL keys to select a range of items and select or deselect a single item without affecting other items.

In addition, the mouse buttons are used for special functions described later in this chapter. These functions assume that you have a three-button mouse. If you have a two-button mouse, ensure that it is configured for three-button mouse simulation (the middle mouse button is simulated by pressing or holding down both buttons simultaneously).

2.2 Starting Maestro

Before starting Maestro, you must first set the `SCHRODINGER` environment variable to point to the installation directory. To set this variable, enter the following command at a shell prompt:

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

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

You might also need to set the `DISPLAY` environment variable, if it is not set automatically when you log in. To determine if you need to set this variable, enter the command:

```
echo $DISPLAY
```

If the response is a blank line, set the variable by entering the following command:

csh/tcsh: `setenv DISPLAY` *display-machine-name*`:0.0`

bash/ksh: `export DISPLAY=`*display-machine-name*`:0.0`

After you set the `SCHRODINGER` and `DISPLAY` environment variables, you can start Maestro using the command:

`$SCHRODINGER/maestro` *options*

If you add the `$SCHRODINGER` directory to your path, you only need to enter the command `maestro`. Options for this command are given in Section 2.1 of the *Maestro User Manual*.

The directory from which you started Maestro is Maestro's current working directory, and all data files are written to and read from this directory unless otherwise specified (see Section 2.8 on page 27). You can change directories by entering the following command in the command input area (see page 8) of the main window:

`cd` *directory-name*

where *directory-name* is either a full path or a relative path.

2.3 The Maestro Main Window

The Maestro main window is shown in Figure 2.1 on page 7. The main window components are listed below.

The following components are always visible:

- **Title bar**—displays the Maestro version, the project name (if there is one) and the current working directory.
- **Auto-Help**—automatically displays context-sensitive help.
- **Menu bar**—provides access to panels.
- **Workspace**—displays molecular structures and other 3D graphical objects.

The following components can be displayed or hidden by choosing the component from the Display menu. Your choice of which main window components are displayed is persistent between Maestro sessions.

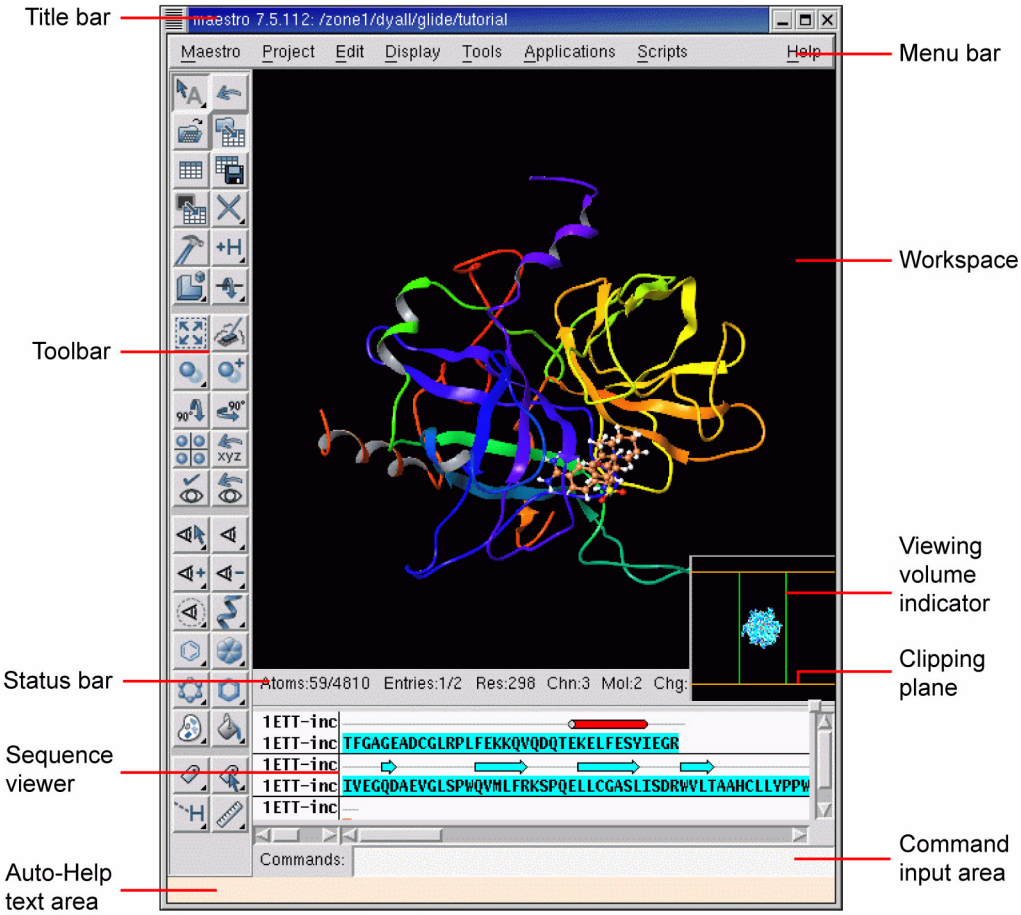

Figure 2.1. The Maestro main window.

- **Toolbar**—contains buttons for many common tasks and provides tools for displaying and manipulating structures, as well as organizing the Workspace.
- **Status bar**—displays information about a particular atom, or about structures in the Workspace, depending on where the pointer pauses (see Section 2.5 of the *Maestro User Manual* for details):
  - **Atom**—displays the chain, residue number, element, PDB atom name, formal charge, and title or entry name (this last field is set by choosing Preferences from the Maestro menu and selecting the Feedback folder).
  - **Workspace**—displays the number of atoms, entries, residues, chains, and molecules in the Workspace.

- **Clipping planes window**—displays a small, top view of the Workspace and shows the clipping planes and viewing volume indicators.
- **Sequence viewer**—shows the sequences for proteins displayed in the Workspace. See Section 2.6 of the *Maestro User Manual* for details.
- **Command input area**—provides a place to enter Maestro commands.

When a distinction between components in the main window and those in other panels is needed, the term *main* is applied to the main window components (e.g., main toolbar).

You can expand the Workspace to occupy the full screen, by pressing CTRL+=. All other components and panels are hidden. To return to the previous display, press CTRL+= again.

2.3.1 The Menu Bar

The menus on the main menu bar provide access to panels, allow you to execute commands, and control the appearance of the Workspace. The main menus are as follows:

- Maestro—save or print images in the Workspace, execute system commands, save or load a panel layout, set preferences, set up Maestro command aliases, and quit Maestro.
- Project—open and close projects, import and export structures, make a snapshot, and annotate a project. These actions can also be performed from the Project Table panel. For more information, see Section 2.4 on page 13.
- Edit—undo actions, build and modify structures, define command scripts and macros, and find atoms in the Workspace.
- Display—control the display of the contents of the Workspace, arrange panels, and display or hide main window components.
- Tools—group atoms; measure, align, and superimpose structures; and view and visualize data.
- Applications—set up, submit, and monitor jobs for Schrödinger's computational programs. Some products have a submenu from which you can choose the task to be performed.
- Scripts—manage and install Python scripts that come with the distribution and scripts that you create yourself. (See Chapter 13 of the *Maestro User Manual* for details.)
- Help—open the Help panel, the PDF documentation index, or information panels; run a demonstration; and display or hide Balloon Help (tooltips).

2.3.2 The Toolbar

The main toolbar contains three kinds of buttons for performing common tasks:

Action—Perform a simple task, like clearing the Workspace.

Display—Open or close a panel or open a dialog box, such as the Project Table panel.

Menu—Display a *button menu*. These buttons have a triangle in the lower right corner.

There are four types of items on button menus, and all four types can be on the same menu (see Figure 2.2):

- **Action**—Perform an action immediately.
- **Display**—Open a panel or dialog box.
- **Object types for selection**—Choose Atoms, Bonds, Residues, Chains, Molecules, or Entries, then click on an atom in the Workspace to perform the action on all the atoms in that structural unit.

  The object type is marked on the menu with a red diamond and the button is indented to indicate the action to be performed.
- **Other setting**—Set a state, choose an attribute, or choose a parameter and click on atoms in the Workspace to display or change that parameter.

The toolbar buttons are described below. Some descriptions refer to features not described in this chapter. See the *Maestro User Manual* for a fuller description of these features.

Figure 2.2. The Workspace selection ***button menu and the*** Adjust distances, angles or dihedrals ***button menu.***

Workspace selection
– Choose an object type for selecting
– Open the Atom Selection dialog box

Undo/Redo
Undo or redo the last action. Performs the same function as the Undo item on the Edit menu, and changes to an arrow pointing in the opposite direction when an Undo has been performed, indicating that its next action is Redo.

Open a project
Open the Open Project dialog box.

Import structures
Open the Import panel.

Open/Close Project Table
Open the Project Table panel or close it if it is open.

Save as
Open the Save Project As dialog box, to save the project with a new name.

Create entry from Workspace
Open a dialog box in which you can create an entry in the current project using the contents of the Workspace.

Delete
– Choose an object type for deletion
– Delete hydrogens and waters
– Open the Atom Selection dialog box
– Delete other items associated with the structures in the Workspace
– Click to select atoms to delete
– Double-click to delete all atoms

Open/Close Build panel
Open the Build panel or close it if it is open.

Add hydrogens
– Choose an object type for applying a hydrogen treatment
– Open the Atom Selection dialog box
– Click to select atoms to treat
– Double-click to apply to all atoms

Local transformation
– Choose an object type for transforming
– Click to select atoms to transform
– Open the Advanced Transformations panel

Adjust distances, angles or dihedrals
– Choose a parameter for adjusting
– Delete adjustments

Fit to screen
Scale the displayed structure to fit into the Workspace and reset the center of rotation.

Clear Workspace
Clear all atoms from the Workspace.

Set fog display state
Choose a fog state. Automatic means fog is on when there are more than 40 atoms in the Workspace, otherwise it is off.

Enhance depth cues
Optimize fogging and other depth cues based on what is in the Workspace.

Rotate around X axis by 90 degrees
Rotate the Workspace contents around the X axis by 90 degrees.

Rotate around Y axis by 90 degrees
Rotate the Workspace contents around the Y axis by 90 degrees.

Tile entries
Arrange entries in a rectangular grid in the Workspace.

Reset Workspace
Reset the rotation, translation, and zoom of the Workspace to the default state.

Save view
Save the current view of the Workspace: orientation, location, and zoom.

Restore view
Restore the last saved view of the Workspace: orientation, location, and zoom.

Display only selected atoms
– Choose an object type for displaying
– Click to select atoms to display
– Double-click to display all atoms

Display only
– Choose a predefined atom category
– Open the Atom Selection dialog box

Also display
– Choose a predefined atom category
– Open the Atom Selection dialog box

Undisplay
– Choose a predefined atom category
– Open the Atom Selection dialog box

Display residues within N angstroms of currently displayed atoms
– Choose a radius
– Open a dialog box to set a value

Show, hide, or color ribbons
– Choose to show or hide ribbons
– Choose a color scheme for coloring ribbons

Draw bonds in wire
– Choose an object type for drawing bonds in wire representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms

Draw atoms in CPK
– Choose an object type for drawing bonds in CPK representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms

Draw atoms in Ball & Stick
– Choose an object type for drawing bonds in Ball & Stick representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms

Draw bonds in tube
– Choose an object type for drawing bonds in tube representation
– Open the Atom Selection dialog box
– Click to select atoms for representation
– Double-click to apply to all atoms

Color all atoms by scheme
Choose a predefined color scheme.

Color residue by constant color
– Choose a color for applying to residues
– Click to select residues to color
– Double-click to color all atoms

Label atoms
– Choose a predefined label type
– Delete labels

Label picked atoms
– Choose an object type for labeling atoms
– Open the Atom Selection dialog box
– Open the Atom Labels panel at the Composition folder
– Delete labels
– Click to select atoms to label
– Double-click to label all atoms

Display H-bonds
– Choose bond type:
intra—displays H-bonds within the selected molecule
inter—displays H-bonds between the selected molecule and all other atoms.
– Delete H-bonds
– Click to select molecule

Measure distances, angles or dihedrals
– Choose a parameter for displaying measurements
– Delete measurements
– Click to select atoms for measurement

2.3.3 Mouse Functions in the Workspace

The left mouse button is used for selecting objects. You can either click on a single atom or bond, or you can drag to select multiple objects. The right mouse button opens shortcut menus, which are described in Section 2.7 of the *Maestro User Manual*.

The middle and right mouse buttons can be used on their own and in combination with the SHIFT and CTRL keys to perform common operations, such as rotating, translating, centering, adjusting, and zooming.

Table 2.1. Mapping of Workspace operations to mouse actions.

Mouse Button	Keyboard	Motion	Action
Left		click, drag	Select
Left	SHIFT	click, drag	Toggle the selection
Middle		drag	Rotate about X and Y axes Adjust bond, angle, or dihedral
Middle	SHIFT	drag vertically	Rotate about X axis
Middle	SHIFT	drag horizontally	Rotate about Y axis
Middle	CTRL	drag horizontally	Rotate about Z axis
Middle	SHIFT + CTRL	drag horizontally	Zoom
Right		click	Spot-center on selection
Right		click and hold	Display shortcut menu
Right		drag	Translate in the X-Y plane
Right	SHIFT	drag vertically	Translate along the X axis
Right	SHIFT	drag horizontally	Translate along the Y axis
Right	CTRL	drag horizontally	Translate along the Z axis
Middle & Right		drag horizontally	Zoom

2.3.4 Shortcut Key Combinations

Some frequently used operations have been assigned shortcut key combinations. The shortcuts available in the main window are described in Table 2.2.

Table 2.2. Shortcut keys in the Maestro main window.

Keys	Action	Equivalent Menu Choices
CTRL+B	Open Build panel	Edit > Build
CTRL+C	Create entry	Project > Create Entry From Workspace
CTRL+E	Open Command Script Editor panel	Edit > Command Script Editor
CTRL+F	Open Find Atoms panel	Edit > Find
CTRL+H	Open Help panel	Help > Help
CTRL+I	Open Import panel	Project > Import Structures
CTRL+M	Open Measurements panel	Tools > Measurements
CTRL+N	Create new project	Project > New
CTRL+O	Open project	Project > Open
CTRL+P	Print	Maestro > Print
CTRL+Q	Quit	Maestro > Quit
CTRL+S	Open Sets panel	Tools > Sets
CTRL+T	Open Project Table panel	Project > Show Table
CTRL+W	Close project	Project > Close
CTRL+Z	Undo/Redo last command	Edit > Undo/Redo
CTRL+=	Enter and exit full screen mode (Workspace occupies full screen)	None

2.4 Maestro Projects

All the work you do in Maestro is done within a *project*. A project consists of a set of *entries*, each of which contains one or more chemical structures and their associated data. In any Maestro session, there can be only one Maestro project open. If you do not specify a project when you start Maestro, a *scratch* project is created. You can work in a scratch project without saving it, but you must save it in order to use it in future sessions. When you save or close a project, all the view transformations (rotation, translation, and zoom) are saved with it. When you close a project, a new scratch project is automatically created.

Likewise, if there is no entry displayed in the Workspace, Maestro creates a *scratch* entry. Structures that you build in the Workspace constitute a scratch entry until you save the structures as project entries. The scratch entry is not saved with the project unless you explicitly add it to the project. However, you can use a scratch entry as input for some calculations.

To add a scratch entry to a project, do one of the following:

- Click the Create entry from Workspace button:

- Choose Create Entry from Workspace from the Project menu.
- Press CTRL+C.

In the dialog box, enter a name and a title for the entry. The entry name is used internally to identify the entry and can be modified by Maestro. The title can be set or changed by the user, but is not otherwise modified by Maestro.

Once an entry has been incorporated into the project, its structures and their data are represented by a row in the Project Table. Each row contains the row number, an icon indicating whether the entry is displayed in the Workspace (the In column), the entry title, a button to open the Surfaces panel if the entry has surfaces, the entry name, and any entry properties. The row number is not a property of the entry.

Entries can be collected into groups, and the members of the group can be displayed or hidden. Most additions of multiple entries to the Project Table are done as entry groups.

You can use entries as input for all of the computational programs—Glide, Impact, Jaguar, Liaison, LigPrep, MacroModel, Phase, Prime, QikProp, QSite, and Strike. You can select entries as input for the ePlayer, which displays the selected structures in sequence. You can also duplicate, combine, rename, and sort entries; create properties; import structures as entries; and export structures and properties from entries in various formats.

To open the Project Table panel, do one of the following:

- Click the Open/Close Project Table button on the toolbar

- Choose Show Table from the Project menu
- Press CTRL+T.

The Project Table panel contains a menu bar, a toolbar, and the table itself.

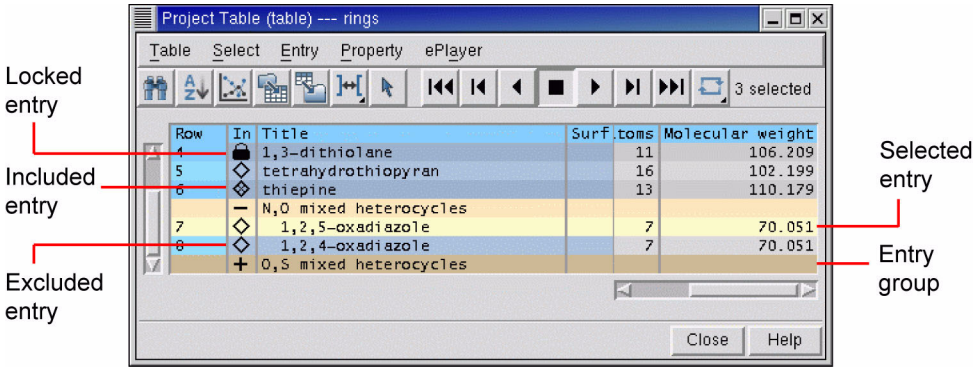

Figure 2.3. The Project Table ***panel.***

2.4.1 The Project Table Toolbar

The Project Table toolbar contains two groups of buttons and a status display. The first set of buttons opens various panels that allow you to perform functions on the entries in the Project Table. The second set of buttons controls the ePlayer, which "plays through" the selected structures: each structure is displayed in the Workspace in sequence, at a given time interval. See Section 2.3.2 on page 9 for a description of the types of toolbar buttons. The buttons are described below.

Find
Open the Find panel for locating alphanumeric text in any column of the Project Table, except for the row number.

Sort
Open the Sort panel for sorting entries by up to three properties.

Plot
Open the Plot panel for plotting entry properties.

Import Structure
Open the Import panel for importing structures into the project.

Export Structure
Open the Export panel for exporting structures to a file.

Columns
Choose an option for adjusting the column widths.

Select only
Open the Entry Selection dialog box for selecting entries based on criteria for entry properties.

Go to start
Display the first selected structure.

Previous
Display the previous structure in the list of selected structures.

Play backward
Display the selected structures in sequence, moving toward the first.

Stop
Stop the ePlayer.

Play forward
Display the selected structures in sequence, moving toward the last.

Next
Display the next structure in the list of selected structures.

Go to end
Display the last selected structure.

Loop
Choose an option for repeating the display of the structures. Single Direction displays structures in a single direction, then repeats. Oscillate reverses direction each time the beginning or end of the list is reached.

The status display, to the right of the toolbar buttons, shows the number of selected entries. When you pause the cursor over the status display, the Balloon Help shows the total number of entries, the number shown in the table, the number selected, and the number included in the Workspace.

2.4.2 The Project Table Menus

- Table—find text, sort entries, plot properties, import and export structures, and configure the Project Table.
- Select—select all entries, none, invert your selection, or select classes of entries using the Entry Selection dialog box and the Filter panel.
- Entry—include or exclude entries from the Workspace, display or hide entries in the Project Table, and perform various operations on the selected entries.
- Property—display and manipulate entry properties in the Project Table.
- ePlayer—view entries in succession, stop, reverse, and set the ePlayer options.

2.4.3 Selecting Entries

Many operations in Maestro are performed on the entries selected in the Project Table. The Project Table functions much like any other table: select rows by clicking, shift-clicking, and control-clicking. However, because clicking in an editable cell of a selected row enters edit mode, you should click in the Row column to select entries. See Section 2.4.5 on page 18 for more information on mouse actions in the Project Table. There are shortcuts for selecting classes of entries on the Select menu.

In addition to selecting entries manually, you can select entries that meet a combination of conditions on their properties. Such combinations of conditions are called *filters*. Filters are Entry Selection Language (ESL) expressions and are evaluated at the time they are applied. For example, if you want to set up a Glide job that uses ligands with a low molecular weight (say, less than 300) and that has certain QikProp properties, you can set up a filter and use it to select entries for the job. If you save the filter, you can use it again on a different set of ligands that meet the same selection criteria.

To create a filter:

1. Do one of the following:
   - Choose Only, Add, or Deselect from the Select menu.
   - Click the Entry selection button on the toolbar.

2. In the Properties folder, select a property from the property list, then select a condition.
3. Combine this selection with the current filter by clicking Add, Subtract, or Intersect. These buttons perform the Boolean operations OR, AND NOT, and AND on the corresponding ESL expressions.
4. To save the filter for future use click Create Filter, enter a name, and click OK.
5. Click OK to apply the filter immediately.

2.4.4 Including Entries in the Workspace

In addition to selecting entries, you can also use the Project Table to control which entries are displayed in the Workspace. An entry that is displayed in the Workspace is *included* in the Workspace; likewise, an entry that is not displayed is *excluded*. Included entries are marked by an X in the diamond in the In column; excluded entries are marked by an empty diamond. Entry inclusion is completely independent of entry selection.

To include or exclude entries, click, shift-click, or control-click in the In column of the entries, or select entries and choose Include or Exclude from the Entry menu. Inclusion with the mouse works just like selection: when you include an entry by clicking, all other entries are excluded.

It is sometimes useful to keep one entry in the Workspace and include others one by one: for example, a receptor and a set of ligands. You can fix the receptor in the Workspace by selecting it in the Project Table and choosing Fix from the Entry menu or by pressing CTRL+F. A padlock icon replaces the diamond in the In column to denote a *fixed* entry. To remove a fixed entry from the Workspace, you must exclude it explicitly (CTRL+X). It is not affected by the inclusion or exclusion of other entries. Fixing an entry affects only its inclusion; you can still rotate, translate, or modify the structure.

2.4.5 Mouse Functions in the Project Table

The Project Table supports the standard use of shift-click and control-click to select objects. This behavior applies to the selection of entries and the inclusion of entries in the Workspace. You can also drag to resize rows and columns and to move rows.

You can drag a set of non-contiguous entries to reposition them in the Project Table. When you release the mouse button, the entries are placed after the first unselected entry that precedes the entry on which the cursor is resting. For example, if you select entries 2, 4, and 6, and release the mouse button on entry 3, these three entries are placed after entry 1, because entry 1 is the first unselected entry that precedes entry 3. To move entries to the top of the table, drag them above the top of the table; to move entries to the end of the table, drag them below the end of the table.

A summary of mouse functions in the Project Table is provided in Table 2.3.

Table 2.3. Mouse operations in the Project Table.

