

# Prime 3.1

## User Manual

Prime User Manual Copyright © 2012 Schrödinger, LLC. All rights reserved.

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

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

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

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

Revision A, September 2012

# Contents

---

Document Conventions .....	xi
Chapter 1: Introduction .....	1
1.1 The Prime Suite in Maestro .....	1
1.2 Prime Documentation .....	3
1.3 Running Schrödinger Software .....	4
1.4 Citing Prime in Publications .....	6
Chapter 2: Using Prime–Structure Prediction .....	7
2.1 Prime Runs and Maestro Projects .....	7
2.2 General Panel Layout .....	8
2.3 The Menu Bar .....	10
2.3.1 The File Menu .....	10
2.3.2 The Edit Menu .....	11
2.3.3 The Display Menu .....	11
2.3.4 The Step Menu .....	14
2.4 The Prime Toolbar .....	14
2.5 The Prime Sequence Viewer .....	15
2.6 Prime Job Options and Control .....	18
2.6.1 The Job Options Dialog Box .....	18
2.6.2 Setting Up and Launching Jobs .....	18
2.6.3 Monitoring Jobs .....	19
2.6.4 Adding Structures to the Project Table .....	19
Chapter 3: Structure Prediction: Initial Steps .....	21
3.1 Input Sequence Step .....	21
3.2 Find Homologs Step .....	23
3.2.1 Importing Homologs .....	23
3.2.2 Searching for Homologs .....	23
3.2.3 Find Homologs Search Options .....	25

3.2.4	Search Results.....	26
3.2.5	Multiple Templates and Structure Alignment.....	29
3.2.6	Continuing to the Next Step .....	30
3.2.7	Saving Runs With Different Templates.....	30
<b>Chapter 4:</b>	<b>Structure Prediction—Edit Alignment.....</b>	<b>31</b>
<b>4.1</b>	<b>Overview of the Edit Alignment Step.....</b>	<b>31</b>
<b>4.2</b>	<b>Using Existing Alignments .....</b>	<b>31</b>
<b>4.3</b>	<b>Generating an Alignment .....</b>	<b>33</b>
4.3.1	Generating SSPs .....	33
4.3.2	Importing and Exporting SSPs .....	33
4.3.3	Adjusting SSPs .....	34
4.3.4	Running the Alignment.....	35
4.3.5	Align Program Technical Details .....	36
<b>4.4</b>	<b>Manually Editing an Alignment.....</b>	<b>37</b>
<b>4.5</b>	<b>Error Messages.....</b>	<b>37</b>
<b>Chapter 5:</b>	<b>Structure Prediction—Build Structure .....</b>	<b>39</b>
<b>5.1</b>	<b>The Build Process .....</b>	<b>39</b>
<b>5.2</b>	<b>Choosing the Model Type .....</b>	<b>39</b>
5.2.1	Building a Composite Model .....	40
5.2.2	Building a Consensus Model .....	41
5.2.3	Building Multimers.....	41
<b>5.3</b>	<b>Preparing to Build .....</b>	<b>42</b>
5.3.1	Setting Constraints.....	42
5.3.2	Including Ligands and Cofactors.....	42
5.3.3	Build Options.....	42
<b>5.4</b>	<b>Building the Structure .....</b>	<b>44</b>
<b>5.5</b>	<b>Technical Notes for the Build Structure Step.....</b>	<b>45</b>

---

Chapter 6: Refining Protein Structures .....	47
6.1 Preparing Structures for Refinement.....	47
6.2 Using the Refinement Panel .....	51
6.3 Setting Up an Implicit Membrane .....	52
6.4 Refining Loops.....	54
6.5 Predicting Side Chains .....	56
6.6 Minimizing Structures .....	57
6.7 Analyzing the Energy.....	58
6.8 Setting Refinement Options .....	58
6.8.1 General Options.....	59
6.8.2 Dielectric Options.....	59
6.8.3 Side-Chain Prediction Options.....	60
6.8.4 Loop Refinement Options .....	61
6.9 Applying Constraints.....	63
6.10 Technical Details for Refinement .....	66
6.10.1 Loop Refinement.....	66
6.10.2 Cooperative Loop Refinement .....	67
6.10.3 Side-Chain Prediction .....	68
6.11 Refinement with Nonstandard Residues .....	69
6.12 Output of Refinement Jobs .....	69
Chapter 7: Maestro Protein Structure Alignment .....	71
7.1 The Protein Structure Alignment Panel.....	71
7.2 The Align Binding Sites Panel .....	73
Chapter 8: Docking Covalently Bound Ligands.....	77
8.1 Running Covalent Docking from Maestro .....	77
8.1.1 Selecting the Ligands.....	78
8.1.2 Defining the Ligand Reactive Groups .....	79

8.1.3	Specifying the Receptor Bond to Break .....	79
8.1.4	Specifying Receptor Residues to Sample .....	80
8.1.5	Setting the Number of Output Poses .....	80
8.1.6	Running the Job .....	80
8.1.7	Output .....	81
<b>8.2</b>	<b>Running Covalent Docking from the Command Line .....</b>	<b>81</b>
<b>Chapter 9: Prime MM-GBSA .....</b>		<b>83</b>
<b>9.1</b>	<b>Running Prime MM-GBSA from Maestro .....</b>	<b>83</b>
<b>9.2</b>	<b>Running Prime MM-GBSA from the Command Line .....</b>	<b>85</b>
9.2.1	Command Input File .....	88
9.2.2	ASL Expressions .....	88
<b>9.3</b>	<b>Prime MM-GBSA Output .....</b>	<b>88</b>
<b>Chapter 10: Command Syntax .....</b>		<b>91</b>
<b>10.1</b>	<b>blast—Run Blast Searches .....</b>	<b>91</b>
<b>10.2</b>	<b>prime—Build and Refine Protein Structures .....</b>	<b>94</b>
10.2.1	Command Input File .....	94
10.2.2	Constraint Syntax .....	103
10.2.3	Restarting an Incomplete Loop Refinement Job .....	104
10.2.4	Environment Variables .....	104
10.2.5	Examples .....	105
10.2.5.1	Model Building .....	105
10.2.5.2	Energy calculation .....	105
10.2.5.3	Side-Chain Prediction .....	106
10.2.5.4	Minimization .....	106
10.2.5.5	Loop Prediction .....	106
<b>10.3</b>	<b>ska—Protein Structure Alignment .....</b>	<b>107</b>
<b>10.4</b>	<b>ssp—Secondary Structure Prediction .....</b>	<b>112</b>
<b>10.5</b>	<b>sta—Single Template Alignment .....</b>	<b>113</b>
<b>10.6</b>	<b>tfm—Tertiary Folding Module .....</b>	<b>116</b>

---

Chapter 11: Command-Line Utilities .....	119
11.1 Multiple Template Alignment: <code>structalign</code> .....	119
11.2 Format Conversion: <code>seqconvert</code> .....	121
11.3 Structure Alignment: <code>align_binding_sites</code> .....	122
11.4 Database Update Utilities .....	124
11.4.1 <code>update_BLASTDB</code> .....	124
11.4.2 <code>rsync_pdb</code> .....	125
Appendix A: Third-Party Programs .....	127
A.1 Location of Third-Party Programs .....	127
A.1.1 BLAST/PSI-BLAST, Pfam, and the PDB .....	127
A.1.2 Secondary Structure Prediction .....	128
A.2 Third-Party Program Use Agreement .....	128
Appendix B: Environment Variables .....	129
Appendix C: File Formats .....	131
C.1 Input File Formats .....	131
C.2 Output File Formats .....	131
Appendix D: Error Messages and Warnings .....	133
D.1 Input Sequence .....	133
D.2 Find Homologs .....	133
D.2.1 Error Messages .....	133
D.2.2 Warnings in Homologs Table .....	134
D.3 Edit Alignment .....	135
D.4 Build Structure .....	135
D.5 Refinement .....	135
Getting Help .....	137

Glossary..... 141

Index..... 143



---

# Document Conventions

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

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

Links to other locations in the current document or to other PDF documents are colored like this: [Document Conventions](#).

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

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

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

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

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



# Introduction

Prime 3.1 is a highly accurate protein structure prediction suite of programs. The Homology Modeling workflow incorporates the complete protein structure prediction process from template identification, to alignment, to model building. Refinement can then be done from a separate panel, and involves side-chain prediction, loop prediction, and minimization.

The Prime interface was designed to accommodate both novice and expert users, while the underlying programs were designed to produce superior results in a variety of applications, including:

- High-resolution homology modeling
- Refinement of active sites
- Induced-fit optimization
- Structure-based functional annotation
- Generation of alternate loop conformations

## 1.1 The Prime Suite in Maestro

To run Prime you use Maestro, the graphical user interface (GUI) for all of Schrödinger's products. For help with a Maestro panel, click the Help button. For more information on the interface itself, see the [Maestro User Manual](#). Maestro has several "profiles" that present different layouts of the interface. For protein modeling the main profiles are the main Maestro profile and the BioLuminate profile.

The Prime suite provides the tools for four main tasks: homology modeling, structure refinement, covalent docking, and binding energy estimation, all of which are accessible from the Maestro interface.

**Homology modeling** can be done through one of two interfaces, the Structure Prediction wizard, and the Multiple Sequence Viewer. In both you can perform the required steps leading from a protein sequence to the construction of a 3D structure.

To open the Structure Prediction wizard interface:

From Maestro:

- Choose Applications → Prime → Structure Prediction or Tasks → Homology Modeling in the main window, then click the Structure Prediction Wizard icon in the Homology Modeling panel.



From BioLuminate:

- Choose Tasks → Homology Modeling → Advanced Homology Modeling in the main window.

To open the Multiple Sequence Viewer interface:

From Maestro:

- Choose Applications → Prime → Structure Prediction or Tasks → Homology Modeling in the main window, then click the Multiple Sequence Viewer icon in the Homology Modeling panel.



From BioLuminate or Maestro:

- Choose Tools → Multiple Sequence Viewer in the main window.

For information on using this interface see the [Multiple Sequence Viewer](#) document.

**Structure refinement** is a facility for refining protein structures that have been imported into Maestro or added to the Project Table from the Homology Modeling workflow or another Schrödinger application.

To open the Refinement panel:

From Maestro:

- Choose Applications → Prime → Refinement or Tasks → Protein Refinement in the main window.

From BioLuminate:

- Choose Tasks → Loop and Side-Chain Prediction in the main window.

**Covalent docking** is a facility for “docking” ligands that are covalently bound to the receptor. It uses designated reactive bonds in the ligand and a receptor bond to sample the possible attachment points, and performs a loop prediction to sample the ligand and specified parts of the receptor. The result is a set of “poses” of the ligand, docked to the receptor.

To open the Covalent Docking panel:

From Maestro:

- Choose Applications → Prime → Covalent Docking or Tasks → Docking → Covalent Docking in the main window.

From BioLuminate:

- Choose Tasks → Ligand Tasks → Docking → Covalent Docking in the main window.

Binding energy estimation can be done with Prime MM-GBSA which is a facility for calculating ligand binding energies within the MM-GBSA continuum solvation model.

To open the Prime MM-GBSA panel:

From Maestro:

- Choose Applications → Prime → MM-GBSA or Tasks → Binding Energy Estimation → MM-GBSA in the main window.

From BioLuminate:

- Choose Tasks → Ligand Tasks → Binding Energy Estimation → MM-GBSA in the main window.

## 1.2 Prime Documentation

The *Prime User Manual* describes the Prime suite, including information about:

- Essentials of using the Prime–SP module: runs and projects, job control, and the layout and common features of step panels
- Running a structure prediction: the Input Sequence, Find Homologs, Edit Alignment, and Build Structure steps
- Using the Prime–Refinement module
- Performing covalent docking
- Running Prime MM-GBSA calculations
- Running Prime jobs from the command line

This document expands on the information provided in the Maestro online help. In addition to describing steps, features, and options, this document suggests ways to determine which choices are the most useful, describes results, and provides some background and technical detail. Terms pertinent to Prime are defined in the [Glossary](#).

For a tutorial introduction to Prime, see the [Prime Quick Start Guide](#). For information on installing Prime and the related third-party products, see the [Installation Guide](#).

For information about Schrödinger's Induced Fit Docking protocol, which uses Prime and Glide to optimize docking by including ligand-induced flexibility in the receptor active site, see the document [Induced Fit Docking](#).

Maestro online help includes topics for Prime. See [page 137](#) for a discussion of help in Maestro, including tool tips (Balloon Help) and technical support contacts. For a fuller description of the help facility, see [Chapter 17](#) of the *Maestro User Manual*.

### 1.3 Running Schrödinger Software

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

#### Linux:

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

```
csh/tcsh:      setenv SCHRODINGER installation-directory  
bash/ksh:      export SCHRODINGER=installation-directory
```

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

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

You can start the Maestro interface with the following command:

```
$SCHRODINGER/maestro &
```

It is usually a good idea to change to the desired working directory before starting Maestro. This directory then becomes Maestro's working directory.

**Windows:**

The primary way of running Schrödinger applications on a Windows platform is from a graphical interface. To start the Maestro interface, double-click on the Maestro icon, on a Maestro project, or on a structure file; or choose Start → All Programs → Schrodinger-2012 > Maestro. You do not need to make any settings before starting Maestro or running programs. The default working directory is the Schrodinger folder in your documents folder (Documents on Windows 7/Vista, My Documents on XP).

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

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

You can open these shells from Start → All Programs → Schrodinger-2012. You do not need to include the path to a program or utility when you type the command to run it. If you want access to Unix-style utilities (such as `awk`, `grep`, and `sed`), preface the commands with `sh`, or type `sh` in either of these shells to start a Unix-style shell.

**Mac:**

The primary way of running Schrödinger software on a Mac is from a graphical interface. To start the Maestro interface, click its icon on the dock. If there is no Maestro icon on the dock, you can put one there by dragging it from the SchrodingerSuite2012 folder in your Applications folder. This folder contains icons for all the available interfaces. The default working directory is the Schrodinger folder in your Documents folder (`$HOME/Documents/Schrodinger`).

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

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

To launch Structure Prediction jobs from Maestro, you must have access to the PDB, either in a local installation or to the web site. You must set an option in the Jobs tab of the Preferences panel in Maestro to use web-based searching—see [Section 13.2.12](#) of the *Maestro User Manual*. If you are using a local installation and the PDB database is not installed into `$SCHRODINGER`, you must set the `SCHRODINGER_PDB` environment variable to the path to the PDB installation. For information on how to set environment variables, see [Appendix A](#) of the *Installation Guide*.

## **1.4 Citing Prime in Publications**

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

Prime, version 3.1, Schrödinger, LLC, New York, NY, 2012.



# Using Prime–Structure Prediction

This chapter presents the essentials of using Prime–Structure Prediction (Prime–SP):

- Starting, naming, and saving Prime–SP runs in Maestro projects
- Navigating within and between runs
- Job control, job monitoring, and Prime–SP job options
- Using Prime–SP step panels, including their common layout and features
- Using the Prime menu bar and Prime toolbar

## 2.1 Prime Runs and Maestro Projects

A single execution of the Prime structure prediction workflow using a particular set of choices of templates, paths, and settings is called a *run*.

Prime runs are stored within Maestro projects. Structures generated in completed Prime runs are added to the Maestro Project Table. If you are unfamiliar with Maestro projects and the Maestro Project Table, you may want to review the Project Table and Project Facility online help topics. For more information, see the [Maestro User Manual](#).

Maestro always has a project open. If you begin to work without selecting or naming a project, Maestro creates a *scratch* (unnamed) project.

Prime always has a run open, which belongs to the current project (whether it is a named project or a scratch project). If you begin the Prime workflow without specifying an existing run, the default name for the current run is `run1`. Multiple runs can be performed within a project, but you must save the project in order to save the data in the runs.