Task	Mouse Operation
Change a Boolean property value	Click repeatedly in a cell to cycle through the possible values (On, Off, Clear)
Display the Entry menu for an entry	Right-click anywhere in the entry. If the entry is not selected, it becomes the selected entry. If the entry is selected, the action is applied to all selected entries.
Display a version of the Property menu for a property	Right-click in the column header
Edit the text or the value in a table cell	Click in the cell and edit the text or value
Include an entry in the Workspace, exclude all others	Click the In column of the entry

Table 2.3. Mouse operations in the Project Table. (Continued)

Task	Mouse Operation
Move selected entries	Drag the entries
Paste text into a table cell	Middle-click
Resize rows or columns	Drag the boundary with the middle mouse button
Select an entry, deselect all others	For an unselected entry, click anywhere in the row except the In column; for a selected entry, click the row number.
Select or include multiple entries	Click the first entry then shift-click the last entry
Toggle the selection or inclusion state	Control-click the entry or the In column

2.4.6 Project Table Shortcut Keys

Some frequently used project operations have been assigned shortcut key combinations. The shortcuts, their functions, and their menu equivalents are listed in Table 2.4.

Table 2.4. Shortcut keys in the Project Table.

Keys	Action	Equivalent Menu Choices
CTRL+A	Select all entries	Select > All
CTRL+F	Fix entry in Workspace	Entry > Fix
CTRL+I	Open Import panel	Table > Import Structures
CTRL+N	Include only selected entries	Entry > Include Only
CTRL+U	Deselect all entries	Select > None
CTRL+X	Exclude selected entries	Entry > Exclude
CTRL+Z	Undo/Redo last command	Edit > Undo/Redo in main window

2.5 Building a Structure

After you start Maestro, the first task is usually to create or import a structure. You can open existing Maestro projects or import structures from other sources to obtain a structure, or you can build your own. To open the Build panel, do one of the following:

- Click the Open/Close Build panel button in the toolbar:

- Choose Build from the Edit menu.
- Press CTRL+B.

The Build panel allows you to create structures by drawing or placing atoms or fragments in the Workspace and connecting them into a larger structure, to adjust atom positions and bond orders, and to change atom properties. This panel contains a toolbar and three folders.

2.5.1 Placing and Connecting Fragments

The Build panel provides several tools for creating structures in the Workspace. You can place and connect fragments, or you can draw a structure freehand.

To place a fragment in the Workspace:

1. Select Place.
2. Choose a fragment library from the Fragments menu.
3. Click a fragment.
4. Click in the Workspace where you want the fragment to be placed.

To connect fragments in the Workspace, do one of the following:

- Place another fragment and connect them using the Connect & Fuse panel, which you open from the Edit menu on the main menu bar or with the Display Connect & Fuse panel on the Build toolbar.

- Replace one or more atoms in the existing fragment with another fragment by selecting a fragment and clicking in the Workspace on the main atom to be replaced.
- Grow another fragment by selecting Grow in the Build panel and clicking the fragment you want to add in the Fragments folder.

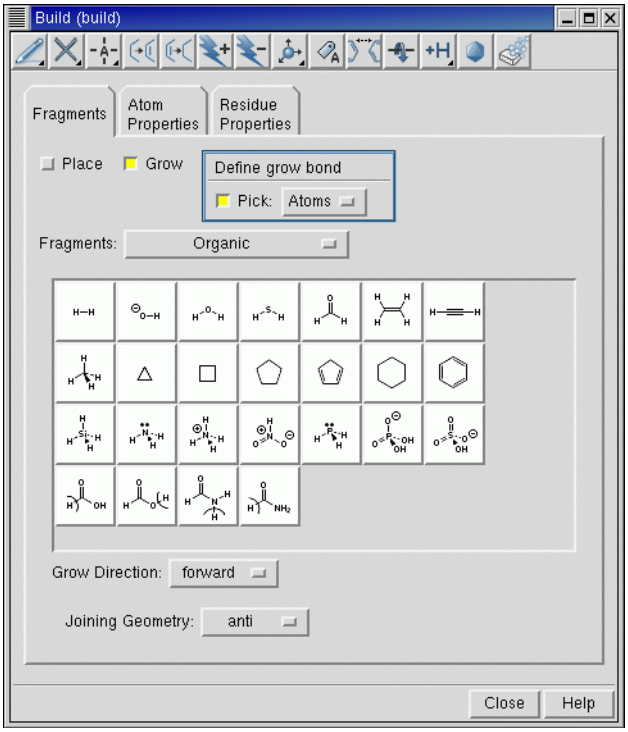

Figure 2.4. The Build ***panel.***

Grow mode uses predefined rules to connect a fragment to the *grow bond*. The grow bond is marked by a green arrow. The new fragment replaces the atom at the head of the arrow on the grow bond and all atoms attached to it. To change the grow bond, choose Bonds from the Pick option menu in the Build panel and click on the desired grow bond in the Workspace. The arrow points to the atom nearest to where you clicked.

To draw a structure freehand:

1. Choose an element from the Draw button menu on the Build panel toolbar:

2. Click in the Workspace to place an atom of that element.
3. Click again to place another atom and connect it to the previous atom.
4. Continue this process until you have drawn the structure.
5. Click the active atom again to finish drawing.

2.5.2 Adjusting Properties

In the Atom Properties folder, you can change the properties of the atoms in the Workspace. For each item on the Property option menu—Element, Atom Type (MacroModel), Partial Charge, PDB Atom Name, Grow Name, and Atom Name—there is a set of tools you can use to change the atom properties. For example, the Element tools consist of a periodic table from which you can choose an element and select an atom to change it to an atom of the selected element.

Similarly, the Residue Properties folder provides tools for changing the properties of residues: the Residue Number, the Residue Name, and the Chain Name.

To adjust bond lengths, bond angles, dihedral angles, and chiralities during or after building a structure, use the Adjust distances, angles or dihedrals button on the main toolbar:

You can also open the Adjust panel from this button menu, from the Display Adjust panel button on the Build panel toolbar (which has the same appearance as the above button) or from the Edit menu in the main window.

2.5.3 The Build Panel Toolbar

The toolbar of the Build panel provides quick access to tools for drawing and modifying structures and labeling atoms. See Section 2.3.2 on page 9 for a description of the types of toolbar buttons. The toolbar buttons and their use are described below.

Free-hand drawing
Choose an element for drawing structures freehand in the Workspace (default C). Each click in the Workspace places an atom and connects it to the previous atom.

Delete
Choose an object for deleting. Same as the Delete button on the main toolbar, see page 10.

Set element
Choose an element for changing atoms in the Workspace (default C). Click an atom to change it to the selected element.

Increment bond order
Select a bond to increase its bond order by one, to a maximum of 3.

Decrement bond order
Select a bond to decrease its bond order by one, to a minimum of 0.

Increment formal charge
Select an atom to increase its formal charge by one.

Decrement formal charge
Select an atom to decrease its formal charge by one.

Move
Choose a direction for moving atoms, then click the atom to be moved. Moves in the XY plane are made by clicking the new location. Moves in the Z direction are made in 0.5 Å increments.

Label
Apply heteroatom labels as you build a structure. The label consists of the element name and formal charge, and is applied to atoms other than C and H.

Display Connect & Fuse panel
Open the Connect & Fuse panel so you can connect structures (create bonds between structures) or fuse structures (replace atoms of one structure with those of another).

Display Adjust panel
Open the Adjust panel so you can change bond lengths, bond angles, dihedral angles, or atom chiralities.

Add hydrogens
Choose an atom type for applying the current hydrogen treatment. Same as the Add hydrogens button on the main toolbar, see page 10.

Geometry Symmetrizer
Open the Geometry Symmetrizer panel for symmetrizing the geometry of the structure in the Workspace.

Geometry Cleanup
Clean up the geometry of the structure in the Workspace.

2.6 Selecting Atoms

Maestro has a powerful set of tools for selecting atoms in a structure: toolbar buttons, picking tools in panels, and the Atom Selection dialog box. These tools allow you to select atoms in two ways:

- Select atoms first and apply an action to them
- Choose an action first and then select atoms for that action

2.6.1 Toolbar Buttons

The small triangle in the lower right corner of a toolbar button indicates that the button contains a menu. Many of these buttons allow you to choose an object type for selecting: choose Atoms, Bonds, Residues, Chains, Molecules, or Entries, then click on an atom in the Workspace to perform the action on all the atoms in that structural unit.

For example, to select atoms with the Workspace selection toolbar button:

1. Choose Residues from the Workspace selection button menu:

   The button changes to:

2. Click on an atom in a residue in the Workspace to select all the atoms in that residue.

2.6.2 Picking Tools

The picking tools are embedded in each panel in which you need to select atoms to apply an operation. The picking tools in a panel can include one or more of the following:

- Pick option menu—Allows you to choose an object type. Depending on the operation to be performed, you can choose Atoms, Bonds, Residues, Chains, Molecules, or Entries, then click on an atom in the Workspace to perform the action on all the atoms in that structural unit.

  The Pick option menu varies from panel to panel, because not all object types are appropriate for a given operation. For example, some panels have only Atoms and Bonds in the Pick option menu.

- All button—Performs the action on all atoms in the Workspace.
- Selection button—Performs the action on any atoms already selected in the Workspace.
- Previous button—Performs the action on the most recent atom selection defined in the Atom Selection dialog box.
- Select button—Opens the Atom Selection dialog box.
- ASL text box—Allows you to type in an ASL expression for selecting atoms.

  ASL stands for Atom Specification Language, and is described in detail in the *Maestro Command Reference Manual*.

- Clear button—Clears the current selection

- Show markers option—Marks the selected atoms in the Workspace.

For example, to label atoms with the Label Atoms panel:

1. Choose Atom Labels from the Display menu.
2. In the Composition folder, select Element and Atom Number.
3. In the picking tools section at the top of the panel, you could do one of the following:
   - Click Selection to apply labels to the atoms already selected in the Workspace (from the previous example).
   - Choose Residues from the Pick option menu and click on an atom in a different residue to label all the atoms in that residue.

2.6.3 The Atom Selection Dialog Box

If you wish to select atoms based on more complex criteria, you can use the Atom Selection dialog box. To open this dialog box, choose Select from a button menu or click the Select button in a panel. See Section 5.3 of the *Maestro User Manual* for detailed instructions on how to use the Atom Selection dialog box.

2.7 Scripting in Maestro

Although you can perform nearly all Maestro-supported operations through menus and panels, you can also perform operations using Maestro commands, or compilations of these commands, called *scripts*. Scripts can be used to automate lengthy procedures or repetitive tasks and can be created in several ways. These are summarized below.

2.7.1 Python Scripts

Python is a full-featured scripting language that has been embedded in Maestro to extend its scripting facilities. The Python capabilities within Maestro include access to Maestro functionality for dealing with chemical structures, projects, and Maestro files.

The two main Python commands used in Maestro are:

- `pythonrun`—executes a Python module. (You can also use the alias `pyrun`.) The syntax is:

  `pythonrun` *module*.*function*

- `pythonimport`—rereads a Python file so that the next time you use the `pythonrun` command, it uses the updated version of the module. (You can also use the alias `pyimp`.)

From the Maestro Scripts menu you can install, manage, and run Python scripts. For more information on the Scripts menu, see Section 13.1 of the *Maestro User Manual*.

For more information on using Python with Maestro, see *Scripting with Python*.

2.7.2 Command Scripts

All Maestro commands are logged and displayed in the Command Script Editor panel. This means you can create a command script by performing the operations with the GUI controls, copying the logged commands from the Command History list into the Script text area of the panel, then saving the list of copied commands as a script.

To run an existing command script:

1. Open the Command Script Editor panel from the Edit menu in the main window.
2. Click Open Local and navigate to the directory containing the desired script.
3. Select a script in the Files list and click Open.

   The script is loaded into the Script window of the Command Script Editor panel.
4. Click Run Script.

Command scripts cannot be used for Prime operations.

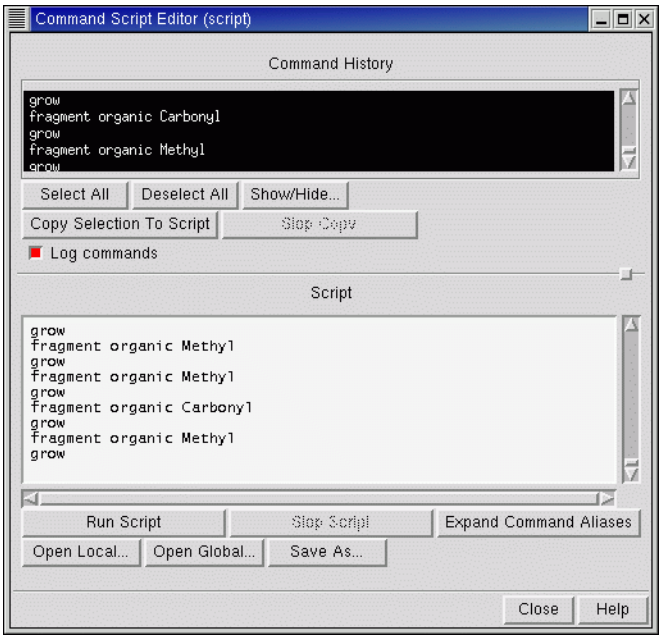

Figure 2.5. The Command Script Editor ***panel.***

2.7.3 Macros

There are two kinds of macros you can create: named macros and macros assigned to function keys F1 through F12.

To create and run a named macro:

1. Open the Macros panel from the Edit menu in the main window.
2. Click New, enter a name for the macro, and click OK.
3. In the Definition text box, type the commands for the macro.
4. Click Update to update the macro definition.
5. To run the macro, enter the following in the command input area in the main window:

   `macrorun` *macro-name*

   If the command input area is not visible, choose Command Input Area from the Display menu.

To create and run a function key macro:

1. Open the Function Key Macros panel from the Edit menu in the main window.
2. From the Macro Key option, select a function key (F1 through F12) to which to assign the macro.
3. In the text box, type the commands for the macro.
4. Click Run to test the macro or click Save to save it.
5. To run the macro from the main window, press the assigned function key.

For more information on macros, see Section 13.5 of the *Maestro User Manual*.

2.8 Specifying a Maestro Working Directory

When you use Maestro to launch LigPrep jobs, Maestro writes job output to the directory specified in the Directory folder of the Preferences panel. By default, this directory (the file I/O directory) is the directory from which you started Maestro.

To change the Maestro working directory:

1. Open the Preferences panel from the Maestro menu.
2. Click the Directory tab.
3. Select the directory you want to use for reading and writing files.

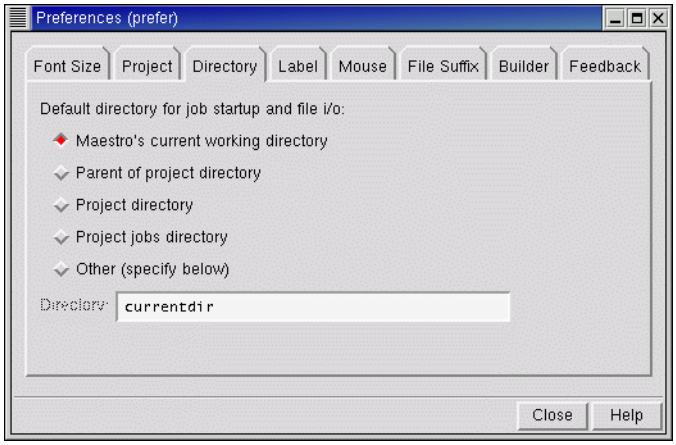

Figure 2.6. The Directory ***folder of the*** Preferences ***panel.***

You can also set other preferences in the Preferences panel. See Section 12.2 of the *Maestro User Manual* for details.

2.9 Undoing an Operation

To undo a single operation, click the Undo button in the toolbar, choose Undo from the Edit menu, or press CTRL+Z. The word Undo in the menu is followed by text that describes the operation to undo. Not all operations can be undone: for example, global rotations and translations are not undoable operations. For such operations you can use the Save view and Restore view buttons in the toolbar, which save and restore a molecular orientation.

2.10 Running and Monitoring Jobs

Maestro has panels for each product for preparing and submitting jobs. To use these panels, choose the appropriate product and task from the Applications menu and its submenus. Set the appropriate options in the panel, then click Start to open the Start dialog box and set options for running the job. For a complete description of the Start dialog box associated with your computational program, see your product's User Manual. When you have finished setting the options, click Start to launch the job and open the Monitor panel.

The Monitor panel is the control panel for monitoring the progress of jobs and for pausing, resuming, or killing jobs. All jobs that belong to you can be displayed in the Monitor panel, whether or not they were started from Maestro. Subjobs are indented under their parent in the job list. The text pane shows output information from the monitored job, such as the contents

of the log file. The Monitor panel opens automatically when you start a job. If it is not open, you can open it by choosing Monitor from the Applications menu in the Maestro main window.

While jobs are running, the Detach, Pause, Resume, Stop, Kill, and Update buttons are active. When there are no jobs currently running, only the Monitor and Delete buttons are active. These buttons act on the selected job. By default, only jobs started from the current project are shown. To show other jobs, deselect Show jobs from current project only.

When a monitored job ends, the results are incorporated into the project according to the settings used to launch the job. If a job that is not currently being monitored ends, you can select it in the Monitor panel and click Monitor to incorporate the results. Monitored jobs are incorporated only if they are part of the current project. You can monitor jobs that are not part of the current project, but their results are not incorporated. To add their results to a project, you must open the project and import the results.

Further information on job control, including configuring your site, monitoring jobs, running jobs, and job incorporation, can be found in the *Job Control Guide* and the *Installation Guide*.

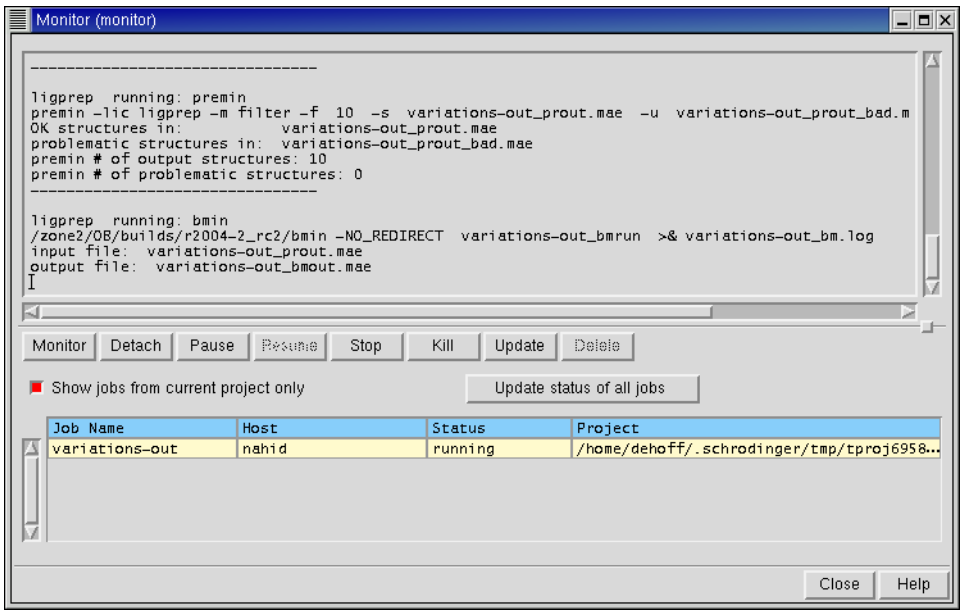

Figure 2.7. The Monitor ***panel.***

2.11 Getting Help

Maestro comes with automatic, context-sensitive help (Auto-Help), Balloon Help (tooltips), an online help facility, and a user manual. To get help, follow the steps below:

- Check the Auto-Help text box at the bottom of the main window. If help is available for the task you are performing, it is automatically displayed there. It describes what actions are needed to perform the task.
- If your question concerns a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Help menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- If you do not find the help you need using either of the steps above, click the Help button in the lower right corner of the appropriate panel. The Help panel is displayed with a relevant help topic.
- For help with a concept or action not associated with a panel, open the Help panel from the Help menu or press CTRL+H.

If you do not find the information you need in the Maestro help system, check the following sources:

- The *Maestro User Manual*
- The Frequently Asked Questions page on the Schrödinger Support Center.

You can also contact Schrödinger by e-mail or phone for help:

- E-mail: help@schrodinger.com
- Phone: (503) 299-1150

2.12 Ending a Maestro Session

To end a Maestro session, choose Quit from the Maestro menu. To save a log file with a record of all operations performed in the current session, click Quit, save log file in the Quit panel. This information can be useful to Schrödinger support staff when responding to any problem you report.

Running LigPrep

The main objective of LigPrep is to take 2D or 3D structures and produce corresponding low-energy 3D structures for use in further computational studies. The input and output files can be in SD or Maestro format. LigPrep can also produce multiple output structures for each input structure, a process referred to as *expansion*, by generating variations on the ionization state, tautomers, stereochemistry, and ring conformations.

3.1 Running LigPrep From Maestro

You can set up a LigPrep job from the LigPrep panel in Maestro. To open the LigPrep panel, choose LigPrep from the Applications menu in the main window.

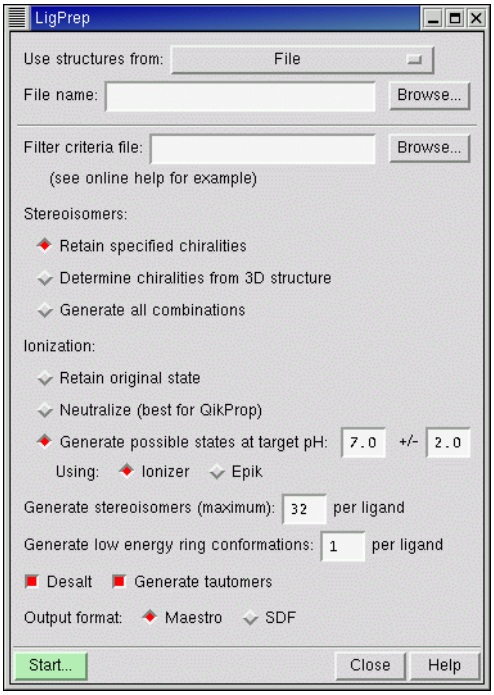

Figure 3.1. The** LigPrep **panel.

3.1.1 Job Input

The upper portion of the LigPrep panel contains settings that control the input structures.

Use structures from

The input for LigPrep can come from the Workspace, selected entries in the Project Table, or from a file. If you select Workspace (included entries) or Project Table (selected entries) as the source of job input, the structures are written to a file in Maestro format, named *jobname*`.mae`. If you select File, you can type a file name in the File name text box, or click Browse to navigate to the file and select it.

If you decide to use a file for your job input, the results are written to a file in the format you specify under Output format (in the lower left of the panel). The file name is *jobname*.*ext*, where *ext* is `mae` for Maestro format, `sdf` for SD format, or `smi` for SMILES format.

3.1.2 Options for Structural Variation

Stereoisomers

The option you select affects the treatment of stereochemical information in the input and the generation of stereoisomers in the output.

- Retain specified chiralities keeps the information on chiralities from the input file and fixes these chiralities for the entire calculation. Chirality information includes parities and bond directions from SD files and the chirality property from Maestro files. (Maestro chiralities are written only by `sdconvert` and the `stereoizer`.) If the configuration or chirality of a chiral center is not specified, the two possible chiralities are generated in the output.
- Determine chiralities from 3D structure discards any information from the input file and determines the chirality from the 3D geometry. These chiralities are held fixed. For centers whose chirality is indeterminate, the two possible chiralities are generated in the output.
- Generate all combinations discards chirality information and generates all possible configurations that result from the combination of chiralities on each chiral center.

For any of these settings, you can limit the number of stereoisomers generated for the atoms whose chiralities are not assigned by entering a limit in the Generate stereoisomers (maximum) text box. Stereoisomer generation starts by setting all configurations to R and systematically varying the configurations.

Ionization States

The option you select controls the generation of ionization states of the ligand. The ionization states are generated by adding or removing protons from the ligand. Some programs require neutral molecules as input. For these programs, you should select Neutralize. For other programs, such as Glide, it can be useful to generate a range of ionization states that are populated in a given pH range. The Generate possible states option selects the generation of ionization states that are significantly populated in the specified pH range. If you do not want the ionization states to be altered, choose Retain original state. However, even if you select Retain original state, deprotonated acids can still be protonated by `applyhtreat` if the formal charges in the acid are incorrect.

To perform the ionization, you can select either the Ionizer or Epik. Epik is a separate product, and must be purchased and installed if you want to use it with LigPrep. The Ionizer is included with LigPrep.

Ring Conformations

From a 2D structure, it is not immediately obvious which ring conformations give the lowest energy or are preferred for binding to an active site. LigPrep can generate ring conformations and evaluate an approximate conformational energy to determine which conformations are likely to be the lowest in energy. The value in the Generate low energy ring conformations text box determines how many conformations are kept, counting from the lowest in energy.

Cofactors

Structures from some databases can consist of multiple molecules, one of which is the ligand. In many cases, the extra molecules are water molecules or counter ions. To remove these molecules, select the Desalt option. LigPrep removes all but the largest molecule when this option is selected. In most cases, this procedure removes the correct molecules. If the desired molecule is not the largest, you may have to edit the structure in Maestro before passing it to LigPrep.

Tautomers

For some ligand molecules, a tautomer of the commonly-given structure binds to the active site. To generate up to 8 tautomers per input structure, select Generate Tautomers. LigPrep performs keto-enol tautomerization and the analogous sulfur and nitrogen tautomerizations, and histidine and DNA base tautomerizations. For more information, see Chapter 7.

Output

If you use a file for your job input, or choose not to incorporate your results in the Project Table (see Output Settings below), the results are written to a file in the format you specify. The file name is *jobname*.*ext*, where *ext* is `mae` for Maestro format or `sdf` for SD format.

3.1.3 Job Submission

Once you have selected the structural variation options, click Start to open the Start dialog box (see Figure 3.2) and choose options for how and where the job is run.

Output Settings

- Incorporate—If your job input came from the Project Table or the Workspace, the results can be incorporated into the Project Table when the job finishes. You can either append the new entries to the Project Table or replace the existing entries. If you choose not to incorporate the results, the results are written to a file in the format you specified under Output format (see Output above). The file name is *jobname*.*ext*, where *ext* is `mae` for Maestro format or `sdf` for SD format.

Job Settings

- Name—Enter a name for the computational job. This name is used as a prefix for all files created during the calculation.

  Note: Maestro does *not* automatically assign new names to jobs or files. However, it does display a warning if files of the same name already exist.

- Host—Choose a host, if you want to run the job on a remote machine. This option menu displays all the hosts defined in the `$SCHRODINGER/schrodinger.hosts` file, with the number of processors on the host in parentheses. For more information on hosts, host settings, and configuring this file, see the *Job Control Guide* or the *Installation Guide*.

- Username—Enter your user name, if it is required for running the job on remote machines. The default value is the user name of the user who started Maestro. If this user name is not correct for the selected host, you can change it in this text box.

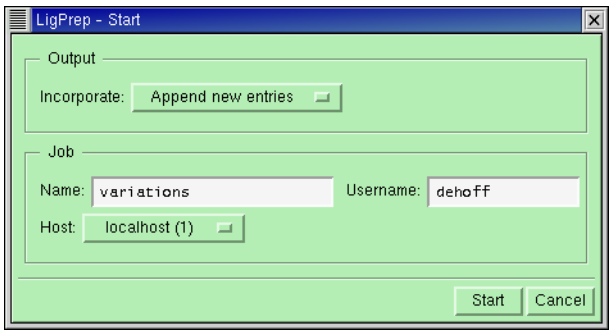

Figure 3.2. The LigPrep - Start ***dialog box.***

Starting the Job

When you have finished choosing the output and job settings, click Start to start the job. The LigPrep job is submitted to the selected host, and the Maestro Monitor panel is displayed. This panel provides information on the status of the job. Output for each step of the LigPrep process is stored in a temporary directory. The log file for the job is stored in the job submission directory (the current working directory in Maestro). This file is updated as the job progresses and is displayed in the text area of the Monitor panel. For more information on job monitoring, see Section 2.10 on page 28 or the *Job Control Guide*.

3.2 The ligprep Command

To run LigPrep from the command line, use the `ligprep` command. The syntax of the `ligprep` command is as follows.

> `$SCHRODINGER/ligprep` [*options* | *meta-options*] {`-imae`|`-isd`|`-ismi`} *input_filename*{`-omae`|`-osd`} *output_filename*

You must specify an input file using either `-imae` for input files in Maestro format, `-isd` for input files in SD file format, or `-ismi` for input files in SMILES format. The output file is specified with `-omae` for Maestro format or `-osd` for SD format.

The `ligprep` command has a number of options that allow you to control various stages of the process. The options and their descriptions are listed in Table 3.1 and can be displayed by entering the command `ligprep -h`.

Table 3.1. Options for the `ligprep` *command*

Option	Definition
`-a`	Append structures to the output file. Default is to overwrite the output file if it already exists.
`-ac`	Do not respect existing chirality properties or use chiralities from the input geometry. Generate stereoisomers for all chiral centers up to the number permitted (specified using the `-s` option).
`-bff` *number*	Instruct `bmin` to use the force field specified. Only 10 (MMFFs), 11 (OPLS_2001), and 14 (OPLS_2005, default) are supported.
`-bvac`	Perform `bmin` minimization in a vacuum. By default `bmin` minimizations are carried out using a distance-dependent dielectric.
`-br`	Instruct `bmin` to retain structures with incorrect chiralities.
`-bns`	Instruct `bmin` to perform minimizations only. By default, `bmin` performs short conformational searches on distorted structures.

Table 3.1. Options for the `ligprep` *command (Continued)*

Option	Definition
`-btc` *number*	Torsional constraints to use in `bmin`. `-1` Do not use torsional constraints. `0`, `1` Torsional constraints for C=C, carboxylic acids, esters and amides. `2` Torsional constraints for C=C.
`-DEBUG`	Show details of operation of the top-level script.
`-epik`	Use Epik for the ionization and tautomerization stages.
`-f` *filename*	Filter structures using `ligparse` with specifications from *filename*. Default: do not filter.
`-fc` *filename*	Use custom composite descriptors for ligparse from *filename*. Only valid when used with `-f`. Default: use composite descriptors from `composite.types` file in the installation. See Section 8.1 on page 101 for more information.
`-fs` *filename*	Use custom SMARTS patterns (descriptors) for ligparse from *filename*. Only valid when used with `-f`. Default: use composite descriptors from `str_keys.types` file in the installation. See Section 8.1 on page 101 for more information.
`-g`	Respect chiralities from input geometry when generating stereoisomers. For chiral centers whose chirality cannot be determined from the input geometry, stereoisomers will be generated.
`-h[elp]`, `-HELP`	Show usage message.
`-i` *number*	Ionization treatment `0` Do not run `neutralizer` or `ionizer`. `1` Run neutralizer. `2` Run `neutralizer` and `ionizer`. Default: 1
`-is` *filename*	Use non-standard ionizer specification file, named *filename*.
`-kp` | `-keep_props`	Transmit properties from input to output at each stage, and retain properties in the output structure file. The `ionizer` and `bmin` delete properties by default.
`-l` *number*	Control how `ring_conf` handles atoms that are not in flexible rings. `1` Use input geometries. `2` Use idealized geometries even for rigid rings. `3` Use idealized geometries for non-ring portions and input geometries for rigid rings. Default: 3
`-ma` *number*	Skip over structures containing more than this number of atoms in some steps. Default: 150.

Table 3.1. Options for the `ligprep` *command (Continued)*

Option	Definition
`-mg` *number*	Instruct `ionizer` to skip structures that have more than this number of ionizable groups. Default: 10
`-n` *structs*	Specify a comma-delimited series of colon-separated ranges and single values to convert. For example: `1:10,14` Structures 1 through 10, and 14 `2:` Structures 2 through end of file `:5,13:18` Structures 1 through 5, and 13 through 18 Default: `1:` (convert all structures)
`-nc`	Do not remove intermediate files. Default is to remove intermediate files.
`-nd`	Do not use the `desalter`. Default is to use the `desalter`.
`-np`	Remove files containing problematic structures. Default is to keep problematic structure files.
`-nt`	Do not use the `tautomerizer` to generate tautomers. Default is to use the `tautomerizer`.
`-r` *number*	Generate up to *number* ring conformations per input structure in `ring_conf`. Default: 1
`-re` *number*	Generate ring conformations with energies lower than *number* (kJ/mol) relative to the lowest energy conformer. This number sets the maximum number of ring conformers to 8 unless a different number is provided using the `-r` option.

`-R {b|c|d|e|f|h|i|n|p|r|s|t}`	Run only the stage of ligand preparation specified by the letter code:
	`b` `bmin` minimization
	`c` `sdconvert`
	`d` `desalter`
	`e` `epik`
	`f` filter (`ligparse`)
	`h` `applyhtreat`
	`i` `ionizer`
	`n` `neutralizer`
	`p` `premin`
	`r` `ring_conf`
	`s` `stereoizer`
	`t` `tautomerizer`

`-s` *number*	Generate up to *number* stereoisomers per input structure in `stereoizer`. Default: 32
`-t` *number*	Generate up to *number* tautomers per input structure in `tautomerizer`. Default: 8
`-ts` *filename*	Specify a custom tautomer database.

Table 3.1. Options for the `ligprep` *command (Continued)*

Option	Definition
`-tp` *number*	Set the minimum tautomerization probability to *number*, where 0 < *number* < 1.0. Structures with net tautomerization probabilities lower than this value are not retained. Default: 0.01
`-verb`	Report on the progress of the ligand preparation.

`-W` *string*	Options for various stages of ligand preparation. *string* consists of comma-separated designators. The first designator specifies the program that options are to be passed to and must be one of:
	`e` `epik`
	`i` `ionizer`
	`r` `ring_conf`
	`s` `stereoizer`
	`f` filter (`ligparse`)
	The remaining designators are passed to the program in the order specified with the commas removed. For example: `r,-l,1,-n,16` passes the arguments `-l 1 -n 16` to `ring_conf`. Multiple `-W` options are permitted, but only the last one pertaining to a particular program is used. For `ring_conf` and `stereoizer`, these options are used in addition to the default options and override redundant specifications. When `ligparse` is in use, it is passed the following arguments by default: `-any -j 5` (filter by rejecting structures if any criteria listed in the filter file are matched). If the `-W` option is used to pass arguments to `ligparse`, the default set of arguments is not used in `ligparse` processing.
	Note: Not all `ionizer` options are supported with the `-W` option. See Section 6.1 on page 81 for more information.

The `ligprep` command also has a number of *meta-options*, which set a group of options with particular values. The meta-options are listed in Table 3.2, and their actions are described in Table 3.3. You can use options together with meta-options; the options override any values set by the meta-options.

The `adjust`, `vary` and `expand` meta-options control the range of structures generated. You can use all three of these meta-options in the same `ligprep` command, but `c`, `i` and `t` must not appear more than once. For instance, you could specify `-adjust_t -vary_c -expand_i`, but not `-adjust_t -vary_c -expand_it`.

The `retain` meta-option requires that at least one structure produced for each input structure retains the characteristics denoted by the suffix. For instance, `-retain_t` means that the tautomerization state in the input structure will be present in at least one of the output structures for that molecule. The `retain` meta-option can be used with any of the `adjust`, `vary` and `expand` meta-options.

Table 3.2. Meta-options for the `ligprep` *command. Suffix values are given for each meta-option that requires a suffix.*

Option	Description
`-adjust_`*suffix*	Adjust to a suitable state. Any combination of `c` (chirality), `i` (ionization states) or `t` (tautomers) can be used for *suffix*.
`-expand_`*suffix*	Aggressively expand states. Any combination of `c` (chirality), `i` (ionization states) or `t` (tautomers) can be used for *suffix*.
`-retain_`*suffix*	Retain the original state in the output. Any combination of `c` (chirality), `i` (ionization states) or `t` (tautomers) can be used for *suffix*. Suffix values determine which state is kept. If *suffix* is `i`, both the original ionization state and the original tautomer are retained.
`-qik`	Set defaults appropriate for QikProp. Turns on the `desalter`, the `neutralizer`, and the `tautomerizer`.
`-unt`	Run `premin` and `bmin` to untangle structures. Used by CombiGlide.
`-vary_`*suffix*	Generate a small number of states. Any combination of `c` (chirality), `i` (ionization states) or `t` (tautomers) can be used for *suffix*.

Table 3.3. Action performed by meta-options.

Suffix	Action of `adjust`	Action of `vary`	Action of `expand`
`c`	Produce only one stereoisomer	Produce at most 2 stereoisomers	Produce at most 32 stereoisomers
`i`	Produce the ionization state with the lowest overall penalty	Set pH tolerance to 1.0	Set pH tolerance to 2.0
`t`	Produce only the most probable tautomer	Produce at most the 2 most probable tautomers	Produce at most the 8 most probable tautomers

In addition to the options for the program, the ligprep command has job control options and diagnostic options, most of which are common to Schrödinger products. These options are listed in Table 3.4 and Table 3.5.

LigPrep jobs run in the background under Schrödinger's job control facility. The job output is normally written to a temporary directory and then copied back to the job submission directory when the job finishes. You can direct the temporary files and the output to the job submission directory with the `-LOCAL` option of the `ligprep` command. The `.log` file is written to the job submission directory and is updated during the run. For more information on the Job Control facility, see the *Job Control Guide*.

Table 3.4. Job control options for the `ligprep` *command.*

Option	Description
`-HOST` *hostname*	The name of the host to use for this run. The host name must be defined in the `schrodinger.hosts` file.
`-HOSTFILE` *filename*	The name of the host file to use for this run. The default host file is the installed version of `schrodinger.hosts`.
`-INTERVAL` *sec*	The maximum time in seconds between updates of the *jobname*`.log` file.
`-LOCAL`	Do not use a temporary directory for intermediate files. Keep files in the current working directory.
`-NICE`	Run the job at reduced priority.
`-PROJ` *name*	Assign job to the Maestro project *name*.
`-TMPDIR` *directory*	The name of the directory used to store files temporarily during a job.
`-USER` *name*	Launch job as user *name*.
`-WAIT`	Do not return a prompt until the job finishes.

Table 3.5. Diagnostic options for the `ligprep` *command.*

Option	Description
`-ALL`	Ignore platform compatibility (list all platforms).
`-ENTRY`	List the relevant host entries from the hosts file.
`-HOSTS`	List the hosts available for remote jobs.
`-LIST`	List all the platform-compatible versions of product.
`-WHICH`	Report the product exec directory and `MMSHARE_EXEC`.
`-WHY`	Show details of how the selected executable directory was chosen.

3.3 Distributing LigPrep Jobs with para_ligprep

You can significantly reduce the amount of time for LigPrep tasks involving many ligands by distributing these jobs across multiple processors. The `para_ligprep` utility is provided to facilitate this task. You can also use `para_ligprep` to split very large input structure files into smaller files that can be processed separately without exceeding the 2 GB file size limit (see page 45). The number of tasks may be greater than or equal to the number of processors requested, so that processors carry out more than one task before the job finishes. By distributing the overall calculation across multiple processors, a form of parallel processing, you can significantly reduce the wall-clock time.

Note: Distributed LigPrep jobs must be run from the command line using the `para_ligprep` command, except when they are run from the Phase panels.

3.3.1 Load Balancing

Load balancing involves giving different processors different amounts of work depending on how rapidly they carry out tasks assigned to them, in a manner intended to reduce the overall time between requesting the overall calculation and having it completed. Often if the overall calculation requires a lot of CPU time, there is a distinct advantage in setting the number of tasks to a small multiple (e.g. greater than 3) of the number of processors requested, particularly when the processors differ in inherent speed and load (number of programs contending for CPU time). When a processor finishes a task, it is assigned a new one from the pool of unprocessed tasks until the overall calculation is complete. As a result, processors that return results faster are assigned a greater portion of the overall calculation, which in turn is completed sooner.

3.3.2 Running para_ligprep

To run the `para_ligprep` script from the command line, use the following syntax:

`$SCHRODINGER/utilities/para_ligprep` [*options*]

The options are described in Table 3.6. You can also use the same diagnostic options as for the `ligprep` command—see Table 3.5.

Table 3.6. Options for the `para_ligprep` *command, including job control options.*

Option	Description
`-DEBUG`	Show details of operation of the top-level script.
`-first` *firstlig*	First ligand to include. Default 1.
`-HELP | -h[elp]`	Print usage message and exit.
`-HOST` *host_list*	*host_list* is a list of one or more hosts on which to run the job. The list must be quoted if multiple hosts are specified: for example, `"hostname1:nprocs1 hostname2:nprocs2 ..."` Default: `localhost:1`
`-HOSTFILE` *host_file_name*	The name of the host file to use for this run.
`-INTERVAL` *sec*	The maximum time in seconds between updates of the *jobname*`.log` file.
`-j` *jobnum*	Subjob number to prepare. Default 0, meaning all subjobs.

Table 3.6. Options for the `para_ligprep` *command, including job control options.*

Option	Description
`-JOBCTS` *maxctsjob*	Ensure that each subjob has no more than this many structures to process. Defaults to 10000.
`-last` *lastlig*	Last ligand to include. Default 0, meaning the last ligand the file.
`-LOCAL`	Do not use a temporary directory to store the files. Store files in the local directory.
`-NICE`	Run the job at reduced priority
`-nproc` *njobs* | `-NJOBS` *njobs*	Divide the overall job into *njobs* subjobs.
`-nx`	Just create input files for subjobs.
`-OUTPUT_ORG BY_SUBJOB`	Produce one output file for each subjob.
`-PROJ` *name*	Assign job to the Maestro project *name*.
`-TMPDIR` *directory*	Specify the directory used for temporary storage during a job.
`-v[er[sion]]`	Report the version number for `para_ligprep`
`-WAIT`	Do not return a prompt until the job finishes.

Any other options or arguments you enter on the command line are passed to the `ligprep` command for each subjob. The LigPrep input structure files must be specified with `-isd`, `-imae`, or `-ismi`, and the output structure file must be specified with `-omae` or `-osd`. The jobname for the overall `para_ligprep` run is derived from the output structure file name by removing everything after the last "." if present. The `-n` option in `ligprep` cannot be used with `para_ligprep`, but you can use the `maesubset` or `sdsubset` commands to select the desired ligands before running `para_ligprep`.

`para_ligprep` splits an input ligand file into *njobs* smaller input files, each of which is automatically processed using a separate `ligprep` calculation. The ligands between *firstlig* and *lastlig* (inclusive) are separated into *njobs* equal-sized batches. The `-j` option is useful for preparing and running a single subjob. A value of `0` is equivalent to not specifying `-j` at all, and all subjobs are processed. Negative values are not permitted. If the `-nx` option is given, the subjob files are prepared but not run. `para_ligprep` starts up a parent process on the first computer listed after `-HOST` or the localhost if `-HOST` is not used. This parent process runs the subjobs as child processes and organizes the output when all subjobs are completed.

It is common to specify both the number of tasks (using `-NJOBS`) and the number of processors (using `-HOST`) to use. For instance, if the input structure file, `many.sd`, contains 100,000 structures, then the command:

```
$SCHRODINGER/utilities/para_ligprep -nproc 20 -HOST big_host:5
-isd many.sd -osd many-out.sd
```

produces 20 subdirectories, each containing a separate 5,000-structure portion of the input file. Up to five subjobs at a time run on the computer `big_host` until all 20 subjobs are completed.

The jobname for `para_ligprep` is the stem of the output structure file name, formed by removing any `.mae`, `.sd`, or `.sdf` extension. By default, all output structures are placed in the output structure file listed on the command line. If the total size of the output structures would exceed 2GBytes, it is better to retain a separate output structure file, named *jobname*_subjob_*subjobnum*.mae, for each subjob by using the `-OUTPUT_ORG BY_SUBJOB` option.

Behind the scenes, `para_ligprep` jobs can involve many files in a number of remote directories (one for each subjob), which you usually do not need to examine. However, sometimes you may wish to examine these files. `para_ligprep` saves all these files in a single gzipped tar file called *jobname*.tar.gz. The job files can be extracted using the commands:

```
gunzip jobname.tar.gz
tar -xvf jobname.tar
```

3.4 LigPrep Files

LigPrep generates many temporary files due to the many stages of the LigPrep process. Most of these files are automatically deleted unless the program is instructed to keep them.

At each stage there is usually one input structure file, one output structure file, and one log file. For some stages the log files are normally empty, while for others they can be large. In general, the output structure file from one stage is used as the input structure for the next stage. When a stage is complete, the input structure file is removed unless it is the first stage in the process, the only stage that has been requested (with the `-R` option), or the `-nc` option (do not clean up files) has been given. If a stage fails in a manner that causes `ligprep` to terminate, none of the files for that stage are deleted. Log files from all stages are retained.

Most intermediate files are named by adding a suffix to the job name. The job name for all tasks in the LigPrep process is the stem of the output structure file name (the part prior to the last '.' or the entire name if no '.' is present). For example, the name of the output structure file for `sdconvert` is *jobname*_sdout.mae. File name suffixes and other files used are described in Table 3.7. The output structure file for the last stage of a successful LigPrep run is renamed to the output structure file name specified on the command line.

Table 3.7. File name suffixes

Stage	Log File	Output Structure Files	Other Files
sdconvert	_sd.log	_sdout.mae	None
maesubset	None	_msout.mae	None
applyhtreat	_ht.log	_htout.mae	None
desalter	_ds.log	_dsout.mae	None
neutralizer	_nu.log	_nuout.mae	None
ionizer	_ion.log	_ionout.mae	Output: _ionout-bad.mae This file contains structures flagged as problematic by ionizer. These structures do not appear in the final LigPrep output structure file.
tautomerizer	_ta.log	_taout.mae	None
epik	_epik.log	_epikout.mae	None
ligparse	_lp.log	-lpout.mae	Input: filter.info. This file contains the conditions used to eliminate molecules and must follow the -f ligparse command-line option. Output: _lp.out This files contains statistics regarding the input and output sets of structures.
stereoizer	_st.log	_stout.mae	None.
ring_conf	_rc.log	_rcout.mae	Output: _rcout_tmpl_att_prob.mae This file contains copies of structures with missing ring system templates or missing attachment axial vs. equatorial energies. _rcout_misc_prob.mae This file contains copies of structures with other types of problems. ring_conf tries to include all structures in the _rcout.mae file, even those for which copies are saved in these two problem files.