Each project can be displayed in the Project Table panel. In Prime, Project Table entries are usually finished model structures with their properties. Prime stores data in the project as each step in a run is performed, but most of this data is not displayed in the Project Table, which can remain empty until a run is completed. For this reason you should save scratch projects in which you have done work, even if the Project Table is empty.

Two Maestro projects can be merged into one. If you merge two Maestro projects, each of which contains one or more Prime runs, all the runs are included in the merged project. If two runs have the same name, Maestro automatically makes the names unique by appending `-mrg1` to one of them.

Maestro projects containing multiple runs are useful for comparing the results of workflows in which different selections and options were chosen: for example, a different template selected for building, or a job performed with a different set of options. Creating these multiple runs may involve navigating to an earlier step in your current run, making different selections, and then navigating forward again in a new run. For a description of the features used in navigation (the Next button, the Back button, and the Guide), see [Section 2.2 on page 8](#).

If you return to an earlier step in your current run and make different choices as you proceed forward, you will be asked whether you want to overwrite your existing data, ignore changes, or cancel. If you want to use the new choices but keep the data in your current run for comparison rather than overwriting it, you can start a new run. The commands for saving the old run, starting a new one, and switching between Prime runs are on the File menu.

The File menu commands for managing runs are described in [Section 2.3.1 on page 10](#).

### **To backtrack and try a different set of choices without overwriting your current workflow:**

1. Choose Save As from the File menu.

This saves the current run under the old name (`run1` is the default for the first run in a new project) and creates a copy under the new name.

You are now in a new workflow identical to the first except for its name.

2. Navigate to the step where you want to make a different choice. If you want to go back more than one step, click the step name in the Guide.

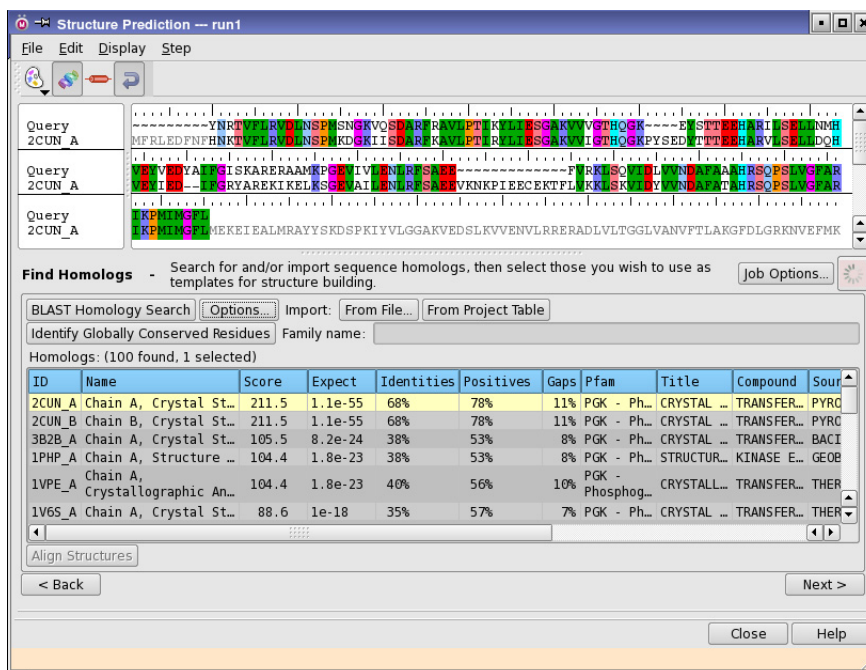
## 2.2 General Panel Layout

Each step in the Structure Prediction panel has a common set of features as well as features that are unique to the step. This section discusses these common elements, their location, and their function. An example of a Prime–SP panel is shown in [Figure 2.1](#).

In the upper part of the panel are the menu bar, the toolbar, and the Prime sequence viewer. These are described in the following sections.

The center of the panel contains the Step View, which includes the step name and a brief description of the step, followed by a row of buttons, and data relevant to the step.

The step usually includes some action buttons, one of which runs the main action of the step, which may involve launching a job. See [Section 2.6 on page 18](#) for more information on running Prime jobs. An Options button may be located to the right of the action buttons. Most buttons, including those in the Prime toolbar, the Prime Guide, and the Maestro toolbar, have tool tips that you can view by pausing the cursor over the button.



**Figure 2.1. The Find Homologs step of the Structure Prediction panel.**

The job status icon to the right of the step description turns green and spins when you launch a job. When the icon turns pink again, the job is finished. Click the icon to open the Monitor panel. For more information on job control, see [Section 2.6 on page 18](#).

The data relevant to the step is usually shown in table format. You can view the complete text of a truncated table entry as a tool tip by pausing the pointer over the text. To resize a table column, drag a column border. You can sort the tables in the Step View by clicking the column heading, once for ascending order, twice for descending order. You can also export data from any table in CSV or HTML format from the table shortcut menu.

Below the Step View are the Back and Next buttons, the primary way to navigate between steps. You can perform a simple workflow by clicking Next after completing each step. However, it is sometimes useful to go back to previous steps, make different choices, and proceed forward again using the Next button or the Guide, explained next.

The Guide, below the Back and Next buttons, is a diagram of the Prime workflow as a series of steps, each step represented by a button. It is hidden by default. To display the Guide, select the Guide check box in the Step menu.

The Guide has two purposes: to show your current location in the Prime–SP process, and to let you navigate back and review or change the input at previous steps in the process. The current step is highlighted in the Guide diagram. All previously accessed steps, forward or backward, are available. You can go to any of these steps by clicking them in the Guide.

Clicking on an available step button in the Guide displays the panel for that step. If a step is not yet available, its button is dimmed. Available steps include those that have already passed through in the current run, the present step, and (once step-specific criteria have been met) the next step beyond the current location. When the next step is available, clicking on it behaves the same as clicking the Next button; likewise clicking on the previous step behaves the same as clicking the Back button.

When you return to a previous step and change data, then move forward in the workflow to the following step, you are given a choice of what to do with the existing data in the following step. You can choose **Overwrite**, to discard the existing data and use the new data, **Ignore Changes**, to discard the new data and use the existing data, or **Cancel**, to remain in the current step.

**Note:** You cannot launch searches and jobs by clicking the steps in the Guide. To launch actions pertaining to a step, click the appropriate buttons within the step panel.

The Close button appears below the Guide on the lower left of the panel. To the right is the Help button, which opens online help related to the current step. Below these buttons is the status area, which shows text related to the current action or the action performed by a control in the panel.

## 2.3 The Menu Bar

The menu bar consists of four menus: File, Edit, Display, and Step. Some of the options in these menus are also available as toolbar buttons.

### 2.3.1 The File Menu

The File menu contains commands for managing runs, exporting sequences and setting job options. The commands are described below.

#### New

Asks for a new run name (all run names must be unique), and starts a new run at the Input Sequence step.

### Save As

Copies the current run under a new run name and switches to the new run. The original run is saved under the original name. If you open a new project and begin a workflow without explicitly giving the run a name, it will be given the default name of run1.

### Rename

Asks for a new name for the current run. All run names must be unique.

### Delete

Deletes the current run, on confirmation.

### Open

Displays options for switching to another run. The list includes the four most recently used runs. If there are more than four runs in the current project, click **More** to select a run by name.

### Export Sequences

Exports the sequences in the Prime sequence viewer to a file.

### Job Options

Opens the Structure Prediction - Job Options dialog box so that you can set options for running jobs for each stage of the workflow. See [Section 2.6 on page 18](#) for information on job options.

## 2.3.2 The Edit Menu

The Edit menu contains commands for editing sequences and secondary structure predictions. These editing tools are also available from the Prime toolbar, and are described with the toolbar in [Section 2.4 on page 14](#).

## 2.3.3 The Display Menu

The Display menu includes the following commands for controlling the appearance of the sequence viewer and structures in the Maestro Workspace.

### Color Scheme

This command opens a menu of coloring options. In Prime–SP, the coloring schemes act as modes; that is, only one color scheme can be used at any given time. Each step in Prime has a default color scheme. You can use this menu to choose a different one. When amino acid sequences are added to the sequence viewer, they are colored according to the scheme that is currently selected.

When the structure associated with a sequence is displayed in the Workspace, it is generally colored using the same scheme as the sequence. However, protein structures in the Workspace are sometimes colored using other color schemes:

- When a PDB structure is imported, it is displayed in the Workspace using an error reporting color scheme in which complete recognized residues are gray, incomplete residues are red, unknown or nonstandard residues are orange, residues with multiple location indicators are green, and recognized residues with some unrecognized atoms are blue. For more information about conversion of PDB structures, see [Section 3.1.6](#) of the *Maestro User Manual*.
- When the Build Structure step is complete, a color scheme is applied to the structure in which blue atoms are those derived directly from the template, cyan atoms have backbone atoms from the template, and predicted side chains, and red atoms have been predicted or modeled.



The Prime color schemes are:

- None: All sequences are colored standard gray.
- Color by Sequence: The query sequence is colored standard gray. Each of the other sequences uses a different color.
- Residue Type: Each amino acid sequence is colored according to the array of residue type colors.
- Residue Property: Each amino acid sequence is colored according to the array of residue property colors (fewer colors than residue type).
- Residue Matching: The query and templates are colored in a pairwise manner. Matching residues are colored according to their residue type, and nonmatching residues are colored standard gray. If more than one template is chosen, the query will have more colored residues because the effect is cumulative. If only the query sequence is shown, it is colored standard gray for all residues.
- Residue Homology: Aligned query and template residues that are identical (highly conserved) are colored red. Aligned query and template residues that are similar according to the BLOSUM62 scoring matrix (conserved) are colored yellow. Gray is used for residues with no homology.
- Secondary Structure: The query sequence is colored standard gray. All other AA sequences should have SSAs, which are colored according to the SSA colors.

- **Template ID:** Only available for the Comparative Modeling path. The query sequence is colored standard gray. The predicted structure is colored according to template ID for each region (using the cycle colors). Each template sequence is colored with its color, but only in the region that was used (the rest of the template is colored dark gray to indicate it is not used).
- **Proximity:** At any step in which the sequence viewer is displaying sequences that have associated structure (all steps except Input Sequence), color by proximity mode is available. When you select this color scheme, the current sequence colors do not change immediately. Instead, when the mouse is over a sequence that has structure, the cursor changes to indicate that proximity can be calculated. You can then select one or more contiguous residues in the sequence, and the system will color that sequence based on the proximity to the selected residues. No other sequences will have modified colors (even if they have structure too and are in proximity of the selected sequence).

Proximity is calculated based on the shortest distance between any two atoms in two residues, not including the bonded C and N atoms in the residues next to the selected residues.

To change the proximity cutoff, choose Preferences from the Display menu and change the number in the Proximity cutoff text box.

### Font Size

The font size of the text in the sequence viewer can be changed using this option. The menu offers a selection of point sizes for the text. The default is 10 points.

### View Template Structure

View the template structures in the Maestro Workspace.

### View SSA

This option controls whether secondary structure assignments are shown in the sequence viewer. The default is not to show secondary structure assignments.

### View Full Pfam Annotation

Select this option to view the full Pfam annotation. By default, only the highly conserved residues are shown in the annotation.

### Legend

Opens the Legend panel, which includes the assignment of each color and symbol used in each color scheme and sequence representation. Click the Colors tab and select a Color scheme

from the list, or click the Symbols tab and select a type of Sequence data, such as Pfam/HMMER or SSP. This option is also available when you right-click on a sequence in the sequence viewer.

### Preferences

Opens the Preferences dialog box, which has a single item:

- Proximity cutoff—The value entered in this text box defines proximity for the Proximity color scheme. The default is 4.00 Å.

### 2.3.4 The Step Menu

The Step menu provides another way to navigate between steps in Prime. All the steps are listed on the menu. The current step is selected. To go to a different step, select that step on the menu. Steps that are not available are dimmed.

You can also show or hide the Guide, by selecting Guide on this menu. When the check box is selected, the Prime Guide is displayed. Deselect the check box to hide the Guide. The Guide is hidden by default.

## 2.4 The Prime Toolbar

The Prime toolbar is the row of buttons just under the Prime menu bar. The Prime toolbar provides convenient access to commonly used Prime tasks. Pause the pointer over a button to see its name in a tool tip and its description in the status bar. These tasks are also available from the Edit and Display menus. The toolbar buttons are shown below as they appear on the toolbar, with their names, the corresponding menu, and their functions.



Select (Edit menu): Select a region of the sequence to see the corresponding residues highlighted in the 3D Workspace.



Crop (Edit menu): The cropping tool, which can be used in the Build Structure step to select ends of the query sequence that are not to be used for building.



Edit SSP (Edit menu): Edit secondary structure predictions. Highlight a region of the SSP, and then type either H for  $\alpha$ -helix, E for  $\beta$ -strand, or - for loop.



Edit Query Sequence (Edit menu): Edit the query sequence, for example to remove extra terminal residues, remove insertions, or mutate residues. Opens a dialog box in which you can edit the sequence as a text string.





**Slide Freely (Edit menu):** Use this button to edit the alignment between query and template. Selecting a residue and sliding to the right creates N-terminal gaps. Selecting a residue and sliding to the left creates C-terminal gaps.



**Slide As Block (Edit menu):** Use this button to edit the alignment between query and template. Selecting a residue and sliding to the right creates N-terminal gaps. Selecting a residue and sliding to the left slides all residues as a block to the left.



**Add Anchors (Edit menu):** Set alignment constraints by setting anchors at residue positions. To remove anchors, click on the same residue again.



**Lock Gaps (Edit menu):** To prevent gaps from being collapsed during manual editing, lock the gaps using this button. Locked gaps are indicated with a - symbol.



**Unlock Gaps (Edit menu):** To allow gaps to collapse during manual editing, unlock the gaps using this button. Unlocked gaps are indicated with a ~ symbol.



**Color Scheme (Display menu):** Select a color scheme for coloring sequences in the sequence viewer. If there is a structure in the 3D Workspace associated with the sequence, it will also be colored.



**View Template Structure (Display menu):** Use this button to display and undisplay template structures in the Workspace.



**View SSA (Display menu):** Use this button to turn the secondary structure assignment (SSA) in the sequence viewer on and off.



**Wrap Sequences (Display menu):** Wrap sequences that are longer than the display area of the sequence viewer.

## 2.5 The Prime Sequence Viewer

The sequence viewer displays the sequences for the query and templates, and in some steps it shows secondary structure assignments (SSAs) and secondary structure predictions (SSPs). It is divided into two sections. The name of the sequence, SSA, or SSP is given in the left section, and the sequence, SSP, or SSA itself is displayed in the right section. The sequence viewer includes a shortcut menu (right click) that provides access to some functions.

To resize the sequence viewer, drag the dotted line below the sequence viewer. If you cannot enlarge it any further, drag the bottom border of the panel to make the panel larger, and then you will be able to enlarge the sequence viewer some more. You can also drag the dotted line between the right and left sections of the sequence viewer to resize the sections.

The sequence viewer can be configured to display a single line for each sequence that can be scrolled horizontally, or the sequences can be wrapped so that they scroll vertically. The configuration is controlled by the **Wrap Sequences** option on the sequence viewer shortcut menu.

The sequence viewer also includes a “ruler”, which marks the residue position relative to the leftmost residue displayed in the sequence viewer. This residue position is displayed in parentheses in the tool tip for each residue. The positions are arbitrary, and can change when you display residues in the viewer. They are intended to help you keep track of the sequence as you scroll the sequence viewer.

You can select a single residue in a sequence by clicking on it, and you can use control-click to select or deselect individual residues. To select a range of residues, you can click the first and shift-click the last, or drag over the residues. The sequence in which the residues are selected is determined by where you start to drag. Likewise, you can invert the selection of multiple residues by holding down **CTRL** when you drag.