Table 3.7. File name suffixes (Continued)

Stage	Log File	Output Structure Files	Other Files
premin	_pr.log	_prout.mae	Output: _prout_bad.mae This file contains structures flagged as problematic by premin. These structures do not appear in the final LigPrep output structure file.
bmin	_bm.log	_bmout.mae	Output: _bmout.ouL This file becomes _bmout.mae at the end of a successful bmin run. If bmin fails, then the structures processed up to the failure are present in the former file.
sdconvert	_sdout.log	_sdout.sd	None
smiles_to_mae	_smi.log	_smiout.mae	None

3.5 LigPrep Limitations

The limitations in this sections apply to the LigPrep process as a whole. Information on the limitations of individual tools is given in the sections for each tool.

Target Molecule Classes

LigPrep is intended for use with largely organic, drug-like molecules. While it may be useful for molecules outside this class of compounds it has not been developed for or extensively tested with such systems.

By default some stages of LigPrep pass over molecules containing more than 150 atoms. This limit can be adjusted using the -ma option.

Processing Problems May Eliminate Molecules From the Output Data Set

It may not be possible for the various stages in the LigPrep process to accommodate all input structures. Under such circumstances, these problematic structures might not be retained in the output structure file for that stage and thus might be absent in the final output structure file from the LigPrep process. To learn more about this issue, see the descriptions of the individual programs within this document.

File Size Limits

While most computers can work with large files, such as those exceeding 2 GBytes, working with them often puts heavy strains on operating systems and hardware. Maestro files of this

size can contain about 150,000 ligand-like molecules. Some LigPrep process and option selections can significantly increase the number of output structures. Given these considerations we recommend processing around 50,000 structures in a single LigPrep run.

The utility `para_ligprep` provides a convenient way to split and process a large structure file. For more information see Section 3.3 on page 40.

Using LigPrep

Although LigPrep contains a variety of options, you will probably use only a subset of them regularly. The options you choose will depend on your goals, the types of input structures you are processing, and the types of subsequent processing that will be carried out. For a complete list of LigPrep options, see Section 3.2 on page 35. A general description of how LigPrep processes structures is provided in Section 1.2 on page 2.

This chapter provides a few examples showing how to use LigPrep. The first section demonstrates the use of the LigPrep panel in Maestro, with input structure files provided in `$SCHRODINGER/macromodel-v`*version*`/ligprep/samples/examples`. A range of command-line examples is given in the second section.

In general, we recommend limiting the size of the input structure file to 20,000 to 50,000 structures. For larger input files, we recommend using `para_ligprep` to split up and process the structures in smaller batches. Some LigPrep stages do not adjust structures with more than 150 atoms, to avoid spending excessive processor time on molecules that are atypical for ligands. This limit is adjustable using the LigPrep `-ma` option. However, increasing it significantly will result in the processing of ligands larger than LigPrep is tested for, and you can encounter problems, including much slower processing of structures. For information on file size issues, see Section 3.5 on page 45.

Since LigPrep uses Schrödinger's job control facility, jobs are run in the background and the command-line prompt returns immediately after you enter the `ligprep` command. Processing continues until the job has finished, at which time the output files appear in the job submission directory. For more information on the Job Control facility, see the *Job Control Guide*.

4.1 LigPrep Panel Examples

The LigPrep panel in Maestro enables you to prepare and run a variety of ligand preparation jobs. This section provides examples of three ligands and illustrates the effect that LigPrep has on their structure.

4.1.1 Preparing to Run the Examples

Input files and sample output files are provided in the directory

`$SCHRODINGER/macromodel-v`*version*`/ligprep/samples/examples`

To prepare for the examples:

1. Create a working directory.
2. Copy the files `2D_variations.sdf` and `3D_rings.mae` from the directory

   `$SCHRODINGER/macromodel-v`*version*`/ligprep/samples/examples`

   to your working directory (*workdir*).
3. Start Maestro in your working directory.

   If you want to use an existing Maestro session instead, close the current project and enter `cd` *workdir* in the Commands text box in the main window.

4.1.2 Generating Variations on 2D Structures

The two structures provided in this example illustrate the variations that are generated with the default LigPrep options.

To import and examine the structures:

1. Click the Import structures button on the toolbar.

   The Import panel is displayed.
2. Choose SD from the Format option menu.
3. Select the file `2D_variations.sdf`.
4. Ensure that Import all structures is selected.
5. Click Import.

   The first structure should be displayed in the Workspace. This structure has two ionizable groups—a carboxylic acid and a pyridine—and a quaternary ammonium counter ion. There are three chiral centers, of which two have chirality information, and the hydrogen atoms are implicit.
6. Click Close on the Import panel.
7. To verify that this structure is indeed two-dimensional, click the Rotate around Y axis by 90 degrees button on the toolbar, then click the Reset Workspace button.

8. Display the Project Table by clicking the Open/Close project table button on the toolbar.

9. Click the In column for entry 2.

   The second structure should be displayed in the Workspace. This structure also has two ionizable groups—an imidazole ring and a 4-nitrosophenol group. The neutral state of the imidazole ring has two tautomers.

To set up and run the LigPrep job:

1. In the Project Table, choose All from the Select menu.
2. In the main window, choose LigPrep from the Applications menu.

   The LigPrep panel is displayed.
3. Under Use structures from, select Project Table (selected entries).
4. Check that the default options are selected:
   - Stereoisomers: Retain specified chiralities
   - Ionization: Generate possible states at target pH 7.0 +/- 2.0
   - Using: Ionizer
   - Generate stereoismers (maximum): 32 per ligand
   - Generate low energy ring conformations: 1 per ligand
   - Desalt
   - Generate tautomers
   - Output format: Maestro
5. Click Start to open the Start dialog box.
6. Under Incorporate, select Append new entries.
7. Type `variations` in the Name text box.
8. Click Start.

   The Monitor panel is displayed. This panel logs the progress of the job. When the job finishes, there should be 10 new entries in the Project Table.

To examine the output structures:

1. In the Project Table, click the In column for entry 3, then shift-click the In column for entry 6.

The four structures generated from the first 2D structure are displayed in the Workspace, superimposed. Note the variations in the structures and the absence of the quaternary ammonium ion, which was removed by the `desalter`.

2. Click the Tile entries button on the toolbar and click Yes in the confirmation dialog box.

The four structures are displayed in a square grid in the Workspace. Each has a full complement of hydrogen atoms. The carboxylate group is unprotonated because its pK_a lies outside the target pH range of 5 – 9. In two structures, the pyridine is protonated; in the other two it is unprotonated.

3. From the Label picked atoms button menu on the toolbar, choose Composition.

The Composition folder of the Atom Labels panel is displayed.

4. Select R/S Chirality and clear all other label options.
5. Click All in the Label atoms section.

The chiral centers are labeled R or S. Two of the chiral centers are labeled R in all structures: these chiralities were preserved from the input structures. The remaining center is labeled R in one pair of structures and S in the other. This center did not have chirality information in the input structure, and its chirality has been varied.

6. In the Project Table, click the In column for entry 7, then shift-click the In column for entry 12.

The six structures that were generated from the second 2D structure are displayed in the Workspace, superimposed. Note the variations in the structures.

7. Click the Fit to screen button on the toolbar.

8. In the Project Table panel, click the Select Only button on the toolbar.

The Entry Selection dialog box is displayed, with the Properties folder on top.

9. Choose Included State from the Properties list, then select Included (or fixed).

10. Click Add, then click OK.

    The included entries are now also the selected entries.

11. Click the In column for entry 7.

    This entry is displayed, and the other entries are undisplayed. The phenol group is protonated and the imidazole group is unprotonated in this structure.

12. Click the Next button on the Project Table toolbar.

    The next structure is displayed in the Workspace. Note the tautomerization of the imidazole group.

13. Click the Next button again.

    The imidazole group is protonated in this structure.

14. Click the Next button three more times to display the remaining structures.

    The last three structures have the phenol group deprotonated.

4.1.3 Generating Ring Conformations of a 3D Structure

The structure in this example has two flexible rings and two chiral atoms. The ring conformations will be varied, but the chiralities will be kept.

To import and examine the structure:

1. In the main window, choose Close from the Project menu.
2. Click Discard in the dialog box.

   The Workspace and the Project Table are cleared.

3. If the Import panel is not open, click the Import structures button on the toolbar.

   The Import panel is displayed.

4. Choose Maestro from the Format option menu.
5. Select the file `3D_rings.mae`.
6. Click Import.

   The 3D structure is displayed in the Workspace.

7. If you are continuing from the previous exercise, bring the Atom Labels panel to the front. Otherwise, do the following:

   a. From the Label picked atoms button menu on the toolbar, choose Composition to display the Composition folder of the Atom Labels panel.

   b. Select R/S Chirality and clear all other label options.

8. Click All in the Label atoms section to label the chiral atoms: one is R, the other is S.

To set up and run the LigPrep job:

1. If the LigPrep panel is not displayed, choose LigPrep from the Applications menu in the main window.
2. Under Use structures from, select Workspace (included entries).
3. Under Stereoisomers, select Determine chiralities from 3D structure.
4. Under Ionization, select Retain original state.
5. Type `4` in the Generate low energy ring conformations text box.

   The remaining options do not affect the outcome of this job.

6. Click Start to open the Start dialog box.
7. Under Incorporate, select Append new entries.
8. Type `rings` in the Name text box.
9. Click Start.

   The Monitor panel is displayed. When the job finishes, there should be four new entries in the Project Table, with the first of the new entries displayed in the Workspace.

To examine the output structures:

1. In the Project Table, click the In column for entry 2, then shift-click the In column for entry 5 to display the four structures in the Workspace, superimposed.
2. Click the Tile entries button on the toolbar and click Yes in the confirmation dialog box.

   The four entries are now displayed separately.

3. Choose Molecules from the Local transformation button menu on the toolbar.

4. As necessary, click on each molecule and rotate it (middle mouse button) so that the orientation of the 1,4-dioxane rings is the same in each molecule.

   The 1,4-dioxane ring adopts the chair conformation in all four molecules but is joined to the piperidine ring in different orientations.

5. Now rotate each molecule until the orientation of the piperidine rings match.

   The piperidine ring adopts the chair conformation in two molecules and the boat conformation in the other two molecules. The variation of ring conformations leads to an inversion at the N atom to generate the optimal conformation.

4.2 Command-Line Examples

The LigPrep panel in Maestro provides access to the main `ligprep` command options. For finer control of the options, you must run LigPrep from the command line. If you want to prepare a large number of ligands, you might prefer to run LigPrep from the command line. This section provides a number of examples of the `ligprep` command.

4.2.1 Default Operation

LigPrep can be run using the default settings by specifying only the input and output files. Examples follow showing syntax using different input file types; both examples include output files in Maestro format:

```
$SCHRODINGER/ligprep -isd my_2D_cts.sd -omae my_3D_cts.mae
```

In this example, the input file is in SD format.

```
$SCHRODINGER/ligprep -ismi my_1D_cts.smi -omae my_3D_cts.mae
```

In this example, the input file is in SMILES format.

The above commands do the following:

- Adds hydrogen atoms
- Neutralizes many types of charged functional groups
- Generates up to 8 tautomers for each input structure
- Generates up to 32 stereoisomers for each tautomer
- Generates one set of ring conformations for each stereoisomer
- Minimizes the energy for each stereoisomer

The resulting 3D structures are written to the output file. Only the largest molecule in each input structure is retained.[3] Some of these defaults can be overridden by specifying additional command options.

Valid combinations of chiralities, recorded as parities or bond directions in the input structure file, are retained in almost all cases. Stereoisomers are generated by varying the chiralities of chiral atoms whose chiralities are not specified. It is generally advisable to produce multiple stereoisomers, especially for fused ring systems, in which some combinations of chiralities are unacceptable and may be eliminated from the set of output structures. For Maestro-formatted input structure files the chiralities recorded as properties are respected. To vary the chiralities of all chiral atoms present, use the `-ac` option. If you use this option, it might be desirable or necessary to increase the maximum number of stereoisomers per molecule from its default value of 32 to a higher limit. For example, the option `-s 128` permits up to 128 stereoisomers to be generated.

4.2.2 Producing One Output Structure for Each Input Structure

LigPrep is primarily designed to produce high quality 3D structures and, if possible, to produce multiple output structures for each input structure by introducing variations in the chemical nature of the structures, such as adjusting the stereochemistry or the ionization (protonation) state. However, LigPrep can be used so that it typically produces only one output structure for each input structure.

4.2.2.1 Adjusting Only the Geometry

One extended-conformation, energy-minimized 3D structure can be produced for each 3D input structure from the SD file, `my_3D.sd`, using the command:

```
$SCHRODINGER/ligprep -i 0 -nt -nd -s 1 -g -isd my_3D.sd  -omae
my_improved_3D.mae
```

For the Maestro-formatted input file `my_3D.mae`, use the command

```
$SCHRODINGER/ligprep -i 0 -nt -nd -s 1 -g -imae my_3D.mae -omae
my_improved_3D.mae
```

The options selected with these commands add hydrogens to structures that lack them to produce all atom structures, but neither command should adjust the protonation state of the ionizable groups (`-i 0`). The `-nt` option instructs LigPrep not to adjust the tautomeric state, while the `-nd` option turns off salt elimination. The `-g` option requests that the chiralities, which are needed for ring conformation generation, be determined from the input geometry. If

3. LigPrep removes all molecules except the one with the largest number of atoms in the structure to ensure that co-crystallized counter ions are removed.

parts of the molecule are quite distorted, it might not be possible to determine the chiralities of all the chiral atoms unambiguously. In such cases, LigPrep systematically varies the chiralities that could not be determined to generate possible stereoisomers (up to 32 by default). The `-s 1` option instructs LigPrep to generate only one such structure, with all undetermined chiralities set to R. If there is more than one indeterminable chirality, incompatible chiralities can be selected (for instance, in fused ring systems) and the molecule could be eliminated in subsequent processing.

For 2D input structures, the corresponding command would be:

```
$SCHRODINGER/ligprep -i 0 -nt -nd -s 1 -isd my_2D.sd -omae
my_improved_3D.mae
```

This command has the same options as the previous example except that the `-g` option has been removed, and therefore LigPrep looks for stereochemical specifications in the input structure file to determine chiralities. As with the 3D examples, LigPrep normally would vary the unspecified chiralities to generate multiple stereoisomers. As before, the `-s 1` option forces the output of only one stereoisomer. However, since it is quite common for 2D structures to have multiple chiral centers unspecified, the risk of eliminating structures due to incompatible chiralities is much higher than with 3D structures. You might want to compromise and permit the generation of multiple stereoisomers and thus multiple output structures for some input structures.

4.2.2.2 Adjusting the Chemistry

LigPrep can adjust the chemistry of the input structure by removing counter ions (desalting), neutralizing or ionizing ionizable groups, changing between tautomers and generating stereoisomers. Currently it is not possible to guarantee that each input structure will produce only one output structure when adjusting the chiralities or ionization states. A command that might usefully approximate the goal of producing only one output structure per input structure is:

```
$SCHRODINGER/ligprep -s 1 -i 2 -W i,-ph,7.0,-pht,0.0 -t 1 -imae
input.mae -omae output.mae
```

As before, the `-s 1` option forces the generation of only one stereoisomer, and chiral atoms whose chiralities were not previously defined are classified as R. To avoid eliminating molecules in which such chiralities are incompatible, a practical compromise is to permit the generation of several stereoisomers (e.g., `-s 4` to generate up to 4). The desalter is run by default, so only the largest molecule in each structure is retained. The `-t 1` option results in the generation of the tautomer judged to be the most probable one.

The `-i 2` option requests the running of the `ionizer`, and options are passed to it by means of the `-W i,-ph,7.0,-pht,0.0` option. These options request that functional groups with pK_a values less than or equal to 7.0 be deprotonated, while those with pK_a values greater than or

equal to 7.0 be protonated. For most input structures, this results in only one output structure. However, for molecules that have functional groups with pK_a values of exactly 7, multiple structures are produced, each with a different protonation state.

4.2.3 Generating Variations

Often it is desirable to generate a number of variations on each input structure. The main benefit is that there is a better chance that the resulting collection of structures includes the "best" form for your particular study. However, as more variations are generated, proportionately more CPU time is consumed, both in LigPrep and in subsequent processing. Another drawback of generating more variations is that improbable states can be generated, which at present are not penalized in subsequent processing.

There are many ways to introduce variations in the structures using LigPrep. Here are a few examples showing how to do this.

4.2.3.1 Generating a Small Set of Low-risk Variations

To generate a few low-risk variations on the input structures:

```
$SCHRODINGER/ligprep -r 1 -s 8 -t 4 -i 2 -W i,-pht,1.0 -imae
input.mae -omae output.mae
```

For each structure in the input structure file, LigPrep generates an extended conformation containing only the lowest energy ring conformation for any rings present, up to 8 stereoisomers, up to 4 tautomers, and ionization states highly populated in the pH range 6.0 to 8.0 (7.0 1.0). Typically, far fewer than 8 stereoisomers and 4 tautomers would be produced.

This command assumes that the available information on the stereoisomers is recorded as R/S properties in the input structure file. If the input structures are 3D and have the correct chiralities, you could use the `-g` option to instruct LigPrep to respect the chiralities present in the 3D geometry.

4.2.3.2 Generating Many Variations

To generate many variations:

```
$SCHRODINGER/ligprep -r 4 -s 64 -t 8 -i 2 -W i,-pht,2.0
-imae input.mae -omae output.mae
```

This command generates many variations for ring conformations, stereoisomers, tautomers, and ionization states. Not all variations are necessarily kept in the output: some may be eliminated because they have poor bonding patterns, others because they lie too high in energy. In a typical LigPrep run with these options, the number of structures increases by about a factor of 8. The actual expansion factor might be smaller or larger, depending on your data set.

4.2.3.3 Using Meta-Options for Convenient Selection of Run Characteristics

For some scenarios, a set of nondefault options is needed. Rather than having to understand and specify each option separately, you can use the `ligprep` command's more intuitive meta-options, which include:

`-qik`	Prepare structures for processing with QikProp (see Section 4.2.4 on page 58)
`-unt`	Untangle structures without changing any atoms or bonds.
`-adjust_`*itc*	Adjust to a suitable state.
`-vary_`*itc*	Generate different states.
`-expand_`*itc*	Aggressively generate different states.
`-retain_`*itc*	Retain characteristics of the input structures. Specifying `i` also turns on retention of tautomers.

itc can be replaced by one or more of:

- `c` chiralities
- `i` ionization and neutralization
- `t` tautomerization

The `adjust`, `vary` and `expand` meta-options control the range of structures generated. All three of these meta-options may be used in the same `ligprep` command, but each of `c`, `i` or `t` must only appear in one of them: for instance, you could specify `-adjust_t -vary_c -expand_i`. See Table 3.3 for details on the action of these meta-options. The `retain` meta-option requires that at least one structure produced for each input structure retains the characteristics specified by the suffix. For instance, `-retain_t` reqires the tautomerization state in the input structure to be present in at least one of the output structures for that molecule. `retain` can be used with any of `adjust`, `vary`, and `expand`. `retain` can be used to produce a range of structural variations while keeping the form in the input structure—something that might be useful in lead optimization.

As an example, the following command permits the generation of moderate variations on the ionization and tautomerization state and significant numbers of stereoisomers, while ensuring that the input tautomeric form appears in the output:

```
$SCHRODINGER/ligprep -vary_it -expand_c -retain_t
    -imae input.mae -omae output.mae
```

This command is equivalent to:

```
$SCHRODINGER/ligprep -i 2 -W i,-pht,1.0 -r 1 -t 2 -s 32
    -imae  input.mae -omae output.mae
```

If other options are given with a meta-option, their values override the values set by the meta-option. For instance,

```
$SCHRODINGER/ligprep -vary_it -expand_c -retain_t -W i,-pht,1.5
    -imae input.mae -omae output.mae
```

extends the pH range from that used by `-vary_i`, 1 (pH 6-8), to 1.5 (pH 5.5-8.5).

4.2.4 Preparing Ligands for QikProp

LigPrep can be used to prepare ligand structures for processing with QikProp by using the `-qik` meta-option, as shown in the following command:

```
$SCHRODINGER/ligprep -qik -isd my_2D.sd -omae my_3D.mae
```

This command creates all-atom, 3D, energy-minimized structures with uncharged functional groups, produced by adding or removing protons where possible. Up to 8 tautomers are generated for each input structure and up to 32 stereoisomers are generated for each tautomer by default, but only one ring conformation is generated for each stereoisomer. The `-t`, `-s`, and `-r` options can be used to adjust the number of tautomers, stereoisomers, and ring conformations generated.

4.2.5 Filtering Structures

During LigPrep processing, structures matching certain conditions can be removed using `ligparse`. A file containing a list of these conditions must be created before running LigPrep. For example, to remove molecules with a molecular weight greater than 650 and molecules that have too many hydrogen bond acceptors, you could create a file with the following content:

```
##Remove Molecules that have a molecular weight of more than 650 AMU.
Molecular weight >650.0
## Remove molecules with too many H-bond acceptors
Num neutral acceptor groups       >3.0000
Num charged acceptor groups       >3.0000
```

The following command uses LigParse to filter the input structures using the above file, saved as `filter.info`:

```
$SCHRODINGER/ligprep -f filter.info -isd my_2D.sd
-omae my_3D.mae
```

For more information on using `ligparse` and creating filtering files, see Chapter 8.

4.2.6 Running LigPrep on a Remote Host

One advantage of the job control facility is that jobs can be submitted to remote hosts that have been prepared to run Schrödinger software. For instance, to run LigPrep on a host called `fast_cpu`, you can add the `-HOST` option to a command:

```
$SCHRODINGER/ligprep -HOST fast_cpu -s 1 -imae my_2D.mae -omae
my_3D.mae
```

When the LigPrep job finishes on `fast_cpu`, the output files are copied back to the directory from which you started the job. For information on the Job Control facility and on preparing hosts and batch queues to run Schrödinger software, see the *Job Control Guide*.

4.3 Chirality Manipulations Using Maestro and LigPrep

Both Maestro and LigPrep have the ability to manipulate the stereogenic composition of a three-dimensional molecular structure. Maestro includes graphical features that can exchange the chirality of a single center, as well as easily generate the enantiomer of a displayed structure. Correspondingly, a LigPrep preparation has the ability to sample some or all of the chiral centers, in one or multiple input structures, during a given preparative computation. Finally, the combination of the Project Table and LigPrep enables the expansion or inversion of selected subsets of chiral centers in one or more structures easily and quickly.

Simple inversions of a single chiral center, located in a structure displayed by Maestro, can be easily accomplished with Maestro's Adjust Chirality feature. To open this panel, choose Chirality from the Adjust submenu of the Edit menu in the main window. To reverse the chirality of a given stereogenic center, just select the chiral atom, followed by selecting two other "non-moving" atoms connected to the chiral center. The remaining two groups connected to the selected chiral atom are switched, resulting in an effective inversion of the chirality at the atom. This operation can only be performed on chiral atoms that are not a part of a ring system. To invert the chirality of an individual ring system atom, use LigPrep as described below.

The same Adjust Chirality feature also enables the generation of the corresponding enantiomeric structure. Click All in the Chirality folder of the Adjust panel to generate the mirror image of the structure in the Workspace. This operation inverts all the chiral centers and produces the enantiomer. The z-coordinate values for each atom in the structure are inverted by this procedure.

In addition to Maestro's built-in graphical tools, which modify individual chiral centers and produce enantiomers, LigPrep can also generate and sample alternative chiralities on a broader scale. Generally, in addition to 2D to 3D conversion, LigPrep can sample and expand ionization states, tautomers, ring conformations, and alternative chiralities. For the latter, both

Maestro-formatted files and SD-formatted files have the internal ability to indicate the specified chirality at some or all stereogenic centers, usually "R" or "S". LigPrep and the Stereoizer have three basic usage modes that utilize this information:

- Retain specified chiralities—For structures with *specified* chirality information, LigPrep attempts to generate output structures that match the chirality indications at all such stereogenic centers. LigPrep tries to match the specification whether or or not the input 3D geometry about the chiral centre (if any) matches this specification. All unspecified chiral centers in the structure will be expanded.
- Determine chiralities from 3D structure—Specified chiralities are ignored, and no alternative chiralities are generated. Output structures have the same stereogenic makeup as the input, and all chiral centers have their specification changed to match the input structure.
- Generate all combinations—The input specifications are ignored, and and geometrically feasible stereoisomers (up to the maximum number requested) are generated.

The specifications discussed can be observed in the Project Table as properties with titles such as Chirality 1 and specifications like `9_R_8_13_10_37`. In this example, atom number 9 has the "R" configuration based on the listed atoms with CIP priorities. MDL SD files have their own chirality-indicating format, which is converted to the Maestro format upon importing the SD-formatted structures.

If the chirality specifications were not included in the original structure file, or the molecule was constructed in Maestro, the specifications can be easily assigned by simply running LigPrep with the Stereoizer set to Determine chiralities from 3D structure, and setting the output to Replace existing entries or Append new entries. You may select to turn off the Ionizer, Tautomerizer, and Desalter if you prefer no additional expansion when only specifying chiralities. After incorporation, the chirality specifications are listed in the Project Table.

Once the chirality specifications are listed in the Project Table for the structures of interest, the combination of the Project Table and LigPrep provide a simple, easy method to invert or sample a subset of chiral centers in one or more structures.

To invert a chiral center using LigPrep, simply adjust the specification in the appropriate property cell for the structure of interest to the other "R" or "S" indication, for instance from `9_R_8_13_10_37` to `9_S_8_13_10_37`. You may want to duplicate the entry in the Project Table prior to adjusting the designation. Using the modified entry in a subsequent LigPrep computation with the Retain specified chiralities mode, generates the requested chirality change. The user can make changes to multiple structures by modifying any of the specified chiralities, selecting all of the modified structures in the Project Table, and running the LigPrep preparation with Use structures from set to Selected entries.

In contrast, to sample or expand alternate chiralities once the chiral centers are specified, simply delete or remove the specification from the Project Table cell. A subsequent LigPrep preparation, using the Retain specified chiralities mode, maintains the other, indicated chiralities, but expands the chiralities of the centers with specifications removed. Multiple structures can be expanded in this way using selected entries as the input.

Tools Used by LigPrep

This chapter describes the tools used by LigPrep. A description of the purpose, the syntax, and the options is provided for each tool. The options for the tools cannot be used directly as options for `ligprep`. The `ligprep` options can be used to control the behavior or the tools, and for some tools, options can be passed from the `ligprep` script to the tool with the `ligprep -W` option. The descriptions are provided so that you can understand how these tools function in the context of LigPrep, and so that you can use these tools separately from LigPrep.

LigPrep can use a separate product, Epik, instead of the `ionizer` and the `tautomerizer`. For more information on Epik, see the *Epik User Manual*.

5.1 sdconvert

This utility converts between MDL SD, Maestro, and MacroModel format files.

Syntax:

`$SCHRODINGER/utilities/sdconvert` [*options*]
`{-isd|-imm|-imae}` *inputfile* `{-osd|-omm|-omae}` *outputfile*

`-isd`	Input file in SDfile format
`-imm`	Input file in MacroModel format
`-imae`	Input file in Maestro format
`-osd`	Output file in SDfile format
`-omm`	Output file in MacroModel format
`-omae`	Output file in Maestro format

Options:

`-a`	Append structures to the output file.
`-all`	Convert all structures (synonym for `-n 1:`). This is the default.
`-h`	Show usage summary.

`-n` *structs* Specify a comma-delimited series of colon-separated ranges and single values to convert, for example:

`1:10,14`	Structures 1 through 10 and 14
`2:`	Structure 2 through end of file
`:5,13:18`	Structures 1 through 5, and 13 through 18

`-noarom` Do not convert aromatic type 4 bonds to single and double bonds, according to Maestro and OPLS conventions.

`-nolewis` Do not fix bond orders and formal charges of nitro and carboxylate groups as appropriate for OPLS use.

`-st` When files are converted from SD to Maestro format, convert parity and bond-direction information into chirality properties, which are recorded in the structures produced. See Section 5.10 on page 70 (`stereoizer`) for more information on chirality properties.

`-title` *prop* For SD to Maestro conversion, define SD property *prop* as the source for assigning the Maestro title.

`-v` Show version number.

If multiple `-n` options are given, the rightmost range takes precedence. The `-all` option overrides any `-n` settings.

Example:

```
$SCHRODINGER/utilities/sdconvert -n 1:500 -isd lig.sdf -omae
lig.mae
```

5.2 smiles_to_mae

This utility converts a SMILES file to a Maestro-formatted file. This utility should be used in conjunction with LigPrep.

Syntax:

`$SCHRODINGER/utilities/smiles_to_mae` [*options*] *inputfile*`.smi` *outputfile*`.mae`

The first file will be taken as the input file and the second as the output file.

Options:

`-h` Show usage summary.

`-v` Show version number.

5.3 maesubset

This utility selects a subset of the structures present in a Maestro-formatted file. The selected structures are printed on standard output.

Syntax:

`$SCHRODINGER/utilities/maesubset` [*options*] *input_file*`.mae >` *output_file*`.mae`

Options:

`-h`	Show usage summary.
`-n` *range*	Specify a comma-delimited series of colon-separated ranges and single values to process. For example: `1:200` Structures 1 through 200 `50:` Structures 50 through end of file `1:5,14` Structures 1 through 5 and 14
`-v`	Show version number.

5.4 applyhtreat

The `applyhtreat` utility can be used to delete or add hydrogen atoms and lone pairs to one or more structures in a Maestro-formatted file, consistent with the hydrogen treatment option that you provide. A *hydrogen treatment* is a protocol that determines which atoms are to have hydrogens and lone pairs attached. Several treatments are supplied. Each of these treatments is associated with a particular molecular mechanics force field, but some treatments are suitable for several force fields. See Section 4.8 of the *Maestro User Manual* for information on the relationship between force fields and hydrogen treatments.

Syntax:

`$SCHRODINGER/utilities/applyhtreat` *input-file* *output-file* [*treatment-name*] [*ASL-expression* [*adjust-chirality*]]

The treatment name must be one of the following names, and must be enclosed in quotes:

```
All-atom with Osp3/Nsp3-Lp
All-atom with No-Lp
Csp3 United-atom with S-Lp
Csp3 United-atom with No-Lp
All-atom with S-Lp
```

```
C sp2/sp3 United-atom with No-Lp
C sp2/sp3, N,O,S United-atom with No-Lp
```

Default treatment: `All-atom with No-Lp`

The ASL expression is a Maestro Atom Specification Language expression for selecting a subset of the atoms in each structure processed. See the *Maestro Command Reference Manual* for information on ASL expressions.

The *adjust-chirality* argument can be set to 0 if the chirality information should not be adjusted, or to 1 for adjustment of chirality information. For LigPrep applications the chirality information should be adjusted. To use this option you must provide an ASL expression.

Example:

```
$SCHRODINGER/utiltities/applyhtreat input.mae output.mae
"All-atom with No-Lp" all 1
```

applies the `"All-atom with No-Lp"` treatment to all atoms and adjusts the chirality information.

Notes:

- Hydrogen atom addition or deletion takes place only as necessary to make the structure consistent with the selected hydrogen treatment. For example, if the specified treatment option calls for all atoms to have hydrogen atoms attached and the structure in the input file already has hydrogen atoms attached to all atoms, no changes are performed.
- For LigPrep processing it is important to request adjustment of chirality information, otherwise the atom numbers in the chirality properties remain unchanged. This can result in corruption of the chirality specifications and lead to incorrect results.

5.5 desalter

Individual structures present in many databases contain multiple molecules. In many cases these are co-crystallized components such as counter ions. These components often need to be removed before further calculations are performed. The `desalter` eliminates all molecules except the molecule with the largest number of atoms (including hydrogen atoms, if present) in each structure. If there is more than one molecule containing the largest number of atoms, the molecule with the lowest numbered atom within the structure is retained.

You must have a LigPrep license to run the `desalter` .

Syntax:

`$SCHRODINGER/utilities/desalter [-h] [-v]` *input_file*`.mae` *output_file*`.mae`

Options:

`-h` Show usage summary.

`-v` Show version number.

5.6 neutralizer

Ionizable groups present in structures could be in any of a number of charge states. Some programs, such as QikProp and the `ionizer`, do not successfully process input structures containing charged groups or molecules. The `neutralizer` neutralizes functional groups where possible by adding or removing protons. The following replacements and additions are made in the neutralization process:

- Sodium or potassium ions with a single bond are replaced by hydrogens bound to the same atom.
- $(R)_x$–NH^+ is replaced by $(R)_x$–N.
- A proton is added to O^- or S^- atoms bound to C, S or P atoms.
- A proton is added to O^- when it is bound to a N that has a formal charge of 0. Thus, nitro groups and pyridine N-oxides are not protonated, but a hydroxamate is protonated.
- Dummy atoms are removed.
- A proton is removed from H–O^+R_2.
- A proton is removed from H–O–N^+(=O)–R and an electron to O–N^+(=O)–R, to produce O^-–N^+(=O)–R.
- A proton is removed from a protonated pyridine N-oxide.
- A proton is added to the nitrogen in sulfonamide O=S(=O)-N^- anions.
- A proton is added to the negatively charged N atom in histidine anions.

You must have a LigPrep license to run the `neutralizer`.

Syntax:

`$SCHRODINGER/utilities/neutralizer [-d] [-h] [-v]` *input_file*`.mae` *output_file*`.mae`

Options:

`-d`	Remove dependency information for chiralities. For use only prior to an ionizer treatment.
`-h`	Show usage summary.
`-retain`	Retain the input structure.
`-retain_lab`	Retain the input structure and add a label property to the retained input structure that identifies it as the input structure.
`-v`	Show version number.

Limitations:

Some groups, such as quaternary N atoms, are not neutralized.

5.7 ionizer

The `ionizer`, or ionization state expander, produces multiple structures for each input structure with different combinations of ionized states based on the ionizable groups present. By default, the `ionizer` produces both protonated and deprotonated forms of groups that have a pK_a between 5 and 9. Groups with a higher pK_a are protonated; groups with a lower pK_a are deprotonated. These defaults ensure that the appropriate structures are available for physiological conditions. For more information, see Chapter 6.

The `ionizer` is not called from the `ligprep` script by default. To run the `ionizer` with LigPrep, select Generate Possible States under Ionization in the LigPrep panel, or add the `-i 2` option of the `ligprep` command. The `ionizer` is also called from `ligprep` if you use the `-expand_itc` option. The `ionizer` can be run on its own from the `ligprep` script with the `-R i` option. You can pass options to the `ionizer` from `ligprep` with the `ligprep -W i` option. Not all `ionizer` options are available from `ligprep`; see Section 6.1 on page 81 for more information.

If an input structure contains more than a prescribed number of ionizable groups, the structure is skipped and passed, unchanged, to the output file. To set the limit on the number of ionizable groups, use the `-mg` option of the `ligprep` command. You cannot set the limit to a value greater than 31. The default for running the `ionizer` from `ligprep` is 10.

5.8 tautomerizer

Some isomeric forms of molecules coexist and are interchangeable under physiological conditions. Tautomers constitute an important class of such isomers. For some types of calculations, such as docking with Glide, considering the appropriate tautomeric forms of ligands is nearly as important as examining the appropriate ionization state. LigPrep uses the `tautomerizer` by default to generate probable tautomeric states. To skip the generation of tautomers, clear the Generate Tautomers option in the LigPrep panel of Maestro, or use the `-nt` option of the `ligprep` command.

The `tautomerizer` assigns a probability to each tautomeric form. The tautomers for a given input structure are produced in order of probability, the most probable forms first. By default, up to eight tautomers are generated for each structure. The maximum number of tautomers to generate for each structure can be specified with the `-t` *number* option of the `ligprep` command. In addition, the generation of improbable tautomers can be skipped by setting a threshold probability with the `-tp` *number* option. Only tautomeric forms with overall probabilities higher than this threshold will be produced. The default threshold is 0.01. If all forms have probabilities lower than the threshold, the most probable tautomer is still recorded to avoid eliminating that molecule completely from further consideration.

You must have a LigPrep license to run the `tautomerizer`. For more information on tautomers and the `tautomerizer`, see Chapter 7.

5.9 ligparse

`ligparse` can be used to characterize molecular databases, to select molecules meeting certain criteria, to eliminate molecules meeting certain criteria, or to select subsets of the molecules present possessing certain statistical characteristics.

LigPrep uses `ligparse` to filter molecules, usually to eliminate those matching certain conditions (e.g., molecular weight > 650 AMU), although other types of filtering are possible. This section briefly documents `ligparse` as it is used within LigPrep. For a more complete description, see Chapter 8.

`ligprep` does not call `ligparse` unless `ligprep` is given the `-f` option, which accepts the name of a file containing criteria to use for filtering. For example, to remove molecules with a molecular weight greater than 650 and molecules that have too many hydrogen bond acceptors you could create a file with the following content:

```
##Remove Molecules that have a molecular weight of more than 650 AMU.
Molecular weight >650.0
## Remove molecules with too many H-bond acceptors
Num neutral acceptor groups       >3.0000
Num charged acceptor groups       >3.0000
```

Note: The numeric arguments should be floating point values. The following command uses `ligparse` to filter the input structures using the above file, saved as `filter.info`:

```
$SCHRODINGER/ligprep -f filter.info -isd input.sd
-omae output.mae
```

When `ligparse` is called from `ligprep` the `-any` and `-j 5` options are passed to `ligparse` by default. With these options, if any of the criteria given (`-any`) match for a particular molecule, the molecule is eliminated (`-j 5`) from processing. If options are passed to `ligparse` using the `ligprep -W` option, the `-any` and `-j 5` options are no longer passed by default, so that other filtering strategies can be used, such as retaining only the molecules that match the criteria specified.

The output files from `ligparse` are renamed by LigPrep. The `_cull.mae` file is renamed `_lpout.mae` and the `.out` file is renamed `_lp.out`.

You must have a LigPrep license to run `ligparse`.

5.10 stereoizer

The `stereoizer` has two modes: label mode and stereoisomer mode. In label mode, various stereochemical features in the input structures are labeled, such as atom chiralities (R/S) or double-bonded atom conformations (E/Z for groups of atoms connected by an odd number of bonds or P/M for groups of atoms connected by an even number of bonds). In stereoisomer mode, the `stereoizer` generates multiple output structures for each input structure, based upon the chiral properties of the chiral atoms in the structure. The coordinates of the atoms are not modified in the copies; only the chiral properties included in the structure for the chiral atoms are distinct. The structures produced in stereoisomer mode should be processed by a LigPrep-licensed version of MacroModel to adjust the atomic coordinates to match those required for the various stereoisomers and to eliminate combinations of chiralities that are very unfavorable. To use stereoisomer mode, set the `-n` option described under Options on page 71. To use the label mode, omit the `-n` option.

In stereoisomer mode, the chiralities of the atoms specified in the input structure or that can be determined from the geometry are retained. Only the chiralities of the other atoms are varied. If the `-a` option is specified, the input chiralities are ignored and the chiralities of all chiral atoms are varied as follows. The assigned chiralities are all set to R in the initial stereoisomer and systematically varied until either all possibilities are sampled or until the maximum number of stereoisomers per input structure is reached.

When LigPrep calls the `stereoizer`, it uses `-n 32`. If you are running LigPrep with more than 5 chiral centers in any structure and want to generate all combinations, use the `-s` option

of the `ligprep` command to specify the number of stereoisomers, or the `-n` option of the `stereoizer` command.

You must have a LigPrep license to run the `stereoizer`.

Syntax:

`$SCHRODINGER/utilities/stereoizer [`*options*`]` *input-file*`.mae` *output-file*`.mae`

Options:

`-a`	Overwrite existing chirality properties.
`-g`	Overwrite existing chirality properties. Assign chiralities using geometry where possible. This option is only active when used with the `-n` option.
`-h`	Show usage summary.
`-n` *number*	Generate up to this many stereoisomers (stereoisomer mode). If omitted, existing stereoisomers are labeled and no new stereoisomers are generated (label mode).
`-verb` *level*	Controls `MMERR` level of reporting. 0 `MMERR_FATAL`—only report errors. 1 `MMERR_WARNING`—only report errors and warnings. 2 `MMERR_INFO`—report errors, warnings, and processing information. 3 `MMERR_DEBUG`—report errors, warnings, processing information, and debugging information.
`-v`	Show version number.

5.10.1 Stereochemical Properties

Stereochemical properties created and recognized by the `stereoizer` include those related to atom chiralities and to the conformations of doubly bonded series of C and N atoms.

Chiral atom properties have the form:

A_N_C_N_1_N_2_N_3_N_4

where:

- A_N is the atom number for the chiral atom.
- C specifies the chirality of atom A_N and is either ‘R’ or ‘S’.
- N_1 through N_4 give the atom numbers of the atoms attached to atom A_N in decreasing

order of their Cahn-Ingold-Prelog (CIP) priorities. If only 3 atoms are bonded to atom A_N, then $_N_4$ is missing.

For example, the property 7_S_15_1_8_62 indicates that atom 7 has a 'S' chirality and that the four atoms attached to it in order of decreasing CIP priority are: 15, 1, 8, and 62.

The isomers of doubly bonded series of C or N atoms with distinct attachments at each end are designated with properties of the form:

$N_1_N_2_...._N_i_G$

where:

- N_2 through N_{i-1} are the atom numbers of the doubly bonded series of atoms in the order in which they are bonded together.
- N_1 and N_i are the highest priority atoms bonded to N_2 and N_{i-1} respectively, according to the CIP rules.
- G is a character used to designate the overall geometry. If there is an even number of atoms specified then all the atoms are coplanar with N_1 and N_i lying on either the same, G='Z', or opposite, G='E', sides of the doubly bonded atoms. If there is an odd number of atoms then N_1 and N_i lie roughly perpendicular to each other. If on moving from one end of the series of doubly bonded atoms to the other the highest priority atom lies counter clockwise then G='P'. If they lie clockwise then G='M'.

In problematic cases, C or G appear as "???" if the structure was distorted enough to prevent unambiguous designation of the stereochemistry.

5.10.2 Known Limitations

The `stereoizer` is a fairly strict implementation of the standard CIP rules for chiralities and double-bond geometries as described in Prelog, V.; Helmchen, G. Basic Principles of the CIP-System and Proposals for a Revision. *Angew. Chem., Int. Ed. Engl.* **1982,** *21,* 567-583, except that prioritization of atoms by chiralities is not implemented. For instance, for prioritizing groups attached to the terminal atom of double bonded systems of atoms, if the groups are otherwise equivalent the CIP rules indicate that they may be prioritized if they differ in the chiralities of atoms within them. The `stereoizer` does not do this.

In stereoisomer mode, no attempt is made to avoid combinations of chiralities that are not achievable. For instance, in bicyclic systems the atoms connecting the rings are often chiral and often only specific combinations of their chiralities are found in nature. The `stereoizer` does not attempt to distinguish realistic from unrealistic chiral atoms combinations. The unrealistic combinations can be largely eliminated in the `bmin` stage of LigPrep processing.

5.11 ring_conf

It can be difficult, or at least computationally demanding, to find low-energy conformations for the flexible ring systems within a structure. The `ring_conf` tool attempts to do so by identifying ring systems in the structure, then seeking suitable ring templates from its collection, and using these templates to generate good ring conformations.

Briefly, `ring_conf` examines the molecule to identify groupings of flexible rings, referred to as ring systems, that require a template. Many aromatic ring systems are recognized as such and no template will be sought for them and the input geometry will be used. However, aromatic rings within ring systems containing flexible rings need special attention. If they share 2 or more atoms with a flexible ring then they influence the conformation of the flexible ring and thus must be considered part of the ring system for template matching. However, if they do not share 2 or more atoms with a flexible ring they are not included in ring system that must be matched by a template. In this way, ring systems in which flexible rings are connected by non-flexible rings that do not overlap with flexible rings may be split into separate ring systems for templating purposes. Once the process of identifying portions of the molecule that require ring system templates are found, templates are sought separately for each such portion. Ring systems for which templates cannot be found retain the input geometry for the ring system. Template matching has some flexibility in that enantiomers may be matched and protonation states of sp^2 or sp^3 nitrogen atoms need not match. The template library has more than 350 templates, many of which have multiple conformations. Templates in the library include inversions of non-quaternary sp^3 nitrogen atoms in their collections of conformations.

After the template identification for the ring systems is complete, the conformational energies that depend on the ring conformation are estimated in a crude manner. The energy of each ring system is estimated separately based upon the ring conformation energies recorded in the template database, the axial-equatorial energies of the attachments and a pair-wise short range repulsion between base atoms for each attachment (the atoms directly bonded to atoms in the rings). The total ring system energy for a particular selection of conformations for all ring systems in a structure is obtained by adding together the energies for these conformations in each ring system.

By default, only the structure with the lowest estimated energy for each input structure is written to the output structure file. However, multiple structures with different ring conformations can be saved (see options `-e`, `-n`, and `-N`) using different selection criteria (see option `-c`).

While `ring_conf` is focused on providing conformations for the flexible ring systems in the structure provided, it can also attempt to adjust the conformations of portions of the structures outside such systems, depending on the linear generation method (see the `-l` option below). We generally recommend using `-l 3`, which instructs ring_conf to adjust the linear portions of the molecules so that they adopt extended conformations.

You must have a LigPrep license to run `ring_conf`.

Syntax:

`$SCHRODINGER/utilities/ring_conf [`*options*`]` *input-file*`.mae` *output-file*`.mae`

Options:

`-c` *conf_method*
- `1` Produce conformations in order of increasing energy (default).
- `2` Always provide the lowest energy conformation.
- `3` Select conformations at random.
- `4` Provide conformations with a probability proportional to their Boltzmann weight.
- `5` Systematically provide ring conformations.

`-e` *energy_limit* Set the maximum energy (kJ/mol) relative to the lowest energy ring conformation for output conformers (default 50.0 kJ/mol). This option sets *num_conf* to 8 unless the `-n` option is used to specify another value.