The commands that are available on the shortcut menu are:

### Show All/Hide All

This command is present when you right-click on a sequence or an SSP. It can be used to show all the family and SSP information that is associated with a sequence. It is equivalent to clicking the + or – sign to the left of the sequence name.

### Run SSP

Run a secondary structure prediction for a sequence. This command is only present for sequences, in the **Find Homologs**, **Fold Recognition**, and **Edit Alignment** steps.

### View SSA

Show or hide the secondary structure assignment. This command is only present when you right-click on an SSA or a sequence that has an SSA.

### View Full Pfam Annotation

Show the full Pfam annotation. By default, only the highly conserved residues are shown in the annotation. This command is only present when you right-click on a Pfam annotation.

### Revert

Revert the SSP to that generated by the SSP program. This command is useful to restore the original SSP after making changes. It is only present for SSPs, and is only available in the **Edit Alignment** step.

### Delete

Delete the SSP that you right-clicked on. This command is only present for SSPs, and is only available in the Edit Alignment step.

### Export Single Sequence

Export an SSP to a file. A file selector opens, in which you can navigate to a location and name the file. This command is only present for SSPs.

### Save Image

Save an image of the sequence viewer. A file selector opens, in which you can select a format, from PNG, TIFF, and JPEG, and save the image. This command is always present on the shortcut menu.

### Export Sequences

Export the sequences to a single FASTA file. A file selector opens, in which you can navigate to and name the file. You can also export sequences by choosing File → Export Sequences. This command is always present on the shortcut menu.

### View Template Structure

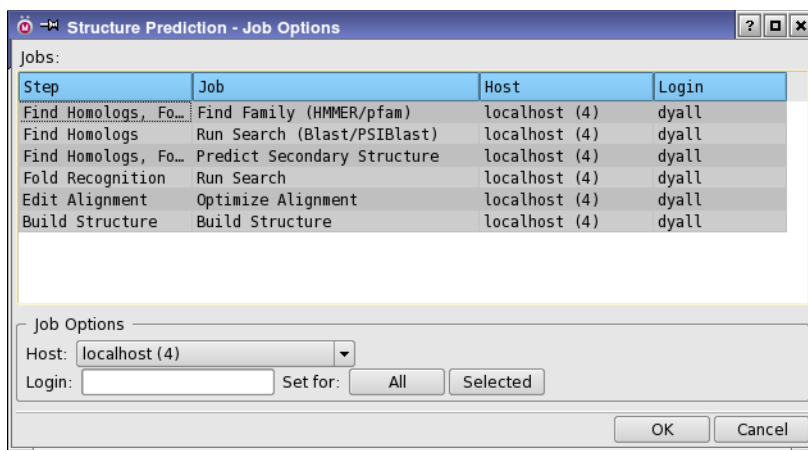
Display the structure for the sequence in the Workspace. This command is only present when there is a structure for the sequence. The query sequence does not have a structure associated with it until a structure has been built. The template sequences do have structures, which can be displayed.

### Wrap Sequences

When this option is selected, the sequences in the sequence viewer are wrapped, so that you can scroll vertically through the sequences. When it is deselected, the sequences are displayed on a single line, which you can scroll horizontally. This command is always present on the shortcut menu.

### Legend

Open the Sequence Viewer Legend panel, which explains the color schemes and the symbols used in the sequence viewer. This command is always present on the shortcut menu.



**Figure 2.2. The Structure Prediction–Job Options dialog box.**

## 2.6 Prime Job Options and Control

Before launching a Prime calculation from a step panel, you may want to choose which machine host the job will run on and check or modify job settings. When a run is complete, you can incorporate the results into the Maestro Project Table and save them as a project.

### 2.6.1 The Job Options Dialog Box

The Structure Prediction–Job Options dialog box, shown in [Figure 2.2](#), allows you to specify which machine each type of Prime job will run on, and under which user name. To open the dialog box, choose Job Options from the File menu on the Prime menu bar. The Job Options dialog box lists 6 types of jobs, from Find Family (run from the Find Homologs step) to Build Structure (run from the Build Structure step). Specify the host machine and the user name for each type of job. The default host is localhost. Make sure your `schrodinger.hosts` file contains a list of available machines to run on. See the [Installation Guide](#) for information on the `schrodinger.hosts` file.

### 2.6.2 Setting Up and Launching Jobs

Though you specify the machines used to run jobs in the Job Options panel, you select Prime job settings and launch jobs from the Prime step panels. This can be as simple as clicking an action button, such as Search. In other steps, you select the type of job from a Task menu and then launch the job by clicking Run.

You can modify settings for step actions or tasks using the Options button. You select the residues or secondary structure features on which a refinement task will operate from lists, tables, and the Atom Selection dialog box in the Refine Structure step panel. Until you make a selection, the Run button remains unavailable.

For more information on setting up and launching jobs, see the discussions of specific steps in the chapters that follow.

### 2.6.3 Monitoring Jobs

After you launch a job, the job status icon in the upper right corner of the step panel turns green and spins.

To monitor a job using the Monitor panel, click the job status icon. You can use this panel to monitor jobs from the current session or your previous sessions.

To kill a job, select the job name and then click Kill. If the job still appears to be running, you may have to go to the Project directory and remove the job file.

For more information about the Monitor panel and monitoring jobs, see [Section 3.2](#) of the *Job Control Guide* and the online help.

### 2.6.4 Adding Structures to the Project Table

In the Build Structure step, predicted structures can be added to the Maestro Project Table as new entries. This is useful for comparing predicted structures produced from different runs, and for refining structures.



# Structure Prediction: Initial Steps

This chapter outlines the initial steps in the Prime–SP workflow. In the Input Sequence step, a query sequence is imported. In the Find Homologs step, potential templates are identified. Later chapters describe the steps in which the templates are aligned and a model is built.

## 3.1 Input Sequence Step

This step is used to read in an amino acid sequence, edit the sequence if desired, and select the query sequence.

### To read a sequence from a file:

- Click File and specify a file containing the desired sequence.

Several file formats are supported, including FASTA, EMBL, GENBANK, PDB, and PIR. For a complete list, see [Appendix C](#). File formats are auto-detected by the program. Sequences must be in standard uppercase one-letter code format.

### To read a sequence from the PDB:

- Click PDB ID and specify the PDB ID and optionally the chain name.

The PDB file is downloaded and the sequences are read in.

### To enter a sequence as text:

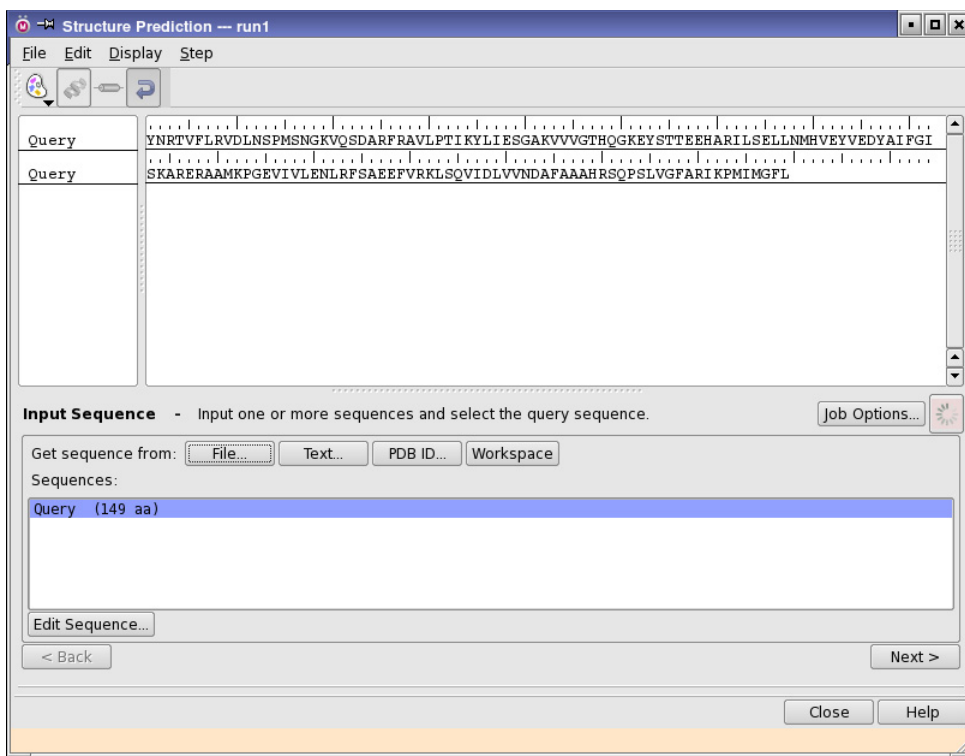
- Click Text and type in the sequence or paste it from the clipboard.

If there are multiple chains with the same sequence, the sequence will be read in only once.

### To read a sequence from the Workspace:

- Click Workspace.

All sequences represented in the Workspace are added to the Sequences table. If there are multiple chains with the same sequence, the sequence is read in only once.



**Figure 3.1. The Input Sequence step, after input.**

After input, the Sequences table in the Step View area shows all available unique sequences with their corresponding chain lengths in the order in which they are read. By default, the first sequence read in is displayed in the sequence viewer in the upper section of the panel. If multiple sequences have been read in, only one can be brought forward to the next step in a given Prime run. If you want to build a multimer, you can do so by combining information from separate runs for each chain in the Build Structure step.

**To select a query from the Sequences table:**

- Click on a sequence to select it for this run.

This brings the selected sequence into the sequence viewer. The sequences are named by extracting the name from either the PDB header or the FASTA file header. Non-unique sequences are ignored. Non-unique names are modified to make them unique.

If you want to modify the sequence before proceeding, for example, to remove extra residues at the termini, delete an insertion, or mutate residues, you can edit the sequence.



**To edit a sequence:**

1. Select the sequence in the Sequences table.
2. Click Edit Sequence.
3. Edit the sequence in the dialog box that opens, and click OK.

**Note:** Sequences longer than 1,000 residues generally include more than one domain. You should consider dividing these sequences into probable domains and treat each domain as an individual query.

**To continue:**

- Click Next to go to the Find Homologs step.

If an error message appears when you click Next, you cannot proceed until the problem is resolved. Error messages are listed in [Appendix D](#). See [Section D.1 on page 133](#) for help with Input Sequence error messages.

## 3.2 Find Homologs Step

This step is used to search for homologs of the query sequence. These homologs can then be used as templates to build a model structure.

### 3.2.1 Importing Homologs

To import homologs, use the Import buttons. You might want to do this for homologs that are not in the NCBI sequence database, or homologs that have been rotated into a common reference frame when multiple templates have been selected. You can import from two sources:

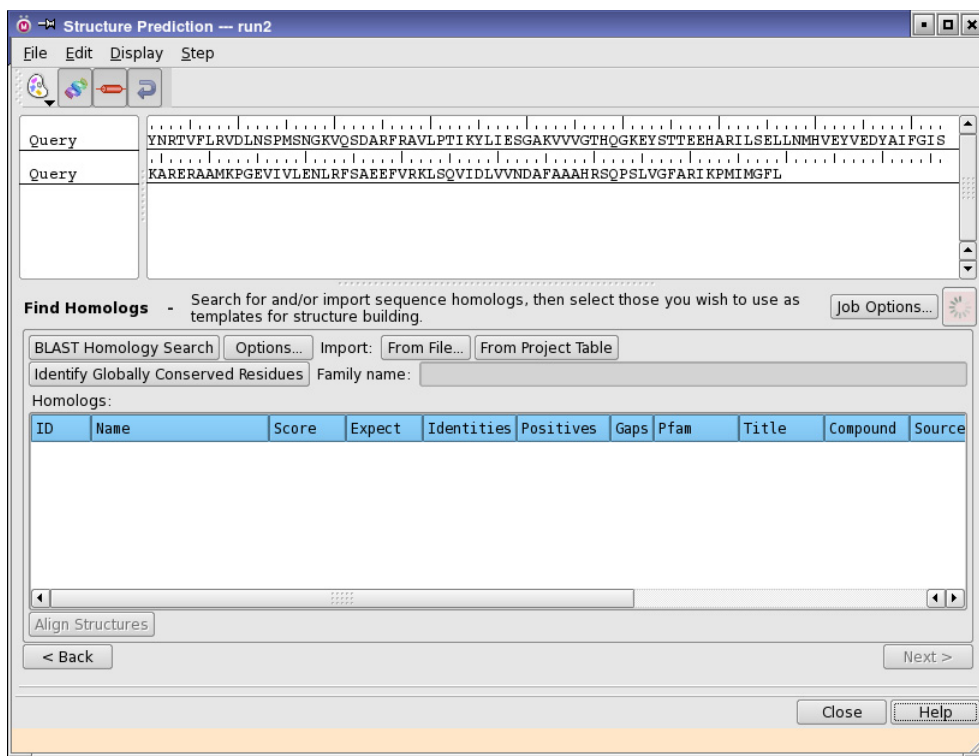
- From File—You can import homologs from a PDB file or a Maestro file.
- From Project Table—Select the desired homologs in the Project Table and click this button. This allows you to import a homolog in any format recognized by Maestro.

Imported homologs are aligned quickly using ClustalW.

### 3.2.2 Searching for Homologs

**To launch a search for homologs:**

- Click BLAST Homology Search (near the middle of the panel).



**Figure 3.2.** The Find Homologs step, initial view.

Depending on what is selected in the Find Homologs–Options dialog box, BLAST or PSI-BLAST is used to search for templates. The default is to use BLAST to search the non-redundant PDB database that has been installed with Prime. If the database is not accessible locally, the BLAST web site can be used to search for templates. To use the web site rather than a local installation, you must allow web searches in the Find Homologs–Options dialog box. For search options, see [Section 3.2.3](#).

**To identify globally conserved residues for the query (optional):**

- Click Identify Globally Conserved Residues to run HMMER on the Pfam database.

A Hidden Markov Model (HMM) is generated from a multiple sequence alignment and used to identify the query family and provide information about which residues are conserved in the consensus sequence. The job should take only a few minutes.

The query family name found appears in the Family name text box with an E-value in parentheses. If the E-value of the family is too large, it is not meaningful to consider the family.

The sequence viewer shows the match between the query and the consensus sequence of the family. If this sequence, labeled *NAME*\_pfam, is not visible, check for a plus sign to the left of the query name: +*NAME*. Click the plus sign to show any hidden sequences. The match sequence displays the residues that are conserved in the multiple sequence alignment used to generate the Hidden Markov Model (HMM). The conserved residues that are present in both the query and the template are colored with the template color. Conserved residues that are present in only one sequence or in neither are not colored. This scheme allows you to easily identify residues in the template or in the query that are not conserved according to the model.

If you want to see the full Pfam annotation, right-click on the annotation and choose View Full Pfam Annotation. The full annotation displays more information about which residues are conserved in the multiple sequence alignment used to generate the HMM. Capital letters indicate highly conserved residues, lowercase letters indicate a match to the HMM, + means the match is conservative, and a blank indicates that the residue does not match the HMM.

### 3.2.3 Find Homologs Search Options

To change search options:

- Click Options to open the Find Homologs–Options dialog box.

You can use this dialog box to specify a PSI-BLAST or BLAST search, change the settings for either type of search, and choose whether to run the search on the BLAST web site or use a local installation of the BLAST database. For more information about BLAST, see the documentation on the NIH web site, at <http://www.ncbi.nlm.nih.gov/books/NBK21097/>.

General options include:

- Database
- Similarity Matrix
- Gap Costs

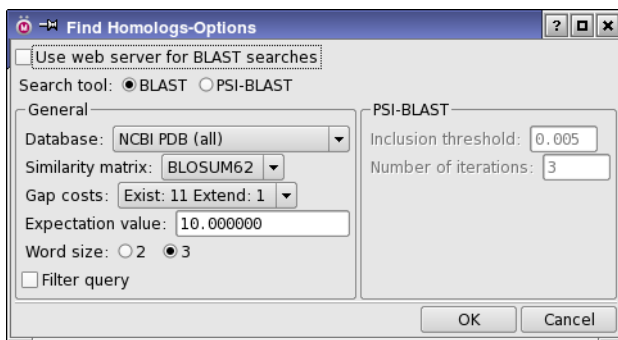


Figure 3.3. The Find Homologs–Options dialog box.

- Expectation value
- Word size
- Filter query (off by default)

PSI-Blast options include:

- Inclusion threshold
- Number of iterations

The default database selection is NCBI PDB (all). Other selections include:

- NCBI PDB (non-redundant)—Does not include PDB files with identical sequences.
- NCBI NR—Recommended for PSI-BLAST runs. With this selection, only the sequences with structures available in the PDB are displayed in the Homologs table.

### 3.2.4 Search Results

The output from the homolog search appears in the table labeled Homologs: ( $N$  found,  $n$  selected). For each potential template, the row contains the ID, the name, score, expectation value, % identities, % positives, % gaps, and header information (if taken from a PDB file). All percentages are rounded to the nearest whole number, which may be zero. A description of the table columns is given in [Table 3.1](#).

*Table 3.1. Homologs table data*

Column Header	Description
ID	The PDB ID code, including chain name. If a chain name is specified, then the chain, not the entire protein, is the potential template.
Name	Sequence name provided by BLAST.
Score	BLAST bit score.
Expect	BLAST expectation value.
Identities (%)	Percentage of residues that are identical between the sequences.
Positives (%)	Percentage of residues that are positive matches according to the similarity matrix selected in the Options dialog (default matrix is BLOSUM62).
Gaps (%)	Percentage of gaps in both query and homolog as returned by BLAST.
Title	From the PDB TITLE field.
Compound	From the PDB COMPND field.
Source	From the PDB SOURCE field.

Table 3.1. Homologs table data (Continued)

Column Header	Description
Experiment	From the PDB EXPDTA field. This gives the experimental technique used to obtain the structure, e.g., X-RAY DIFFRACTION or NMR.
Resolution	From the PDB REMARK2 record, where applicable.
Ligands/Cofactors	From the PDB HET records: hetID (three alphanumeric characters) and name for each HET group.
Warning	A warning appears in this column for sequences that may be unusable for model building. See <a href="#">Section D.2.2</a> , for a list of warnings.

You can view the complete text of each table cell as a tool tip by pausing the pointer over the cell. Columns can be sorted by clicking on the column heading. Click once to sort in descending order, and click twice to sort in ascending order. The column widths can be adjusted by dragging the column border.

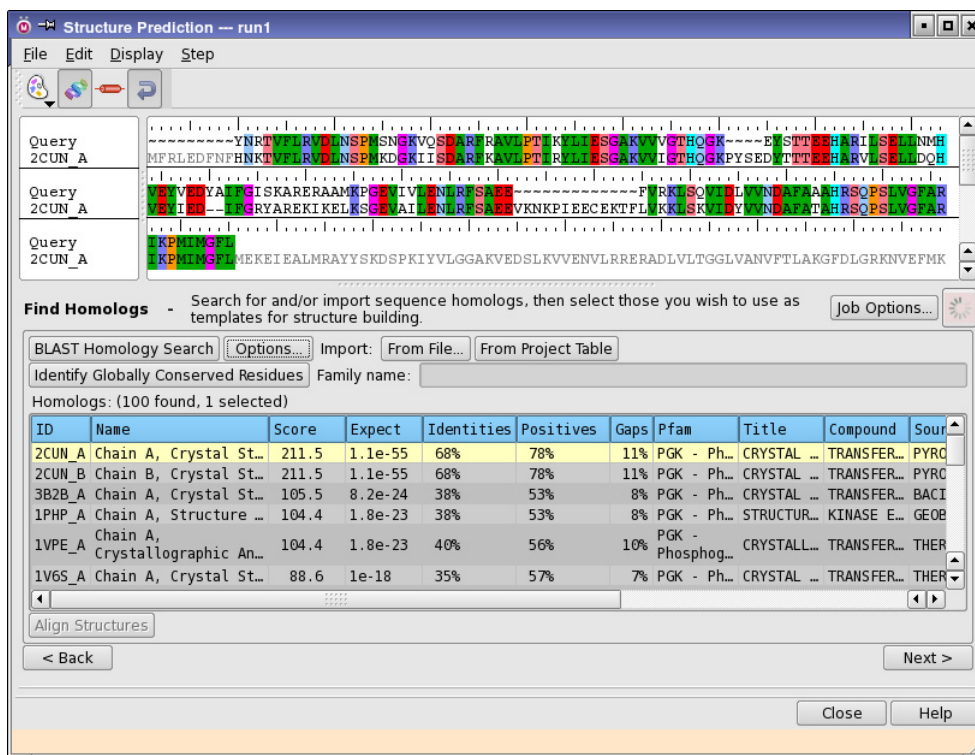


Figure 3.4. The Find Homologs step with Blast search results.

If no coordinates are available for a template, that row is dimmed and cannot be selected.

If a chain ID is given for a homolog (e.g., Chain A in 1EMS\_A), then that chain, not the entire protein, is the potential template. When aligning multiple templates with multiple chains, you may need to extract single chains using the `getpdb` utility, as described in [Section 2.3](#) of the *General Utilities* manual.

Once homologs are found, click on a row to view the structure of that template in the Workspace. Selecting a template brings the sequence alignment of the template and the query sequence into the sequence viewer. Click on another template to deselect the first one and display the selection in the Workspace.

After you review the alignments, select the template that you want to build on by clicking on its row in the table. To select multiple templates, see [Section 3.2.5](#).

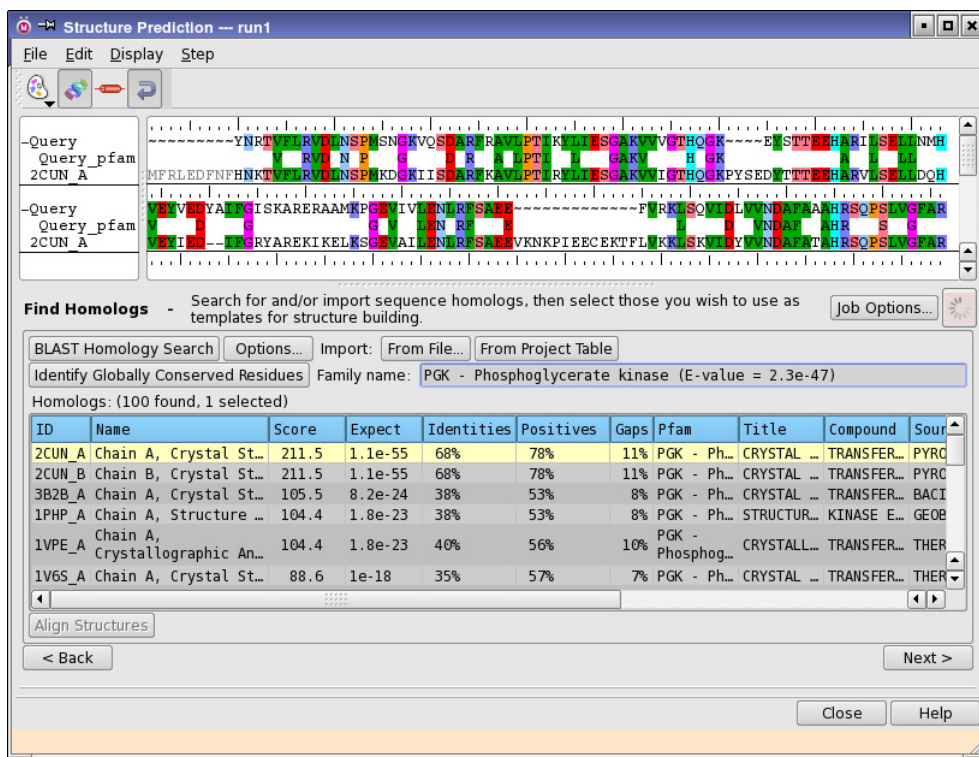


Figure 3.5. The Find Homologs step with Blast and HMMER/Pfam search results.

### 3.2.5 Multiple Templates and Structure Alignment

Multiple templates can be used in several ways in Prime to build a model. For a single query chain, you can build a composite model, where each template contributes to a different part of the structure, or you can build a consensus model, where each template contributes to all parts of the structure. If you want to build a homomultimer, you will need templates for each chain (even though the sequences are identical). For heteromultimers, you will usually only need a single template for each query sequence in a given run. You can select more than one template for building with shift-click and control-click.

If you are building a composite model with more than one template, they will need to be *structurally aligned*, that is, rotated to bring them into a common reference frame, before they can be used in the next step. However, if you are building a multimer of any sort, you should *not* align the templates, because they must already be in the correct spatial relation to each other to construct the multimer from them.

- If you have already created a file with structurally aligned multiple templates, you can import it using the Import button, as described in [Section 3.2.1](#).
- If you have not yet performed structural alignment on the selected templates for a composite model, click Align Structures.

If structure alignment fails or does not give the intended result, the problem may be one of the following:

- The selected templates are structurally too dissimilar for a meaningful alignment.
- One or more of the template structures includes multiple chains, and either
  - Two or more templates are chains from one .pdb file (the structure alignment facility will not align chains from the same file), or
  - The chains that were aligned were not the chains you intended to align.

If the problem involves a template with multiple chains, you can use the `getpdb` utility (see [Section 2.3](#) of the *General Utilities* manual) to extract each chain into a separate .pdb file, then run structure alignment again.

For information on running structure alignment from the command line, see [Section 11.1](#).

### **3.2.6 Continuing to the Next Step**

When you are satisfied with the template (or structurally aligned multiple templates) you have selected, proceed to the Edit Alignment step.

If there is an error, a message appears when you click Next to go to the next step. See [Section D.2 on page 133](#) for a list of error messages for Find Homologs.

### **3.2.7 Saving Runs With Different Templates**

If you decide to try different templates, you can avoid overwriting your existing workflow by starting a new one before you navigate back to the Find Homologs step to choose a different homolog. Choose Save As from the File menu. This creates a copy of the current run, allowing you to select a new template and build on it. You can then compare results from each run.

If you are building a heteromultimer, you must model each unique chain of the multimer, and save a run for each chain. The runs are combined to build the final model.



# Structure Prediction—Edit Alignment

## 4.1 Overview of the Edit Alignment Step

In order to build a model structure of the query, the Build Structure step requires a satisfactory alignment between templates and query. The alignment of templates to query generated in the Find Homologs step is based only on sequence. This alignment usually needs some improvement. The Edit Alignment step allows you to improve alignments using various tools before building a model in the Build Structure step.

There are several ways to produce an alignment for use in Build Structure:

- Accept the BLAST/PSI-BLAST alignment generated in the Find Homologs step and currently displayed in the sequence viewer.
- Import an alignment.
- Generate or import secondary structure predictions, then use the Align program to create a new alignment.
- Edit any of these alignments manually.

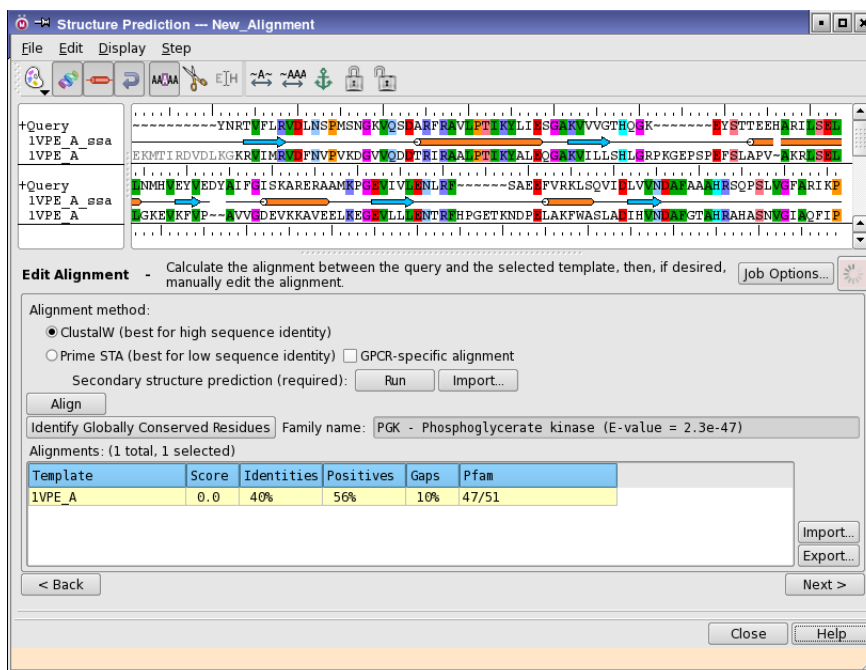
If you want to identify globally conserved residues to assist in the alignment process, but did not do so in the Find Homologs step, click **Identify Globally Conserved Residues** to run HMMER/Pfam. For more information on this action, see [Section 3.2.2 on page 23](#).

When you are satisfied with the alignment, click the **Next** button to proceed to the Build Structure step.

## 4.2 Using Existing Alignments

When you start the Edit Alignment step, the BLAST/PSI-BLAST alignment for the template you selected in the Find Homologs step is displayed in the sequence viewer—see [Figure 4.1](#).

- If you chose a single template and this is the template and the alignment you want to use, click **Next** to go to the Build Structure step.
- If you chose multiple templates, select the alignment you want to use in the alignments table, and click **Next**.



**Figure 4.1. The Edit Alignment panel, initial view.**

- If you want to use an alignment generated in another program (or exported previously from Prime), click Import (to the right of the Alignments table), and navigate to and select the alignment in the Import Alignments file selector.

The sequences in the alignment file must be exactly the same as those used by Prime, and so must the sequence and structure (PDB) names. For details of the allowed formats, see [Appendix C](#).

**Note:** The template PDB ID must be uppercase in the alignment file.

You may want to save the existing alignment for later use before making changes. To do so, click Export, and navigate to the desired location in the Export Alignments file selector and name the file.

## 4.3 Generating an Alignment

If the existing alignments are not satisfactory, you can generate an alignment using one of two programs, listed under Alignment method:

- **ClustalW**—Running this program generates an alignment quickly, from sequence information only. It is best used when there is high sequence identity between the query and the template.
- **Prime STA**—This program generates an alignment based on secondary structure predictions (SSPs) for the query as well as sequence alignments. Once you have these predictions, you can set up and run the alignment program. This program is best used for cases of lower sequence identity.

### 4.3.1 Generating SSPs

To run all available SSP programs, click Run. One secondary structure prediction program (SSpro) is bundled with Prime. However, the optional third-party program PSIPRED is highly recommended for optimal results, especially for GPCRs. PSIPRED is not available on Windows. See [Appendix A](#) for more information on this program. If there is a problem running the structure prediction programs, an error message is displayed.