`-h` Show usage summary.

`-l` *lin_method* Non-flexible ring generation method to use.

- `1` Leave as it was in the input structure (default).
- `2` Reconstruct all portions of the molecule (can lead to distorted rigid rings).
- `3` Generate only linear parts of the molecule. and use original geometry for the rigid rings.

Note: `ligprep` passes a value of 3 to `ring_conf` for this option.

`-N` *num_conf* Produce exactly *num_conf* (see `-n`) conformers. Conformations will be repeated if necessary.

`-n` *num_conf* Produce up to this many conformations for each input conformation (default 1).

`-t` *temperature* The temperature to use in Boltzmann weighting.

`-v` Show version number.

`-verb` *number* Controls `MMERR` level of reporting.

- `0` `MMERR_FATAL`—only report errors
- `1` `MMERR_WARNING`—only report errors and warnings
- `2` `MMERR_INFO`—report errors, warnings, and processing information.
- `3` `MMERR_DEBUG`—report errors, warnings, processing information, and debugging information

Additional output files:

*jobname*_rcout_tmpl_att_prob.mae	Structures for which either templates were missing or the relative energies of axial vs. equatorial geometries for attachments were missing.
*jobname*_rcout_misc_prob.mae	Structures for which ring_conf had other types of problems.

Limitations:

Missing Templates: The collection of templates for flexible ring systems is incomplete. In our trials, typically ring_conf did not find templates for all ring systems present for 8–25% of the valid input structures. However, we expect that the rate of misses will vary dramatically. Flexible ring systems for which templates are not found are simply treated by ring_conf as though they were rigid. At present, ring_conf does not support the ability to add templates.

Attachment Geometries for Missing Templates: Under some circumstances, attachments to flexible ring systems for which templates were not found are placed on top of each other if a value of 2 or 3 is given for the -l option. However, these geometries usually do not cause problems in subsequent processing by premin and bmin within LigPrep protocol.

Misidentification of Smallest Set of Smallest Rings: In very rare instances, the software that identifies the rings within ring systems, particularly fused ring systems, omits some rings. In such cases, it may not be possible to find a suitable template even if one is present.

Overlapping Atoms: ring_conf generates low-energy ring conformations using criteria that depend on features which are local to individual ring systems. The relative positions of portions of a molecule that are not directly associated with a single ring system are not taken into account, and in some instances may lie too close to each other. Within the LigPrep context the follow-up minimization and optional conformational search using MacroModel are expected to eliminate nearly all such problems.

5.12 premin

premin is used in filter mode in LigPrep to eliminate structures that could cause severe problems when processed by Schrödinger applications. Ligands that are processed successfully are written to the output structure file, which by default is called *jobname*-min.mae, while problematic structures are stored in *jobname*-bad.mae. In the LigPrep process, these files are named *jobname*_prout.mae and *jobname*_prout_bad.mae.

When `premin` is run using a LigPrep license, one MacroModel command file is created, named `filter.com`. Limited customization of the filtering process is possible with a modified `filter.com` file if `premin` is run directly from the command line and the `-custom` option is specified. When `premin` is run using a LigPrep license, only a restricted subset of the internal MacroModel commands (*opcodes*) is supported. The following section lists these opcodes.

Syntax:

`$SCHRODINGER/utilities/premin` [*options*] *input*`.mae`

where *input*`.mae` is the input file containing ligand structures.

Options:

`-custom`	Use custom `.com` files.
`-doc`	Show a brief guide to using `premin`.
`-f 11`	Use OPLS_2001 instead of the default force field (MMFFs).
`-f 14`	Use OPLS_2005 instead of the default force field (MMFFs).
`-filter`| `-m filter`	Filtering only (skip minimization).
`-h`	Show usage summary.
`-lic ligprep`	Use LigPrep license instead of MacroModel license.
`-s` *output*	Output file of successfully minimized structures. Default is *input*`-min.mae`.
`-u` *bad*	Output file of structures that could not be minimized. Default is *input*`-bad.mae`.
`-v`	Show version number.

5.13 MacroModel (bmin)

LigPrep uses MacroModel to enforce the chiral properties listed in each structure while producing an energy-minimized structure. When run under a LigPrep license, MacroModel has a very limited set of commands enabled. These commands, or *opcodes*, are targeted specifically for LigPrep use.

The `CHIP` opcode (see below) is used to instruct MacroModel that this calculation is being performed to assist LigPrep. The overall process is formally a serial MCMM search. Each structure is processed separately and sequentially from the input structure file. First the structure is read in and a small random displacement is made to separately to each atom present. Up to 4 improper torsions are placed on each atom indicated as chiral by a chiral property in the

structure to strongly encourage the atom to adopt the chirality specified. This structure is then energy minimized. Then the improper torsional potentials are removed and a second minimization conducted. If the resulting structure has bonds close to their equilibrium length the conformational search is skipped. Otherwise a short conformational search is conducted and the lowest energy structure from the search retained. The same two stage minimization with and without the additional improper torsions is used for each conformation generated during the search. This optional search is used to attempt to correct distorted structures, such as those with connected atoms passing though the central portions of small rings. The lowest energy structure from either the full or truncated search is then checked to see if the requested chiralities were obtained, all bonds lie within 0.25 Å of their ideal length, and all bond angles are greater than 10 degrees. If the structure passes all of these tests then it is added to the output structure file. If not, by default, it is rejected and no structure is recorded in the output structure file. This strategy is used to eliminate unrealistic combination of chiralities, particularly in fused ring systems, which may be generated by the stereoizer.

See the *MacroModel Reference Manual* for `bmin` syntax.

5.13.1 MacroModel Opcodes Enabled for LigPrep

The following opcodes are enabled under a LigPrep license: `READ`, `MINI`, `FFLD`, `BGIN`, `END`, `CONV`, `EXNB`, `MCOP`, `DEBG`, `NANT`, `MCMM`, `MCSS`, `MCNV`, `SOLV`, `BDCO`, `MSYM`, `SPAT`, and `AUTO`. See the *MacroModel Reference Manual* for details. In addition, the `CHIP` opcode, described below, is available only with a LigPrep license.

CHIP — CHIrality Protocol

`CHIP` instructs MacroModel to adjust its behavior appropriately for enforcing chiralities and avoiding common pitfalls encountered for structures that were originally 2D. `CHIP` is intended to be used within a `.com` file that would otherwise be a fairly typical MCMM serial search.

`CHIP` modifies the behavior of the `READ`, `MCMM` and `MINI` opcodes. When `CHIP` has been specified, structures read in with the `READ` opcode are examined for chiral atom properties. For each chiral atom specified up to 4 improper torsional potentials are added to encourage the atom to adopt the chirality requested. In addition, each atom in the structure is displaced by a small random displacement to encourage the system to leave saddle points during minimizations.

The behavior of the `MCMM` opcode is tightly coupled with the `MINI` opcode. In a serial search, when `CHIP` has been specified, the search proceeds as follows:

1. The structure is read using `READ`, whose behavior is modified as described above.
2. An initial minimization is performed on this structure with the additional improper torsional potentials in effect.

3. A second minimization is performed with the additional improper torsional potentials turned off.
4. If the bond lengths deviate from their ideal values by less than a specified value no search is performed and the next step (5) is skipped. Significantly distorted bonds are a good indicator that a portion of the molecule is passing through the center of a small ring.
5. A short MCMM search is performed. Each structure generated is minimized using the same two-stage procedure described in steps 2 and 3.
6. The structure is checked to see if it has adopted the requested chiralities. If so it is saved to the output structure file. If not, by default the structure is not recorded in the output structure file.

arg1 *Chirality enforcement*

−1 Turn on the requirement for LigPrep licensing, but do not turn on chirality enforcement behavior.

0 Turn on the requirement for LigPrep licensing, and turn on chirality enforcement behavior.

arg2 *Recording structures with incorrect chiralities*

−1 Write structures whose chiralities do not match those requested in the output structure file along with structures whose chiralities do match. These structures have the property `b_mmod_Chiralities_Consistent-`*Force_field_name* set to 0 (for FALSE).

0,1 Do not write structures whose chiralities do not match those requested in the output structure file.

arg3 *Search control for strained structures*

Strained structures are defined as those structures with bond lengths differing from the equilibrium bond length by more than the value specified by arg7.

−1 Never perform searches on strained structures.

0,1 Perform an MCMM search on strained structures.

arg4 *Random displacement control*

Displacing atoms by small random amounts is useful in shifting minimizations away from saddle points, a problem that often arises in structures that were originally 2D. The maximum displacement is controlled by arg6.

−1 Do not randomly displace each atom after a new structure has been read in.

0,1 Randomly displace each atom after a new structure has been read in.

arg5 *Chirality enforcement force constant in kJ/mol*

Must be non-negative. The size of the force constant for the improper torsions. Equivalent to arg5 of FXTA. Default value: 1000.0 kJ/mol

arg6 *Maximum size of random displacement in Å*

Must be non-negative. The maximum size of the random change in angstroms. introduced into each of the X,Y, and Z coordinates of each atom in the structure. The random change is chosen with a uniform probability on the range indicated. Default value: 0.05 Å.

arg7 *Extent of bond distortion in Å that triggers a search*

Must be 0 or greater than 0.05. If any bond is distorted away from the equilibrium bond length by more than this amount after the initial minimization then a full conformational search is to be performed provided that arg3 is not −1. Default value: 0.25 Å.

arg8 *CHIP verbosity*

0 Normal level of reporting to the `.log` file.

1 Turn on verbose reporting of `CHIP` processing to the `.log` file.

5.13.2 The LigPrep Command File for MacroModel

`ligprep` creates the `.com` file below for the MacroModel calculations used in the LigPrep process. The overall form of the file is fairly specific for the LigPrep process but follows that for an MCMM serial search.

```
jobname_prout.mae
jobname_bmout.mae
#MMFFs is the default and recommended force field
 FFLD      10      2      0      1     4.0000     0.0000     0.0000     0.0000
 BDCO       0      0      0      0    41.5692 99999.0000     0.0000     0.0000
#Turn on chirality enforcement using the defaults.
```

```
 CHIP       1      0      0      0     0.0000     0.0000     0.0000     0.0000
#Do not consider enantiomers to be equivalent.
 NANT
 READ       0      0      0      0     0.0000     0.0000     0.0000     0.0000
#Short searches (10 steps) are usually sufficient when needed.
 MCMM 20      0      0      0     0.0000     0.0000     0.0000     0.0000
 MCNV       1     10      0      0     0.0000     0.0000     0.0000     0.0000
 MCSS       2      0      0      0    50.0000     0.0000     0.0000     0.0000
 MCOP       1      0      0      0     0.0000     1.0000     0.0000     0.0000
 MSYM       0      0      0      0     0.0000     0.0000     0.0000     0.0000
#Auto identifies comparison atoms, tors and ring closures.
 AUTO                     -1      2                1.0000
 CONV       2      0      0      0     0.0500     0.0000     0.0000     0.0000
 MINI       9      0    500      0     0.0000     0.0000     0.0000     0.0000
```

5.13.3 MacroModel Limitations and Workarounds

AUTOmatic setup and ring closure failures

Automatic setup for the conformational searches sometimes fails to identify all of the ring closure bonds in a conformational search. When this occurs in a serial search MacroModel abandons the current search and the structure is not included in the output structure file. Processing then proceeds from the next structure in the input structure file, if present.

When the specified chiralities are not obtained the structures may be distorted

The strong improper torsions applied to chiral atoms aggressively try to force the molecules to attain the specified chiralities. If this process fails to produce the correct chiralities the resulting structures may be quite distorted. Such failures usually are the result of physically inconsistent chiralities. If the option to save those structures that fail to attain the specified chiralities is given (this is not the default behavior) a significant number of distorted structures might be written to the output structure file.

Ligprep stopping during `bmin` processing

`bmin` processing consumes more than 75% of the CPU time required to process a collection of ligands. If the LigPrep process stops while `bmin` is processing ligands, the output structure file for `ligprep` can be empty or contain only one ligand. However, the file *jobname*_`bmout.ouL` should contain essentially all of the ligands successfully processed up to the stage at which `bmin` stopped.

The ionizer Utility

Under physiological conditions, a molecule with acidic or basic groups can exist in a variety of protonation (ionization) states. Ligand structures are usually provided in only one of these states. To ensure that the important ionization states are sampled, LigPrep includes the `ionizer` utility to generate the states that are significantly populated under a prescribed set of conditions.

The `ionizer`, or ionization state expander, produces multiple structures for each input structure with different combinations of ionized states based on the ionizable groups present. The `ionizer` makes use of a database of substructure patterns (templates) for ionizable groups, which it matches to the input structure to identify candidates for protonation or deprotonation. If a template for the ionizable group is not found, the group is not ionized.

LigPrep sets only a few of the `ionizer` options. For information on these options, see Table 3.1 on page 35 and Section 5.7 on page 68. You can pass a restricted set of `ionizer` options to the `ionizer` with the `ligprep -W i` option. These options are indicated in the option descriptions below.

This chapter describes the use of the `ionizer` as a stand-alone utility.

6.1 Running the ionizer

To run the `ionizer` from the command line, use the syntax described below.

Syntax:

```
$SCHRODINGER/utilities/ionizer [options]
```

Options:

Information options

`-?|-h|-help`	Show usage summary.
`-doc`	
`-v|-ver|-version`	Show program version information.

File options

`-b|-bad|-badfile` *badfile*	Specify file name for input structures that `ionizer` fails to process. Default is *jobname*`-ion-bad.mae`.
`-i|-in|-infile` *infile*	Specify input file. Default is *jobname*`.mae`.
`-j|-job|-jobname` *jobname*	No default (must be specified unless all essential files are specified).
`-l|-log|-logfile` *logfile*	Specify log file name. Default is *jobname*`.log`. To log to screen, replace *logfile* with `-`.
`-o|-out|-outfile` *outfile*	Specify output structure file name. Default is *jobname*`-ion.mae`.

Log file options

`-ll|-loglevel` *level*	Expansion report log level. Use 0 for quietest (default), 1 to log state generations, 2 provides additional details for debugging. Supported with `ligprep -W i`.
`-sf|-showfinal`	Show final ionization candidate list. Supported with `ligprep -W i`.
`-sm|-showmatches`	Show substructure pattern matches. Supported with `ligprep -W i`.
`-ss|-showskips`	Show skipped state generations. Augments log level 1 and higher. Log levels > 1 give skip reasons. Supported with `ligprep -W i`.

Restriction options

`-mg|-maxgroups` *limit*	Maximum number of ionizable groups to be handled. Structures containing more than this number of ionizable groups are recorded in the output structure file with no variation of ionization states. Default 15; `ligprep` default 10. Set by `ligprep -mg`. Cannot be set larger than 31.
`-mi|-maxions` *count*	Maximum number of ionization states. Applies only to pK_a mode. Default 4. Supported with `ligprep -W i`.
`-mo|-maxoutcts` *limit*	Maximum number of output structures per input structure. Default 512. Supported with `ligprep -W i`.
`-mq|-maxabstotq` *charge*	Maximum absolute total charge. Only applies to pK_a mode. Default 2. Supported with `ligprep -W i`.

`-ph` *value*	Effective pH of active site. Default 7.0. Supported with `ligprep -W i`.
`-pht|-phthresh` *maxdiff*	pH threshold for pH-based ion state rejections. Ion states are rejected if the pK_a of the (protonated) ionizable group differs from the pH value by more than this threshold. Default 2.0. Supported with `ligprep -W i`.
`-pkt|-pkthresh` *maxdiff*	pK_a threshold for pK_a-based ion state rejections. Overrides pH-based rejection mode. Ion states are rejected if the maximum pK_a of the deprotonated groups is larger than the minimum pK_a of the protonated groups by more than this threshold. No default. Supported with `ligprep -W i`.

Other options

`-adjust`	Retain only the most favorable structure in the output file.
`-kp|-keep_props`	Keep properties and write to output file. Default is to delete properties. Set by `ligprep -kp`.
`-retain`	Retain the input structure in the output.
`-retain_lab`	Retain the input structure and add a label property to the retained input structure that identifies it as the input structure.
`-rw|-retitle_with` *prefix*	Add ion state number onto structure titles. The prefix precedes the ion state number. Quote *prefix* if there are spaces. By default, structures are not retitled. Examples of retitling: If the original structure title is "TITLE," and the output structure is for ion state 3, then `-rw _` produces the output structure title TITLE_3, `-rw " Ion_state_num="` produces the output structure title "TITLE Ion_state_num=3."
`-s|-spec|-specfile` *specfile*	Use non-standard patterns specification file. See Section 6.7 on page 89. Supported with `ligprep -W i`.
`-strict`	Force termination of `ionizer` if it encounters an unacceptable input structure.

Additional information on the `ionizer` is available in the file:

```
$SCHRODINGER/services-vversion/doc/README.ionizer
```

6.2 Input Structure Requirements

For the `ionizer` to process all input structures successfully, each structure must satisfy the following requirements:

- Hydrogens must be added.
- Atom types must be correct.
- There must be no formal charge on atoms matched as centers of ionizable groups.

It is not necessary for input structures to be minimized fully, as the `ionizer` works in terms of atom types and connectivity. However, bond lengths and bond angles should have reasonable values: unreasonable structures can cause processing failures. Coincident atomic coordinates can cause serious problems, especially if atoms coincide within an ionizable group.

By default, structures that the `ionizer` cannot process are passed unchanged to the output file. It also records these structures in the file *jobname*`-ion-bad.mae` (or the file specified by the `-b` option). If you set the `-strict` option, the `ionizer` stops if if encounters a structure that it cannot process.

6.3 Restricting Ionization State Generation

The `ionizer` has two methods for restricting the generation of ion states to ensure that each state has a significant population under the chosen conditions. The default method is to screen values according to a pH range. This is done by specifying a target pH value with the `-ph` option and a threshold value with the `-pht` option. The default pH is 7, and the default threshold is 2. For any ionizable group whose pK_a lies within the threshold value around the target pH value ($|pK_a - pH| \leq$ threshold), both protonated and deprotonated forms are generated. If the pK_a is smaller than the target pH value by more than the threshold value, only the deprotonated form is generated, and if the pK_a is larger than the target pH value by more than the threshold value, only the protonated form is generated. To generate all possible ionization states, you can set the threshold to a large value.

The alternative method is to use the `-pkt` option to specify a relative threshold for the pK_a values to screen out improbable states. In this method, a structure generated by the `ionizer` is rejected if the maximum pK_a value of any deprotonated group is greater than the minimum pK_a value of any protonated group by the specified threshold.

For example, if the threshold is 3.0 and the structure has two ionizable groups with pK_a values of 6.0 and 10.0, the structure with the first group protonated and the second group deprotonated is rejected. Likewise, if the threshold is 2.0 and the structure has 4 ionizable groups, with pK_a values of 4.0, 5.5, 7.0, and 8.5, the structure with the first and fourth groups deprotonated and the second and third groups protonated is rejected. In this case the maximum pK_a of the depro-

tonated groups is 8.5 and the minimum pK_a of the protonated groups is 5.5. The difference is greater than the threshold value of 2.0.

This method treats each structure independently and ensures that the totally protonated form and the totally deprotonated form are always included in the output. There is no default value for the threshold, and as for the pH-based method, it is possible to specify a pK_a threshold large enough so that no ion state is rejected.

If you specify the pK_a threshold, you can also specify a maximum number of groups ionized and a maximum total charge, to limit the number of structures generated. Setting the `-mi` option limits the maximum number of ionized groups to the value specified. The default is 4. If a possible ion state has more individual groups ionized than this value, the state is rejected. This limit applies only to the ionizable groups detected by the `ionizer`. Setting the `-mq` option limits the absolute value of the total charge to a value less than or equal to the value specified. The default is 2. If the structure already has a formal charge, this charge is taken into account when the restriction is applied.

6.4 Structural Output

If an input structure contains a large number of ionizable groups, the number of output structures will be very large (2^N combinations) and the `ionizer` will take a long time to run. The `ionizer` has a limit on the number of ionizable groups in any structure that can be set, above which the `ionizer` skips its normal analysis and expansion of the structure and passes it unchanged into the output file. To set this limit, use the `-mg` option (see page 82). The default value is 15 for standalone use, and 10 for use within LigPrep. You cannot set the limit higher than 31. An input structure that contains close to 31 ionizable groups could take hours or days to be treated by the `ionizer`. You should therefore set this limit well below 31 if you do not want long run times.

In addition, structures that cannot be processed are passed unchanged into the output file and are recorded in a separate file, named *jobname*`-ion-bad.mae` by default. The name of this file can be set with the `-b` option.

6.5 Log File Output

As the `ionizer` expands input structures into ionized output structures, it records various status messages in the log file, which by default is named *jobname*`.log`. You can specify the log file name with the `-l` option. If you want to direct the log file to your screen, set the log file name to a hyphen: `-l -`. Examining the log file output can be useful if you want to determine how a particular structure was handled.

Substructure matching: If you use the `-sm` option of the `ionizer` command, the substructure pattern matches are logged on each input structure as it is processed. The information logged for each match includes the line number in the pattern file where the substructure pattern is given, the pattern itself, the ion center atom number where the match is anchored, the acid/base character of the ionizable group, and a comment as to whether the match modifies an earlier match at the same atom.

Final ionization candidate list: If you use the `-sf` option of the `ionizer` command, the final list of ion centers identified on each input structure is logged as it is processed. The information logged for each ion center includes the ion center atom number, the acid/base character of the group, and the pK_a value associated with this group.

Log level: The `-ll` option gives general control over the amount of information provided in the log file. Each level adds information to the previous level.

At level 0, the report on each input structure processed includes the structure number (in the input file), structure title, the number of distinct ionizable groups identified, the number of ionization states to consider generating, and the number of states actually generated after imposition of restrictions.

At level 1, the report contains a line on each ion state generated, showing the state's index and a brief description of the state, listing the original atom numbers in the input molecule which become ionized in the output.

At level 2, the report provides more details which may be helpful in debugging in conjunction with Schrödinger Technical Support, and increases the run time slightly.

For production runs, you should use level 0 if possible and level 1 only if you need to see a list of the ion states generated or skipped.

Skipped states: If you use the `-ss` option of the `ionizer` command, a line is written for each ion state skipped—that is, rejected from generation due to imposition of some restriction. This option affects only log levels greater than 0. If this option is specified when running at level 2, the reasons for rejecting an ion state are shown. You might want to see these reasons spelled out, for example, when adjusting the restriction parameters to see which rejections they lead to in practice.

If structures are skipped because the structure itself is faulty, error messages are printed to the log file. In the default fault-tolerant mode, each set of error messages resulting from input structure problems is followed by a message indicating that the otherwise fatal errors have been tolerated, and the structure passed to the output file unchanged.

6.6 Maestro Properties

When the ionization state changes, most if not all of the properties of the structure are no longer valid. By default, the `ionizer` does not copy the properties of the input structures to the output. In some cases, the properties of the parent structure are useful for later screening or identification—for example, QikProp properties. If you want to ensure that properties are inherited by the output structures, you can add the `-kp` option to the command line.

The `ionizer` generates some Maestro properties, which are summarized in Table 6.1.