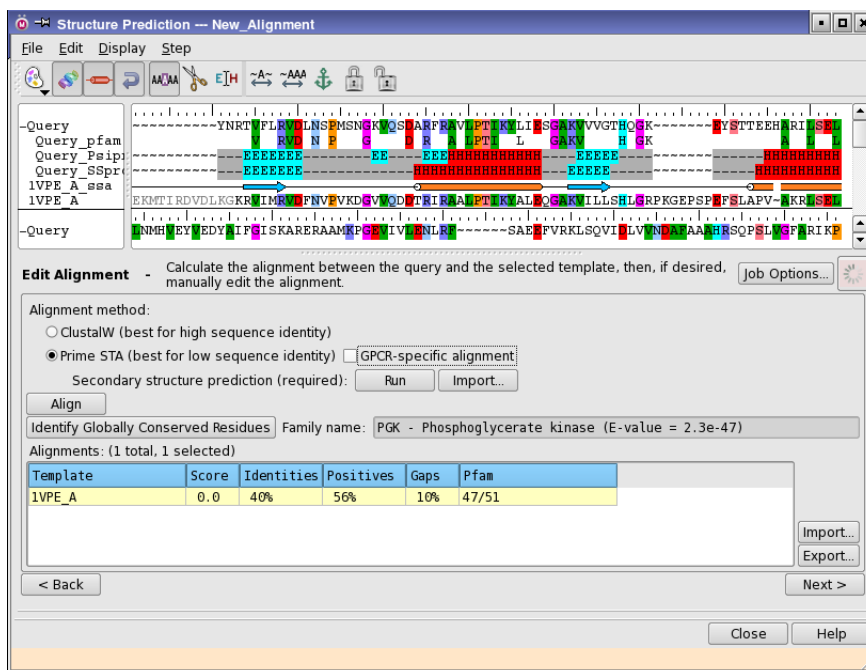
When the SSP run is finished, the secondary structure predictions are displayed in the sequence viewer below the query sequence. If you want to hide the SSPs, right-click on an SSP in the sequence viewer and choose Hide All.

It can be useful to generate or import SSPs even if you perform an alignment with ClustalW, because you might want to use these SSPs to guide a manual adjustment of the alignment.

### 4.3.2 Importing and Exporting SSPs

To import a secondary structure prediction for the query, click the Import button for Secondary structure prediction and select the SSP file, which may be in FASTA or Maestro format. If necessary, use the `seqconvert` utility to change the file format—see [Section 11.2](#) for more information. If the file contains bad data or the sequence doesn't match the query sequence, an error message is displayed.

If you want to save an SSP for later use, right-click on it and choose Export Single Sequence. By default, the SSP is saved in native Schrödinger format. If there is a problem exporting a file, an error message is displayed.



**Figure 4.2. The Edit Alignment step, after running SSPs.**

### 4.3.3 Adjusting SSPs

If you believe an SSP is incorrect, you can delete it entirely or edit the incorrect portions. You can revert to the original SSP if you are not satisfied with your adjustments.

#### To delete an SSP:

- Right-click on the SSP and choose Delete from the shortcut menu.

#### To edit an SSP:

1. Click the Edit SSP toolbar button:

E[H

2. Highlight the part of the SSP you want to change.
3. Type the character e, h, or -.

E represents strand, H represents helix, and - represents loop.

The highlighted region of the prediction is changed to your choice.

**To retrieve the original SSP after editing:**

- Right-click on the SSP and choose **Revert** from the shortcut menu.

### 4.3.4 Running the Alignment

Before running an alignment, if you want to take advantage of the special capabilities for aligning GPCRs, select **GPCR-specific alignment**. These capabilities include fingerprint matching, a customized GPCR sequence database, and identification of transmembrane helices. For GPCR alignment, a set of seven transmembrane helices in the SSP is needed. In general, PSIPRED is more reliable than SSPro in producing a satisfactory SSP. If the SSPs for the GPCR are not satisfactory, you can adjust them manually, by using the instructions in the previous section.

To take advantage of your knowledge of the system, one or more pairs of query/template residues that you know should be aligned can be constrained to remain so, by using alignment constraints.

**To constrain a pair of residues to remain aligned:**

1. Click the **Add Anchors** toolbar button.



2. Select one of the residues in the pair that you want to remain aligned during the Align calculation.

An anchor symbol is displayed in the ruler at the corresponding residue position.

When you are satisfied with the secondary structure predictions for the query, click **Align** to start the alignment job.

After the job finishes, the **Alignments** table is updated with the new alignment score, the percent identity, the percent similarity, and the percent gaps, all rounded to the nearest integer. The changed alignment is also reflected in the Maestro Workspace display of the structure.

If you are satisfied with the new alignment, you can continue to the next step by clicking **Next**. Otherwise, you can edit it manually as described in the next section.

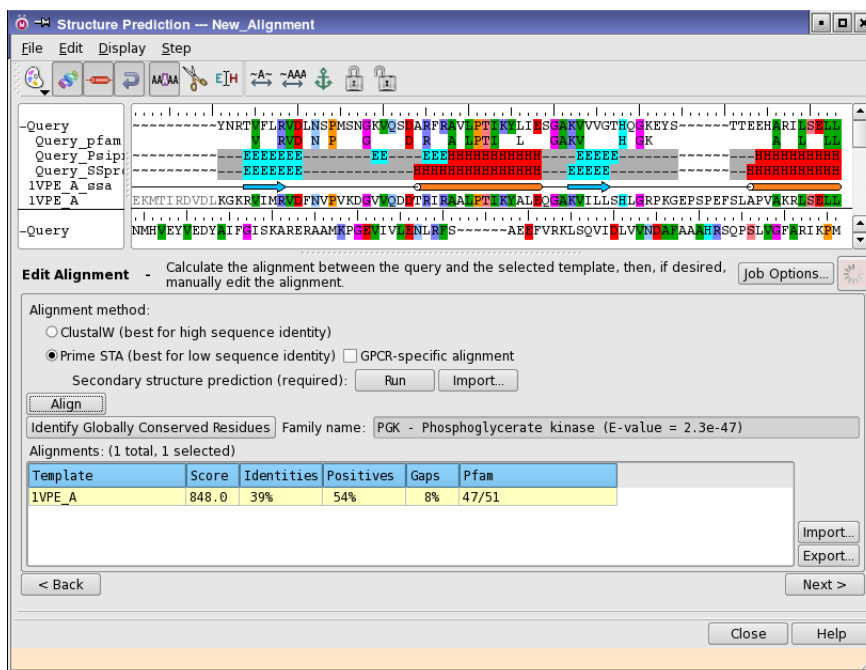


Figure 4.3. The Edit Alignment panel after alignment.

### 4.3.5 Align Program Technical Details

The goal of the program launched from the Edit Alignment step, *sta* (Single Template Alignment), is to generate an accurate alignment between proteins with medium to high sequence identity (20-90%). Features of the program include:

- A position-specific substitutional matrix (PSSM) for the query sequence, derived from PSI-BLAST, is used to match the template sequence.
- In order to minimize the inaccuracy in a single secondary structure prediction, a novel and robust algorithm has been designed to derive a composite secondary structure for the query sequence from multiple predictions. The composite SSP is aligned to the SSA of the template.

See [Section 10.5 on page 113](#) for more information on this program, including instructions for running it from the command line.

## 4.4 Manually Editing an Alignment

You can edit any alignment, however obtained, using combinations of these Prime toolbar buttons: Slide Freely, Slide as Block, Add(/Remove) Anchors, Lock Gaps, and Unlock Gaps.



**Slide Freely (Edit menu):** Edit the alignment between query and template. Selecting a residue and sliding to the right creates N-terminal gaps. Selecting a residue and sliding to the left creates C-terminal gaps.



**Slide As Block (Edit menu):** Edit the alignment between query and template. Selecting a residue and sliding to the right slides all residues as a block to the right. Selecting a residue and sliding to the left slides all residues as a block to the left.



**Add Anchors (Edit menu):** Set alignment constraints by setting anchors at residue positions. To remove anchors, click on the same residue again.



**Lock Gaps (Edit menu):** To prevent gaps from being collapsed during manual editing, lock the gaps using this button. Locked gaps are indicated with a - symbol.



**Unlock Gaps (Edit menu):** To allow gaps to collapse during manual editing, unlock the gaps using this button. Unlocked gaps are indicated with a ~ symbol.

The Workspace shows the template structure colored by query residue property, displayed as a ribbon and side chains. Each template residue is colored according to the property of the corresponding query residue in the alignment, revealing the location of charged, polar, and hydrophobic query residues. You can use this feature to make sure that insertions or gaps are not in the middle of regions of secondary structure such as alpha helices or beta sheets, and to check that hydrophobic residues point into hydrophobic regions and polar residues point towards solvent. As you make changes to the alignment, the Workspace is updated to show the mapping of the query's residue properties onto the template structure.

## 4.5 Error Messages

When you click Next at the end of the Edit Alignment step, the Check Alignment warning dialog box may appear. The warning lists possible alignment problems that you may want to fix before continuing to the next step—for example, gaps in secondary structure elements of the template, or gaps in the query that are aligned to secondary structure elements of the template.

If you do not want to make corrections to the alignment, click Continue to go to the next step. To make corrections based on the information in the error message, make a note of the gap locations given, click Cancel to close the alert box. The Edit Alignment step is still displayed, and you can change the alignment accordingly.





# Structure Prediction—Build Structure

## 5.1 The Build Process

The process of building a structure from one or more templates involves the following steps.

- Copying of backbone atom coordinates for aligned regions and side chains of conserved residues
- Optimization of side chains
- Minimization of non-template residues
- Building of insertions and closing of deletions in the alignment

By default, all of these steps are performed, except that insertions of more than 20 residues are omitted. Also by default, only the protein structure is built: ligands, cofactors and waters are deleted before building. However, you can choose to leave out one or more steps (except the first), and you can choose to build in ligands and cofactors. (Water molecules must be added afterwards.) These choices must be made before you build the structure.

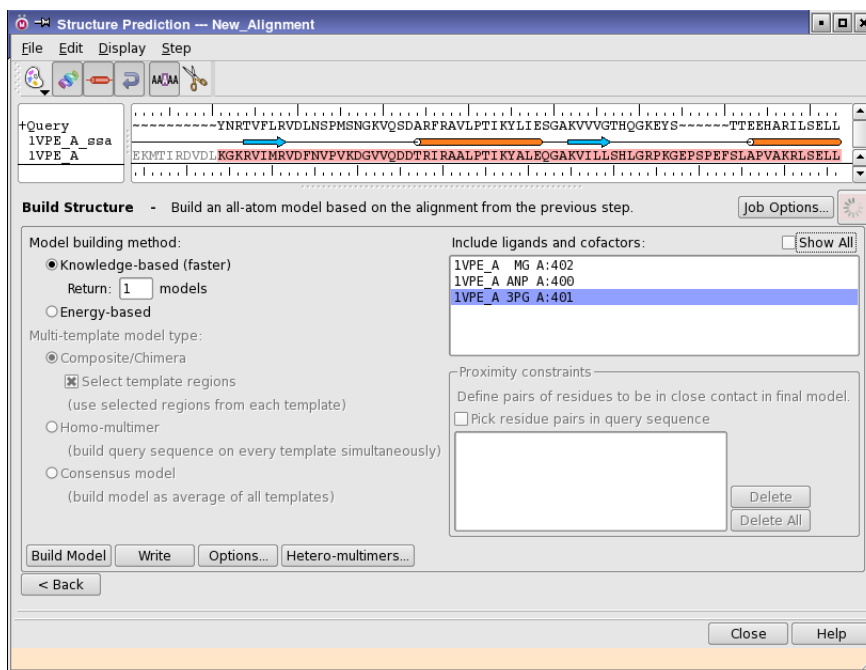
The Build Structure step provides tools for building models of multimers as well as models of single chains, and for building models of single chains from several templates. It also allows you to set constraints and control some aspects of the model building.

The build process does not perform any refinement of the structure. Once you have built a model, you should refine it, by using the Refinement panel, described in [Chapter 6](#).

## 5.2 Choosing the Model Type

If you are using multiple templates, you must choose the type of model you are building: composite, consensus, homo-multimer, or hetero-multimer. For composite models, you must select the regions from each template that are used to build the structure. The following sections provide more detail on the choices to be made.

If you are using a single template, these choices are not available, and you can proceed to the setting of other options.

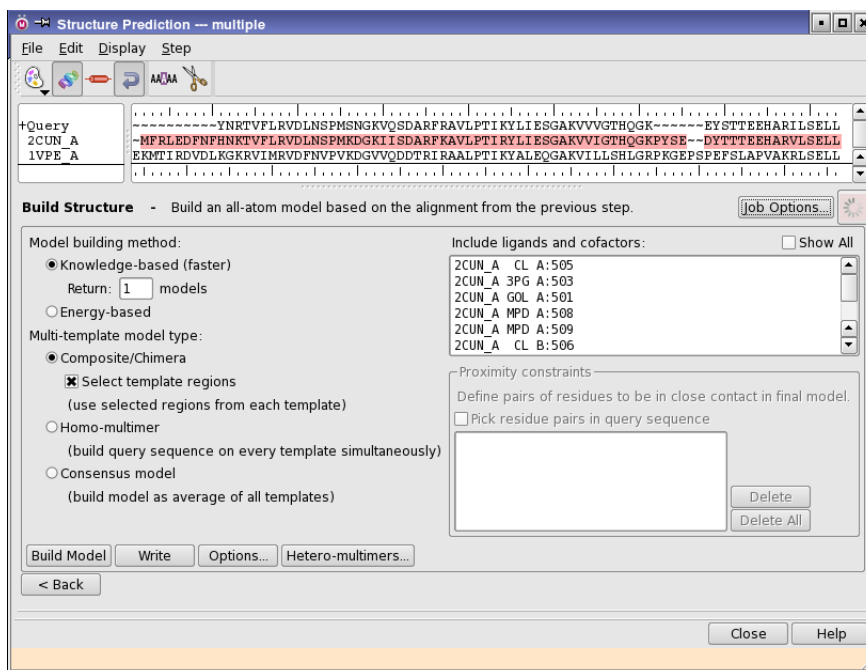


**Figure 5.1. The Build Structure step for a single template.**

### 5.2.1 Building a Composite Model

If you want to build a composite model from the templates you selected in the Find Homologs step, select Composite/Chimera under Multi-template model type section. You must specify which template should be used for each region of the query, otherwise, the first template in the table is used for the entire query.

To specify the regions to be used for each template, first select Select template regions if it is not selected, then select the desired region on each template in the Sequence Viewer. When you begin to select the template regions, only the first template is in use. As you select regions in the second template, the corresponding residues in the first template are automatically deselected. If you select regions for use from a third template, the corresponding regions of the first and second templates are deselected, and so on. Thus, the selection process ensures that only one template residue is assigned to a query residue (the most recent one selected). Regions of a template that are not used to model the query are colored dark gray (regardless of color schemes.)



**Figure 5.2. The Build Structure step with multiple templates.**

## 5.2.2 Building a Consensus Model

If you want to build a consensus model from the templates, select Consensus under Multi-template model type. The model is then built from the consensus between the templates at each residue position: for example, if two templates have a loop in the same spatial region but in the third template, the loop occupies a different region, the model is built from the two that agree. The alignment is done as part of the model-building job, so there is no need to align the structures for the templates.

## 5.2.3 Building Multimers

In addition to building models for a single chain, you can build models for multimers that are composed of copies of the same chain (homomultimers) or of different chains (heteromultimers). To build the multimer, the structures for all chains in the templates must be in the correct location (position, orientation) to build the multimer.

To build homomultimers, select Homo-multimer under Multi-template model type. All chains of the template must be aligned to the query.

To build heteromultimers, you must first model the individual units of the multimer (which can be homomultimers) in separate runs, up to this step. The templates that you choose to model the individual units must be properly structurally aligned to each other, so that the resulting built structure has the desired configuration of the units.

To build the model, click Hetero-multimers, choose the runs in the Hetero-multimers Modeling dialog box, and click Start.

## **5.3 Preparing to Build**

Before you build the structure, you can make settings that affect the build process. These settings are described in the following sections.