Table 6.1. Maestro properties generated by the `ionizer`.

Property Name in Project Table	Name in Maestro Output File	Description
Ion state n	`i_ionizer_Ion_state_n`	Ionization state number. This is an index running from 0 to the number of ionized states, and serves as a unique identifier.
Tot Q	`i_ionizer_Tot_Q`	Total formal charge. This value includes any formal charges in the input structure that have been retained.
Ion centers	`s_ionizer_Ion_centers`	String of atom numbers for new charge centers generated by the `ionizer`.[a]
Ion types	`s_ionizer_Ion_types`	String of MacroModel atom type symbols for new charge centers generated by the `ionizer`.[a]
Ion ctrs in	`s_ionizer_Ion_ctrs_in`	String of atom numbers of the original uncharged atoms in the input structure that correspond to new charge centers in this output structure.
Ion passthru	`s_ionizer_Ion_passthru`	Brief description of the reason an input structure was passed to the output file unchanged. Generated only if an input structure is not processed.
Ionization penalty	`r_ionizer_Ionization_penalty`	This ionization state's total energy penalty[b]
Ionization penalty charging	`r_ionizer_Ionization_penalty_charging`	This ionization state's energy penalty due to charged sites[b]
Ionization penalty neutral	`r_ionizer_Ionization_penalty_neutral`	This ionization state's energy penalty due to neutral sites[b]

a. The strings are empty for the non-ionized state. Neither of these strings is a unique id for the ionization state. It is not uncommon for these strings to match between different ionized variants of the same input.
b. The energy penalties are expressed in kcal/mol units. These properties are not written if running in pK-threshold mode. Details of the calculation are presented below.

Calculation of Ionization State Energy Penalties

For each ionizable group involved in a generated state, we define two energy costs:

Yn: The energy cost of ionizing the group

Zn: The energy cost of keeping the group neutral

Consistent with the Ionizer's other simplified assumptions, we do not account for cooperative effects between multiple groups.

At the given pH, with the group's specified pK_a, if the group is a base, we calculate:

$$Yn = RT \ln(10^{pH-pKa} + 1)$$

$$Zn = RT \ln(10^{pKa-pH} + 1)$$

where R = 0.001987207 kcal/(mol K), and T = 298.15 K.

If the group is an acid, we calculate the alternate pair:

$$Yn = RT \ln(10^{pKa-pH} + 1)$$

$$Zn = RT \ln(10^{pH-pKa} + 1)$$

Now, for each generated ionization state combination, we sum the Yn for all groups that are ionized, calling the total Y, and we sum the Zn for all groups that are not ionized, calling the total Z.

For small values of the pH threshold (controlled by `-pht`), Y is dominated by the energy cost of ionizing weak acids and bases, while Z is dominated by the energy cost of keeping weak acids or bases neutral. For larger values of the pH threshold, these penalties begin to include the corrections for producing stronger acids and bases in atypical forms.

We also compute P = Y + Z, which is the total energy cost of the ionization state combination.

The P, Y, and Z values are expressed in the energy penalty properties for the output structure:

`r_ionizer_Ionization_penalty`

`r_ionizer_Ionization_penalty_charging`

`r_ionizer_Ionization_penalty_neutral`

6.7 Creating a Customized Patterns File

The file that contains the substructure patterns used to identify ionizable groups can be customized. This file contains substructure patterns for protonating N atoms, and deprotonating NH, OH, and SH groups. The standard patterns file can be found at

`$SCHRODINGER/services-v`*version*`/data/ionizer.ini`

If you want to add your own patterns, you should make a copy of the file and edit it. Documentation on the pattern specification commands can be found in comments in this file. The pattern specification commands require use of Schrödinger's linear substructure notation, which is documented in

`$SCHRODINGER/services-v`*version*`/doc/mmsubs_syntax.txt`

Once you have made your changes, you can then use the modified file as input to the `ionizer` in the following ways:

- By providing its name with the `-s` option when you run the `ionizer`, or the `-is` option when you run `ligprep`.
- By copying this file to the directory from which you launch an `ionizer` or `ligprep` job. The file must be named `ionizer.ini`.
- By adding this file to your `$HOME/.schrodinger/mmshare` directory. The file must be named `ionizer.ini`, and becomes your global default patterns file.

Note: Correct use of the syntax is not trivial. You might have to experiment for some time to ensure that you have the correct result, and you may need to contact Schrödinger for assistance.

Warning: Do not change the standard patterns file. If you accidentally change it, the `ionizer` might fail or, worse, might continue to execute but give incorrect results. If you have altered this file, you should re-install the services product before running the `ionizer` again to ensure the correct results.

6.8 Ionizer Limitations

The limitations described in this section apply only to the Ionizer operation, if used in the LigPrep process, or if used on a stand-alone basis. Limitations of the LigPrep process as a whole are described in Section 3.5 on page 45.

- Local Focus in Adjustment of Protonation States

  The Ionizer adjusts the protonation state of each ionizable group without regard for the protonation state of other ionizable groups found in the molecule. In other words, when deciding which states of one ionizable group to generate, the Ionizer does not take into account changes in the group's effective pK_a due to neighboring protonation states. This can lead to generation of the wrong ionization states for molecules containing more than one ionizable group.

  For the specific case of an aliphatic diamine separated by two carbons (for example, piperazines and ethylene diamines) patterns have been introduced in which the pK_a of both nitrogens is given as the average of the two experimental values for the mono and diprotonated species. As a result, when the ionizer is run with the default settings (pH 7 2), four forms are generated: neutral, both monoprotonated forms and the diprotonated species.

  Note: Only the monoprotonated forms are favorable near pH 7. Without these patterns, only the diprotonated form would be generated. The ordering of the penalties for this class of system is not accurate. This behavior can be disabled by commenting out the diamine lines in the `ionizer.ini` file.

  The erroneously excluded states can be generated by running the Ionizer with non-default pH and/or pH threshold settings, but it can take some trial and error. Of course, running with modified settings to produce previously excluded states of one molecule can generate undesired variants of other input molecules.

- Incompatibility with Certain Force Field Implementations

  The Ionizer may generate structures that cannot be used with certain force field implementations. One known case involves the atom type N5 (N^+ sp^3) and AMBER* (Modified AMBER), which arises when ionized forms are generated outside of “normal” pK_a ranges, particularly the protonation of anilines and aromatic heterocyclic nitrogens. There could also be problems with atom type N4 (N^+ sp^2).

- Rotation of Output Molecules

  In some ionized molecules with protonated nitrogens, the output structure will be rotated with respect to the input structure.

- Upper Limit on Number of Ionizable Groups in Input Molecule

  The program is capable of handling up to 31 ionizable groups in an input molecule (though it would take a long time, and a lot of disk space, to so do). The value 31 is a hard upper limit. This is the maximum value allowed to be specified with the command-line option `-mg` or `-maxgroups`.

The tautomerizer Utility

Tautomers are an important class of isomers that can interconvert under physiological conditions. Tautomeric forms have different chemical properties and interact differently. For instance, one tautomeric form may interact with the active site of a protein more strongly than the other forms. Therefore, for some types of calculations, such as docking with Glide, considering the appropriate tautomeric forms of ligands can be important.

There does not seem to be a universal definition for tautomers. In LigPrep, tautomers are defined as isomers that meet the following conditions:

- In aqueous solution tautomers interconvert rapidly enough to be present as mixtures.
- One or more hydrogen atoms are bound to different atoms and the orders of one or more of the bonds between non-hydrogen atoms differs between tautomers.
- The non-hydrogen atom topology of the structure does not change during these interconversions.

LigPrep's definition of tautomers therefore excludes the aldose-hemiacetal ring opening and closing equilibrium in sugars that are sometimes regarded as tautomerizations. An example of the well-known keto-enol tautomerization is shown in Figure 7.1.

The `tautomerizer` utility is intended to generate tautomers for input structures in a practical and flexible manner. The `tautomerizer` relies on a database of tautomeric templates to guide it in generating tautomers. Flexibility is provided in the command-line options and by permitting users to modify the database. For more information on this database, see Section 7.2 on page 93.

The `tautomerizer` is not intended to generate all possible tautomers. The collection of groups of tautomers in the database is not exhaustive. As well, each set of tautomeric forms in the

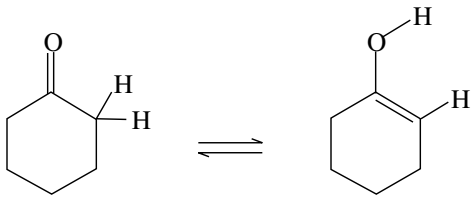

Figure 7.1. Keto-enol tautomerization.

database is limited to those tautomers that are expected to have significant populations in aqueous solution, rather than being a comprehensive list. Tautomers in the database are assigned probabilities to assist in focusing on the most highly populated tautomeric forms.

7.1 Running the tautomerizer

To run the `tautomerizer` from the command line, use the syntax described below. The `tautomerizer` is part of the LigPrep collection of programs and requires a LigPrep license.

Syntax:

`$SCHRODINGER/utilities/tautomerizer [`*options*`]` *input_file*`.mae` *output_file*`.mae`

Options:

`-h`	Show usage summary.
`-m` *number*	Set the maximum number of tautomers generated internally to *number*. Output tautomers are selected from this set. Default: 128
`-ma` *number*	Set the maximum number of atoms to number. Default: 150
`-n` *number*	Set the maximum number of tautomers produced to *number*. Default: 8
`-p` *number*	Set the minimum probability for retaining a tautomer. The most probable tautomeric form for each input structure is always written to the output structure file even if its probability lies below the threshold, to avoid completely eliminating molecules. Default: 0.01
`-retain`	Retain the input structure.
`-retain_lab`	Retain the input structure and add a label property to the retained input structure that identifies it as the input structure.
`-s` *filename*	Specify a custom tautomer database.
`-v`	Show the version number.
`-verb` *level*	Controls `MMERR` level of reporting. 0 `MMERR_FATAL`—only report errors. 1 `MMERR_WARNING`—only report errors and warnings. 2 `MMERR_INFO`—report errors, warnings, and processing information. 3 `MMERR_DEBUG`—report errors, warnings, processing information, and debugging information.

The `tautomerizer` can also be run as part of the LigPrep process. When LigPrep is run from the LigPrep panel in Maestro, the default options are used. When it is run from the command line, the defaults can be overridden by options to the `ligprep` command. See Section 3.2 on page 35 and Section 5.8 on page 69 for more information.

7.2 The Tautomer Database

In LigPrep 2.0 the default tautomer database is not accessible to users. However, you can provide your own file to either completely override the default file, or add patterns to it.

Note: The database file format has changed in LigPrep 2.0. The current format will not work with previous versions of the program although earlier versions of the data file may work with LigPrep 2.0, with some modifications.

At the top level of the tautomer database file the following four items can be present: `name`, `clear_standard`, `group_def`, and `tautomer_set`. These items are described in the following sections. Lines beginning with a `#` are comment lines and are ignored when interpreting the contents of the tautomer database file. Blank lines are also ignored.

7.2.1 name Data Item

`name` specifies the name of the solvent. For example:

```
name: water
```

`water` and `DMSO` are standard names for which the tautomerizer already has information. Currently, the DMSO tautomer information is just a duplicate of that for water.

7.2.2 clear_standard Directive

By default, information in a custom tautomer database file is added to any exising information already available for the solvent specified. Including `clear_standard:` in a tautomer database file clears any values for this solvent accumulated before the current file was read.

7.2.3 group_def Data Structure

The tautomerizer does not support recursive SMARTS. However, a mechanism that supports some of the functionality of recursive SMARTS is provided by the `group_def` data structure. This data structure permits you to define variables that correspond to SMARTS patterns. The variables may be reused in groups and `tautomer_sets` that appear later in the tautomer database file.

Each group contains two items:

`name:` an arbitrary name for the group which is used to reference the group.

`pattern:` The SMARTS pattern for the group. This pattern may refer to previously defined groups using $*groupname*.

Below are some examples of `group_def` data structures:

```
group_def{
        name: Halogens
        pattern: [F,Cl,Br,I]
}

group_def{
        name: Amides
        pattern: [CX3](=[OX1])-[NX3]
}

group_def{
        name: Carbonyls
        pattern: [CX3](=[OX1])
}

group_def{
        name: Carbonyls_only
        pattern: [$Carbonyls;!$Amides]
}
```

7.2.4 tautomer_set Data Structure

`tautomer_set` data structures define sets of interconvertible tautomers. There are more than 150 tautomer sets available by default for water.

Some examples of `tautomer_set` data structures are given below, and the syntax for the data structures is described following these examples.

Note: The entry for `pattern:` values must be a single line. In the examples below, some of the `pattern:` text wraps to the next line due to formatting constraints within this manual. When creating tautomer data structure files in a text editor, ensure that text-wrapping is turned off, or that margins are set wide enough to accommodate single-line entry for this value.

```
tautomer_set{
    name: single-sided_ket-enol
# From: Handbook of organic chemistry

    tautomer{
      name: enol
      pattern: [CX3](-[#1,$Sub_aC])(-[#1,$Sub_aC])=[CX3](-
[#1,$Sub_carbonyl_C])-[OX2]-[#1]
      probability: 0.00005
    }

    tautomer{
      name: ket
      pattern: [CX4](-[#1])(-[#1,$Sub_aC])(-[#1,$Sub_aC])-[CX3](-
[#1,$Sub_carbonyl_C])=[OX1]
      probability: 0.99995
    }
}

tautomer_set{
    name: double-sided_ket-enol
# From: Handbook of organic chemistry

    tautomer{
      name: 1enol
      pattern: [CX3](-[#1,$Sub_aC])(-[#1,$Sub_aC])=[CX3](-[CX4](-
[#1])(-[#1,$Sub_aC])(-[#1,$Sub_aC]))-[OX2]-[#1]
      probability: 0.00000001
    }

    tautomer{
      name: ket
      pattern: [CX4](-[#1,$Sub_aC])(-[#1,$Sub_aC])(-[#1])-[CX3](-
[CX4](-[#1])(-[#1,$Sub_aC])(-[#1,$Sub_aC]))=[OX1]
      probability: .99999998
    }

    tautomer{
      name: 2enol
      pattern: [CX4](-[#1,$Sub_aC])(-[#1,$Sub_aC])(-[#1])-
[CX3](=[CX3](-[#1,$Sub_aC])(-[#1,$Sub_aC]))-[OX2]-[#1]
      probability: 0.00000001
    }
}
```

```
tautomer_set{
    name: imidazole

    tautomer{
      name: form1
      pattern: c1(~[#1,$Sub_c])n(-[#1,$Sub_n])-c(-
[#1,$Sub_c])=[nX2]c1(~[#1,$Sub_c])
      probability: 0.50
    }

    tautomer{
      name: form2
      pattern: c1(~[#1,$Sub_c])[nX2]=c(-[#1,$Sub_c])-n(-
[#1,$Sub_n])c1(~[#1,$Sub_c])
      probability: 0.50
    }
}
```

Each tautomer set contains a `name:` designator and a number of tautomer structures. The `name:` designator is followed by a space and a contiguous non-blank label to identify the class of tautomers described by the set. The label provided does not affect processing. In the examples below, there are three tautomeric sets: `single-sided_enol-ket`, `double-sided_enol-ket`, and `imidazole`.

The `tautomer` structure describes the properties of one tautomeric form. There are three designators that may be used within a `tautomer` structure: `name:`, `probability:`, and `pattern:`.

The `name:` designator provides a label for the tautomer but does not otherwise affect processing.

The `probability:` designator is used to assign a probability or fractional population of this tautomer within this tautomeric set. In many cases, reliable information on the probability of various tautomeric forms is not available and the values entered in the database are simply educated guesses.

The `pattern:` designator is followed by a contiguous SMARTS-like pattern. A difference between this pattern and a normal SMARTS pattern is that instead of using explicit aromatic bonds, explicit single "–" and double "=" bond designators are used to make the corresponding Lewis structures clear. In addition, these patterns may include references to previously defined groups via the $*group_name* mechanism. Information on SMARTS patterns is provided on the web page: http://www.daylight.com/learn[4]. The SMARTS-like pattern is used to detect the corresponding groups of molecules in the input structures and to permit the tautomerizer to understand how the bonding patterns (Lewis structures) differ between tautomers so that they

may be interconverted. For heavy atoms that are expected to carry a formal charge it is advisable to include the charge in the SMARTS pattern. To ensure that the SMARTS patterns are properly interpreted by the `tautomerizer`, the following restrictions must be applied:

- The SMARTS patterns for all tautomers within a tautomer set include the same list of non-hydrogen atoms in the same order.
- All SMARTS patterns must explicitly designate the hydrogens that shift positions in any tautomer within a tautomer set with a `-[#1]` pattern.
- All SMARTS patterns within a tautomer set must contain the same number of explicitly designated mobile hydrogen atoms.
- In both non-aromatic and aromatic portions of the SMARTS pattern, bond orders that change between single and double in any tautomer must be explicitly specified in the SMARTS patterns for all tautomers in a tautomer set.
- In portions of molecules that must be represented by aromatic atom types (e.g., `c` and `n`), only changes in the bond orders of bonds involving `n` atoms in the corresponding Lewis structures are supported. If such a bond changes order in any tautomer in a tautomer set, it must be represented as '`:`' in all the tautomers. See the guanosine tautomer set in the example above.
- Recursive SMARTS are not supported.
- SMARTS patterns within the same tautomer set must all specify the same overall formal charge.

The database provided with this release contains templates for keto-enol tautomers and their sulfur analogues, imine-enamine tautomers, histidine-like tautomers, tautomers of DNA and RNA bases, and a large number of common heteroaromatic rings containing C, S, O, and N.

7.3 Tautomer Processing

Each structure in the input structure file is processed separately. The processing of each structure by the `tautomerizer` can be divided into two stages: tautomer pattern matching and structure generation.

7.3.1 Tautomer Pattern Matching

The current structure is examined for all matches of all tautomeric patterns in the `tautomerizer` database. All matches are cross-examined to see if the atoms are completely

4. Please see the notice regarding third party programs and third party Web sites on the copyright page at the front of this manual.

contained within another match. If so, the match containing the smaller number of atoms is eliminated. If they are the same size, then the match pattern found first is retained. This is the only case where the order of tautomers within a tautomer set or the order of the tautomer sets within the `tautomer_list` file affects processing.

In the example above, there are two related SMARTS patterns for keto-enol tautomerization. The `single-sided_enol-ket` tautomer set handles carbonyl groups that can tautomerize only on one side, such as an aldehyde with at least one H atom bonded to the carbon atom adjacent to the carbonyl carbon, while the `double-sided_enol-ket` tautomer set handles carbonyl groups that can tautomerize on either side, such as ketones with at least one H atom bonded to each carbon atom adjacent to the carbonyl carbon atom. Thus structures that match the `double-sided_enol-ket` pattern also match at least one `single-sided_enol-ket` pattern. However, the single-sided matches for this carbonyl are eliminated because they are completely contained within the double-sided pattern. This is important since the double-sided probabilities are different, and matching an enol form in the double-sided case leads to the consideration of the other enol form during the structure generation stage, which would not happen for a single-sided match.

7.3.2 Structure Generation

For each match, all tautomers in the tautomer set should be considered. Structures in which more than one match has been found have multiple locations in the structure that can be tautomerized, and all combinations of tautomers that are compatible should be considered. Double bonds that can shift to single bonds in any tautomer within a set can shift between E and Z forms, provided that other topological constraints do not prevent this shift.

The `tautomerizer` tries to generate all compatible combinations of all tautomers for each match. For each structure having a high enough probability, all combinations of 180 degree rotations about certain double bonds within each tautomeric pattern are generated. Double bonds are rotated if they involve C or N atoms in which both ends are not attached to two H atoms and in which the two atoms in the bond do not reside in the same ring. No adjustment is made to the probability for the tautomeric form for these double bond rotations. The probability for each structure resulting from this process is estimated as the product of the probabilities for all the tautomer templates used to generate it. The probabilities are normalized amongst all the structures generated for a given input structure.

Tautomers are generated in order of decreasing probability. By default, the maximum number of tautomers generated internally is 128. This limit can be adjusted using the `-m` option of the `tautomerizer` command. Of this collection of tautomers, up to eight tautomers are recorded by default in the output file for each input structure. This limit can be adjusted with the `-n` option of the `tautomerizer` command (`-t` option of the `ligprep` command). The most probable tautomer is always recorded. Except for this tautomer, only tautomers with a probability

higher than 0.01 are recorded in the output structure file. This threshold value can be adjusted using the `-p` option of the `tautomerizer` command (`-tp` of the `ligprep` command). If either the `-retain` or the `-retain_lab` option is used to keep a tautomeric form of a molecule that would normally have a probability lower than the threshold probability value, the probability for this form is set to the threshold value.

Note: The probability for forms of a tautomerizable group is not adjusted for other functional groups inside the molecule, but outside the pattern itself. This means that these probabilities, and thus the overall probabilities for molecular forms (particularly when multiple tautomeric sites are being adjusted), are approximate and intended as a guide.

A chiral atom in a tautomer can switch its chirality if one of its tautomeric forms involves a double bond to this atom. The `tautomerizer` does not vary the chiralities of such atoms and does not add chirality properties for such atoms. However, the `stereoizer` can generate these variations. See Section 5.10 on page 70 for more information. In the LigPrep process, the `stereoizer` is run after the `tautomerizer` and the chiralities of such atoms are varied. Some LigPrep options that restrict the generation of stereoisomers may prevent this variation—see Section 3.2 on page 35.

The ligparse Utility

The `ligparse` utility is a tool for examining the structural composition of a molecular data set in a Maestro- or MDL SD-formatted file. `ligparse` can use this structural information to select subsets of compounds within the original data set that meet some user-defined criteria.

8.1 Descriptors and Composite Descriptors

The `ligparse` utility analyzes the composition of a molecular structure by counting structural components, which include the usual organic functional groups. These structural components are represented by SMARTS patterns. A single SMARTS pattern for a given structural component is called a *descriptor*. Linear combinations of SMARTS patterns can be generated to represent more sophisticated structural entities, such as a hydrogen bond donor or acceptor, or metal-ligating groups. These sums of descriptors are called *composite descriptors*. A descriptor is defined by the combination of a SMARTS pattern with its label.

`ligparse` checks each molecule in turn against all descriptors. The algorithm is "greedy"—each atom or bond matches all applicable SMARTS patterns rather than only the most sophisticated or the most simple pattern. For instance, a carboxylic acid group in a molecule would match one instance of [#6;X3]~([O;X1])~[O;X1], two instances of [#6;X3]~[O;X1], and two instances of [#8].

The default descriptor list is stored in the file `$SCHRODINGER/macromodel-v`*version*`/data/str_keys.type`. You can copy and modify the `str_keys.type` file and use it in place of the default with the `-s` command-line option. The `str_keys.type` file is a free-format file with column descriptions as given below.

```
15   H   H  0.00  [n;X2;!H]  nosel  high  -   #  Num heteroaromatic n w.outH
(1) (2) (3) (4)-  (5)------  (6)--  (7)- (8)  (9)---------------------------
```

Only columns (5), (6), and (9) are relevant to this discussion; however, data in the format given above must exist for all columns. In this example, column (5) gives the SMARTS pattern associated with the title in column (9), `Num heteroaromatic n w.outH`. For more information on SMARTS patterns, see http://www.daylight.com.[5] Any string may be used as a descriptor label. The string `nosel` in column 6 is necessary for atom typing to recognize that this SMARTS pattern is to be used by `ligparse`.