### **5.3.1 Setting Constraints**

If you know that certain residues must be in close contact in the final structure, such as cysteines forming a disulfide bond, you can constrain their positions in the final model by setting proximity constraints on residue pairs. Proximity constraints can only be applied if at least one of the residues in the pair is in a loop, as helices and strands can't be adjusted to satisfy proximity constraints. A disulfide bond is formally constructed between pairs of CYS residues. For other residues, a harmonic constraint is added for the terminal heavy atoms of the side chains. To apply these constraints, select Pick residue pairs in query sequence, and pick the pairs of residues that you want to be in contact in the final model.

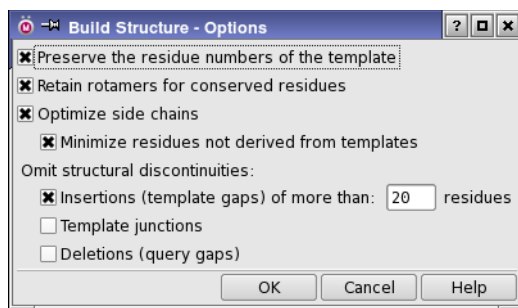
### **5.3.2 Including Ligands and Cofactors**

Any cofactors and ligands present in the template you have chosen (except water) are listed in the Include ligands and cofactors list. To include one or more ligands or cofactors, select them in this list. You can use shift-click and control-click to select multiple rows. The selections are marked in the Workspace. DNA and RNA can be included as cofactors or ligands. By default, only the ligands and cofactors within 5 Å of the chain used for the template are included in the list. To display the full list, select Show All.

### **5.3.3 Build Options**

Click Options to open the Build Structure—Options dialog box and specify which optional model building steps to perform:

- **Preserve the residue numbers of the template**—Number the residues in the built structure the same as in the template, as far as possible. Does not apply to consensus modeling. When using multiple templates for a single chain, the residue numbers are taken from the first template. Sequential numbering from 1 is used if the template numbering fails.



**Figure 5.3. The Build Structure—Options dialog box.**

- Retain rotamers for conserved residues—retain side-chain rotamers for conserved residues.
- Optimize side chains—Optimize the side chains.
- Minimize residues not derived from templates—minimize residues that are not derived from the templates. Only available if Optimize side chains has been selected.

By default, all of these options are selected.

In this dialog box, you can also disable building through structural discontinuities arising from insertions, template junctions, and deletions. The discontinuities are capped with NMA and ACE, so the final structure is discontinuous. The query chain retains its name, regardless of the discontinuities. If you choose any of these options, you should check that your final structure is reasonable.

If there is an insertion in the query that requires the building of a long loop, you can select Insertions (template gaps) under Omit structural discontinuities to cut the query sequence instead, cap it with NMA and ACE, and not build the loop. You can also specify the maximum length of loops that will be built in the text box, so that only longer loops are cut rather than built. When the template that spans the insertion is cut to build the query, the residue on the high side of the cut (higher residue number) is also deleted, to avoid introducing steric clashes. Cutting could be useful where there are long insertions that are not in the region of interest. The option is selected by default, with a length of 20. If it is not selected, all loops are built.

Likewise, if there are deletions in the query (a query gap), or if there is a junction between two templates, you can choose to cap the ends of the two pieces and not try to join them when building the structure. To do this, select Deletions (query gaps) or Template junctions under Omit structural discontinuities.

When the structure is built with discontinuities, the results appear to be misaligned in the sequence viewer. To display the correct alignment, right-click in the sequence viewer and choose Align by Residue Number.

## 5.4 Building the Structure

There are two choices of method for building the structure:

- **Knowledge-based**—Construct insertions and close gaps using segments from known structures. With this option, you can choose to return multiple models of the structure. By default, only one is returned. The models are approximately ordered by the sequence identity and the incidence of clashes in the structure.
- **Energy-based**—Construct and refine residues that do not come from the template based on the energy, including building through structural discontinuities and optimizing side chains.

When the model is built, the structure is automatically incorporated into the project as a project entry, and the model is displayed in the Workspace. For composite and multimer models, the residues are colored according to how the template was used:

- blue—all coordinates were taken from the template
- cyan—the backbone coordinates were taken from the template
- red—the residue was rebuilt

This information is stored in the Homology Status property (`i_psp_Homology_Status`), with values of 0, 1, and 2 respectively, but the color scheme is not stored.

The sequence in the Structure Prediction panel remains as it was before building: it reflects only the state of the sequences brought into the Build Structure step. The Workspace sequence viewer displays the full sequence of the built model.

You can also run the Build Structure job from the command line. To do so, click Write to write the input file. The file is written to the current working directory, and is named according to the run and project. A dialog box informs you of the file name and the command to use, which is:

```
$SCHRODINGER/prime input-file
```

For more information on this command, see [Section 10.2 on page 94](#).

## 5.5 Technical Notes for the Build Structure Step

The energy-based model building method in the Build Structure step uses the following resources and methods:

- The OPLS\_2005 all-atom force field for energy scoring of proteins as well as for ligands and other non-amino-acid residues.
- A Surface Generalized Born (SGB) continuum solvation model for treating solvation energies and effects.
- Residue-specific side-chain rotamer and backbone dihedral libraries, derived from the non-redundant data sets extracted from the PDB, and representing values most commonly observed in protein crystal structures. The libraries allow for both the evaluation of existing rotamer/dihedral values and the systematic prediction of new ones.

Build Structure can treat the 20 standard amino acids as well as modified residues and ligands.

Model building uses the following procedure:

1. Non-conserved residues are mutated to the desired identity. Side chains are added by finding the first rotamer in the library that does not produce a clash.
2. Insertions, deletions, and template transitions are built. These are cases in which the backbone itself needs to be reconstructed, either due to gaps in the alignment or template transitions that produce gaps in the 3D structure. These gaps are closed by reconstruction of the affected region *ab initio*, using a backbone dihedral library. If no reasonable conformation can be generated that closes the loop structure, the region being reconstructed is expanded until closure is possible.
3. Gap reconstruction is done by finding any single loop conformation that closes the structure and is physically reasonable. If multiple conformations are found, the one that deviates from the template structure as little as possible is chosen. No attempt is made to perform an exhaustive optimization of the region.
4. Side-chain prediction (optional). Side chains may be optimized. The two choices are either: (a) optimize all side chains from scratch, or (b) optimize only those that are not in the template.
5. Minimization of non-template regions (optional). Any portions of the backbone that were not derived directly from one of the templates (and were thus reconstructed *ab initio*) are subject to a local minimization as described in [Section 6.6 on page 57](#).





# Refining Protein Structures

Prime–Refinement is a module used to refine protein structures from the Maestro Workspace. The methods can be applied to protein structures from any source, including one built using the Structure Prediction workflow. It runs independently of Prime–SP from its own Maestro panel, the Refinement panel. The Refinement panel offers the following refinement protocols: loop refinement, side-chain prediction, minimization, and a single-point energy calculation at the current geometry of the model structure.

Refinement can be done on protein structures from sources other than Prime, and also with structures added to a project from the Build Structure step in the Prime–SP module. Model structures built in the Build Structure step usually require further refinement. If there are insertions or deletions in the alignment between query and template, it is recommended that you perform a loop refinement, since insertions and deletions are most frequently found in loop regions. As another example, if you have added a water molecule to the binding site of the protein, you can run a refinement on the composite entry.

Refinement can be done on structures with covalently bound ligands. The ligands can be multiply connected, covalently bound to each other and to the protein. This capability allows the refinement of proteins with phosphorylated residues or attached sugars, for example. For more information on covalently bound ligands, see [Chapter 8](#).

Prime–Refinement has an implicit membrane model, in which the membrane is modeled by a slab of low dielectric constant in which the protein is immersed. The width of the slab and the orientation of the protein with respect to the slab can be adjusted.

## 6.1 Preparing Structures for Refinement

Prime–Refinement has a great deal of flexibility in the types of structures it can handle, and can fix many structural problems. For standard residues, Refinement fixes formal charges and bond orders, and correct disparities between the sequence and the structure. For example, if a residue has the coordinates of ALA but is called SER, the 3HB is ignored and the OG and HG added during refinement. The standard residues are the 20 canonical amino acids; ACE and NMA; HOH; CYX (disulfide); and the acid/base variants ASH/AS1 (ASP), GLH/GL1 (GLU), ARN (ARG), LYN (LYS), HIE/HIP/HID (HIS), CYT (CYS), SRO (SER), TYO (TYR).

The input structures must meet certain conditions for a refinement job to be run successfully. Some of these conditions only apply to covalently bound ligands.

For all kinds of structures, the following condition must be met:

- The structure must be an all-atom structure.

This condition applies to the protein and any ligand, waters or cofactors present in the structure: all hydrogens must be present.

You can add hydrogens to a structure by displaying the structure in the Workspace and double-clicking the Add hydrogens button in the toolbar.



However, for structures containing nonstandard residues, you must correct bond orders and formal charges first.

For structures with nonstandard residues, the following conditions must be met:

- No residue can consist of disconnected pieces (“fragmented residues”) without formal bonds. All atoms within a residue must be connected via bonds, including zero-order bonds in the case of metals.
- Bond orders and formal charges must be corrected.

For nonstandard residues the supplied structure will be used, and you must check for correct bond orders and formal charges. Metals and their ligating groups

When you rename residues, you should be aware that OPLS\_2005 parameters are used for standard residues and nonstandard residues.

Proteins can also be used as ligands, and the same conditions apply. No special treatment is needed for two distinct chains connected by a disulfide or similar side-chain linkage. If the “ligand” chain has nonstandard terminal groups or is connected to the main chain by its backbone, two steps are required to make sure that the “ligand” chain is treated correctly:

1. Rename standard residues appearing in the ligand to nonstandard names.
2. Rename the atoms in inter-residue C–N bonds between renamed (ligand) residues. For example, change “\_C\_” to “\_C\*\_”, where the underscores represent spaces. Other atoms can keep their PDB names; only one of C or N needs to be changed.

If your structure does not meet one or more of the conditions outlined above, you must fix it before you can successfully run a refinement. Some of the problems only become apparent when you have run a refinement job, so it can be a useful diagnostic procedure to run a Prime energy calculation first. This job is quick, and produces warnings in the log file, which you can check in the Monitor panel.

Many of the preparation tasks are performed automatically or interactively with the Protein Preparation Wizard. The Protein Preparation Wizard panel can be opened from the Workflows menu on the main toolbar. When the preparation has finished, you should check that all changes made are correct. You can fix any remaining errors with the procedures below. For more information on the Protein Preparation Wizard, see the [Protein Preparation Guide](#).

Procedures for fixing bond orders, formal charges, and atom names are given below. These procedures use the Build toolbar and the Build panel, which you can display by clicking Show/Hide the Build toolbar on the main toolbar or opening the panel from this button menu.



#### To assign bond orders:

1. Choose Assign Bond Orders from the Tools menu.
2. Inspect the residues in the structure for any remaining bond orders that are incorrect.
3. If there are still incorrect bond orders, click the Increment bond order or Decrement bond order button on the Build toolbar.



4. Click the bonds in the Workspace structure that needs correction.

You can also right-click the bonds in the Workspace structure, and choose the correct order from the Order submenu of the shortcut menu.

#### To view and change formal charges:

1. From the Label atoms button menu on the main toolbar, choose Formal Charge. Charges are displayed for atoms that are charged.
2. Click the Increment formal charge or Decrement formal charge button on the Build toolbar.



3. Click on an incorrectly charged atom in the Workspace until its charge is correct.

The formal charge is incremented or decremented by one unit for each click.

### To change PDB atom names:

1. From the Label atoms button menu on the main toolbar, choose PDB Atom Name.



2. In the Atom Properties tab of the Build panel, choose PDB Atom Name from the Property option menu.
3. Ensure that Pick is selected in the Apply PDB atom name section, and that Atoms is chosen from the Pick option menu.
4. Zoom in on one of the residues (middle+right mouse buttons or mouse wheel).
5. For each atom that needs renaming, enter the desired PDB atom name in the PDB atom name text box, then click on the atom.
6. Repeat [Step 4](#) and [Step 5](#) for each residue whose atoms need renaming.

### To change residue names:

1. From the Label atoms button menu on the main toolbar, choose Residue information.



2. In the Residue Properties tab of the Build panel, choose Residue Name from the Property option menu.
3. Ensure that Pick is selected in the Apply residue PDB name section, and that Residues is chosen from the Pick option menu.
4. For each residue that needs renaming, enter the desired PDB name in the Residue PDB name text box, then click on the residue.

You might need to zoom in to select the residues (middle+right mouse buttons or mouse wheel).

## 6.2 Using the Refinement Panel

The four tasks that can be performed in the Refinement panel are single-point energy calculation, loop refinement, side-chain prediction, and minimization. These tasks are discussed in the following sections. The general procedure for running a structure refinement is given below.

### To run structure refinement tasks:

1. Display the structure you want to refine in the Workspace.

If the structure has alternate positions, choose the desired set of alternate positions.

2. Choose Applications → Prime → Refinement from the main Maestro window.

The Refinement panel is displayed.

3. Select a task from the Task menu.

The default task is Refine loops.

4. If you want to use an implicit membrane model, select Use implicit membrane, then click Set Up Membrane to define the thickness and orientation of the membrane.

5. Select the region for refinement, as appropriate.

When you have selected a region for refinement, the Start and Write buttons become available.

6. Click Start.

The Start dialog box is displayed. If you want to run the job later, using `refinestruct`, you can click Write to create the input files.

7. Make changes to the job settings, as appropriate.

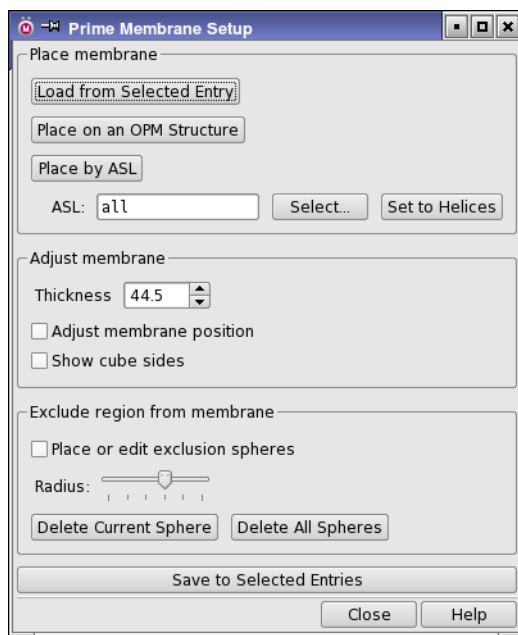
8. Click Start.

The refinement job begins.

The Start and Write buttons remain dimmed until you have specified some part of the structure for refinement. When you have made a selection, the Start and Write buttons become available. The Write button writes the input file for the job, which you can run at a later time

When selecting regions for refinement, you may want to view information about possible structural problems. This information is available in the Protein Reports panel, which can be opened from the Tools menu on the main menu bar. For more information on this panel, see [Section 10.6.2](#) of the *Maestro User Manual*.

The OPLS\_2005 force field is used for all refinement tasks.



**Figure 6.1.** *The Prime Membrane Setup panel.*

## 6.3 Setting Up an Implicit Membrane

The Prime implicit membrane model is intended for proteins embedded in a membrane. The implicit membrane is a low-dielectric slab-shaped region, which is treated in the same way as the high-dielectric implicit solvent region. Hydrophobic groups, which normally pay a solvation penalty for creating their hydrophobic pocket in the high dielectric region, do not have to pay that penalty while in the membrane slab. Conversely, hydrophilic groups lose any short-ranged solvation energy from the high dielectric region when moving into the low dielectric region.

The implicit membrane model is intended for use with proteins that span the membrane. The slab is trimmed for efficiency, so that any structure that is too small or too far away from the membrane is likely to result in too much trimming and might not produce useful results.