5. Please see the notice regarding third party programs and third party Web sites on the copyright page at the front of this manual.

A total of 19 composite descriptors are defined by default in the `composite.types` file in `$SCHRODINGER/macromodel-v`*version*`/data`. They are:

Num rings
Num aromatic rings
Num rotatable bonds
Molecular weight
Num neutral amines
Num divalent oxygen atoms
Num charged donor groups
Num charged acceptor groups
Num acidic hydrogens
Num acceptor groups
Num heteroaromatic rings
Num aliphatic rings
Num atoms
Num chiral centers
Num amide hydrogens
Num neutral donor groups
Num neutral acceptor groups
Num reactive groups
Num donor groups

Of these composite descriptors, the first eight (Num rings, Num heteroaromatic rings, Num aromatic rings, Num aliphatic rings, Num rotatable bonds, Num atoms, Molecular weight, and Num chiral centers) cannot be adjusted by the user. Other descriptors may be added to the file, and descriptors may be removed from the file. You can specify your own `composite.types` file with the `-c` command-line option. The composite descriptors file is in free format, with a single composite descriptor represented as follows:

```
Num neutral donor groups
    +  Num H-N
    +  Num H-O
    -  Charged amines
    -  Charged imines
    -  Charged pyridine
    -  Charged N-C=N on -N
    -  Charged imidazole
    -  Charged HN-imidazole
```

Here `Num neutral donor groups` is a linear combination of the eight descriptors represented by their labels. The + or – is required and indicates whether the descriptor is to be added to or subtracted from the composite descriptor.

8.2 Molecular Database Structural Analysis

The average structural composition of a given database may be examined using job type 3 (`-j 3` command-line option). `ligparse` determines the number of times each SMARTS pattern is recognized in each molecule and maintains a sum of the number of times each

SMARTS pattern has been found in all molecules. After reading the last molecule in a data set, the mean of the number of times each SMARTS pattern has been found is written to the output file. Standard deviations are also given for some measure of the width of the distribution.

More detailed structural information per compound is available with the `-pcsv` command-line option. With this option, a comma-separated file is written that contains the counts of descriptors and composite descriptors per molecule, with one molecule per line. The file name for the comma-separated file can be adjusted with the `-pcsv` *filename* command-line argument. The comma-separated file can be imported into common spreadsheet programs for further processing. In addition, the structures in the input file are copied to a Maestro-formatted output file with the descriptors added as properties.

8.3 Molecular Database Subset Selection: "Culling"

`ligparse` can use four methods for selecting a subset of molecules, which is called *culling*:

- Culling to reproduce the mean value of composite descriptors within the original data set (`-j 2` command-line option)
- Culling to reproduce user criteria for any descriptors (`-j 1` command-line option)
- Culling to retain only compounds which match user criteria (`-j 4` command-line option)
- Culling to remove only compounds which match user criteria (`-j 5` command-line option)

The first two techniques involve optimization of an error function through random molecule substitution. The third and fourth methods allow the use of `ligparse` as a structural filter, removing or retaining molecules that match the user-defined criteria. For job types 4 and 5, if more compounds match the user criteria than is desired to have in the culled subset, molecules are randomly selected from the subset that matched the user criteria.

For a listing of each descriptor and composite descriptor available with the current `str_keys.type` and `composite.types` files, use the `-d` command-line option.

For job types 1, 4, and 5, the user criteria are passed to `ligparse` with the `-i` command-line option. The file containing the user criteria is the `cull.info` file. Any descriptor or composite descriptor may be used in a `cull.info` file. The file is in free format with each line listing the label and target value for a single descriptor or composite descriptor. Any line beginning with "##" is treated as a comment and ignored. Tabs are not recognized and should not be used. For job type 1, all descriptor and composite descriptor target values must be either integer or real positive values.

For filtering with job type 4, only molecules that match all criteria listed in the `cull.info` file are *retained*. For filtering with job type 5, only molecules that match all criteria listed in the `cull.info` file are *removed*. These behaviors may be modified with the `-any` option, which changes the selection behavior from removing or retaining only compounds that match *all* criteria to removing or retaining only compounds that match *any* of the criteria.

To further enhance the filtering of molecules with job types 4 and 5, you can retain compounds with descriptor values greater than a given value (>*X*), less than a given value (<*X*), or within some range (*X*-*Y*). Ranges are defined by $X < value < Y$, thus a molecular weight range of `300.0-400.0` retains only molecules with molecular weights greater than 300 and less than 400. These conditions are valid only when running `-j 4` or `-j 5` jobs.

Two example `cull.info` files are presented below.

```
## Distribution A for job type 1
Num rings                           2.600
Num charged donor groups            0.000
Num charged acceptor groups         0.000
Num reactive groups                 0.000
Molecular weight                  352.247
Num atoms                          41.800


## Filtering for job types 4 or 5
Num rings            0.99-3.01
Neutral Carbonyls    <1.0
```

8.4 The ligparse Command

`ligparse` can be run only from the command line.

Syntax:

`ligparse` {`-mae` *input-file*|`-sdf` *input-file*} [*options*]

You must specify an input file with either the `-mae` or the `-sdf` flag. The output file will be of the same type as the input file. Examples of the `ligparse` command are given below. The order of the specification of the options and the input file is irrelevant.

Options:

`-all` Select molecules that match all descriptors in the `cull.info` file. Only applies to job types 4 and 5, for which it is the default.

`-any` Select molecules that match any descriptor in the `cull.info` file. Only applies to job types 4 and 5.

`-bofix` Correct bond orders and formal charges in structures prior to SMARTS pattern matching.

`-c` *string* Input file name for a user-defined composite descriptor file.

`-csv` *string* Name of comma-separated value file.

`-d` Display a list of all descriptors and composite descriptors, then exit.

`-h` Display usage summary.

`-i` *string* Input file name for the culling information file to be used.

`-j` *int* Job type to be run.

- `1` Select a subset of structures that approximately reproduce user criteria.
- `2` Select a subset of structures to retain original property distributions.
- `3` Analyze structural components.
- `4` Select a subset of structures that match user criteria.
- `5` Remove structures that match user criteria.
- `6` Randomly select user-defined number/percent of structures from a larger dataset (used with `-seed` *integer*, see below).

`-k` *double* Percentage of compounds in original data set to keep after culling (0 to 100).

`-l` *string* Label for output with the culled structures.

`-mae` *string* Input file name for Maestro-formatted input files.

`-n` *int* Number of compounds in original data set to keep after culling (0 to number of compounds in dataset).

`-noprops` Do not include calculated properties in the output structure file.

`-o` *string* Text file to which data from running `ligparse` is sent.

`-pcsv` Write a comma-separated value file (spreadsheet), listing values for each molecule of all descriptors. If a file name is not specified with the `-csv` option, the file is written to *jobname*`.csv` in the working directory.

`-s` *string* Input file name for a user-defined descriptor file to be used.

`-sdf` *string* Input file name for MDL SD-formatted input files.

`-seed` *integer* Select the initial random seed which then directs the selection of compounds when using the `-j 6` option (see above).

`-sort` Sort the output into descriptors that were present in the data set and those that were not.

`-v` Display program version.

For information on how `ligparse` is called by `ligprep`, see Section 5.9 on page 69.

Examples:

To analyze structural components in a data set:

```
ligparse -j 3 {-mae|-sdf} input-file [ -s str-keys -c composite-types
-o output-file -pcsv -csv csv-file -sort -bofix]
```

To select a subset of structures that approximately reproduce user criteria:

```
ligparse -j 1 {-mae|-sdf} input-file -i criteria-file {-k|-n} what-to-keep
[-s str-keys -c composite-types -o output-file -pcsv -csv csv-file -sort
-bofix -l label]
```

To select a subset of structures to retain original property distributions in the data set:

```
ligparse -j 2 {-mae|-sdf} input-file {-k|-n} what-to-keep [-s str-keys
-c composite-types -o output-file -pcsv -csv csv-file -sort -bofix -l label]
```

To select a subset of structures that match user criteria:

```
ligparse -j 4 {-mae|-sdf} input-file -i criteria-file {-k|-n} what-to-keep
[-s str-keys -c composite-types -o output-file -pcsv -csv csv-file -sort
-l label -any -all -bofix]
```

To remove structures that match user criteria:

```
ligparse -j 5 {-mae|-sdf} input-file -i criteria-file {-k|-n} what-to-keep
[-s str-keys -c composite-types -o output-file -pcsv -csv csv-file -sort
-l label -any -all -bofix]
```

To select a random subset of structures:

```
ligparse -j 6 {-mae|-sdf} input-file {-k|-n} what-to-keep [ -o output-file
-pcsv -csv csv-file -l label -any -all -bofix -seed random-seed]
```

8.5 Input and Output Files

Valid input file names for `ligparse` are *jobname*`.mae` for Maestro files, and *jobname*`.sdf` or *jobname*`.sd` for SD files. The stem of the input file name, *jobname*, is used to construct the default output file name. The averages of descriptor values in the input file are written to *jobname*`.out`. The name of this file can be set with the `-o` option.

For job type 3, ligparse copies the structures from the input file to a Maestro output file, and adds the descriptors as properties to this file. The file is named *jobname*`-out.mae`.

For job types 1, 2, 4, and 5, `ligparse` writes a structure file named *jobname-label*`.mae` or *jobname-label*`.sdf`. Here, *label* has the default value of `cull` and can be set with the `-l` option. The input format dictates the file format of the culled structure file. For example, if the input file is a Maestro file, the output contains a subset of molecules in Maestro format. The properties calculated by `ligparse` are added to the output structure files for job types 4 and 5.

If the `-pcsv` option is given, an additional file is written, *jobname*`.csv`. This comma-separated file contains the value of each descriptor per molecule. The file name can be set with the `-csv` option.

Examples of `ligparse` commands and the files generated are given below.

Example 1:

```
ligparse -sdf database.sdf -j 1 -k 10 -i cull.info -l set1
```

Keeps 10 percent of `database.sdf` structures selected to maintain average values of descriptors defined in `cull.info`. Output includes 2 files:

`database.out`	File listing the results of culling and structural information for the starting set of molecules and the culled set of molecules for the descriptors of interest.
`database-set1.sdf`	MDL file of 10 percent of retained structures.

Example 2:

```
ligparse -mae database.mae -j 1 -k 20 -i ../cull.info    -o db1.stat -
pcsv
```

Keeps 20 percent of `database.mae` structures selected to maintain average values of the descriptors listed in the file, `../cull.info`. Output includes 3 files:

`database-cull.mae`	Maestro file of 20 percent of retained structures.
`db1.stat`	File listing the results of culling and structural information for the starting set of molecules and the culled set of molecules for the descriptors of interest.
`database.prop`	File listing the number of times each descriptor was identified per molecule.

Example 3:

```
ligparse -mae database.mae -j 2 -n 500 -l small
```

Keeps 500 `database.mae` structures selected to maintain average values of descriptors found in the original dataset. Output includes 2 files:

`database.out`	File listing the results of culling and structural information for the starting set of molecules and the culled set of molecules for the descriptors of interest.
`database-small.mae`	Maestro file of 50 percent of the retained structures.

Example 4:

```
ligparse -sdf database.sdf -j 3 -c my_composite.types
   -s my_str_keys.types
```

Determines the average values of a number of descriptors for structure in the SD format file, `database.sdf`. The descriptors are read from the `my_str_keys.types` file, with composite descriptor read from the `my_composite.types` file. Output includes 1 file:

`database.out`	File listing average number of various structural keys for compounds in `database.sdf`.

Getting Help

Schrödinger software is distributed with documentation in PDF format. 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*.

Maestro has automatic, context-sensitive help (Auto-Help and Balloon Help, or tooltips), and an online help system. To get help, follow the steps below.

- Check the Auto-Help text box, which is located at the foot of the main window. If help is available for the task you are performing, it is automatically displayed there. Auto-Help contains a single line of information. For more detailed information, use the online help.
- If you want information about a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Help menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- For information about a panel or the folder that is displayed in a panel, click the Help button in the panel. The Help panel is opened and a relevant help topic is displayed.
- For other information in the online help, open the Help panel and locate the topic by searching or by category. You can open the Help panel by choosing Help from the Help menu on the main menu bar or by pressing CTRL+H.

  To view a list of all available LigPrep–related help topics,choose LigPrep from the Categories menu of the Categories tab. Double-click a topic title to view the topic.

If you do not find the information you need in the Maestro help system, check the following sources:

- *Maestro User Manual*, for detailed information on using Maestro
- *Maestro Command Reference Manual*, for information on Maestro commands
- Frequently Asked Questions pages, at https://www.schrodinger.com/LigPrep_FAQ.html

The manuals are also available in PDF format from the Schrödinger Support Center. Information on additions and corrections to the manuals is available from this web page.

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, most of which can be obtained by entering `$SCHRODINGER/machid` at a command prompt:

- All relevant user input and machine output
- LigPrep purchaser (company, research institution, or individual)
- Primary LigPrep user
- Computer platform type
- Operating system with version number
- LigPrep version number
- MacroModel version number
- Maestro version number
- mmshare version number

Index

A

applyhtreat
- brief description ... 2
- full description ... 65

atoms
- chiral ... 71
- overlapping ... 75
- selecting ... 23–25

Auto-Help ... 30, 109

B

Balloon Help ... 30, 109
batch queues ... 59
bmin (MacroModel)
- brief description of LigPrep use ... 3
- full description of LigPrep use ... 76
- options for ... 35–36

Build panel ... 21
building structures ... 20–23
button menu ... 9

C

charged groups, neutralization of ... 67
chiralities
- atom property format ... 71
- change during tautomerization ... 99
- command options ... 35, 36
- enforcement in bmin (MacroModel) ... 77–79, 80
- GUI options ... 32
- retention of ... 32, 70
- unrealistic combinations ... 72

chirality information
- adjustment with hydrogen treatment ... 66
- derived from SD files ... 64

Command Script Editor panel ... 26
command scripts—*see* scripts
composite descriptors ... 101
- custom, ligprep option for ... 36

conformational search ... 77
conformations
- approximate energies ... 73
- double bond ... 70, 72, 98
- non-ring portions of structures ... 73
- ring ... 73
- ring_conf options ... 74

conventions
- document ... ix
- file names ... 43

counter ions, removal of ... 66
- command option ... 37
- GUI option ... 33

culling of structures ... 103
- examples ... 58, 104
- information files for ... 104

current working directory ... 6

D

deprotonation ... 67, 81
desalter
- brief description ... 2
- full description ... 66
- ligprep option for ... 37
- options ... 67

descriptors ... 101
- custom, ligprep option for ... 36
- using conditions with ... 104

directory
- current working ... 6, 27
- job submission ... 35
- output ... 27
- temporary ... 40

distributed processing ... 40
double-bond conformations ... 70, 72, 98

E

enantiomers, matching by ring_conf ... 73
energies
- estimates of conformational ... 73
- minimization of ... 76

entries, Project Table ... 13
- including, excluding, and fixing ... 17
- selecting ... 17
- sorting ... 15

environment variables
- DISPLAY ... 6
- SCHRODINGER ... 4, 5–6

Epik
- passing options to ... 38
- use for ionization and tautomerization . 33, 36

ePlayer ... 15, 16
error messages ... 86

examples
3D structure improvement 54
adjusting chemistry 55
applyhtreat 66
default settings 53
default variations 48
filtering structures 58
generating QikProp input files 58
generating variations 56
ligparse commands 107
one-to-one conversion 54
ring conformation variation 51
excluded entries 17
expansion 31

F

file I/O directory 27
files
composite descriptors 102
composite.types 102
culling information 104
descriptor 101
naming conventions 44
removal of 37, 43
schrodinger.hosts 40
size limits 45
splitting input 40
str_keys.type 101
temporary 43
See also input files, output files
filtering structures 103
command options 36
filters, project entry 17
fixed entries 18
force field
dependence of hydrogen treatment on 65
for bmin (MacroModel) 35
format conversion utilities 4
fragments, building structures from 20
full screen mode 8, 13
function key macros—*see* scripts

G

grow bond 21

H

Help panel 30, 109
host file 40
hydrogen atoms, adding or deleting 65
hydrogen bond acceptors, donors 101

I

included entries 17
input files
ligparse 101, 103
removal of 37, 43
size limits 47
splitting large 40
suffixes 44
valid structure formats 35
installation directory 4
ionizable groups
candidate list 86
customized patterns for 89
limit in ionizer 68, 82, 85
maximum total charge 85
option for maximum number 37
ionization states
command options 36
GUI options 33
restriction by pH or pK_a 84
using Epik for 33
ionizer
brief description 3
full description 81
input restrictions 84
LigPrep use of 68
options 81
output files 85
passing options to 38, 81
thresholds 83

J

job control
facility 39, 47
host file 40
incorporation of results 34
ligprep options 40
submission to remote hosts 34, 59
job submission directory 35
jobs, running in Maestro 28–29

L

license, LigPrep 4
  premin use of 76
  tools requiring 66, 67, 69, 70, 71, 74
  using with bmin (MacroModel) 76
  using with premin (MacroModel) 76
ligparse
  brief description 3
  composite descriptors file 102
  descriptor file 101
  examples 107
  full description 101
  ligprep options for 36
  LigPrep use of 69
  output files 107
  passing options to 38
ligprep
  input file size limit 47
  options 35
LigPrep - Start dialog box 34
LigPrep panel 31
limitations
  LigPrep 45
  MacroModel (bmin) 80
  neutralizer 68
  ring_conf 75
  stereoizer 72
limits
  input structure file 47
  ionizable groups in ionizer 68, 82, 85
  number of atoms 36
  output structures in ionizer 82
  ring conformations 33, 74
  stereoisomers 32, 70
  tautomers 33, 92
log file
  bmin (MacroModel) 79
  ionizer 82, 85
  location of 35
  saving Maestro 30

M

MacroModel
  brief description of LigPrep use 3
  chirality enforcement 80
  full description of LigPrep use 76
  ring closure failures 80
macros—*see* scripts
Maestro
  main window 6, 7
  menus 8
  quitting 30
  running jobs from 28–29
  scratch projects 13
  starting 6
  undoing operations 28
Maestro-formatted files, input 35
maesubset
  LigPrep use of 2
  options 65
main window 7
menu button 9
metal-ligating groups 101
meta-options 38
Monitor panel 29
mouse functions 5
  Project Table panel 18–19
  Workspace 12

N

neutralizer
  brief description 2
  full description 67
  limitations 68
  options 68

O

online help 30
opcodes 76–79
options
  desalter 67
  ionizer 81
  ligprep 35
  maesubset 65
  meta 38
  neutralizer 68
  para_ligprep 41
  passing from ligprep to tools 38
  premin 76
  QikProp input files 39
  ring conformations 37
  ring_conf 74

sdconvert ... 63, 64
stage selection ... 37
stereoisomers ... 35, 36, 37
stereoizer ... 71
tautomerizer ... 92
tautomers ... 37, 38
output files
ionizer ... 85
ligparse ... 107
options ... 35
premin ... 75
ring_conf ... 75
suffixes ... 44

P

para_ligprep
files ... 42
options ... 41
parities ... 54, 64
pH, restriction of ionization states by ... 33, 84
pK_a, restriction of ionization states by ... 84
Preferences panel ... 27, 28
premin
brief description ... 3
full description ... 75
options ... 76
output files ... 75
problematic structures
premin files containing ... 75
removal of ... 37
retention of ... 45
ring_conf files containing ... 75
product installation ... 109
project entries, *see* entries, Project Table
Project Facility, introduction ... 13
Project Table panel ... 15
menus ... 16
mouse functions ... 18–19
shortcut keys ... 19
projects ... 13
protonation ... 67, 81
Python scripts—*see* scripts

Q

QikProp input files
example for ... 58
option for ... 39
quitting Maestro ... 30

R

recursive SMARTS ... 93
remote hosts, submitting jobs to ... 34, 59
restrictions, ionizer input structures ... 84
ring closure failures, bmin (MacroModel) ... 80
ring conformations, generating ... 73
GUI option ... 33
ligprep option ... 37
ring_conf option ... 74
ring_conf
brief description ... 3
full description ... 73
limitations ... 75
options ... 74
output files ... 75
passing options to ... 38
rings, flexible
identification of ... 73
missing templates ... 75
templates ... 73
rings, fused, misidentification ... 75
rings, rigid
identification of ... 73
ligprep option for handling ... 36
ring_conf option for handling ... 74

S

Schrödinger contact information ... 110
schrodinger.hosts file ... 40
scratch entries ... 14
scratch projects ... 13
scripts
function key macros ... 27
macros ... 27
Maestro command ... 26
Python ... 25
sdconvert
LigPrep use of ... 2, 3
options ... 63, 64
SD-formatted files
converting to Maestro format ... 63
input, output of ... 35
parities and bond directions ... 64

selecting objects in the Workspace 9, 23
shortcut keys
main window 13
Project Table panel 19
SMARTS patterns
`ligparse` use 101
recursive..... 93
`tautomerizer` use..... 97
SMILES files, converting to Maestro format.... 64
SMILES format, for input..... 35
solvation model, for `bmin` (MacroModel) 35
stage of LigPrep process, option for selecting .. 37
stereoisomers
command options..... 35, 36, 37
generation of..... 54, 70
GUI options 32
maximum number..... 54
`stereoizer`
brief description..... 3
full description..... 70
limitations..... 72
options 71
passing options to 38
structures
building..... 20–23
culling 103
deprotonated form 84
displaying in sequence..... 15
format conversion 63–64
input, retaining..... 68, 92
`ionizer` input restrictions..... 84
`ionizer` output limit 82
maximum total charge 85
protonated form 84
selecting for conversion..... 37
subset selection utilities..... 4
with many ionizable groups..... 68, 85

T

tautomer database
`ligprep` option for 37
`tautomerizer` option for 92
`tautomerizer`
brief description..... 3
full description..... 91
LigPrep use of..... 69
options 92
pattern matching 97
suppressing use of..... 37
tautomers
command options..... 37, 38
definition..... 91
GUI option..... 33
screening by probability 38, 98
specifying custom database 92
types recognized 97
technical support 30
temporary files 43
thresholds, `ionizer` 83
throughput 1
enhancing with `para_ligprep`..... 40
toolbar
Build panel..... 22–23
main window 9–12
Project Table panel 15–16
torsional constraints, for `bmin` (MacroModel) . 36

U

undoing Maestro operations..... 28
utilities, additional 4

W

Workspace
description 6
full screen mode 8, 13
including, excluding, and fixing entries 17
mouse functions..... 12
scratch entries 14

120 West 45th Street
32nd Floor
New York, NY 10036

101 SW Main Street
Suite 1300
Portland, OR 97204

3655 Nobel Drive
Suite 430
San Diego, CA 92122

Dynamostraße 13
68165 Mannheim
Germany

QuatroHouse, Frimley Road
Camberley GU16 7ER
United Kingdom

SCHRÖDINGER®