The membrane is marked in the Workspace by a semitransparent cube with a white line that is perpendicular to the surfaces of the membrane (which are planar). The membrane surfaces are colored red, and the other faces of the cube (which are interior to the membrane) are colored green. You can control whether the sides of the cube are displayed with the **Show cube sides** option. The direction of the line shows the orientation of the membrane. You can adjust both the thickness and the orientation of the membrane.

Setting up a membrane is done in the Prime Membrane Setup panel, which you open by clicking Set Up Membrane.

There are three sections in the panel, which correspond to three tasks: initial placement, adjustment, and excluding regions from the membrane.

In the Place membrane section, there are three choices for placement.

- **Load from Selected Entry**—Load a membrane from the entry that is selected in the Project Table. The membrane is placed at the same location in space as in the source entry, so you should ensure that the protein in the source entry and the current entry are properly positioned.
- **Place on an OPM Structure**—If you are refining a structure from the OPM database (<http://opm.phar.umich.edu>), click this button to place the membrane according to the information stored with the structure.<sup>1</sup>
- **Place by ASL**—Place the membrane so that the atoms given by the expression in the ASL text box are inside the membrane. The ASL expression can be entered in the text box, or you can click Select to specify the atoms in the Atom Selection dialog box.

To place the membrane on a set of helices, click Set to Helices. The secondary structure assignment is then used to select the helices for placement inside the membrane. The protein must have a secondary structure assignment before you use this button. To do the assignment, choose Tools → Assign Secondary Structure in the main window.

Once you have made an initial placement, you can adjust the position of the membrane. If you started with an ASL expression, you can click Select, and add to the ASL expression used in the Atom Selection dialog box. The Previous button should recall the last ASL expression used. You can adjust the membrane position manually, in the Adjust membrane section, as follows:

1. Select Adjust membrane position.
2. Use the middle mouse button to rotate the membrane and the right mouse button to translate the membrane.
3. Adjust the thickness by entering a value in the text box or using the arrow buttons to adjust the thickness up or down by 0.1 Å at a time.
4. Deselect Adjust membrane position.

This is required to exit membrane adjustment and return to normal Workspace operations.

---

1. The PDB files in this database have an invalid positioning of the remark fields for the membrane information, and must be fixed before use. To fix them, you can use the command  
`perl -pi -e 's#REMARK 1/2#REMARK 1/2# ' *.pdb.`  
Here there are two spaces after the first REMARK and five after the second.

If the protein has pockets that would be occupied by solvent (water), you can select these regions to exclude from the implicit membrane using the tools in the **Exclude region from membrane** section. The solvent dielectric constant is then used for these excluded regions.

Excluded regions are defined by a set of spheres that are placed on atoms adjacent to the region. The spheres can overlap, and should cover the entire excluded region. It does not matter that they overlap the protein, because the protein dielectric is not affected.

- To define a region, select **Place** or **edit exclusion spheres** and pick atoms in the **Workspace** that are adjacent to the region. For each pick a sphere is placed on the atom.
- To adjust the volume of one or more spheres, pick the spheres in the **Workspace**, and use the **Radius** slider. This slider only uses integer values, because the exact size of the sphere isn't critical: the spheres only have to cover the excluded region.
- To delete a sphere, pick the sphere and click **Delete Current Sphere**.

When you have finished, deselect **Place** or **edit exclusion spheres**.

If you want to add the membrane to the entries that are selected in the **Project Table**, click **Save to Selected Entries**. The membrane is stored as a set of coordinates for each end of the white line. The entries that you select should therefore contain proteins that occupy approximately the same region of space as the protein used to define the membrane. Otherwise, you will have to adjust the membrane location for these entries.

## 6.4 Refining Loops

Prime-Refinement is capable of refining loop structures of various lengths, and provides algorithms for different loop lengths. In addition, loops whose structure affects other loops can be cooperatively refined in pairs.

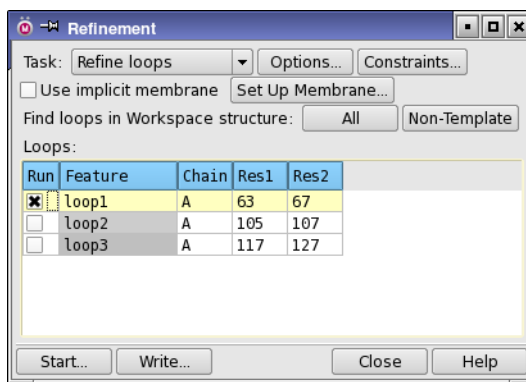
### To refine one or more loops serially:

1. Choose **Refine Loops** from the **Task** menu.
2. Click one of the **Find loops in Workspace** structure buttons.

If the structure in the **Workspace** came from a homology model (the structure has a **Homology Status** atom property), you can click **Non-Template** to load only the loops that did not come from the templates. Otherwise, click **All** to load all loops.

The loop features of the **Workspace** structure appear in the **Loops** table. For each loop, the table includes a **Run** check box, a feature name such as **loop1**, the chain name, and the numbers of the first (**Res1**) and last (**Res2**) residues in the loop. Note that loops of one or two residues are not listed, as they are unsuitable for the refinement protocol.





**Figure 6.2. The Refine loops task in the Refinement panel.**

3. Select a loop by clicking its name in the Feature column.

The loop residues are highlighted in the Workspace. If you are refining a homology model, you should consider residues with a Homology Status of 2 for loop refinement (non-template loops).

4. If necessary, shorten the loop region to be refined by changing the first and last residue numbers (Res1 and Res2).

The time required to refine a loop scales approximately linearly with the length of the loop, and the sampling options recommended for loops longer than five residues are also more time-intensive. The estimated time for refinement is given in the table.

5. (Optional) Click Options to select a sampling method and make settings for this loop in the Structure Refinement Options dialog box.

If the loop is longer than five residues, you should select a sampling method other than the default. See [Section 6.8.4](#) for information on the loop refinement options. The approximate number of subjobs that will be run is indicated along with the sampling method.

6. To designate this loop for refinement, select the check box in the Run column.

The Start and Write buttons becomes available.

7. Repeat [Step 3](#) through [Step 6](#) for each loop you want to refine.

8. (Optional) Set up and enable an implicit membrane model.

9. Click Start.

The Start dialog box opens.

10. Make any job settings, then click Start.

If you are running refinements on longer loops, you should choose a host with multiple processors or cores. The loop refinement job is started when you click Start.

When you choose several loops for refinement, they are refined in series. The first selected loop is refined, the best scoring structure for that loop is determined, that structure is used in refining the next loop, and so on. When all loops have been refined, one final structure is returned. If only one loop is refined, the four highest scoring structures are returned by default.

If you want to refine two loops cooperatively, you should replace [Step 3](#) through [Step 7](#) above with the following steps:

1. Click the Run column for the two loops you want to refine cooperatively.
2. If necessary, shorten the loop region to be refined by changing the first and last residue numbers (Res1 and Res2).
3. Click Options to open the Structure Refinement Options dialog box.
4. Select Cooperative loop sampling in the Loop refinement section.
5. Set any other options, then click OK.

For technical details about loop refinement, see [Section 6.10.1](#).

When a loop refinement begins, a validation program checks the loop features of the structure. Apparent errors are reported in a warning dialog box. You may choose Run All Features, Run Only Valid Features, or Cancel. It is recommended that you make a note of the invalid features, click Cancel, and correct the structure if appropriate.

## 6.5 Predicting Side Chains

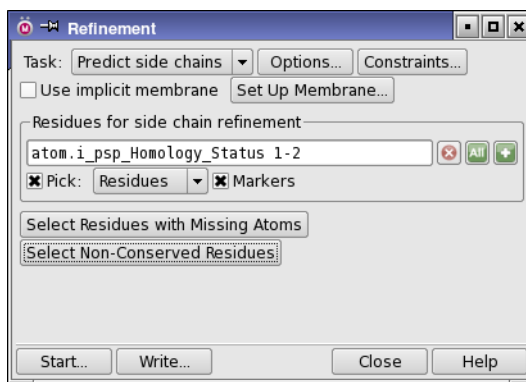
### To predict side-chain conformations:

1. Select Predict side chains from the Task options menu.

The panel displays atom selection options under the heading Residues for side chain refinement.

2. Use the atom selection options to specify the residues for which you want to predict or refine side-chain conformations (See [Chapter 6](#) of the *Maestro User Manual* for information on selecting atoms).

If you imported a PDB structure that has residues with missing atoms, you can select those residues by clicking Select Residues with Missing Atoms. The selection is based on



**Figure 6.3.** The Predict side chains task in the Refinement panel.

the information generated when the structure was imported, not on the current structure. The selection therefore includes any residues that were fixed since import.

If you are refining a homology model, you can select residues whose side-chain coordinates were not obtained from the templates (the Homology Status atom property is 1 or 2) by clicking **Select Non-Conserved Residues**. For residues with a Homology Status of 2 (none of the coordinates came from the template), you should consider doing a loop refinement first.

3. Click **Options** to choose the sampling algorithm, which determines how much of the backbone is allowed to be flexible in the side-chain prediction.
4. Click the **Run** button to start the side-chain prediction job.

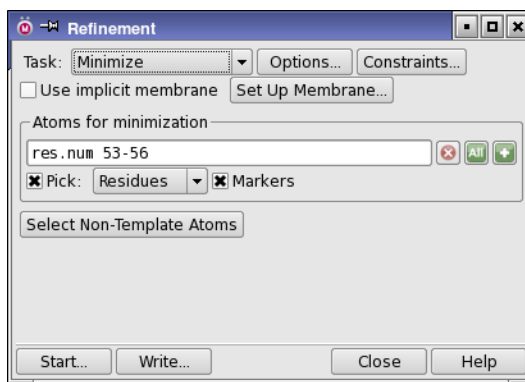
For technical details about side-chain prediction, see [Section 6.10.3](#).

## 6.6 Minimizing Structures

The Minimize refinement task performs a truncated-Newton energy minimization, using the OPLS\_2005 all-atom force field for proteins as well as for ligands and cofactors, and treating solvation energies and effects via the Surface Generalized Born (SGB) continuum solvation model.

### To minimize all or part of the structure selected:

1. Select **Minimize** from the Task menu.  
Atom selection options are displayed under the heading **Atoms** for minimization.
2. (Optional) Set up and use an implicit membrane.



**Figure 6.4. The Minimize task in the Refinement panel.**

3. Specify a set of atoms or residues for minimization with the picking tools, or click **Select Non-Template Atoms** to minimize atoms in a homology model whose coordinates did not come from the templates.

For more information on atom selection, see [Chapter 6](#) of the *Maestro User Manual*.

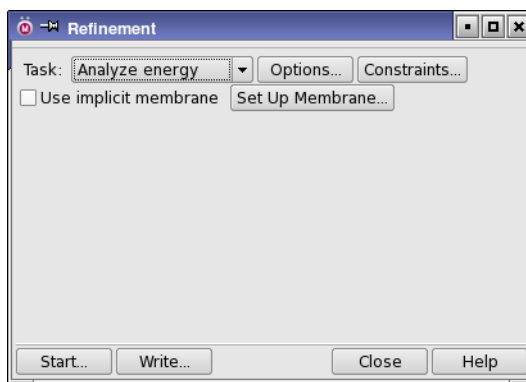
4. Click **Start** to start the minimization job.

## 6.7 Analyzing the Energy

To perform a single-point molecular mechanics energy calculation, choose **Analyze energy** from the Task option menu. The calculation uses the OPLS\_2005 all-atom force field for protein residues as well as for ligands and cofactors. If desired, set up an implicit membrane model. The output includes a breakdown of the energy into covalent, Coulombic, van der Waals and solvation energy contributions. Each of these is added as a Maestro property to the output structure file, named Prime Covalent, Prime Coulomb, Prime vdW, and Prime Solvation. The output also includes a breakdown of the energy on a per-residue basis.

## 6.8 Setting Refinement Options

The Structure Refinement Options dialog box offers three sets of options: General options, Dielectric options, and options for Loop Refinement. Click **Options** in the Refinement panel to view or change the options and settings.



**Figure 6.5.** The *Analyze energy* task in the *Refinement* panel.

### 6.8.1 General Options

The General section supplies options that apply to all types of refinement. The options include specification of a random-number seed for starting the side-chain prediction and use of crystal symmetry.

#### Seed

Specify the seed for the random-number generator used in side-chain predictions. The seed is used for side-chain prediction in both the loop refinement and the side-chain prediction tasks. The options are:

- Random: use a random seed.
- Constant: use the integer given in the text box as the seed. The default is 0.

#### Use crystal symmetry

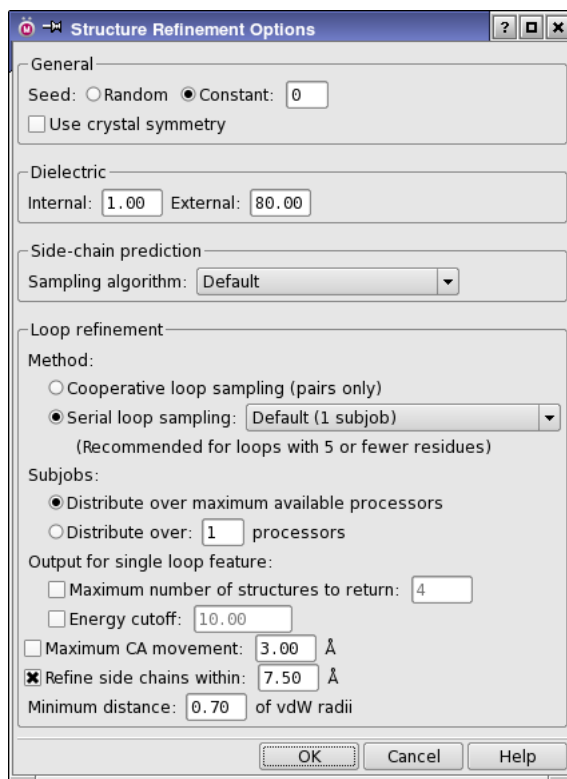
If crystal symmetry is known for this protein, apply periodic boundary conditions so that the crystal symmetry is satisfied.

### 6.8.2 Dielectric Options

The Dielectric section allows you to set the dielectric constants used in the calculations, in the following text boxes.

#### Internal

Set the dielectric constant used for the interior of the protein and any implicit membrane.



**Figure 6.6.** The Structure Refinement Options dialog box.

#### External

Set the dielectric constant for the (continuum) solvent.

### 6.8.3 Side-Chain Prediction Options

The Side-chain prediction section allows you to choose a sampling algorithm for the prediction of side chains. These are primarily aimed at cases where some degree of backbone sampling is likely to be required, such as in homology modeling or cross-docking, where the backbone conformation of the initial structure might not be entirely correct. These methods allow for progressively more backbone movement during the side-chain prediction calculations. There are three choices on the Sampling algorithm option menu:

- **Default**—No backbone sampling or reorientation of the CA – CB bond is performed.

- With CA-CB vector sampling—Vary the orientation of the CA – CB bond by up to 30° from the initial direction.
- With backbone sampling—Sample the backbone by running a loop prediction on a set of 3 residues centered on the residue for which the side chain is being refined.

While both of these algorithms can be run on any arbitrary set of residues in a given structure, the general intent is that they should be run on a relatively small subset of the structure, such as a binding site or other region of interest. Running these calculations on an entire structure is likely to result in more structural changes than were intended. For example, since the CA-CB sampling method involves a minimization of all residues being predicted, using this method on the entire structure results in minimization of the entire structure in the final stage, which is generally undesirable.

If heavy atoms are added to a residue because they were missing, a boolean property `side chain added (b_psp_side_chain_added)` is added to all atoms in the residue. You can use this property to select these side chains for other jobs, such as minimization.

## 6.8.4 Loop Refinement Options

The Loop Refinement section provides options for selecting the sampling method, job options, output options, and some general options.

Prime can perform cooperative loop sampling as well as serial loop sampling. Both the size of the loop and the sampling options chosen affect the time needed for loop refinement. The options are:

### Method

- Cooperative loop sampling—Select this option to refine two loops together. Each loop is refined in turn until the refinement has converged for both loops. You must have only two loops selected to use this option.
- Serial loop sampling—Select this option to refine each loop independently. The option menu provides three levels of sampling accuracy: Default (up to 5 residues), Extended (6 to 11 residues), and Ultra Extended (10 or more residues). The text below the option and menu displays the recommended loop length for the selected accuracy. The menu options indicate how many subjobs will be used for the loop refinement, so you can choose an appropriate number of processors for the job.

Extended sampling samples loop conformations more thoroughly, using a series of loop constraint settings. Up to eight processors can be used simultaneously to perform loop refinement. If you select Ultra Extended, you can only refine one loop at a time.

### Subjobs

These options allow you to set a limit on the number of processors over which to distribute the loop sampling subjobs. Each sampling method runs in several stages, and each stage can use a different number of processors. The maximum number of processors that can be used by the different sampling methods are:

Default	1
Extended	4
Ultra Extended	8*L (L is the loop length)

- To make use of as many processors as are available, select Distribute subjobs over maximum available processors.
- To set the maximum number of processors, select Distribute subjobs over and enter a value in the text box.

### Output for single loop feature

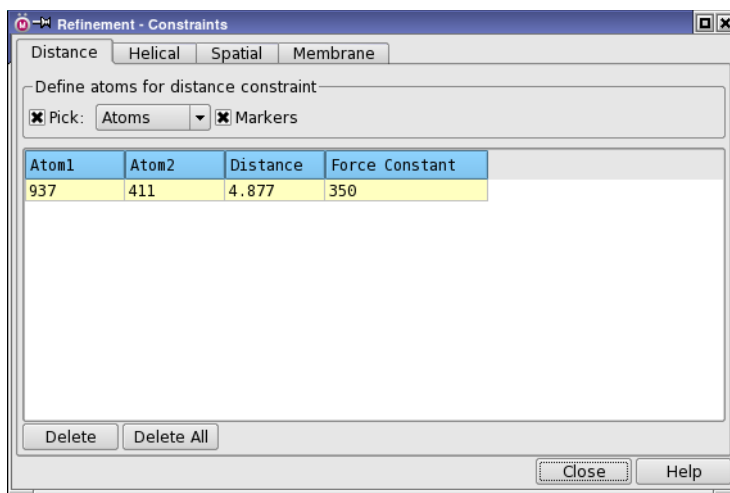
These options determine how many structures are returned:

- Maximum number of structures to return: Select this option to return more than one structure for each loop specified for refinement. The default when this option is selected is to return 4 structures for each loop.
- Energy cutoff: Select this option to prevent higher-energy structures from being returned. Structures whose energy is higher than that of the lowest-energy structure by more than the specified amount are not returned; the default cutoff is 10 kcal/mol.

The following settings are applied to the loops you have selected for refinement.

- Maximum CA movement: To limit the extent to which alpha carbon atoms can move from their original positions, select this option, and enter a limit in the text box. The default distance is 3.0 Å. The extended sampling procedure tests multiple values for Maximum CA movement, so this option is unavailable when extended sampling is selected. This option applies to all alpha carbon atoms. For more control over which alpha carbons are restrained and by how much, use spatial constraints, which are described in the next section.
- Refine side chains within: Select this option to refine side chains within a given distance of the loops, and specify the distance in the text box. Side chains are included in the refinement if the distance between any heavy atom in the side chain and any CB atom in the initial set of loop structures is less than the specified distance. The default distance is 7.5 Å.





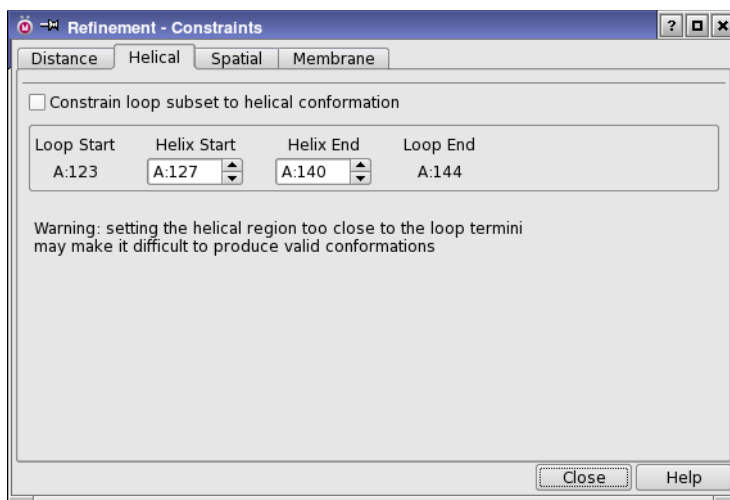
**Figure 6.7.** The Distance tab in the Refinement - Constraints panel.

- **Minimum distance:  $N$  of vdW radii:** Specify the minimum distance between atoms in terms of the fraction of their ‘ideal’ van der Waals radii. The default for this value is 0.70, meaning that the distance between atoms must be at least 70% of their ‘ideal’ van der Waals separation. The extended sampling procedure tests multiple values for the minimum overlap, and therefore this option is unavailable when extended sampling has been selected.

## 6.9 Applying Constraints

Prime provides four kinds of constraints that you can apply in a refinement: distance constraints (also known as pairwise constraints), helical constraints, spatial constraints, and membrane constraints. You can apply the constraints in the Refinement - Constraints dialog box, which you open by clicking Constraints. The various kinds of constraints are set in separate tabs.

Distance constraints can be applied to any pair of atoms, and are applied as harmonic constraints. Normally, these constraints are made between nonbonded atoms, but you can pick bonded atoms as well. To set up a constraint, ensure that Pick is selected in the Define atoms for distance constraint section, and pick two atoms in the Workspace. The atom indexes are recorded in the table, with the current distance and a default force constant of  $350 \text{ kcal mol}^{-1} \text{ \AA}^{-2}$ . You can edit the distance to force the atoms to a new distance, and you can edit the force constant. To delete a constraint, select it in the table and click Delete; to delete all constraints, click Delete All. These constraints set `PAIR_CONSTRAINT_i` keywords in the input file.



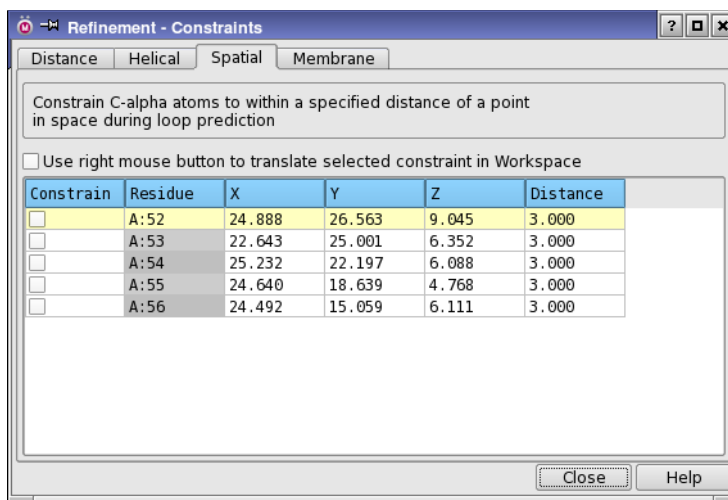
**Figure 6.8. The Helical tab in the Refinement - Constraints panel.**

Helical constraints can only be used in loop refinement. The constraint forces a subset of the residues in the loop to adopt an alpha helical conformation. You can specify the first and last residues in the helix in the Helix Start and Helix End boxes. By default, these are set 4 residues from the end of the loop. If you do not leave enough residues outside the helical region, the loop prediction might not produce good conformations, because it is too constrained. To turn on the constraint, select **Constrain loop subset to helical conformation**. This constraint sets a [HELIX\\_BUILD](#) keyword in the input file.

Spatial constraints are also only available for loop refinement. They specify the maximum distance that any alpha carbon atom in the loop should be allowed to move during refinement. The C-alpha atoms are listed in the table. To apply the constraint, check the check box for the residue in the **Constrain** column. The constraint is represented as a sphere in the Workspace. The current position of the C-alpha atom is listed in the table, and the maximum distance is also given.

You can edit the coordinates to constrain the C-alpha to a different position, and you can edit the maximum distance. You can also select a constraint in the table, select **Translate constraint**, and drag the sphere in the Workspace to a new location to change the position for the constraint. You might want to do this, for example, to force a loop into a different region of space. These constraints set [MAX\\_CA\\_MOVEMENT](#) keywords in the input file.

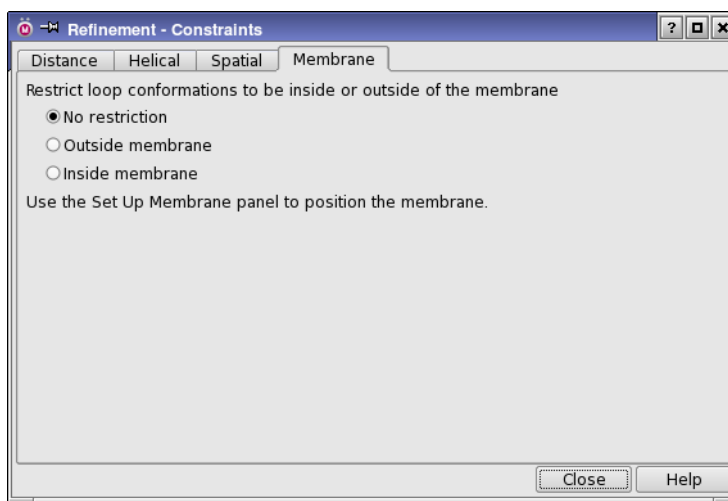
Membrane constraints restrict the loop conformation placement with respect to the implicit membrane. You can restrict the conformation to be outside the membrane (**Outside membrane**) or inside the membrane (**Inside membrane**), or you can allow the loop to be placed anywhere (**No restriction**).



**Figure 6.9. The Spatial tab in the Refinement - Constraints panel.**

The constraints are applied to the C $\alpha$  atoms only: other atoms of the residues are not restricted. Side chains can cross the boundary, for example. The constraints are applied only in the building stage of loop refinement, and subsequent minimization can move the residues across the membrane boundary.

There is no requirement that the implicit membrane model, with the adjusted dielectric constant, is actually used in the loop refinement. You can use the membrane to define a spatial exclusion zone, and then the constraints are just spatial constraints.



**Figure 6.10. The Membrane tab in the Refinement - Constraints panel.**

## 6.10 Technical Details for Refinement

### 6.10.1 Loop Refinement

Loop refinement using the Default option proceeds as follows:

1. The loop is reconstructed using the backbone dihedral library, by building up half from each direction. The resolution starts off very coarsely, becoming finer until it manages to produce a required number of physically realistic loop conformations (based on the length of the loop).
2. The large number of loops generated in the first step are clustered, and representatives of each cluster are selected.
3. These cluster representatives are then scored as follows:
  - a. Side chains are re-added to the loop residues, as well as any residues within a specified cutoff, according to the side-chain optimization procedure.
  - b. The loop and contact side chains are minimized.
  - c. The energy is calculated.
4. The best scoring loop structures are returned.

Loop prediction with extended sampling is specifically designed to overcome sampling problems with long loops, in which the number of residues results in an unwieldy number of possible loop conformations.

Extended sampling uses the following procedure:

1. Two initial predictions are run on the loop. These are intended to coarsely sample conformational space and are carried out with two slightly different sets of prediction parameters. Each initial prediction returns up to four top-scoring conformations, resulting in up to eight structures. These eight structures are intended as initial points for finer sampling of what appears to be favorable regions (basins) of conformational space.
2. Each of the eight structures generated in [Step 1](#) is run through another stage of refinement, with a 4 Å restriction on C $\alpha$  movement. This is intended to sample the basins more finely.
3. The eight refined structures from [Step 2](#) are combined with the original structures from Step 1, and this set of 16 structures is ranked by energy. The five top-scoring structures are then subjected to a final stage of refinement, with a 2 Å restriction on C $\alpha$  movement. This is intended to perform fine-grained sampling of the best basins.

This procedure defines the “Extended High” sampling procedure. The “Extended Medium” and “Extended Low” carry four and two structures, respectively, into [Step 2](#) and [Step 3](#).

In ultra-extended sampling, the three steps of extended sampling are followed by the steps below.

4. The top four structures are selected, and two predictions run on each, with part of the loop frozen. Runs are done with “moving windows” with varying numbers of contiguous residues: all  $n$  residues, the two possible contiguous sets of  $n-1$  residues, the three sets of  $n-2$  residues, and so on. The structures are sorted after each cycle of a given number of residues.
5. [Step 2](#) is repeated with the top eight structures cumulatively generated, then [Step 3](#) is repeated with the top eight structures cumulatively generated.

### 6.10.2 Cooperative Loop Refinement

Cooperative loop refinement proceeds as follows:

1. A list of side chains that will be optimized is generated. By default this list includes any residue with an atom within 7.5 Å of either loop 1 or loop 2.
2. Loop 1 is sampled, allowing side chains in the list from [Step 1](#) to move, holding the backbone of loop 2 fixed at its original conformation. Eight loop conformations are generated.
3. Loop 2 is sampled, allowing side chains in the list from [Step 1](#) to move, holding the backbone of loop 1 fixed at its original conformation. Eight loop conformations are generated. This step runs concurrently with [Step 2](#).
4. Conformations of loop 1 from [Step 2](#) with loop 2 from [Step 3](#) are merged into a set of structures.
5. All side chains in the list determined in [Step 1](#) are reoptimized, keeping only those structures whose energy is below the energy cutoff.
6. All residues in both loops are then minimized, including the backbone, keeping only those structures whose energy is below the energy cutoff.
7. The 4 best new structures are taken as input for resampling, starting from [Step 2](#). The process is repeated until convergence is obtained.

### 6.10.3 Side-Chain Prediction

The structure refinement program uses the following procedure to re-predict conformations for the side chain set that you select. The default algorithm is as follows:

1. Side-chain rotamers are randomized for nonconserved residues (the default) or for all residues.
2. Beginning with the first residue to be predicted, the side-chain rotamer library is used to find the rotamer with the lowest energy while keeping all other side chains fixed. Once the process is complete for the first residue, the next residue is treated, and so forth until all have been done once (a single pass has been completed).
3. Once the pass is complete, [Step 2](#) is repeated from the beginning, and then repeated again until the side-chain rotamers appear to be converged (no more changes are occurring).
4. Minimization is run on all of the side-chain atoms (but not backbone atoms) of the residues being treated.

Two methods are available for situations where some degree of backbone movement is likely to be required to get the correct side-chain conformation. This typically happens when working with homology models or when performing cross-docking studies, where the backbone conformation of the initial structure might not be entirely correct. These two methods allow for progressively more backbone movement during the sidechain prediction calculations, and are described below. They are intended for use on a small set of residues, not the entire structure.

#### **With CA-CB vector sampling**

Even minor discrepancies in backbone conformation can result in a slight shift in the location of the CB atom. This slight shift can often cause subsequent side-chain predictions to fail in placing the sidechain optimally, particularly on large rigid aromatic residues. Allowing for sampling of the CA-CB vector in conjunction with the side-chain prediction can alleviate this problem. The method is turned on in the input file with the keyword `SAMPLE_CBETA`. The algorithm is as follows:

1. CB positions in a conical region are systematically sampled around the initial position with a maximum displacement in the CA-CB vector of 30.0 degrees (settable with input file keyword `MAXCONEANG`).
2. At each position, the optimal sidechain rotamer is selected.
3. The best overall combination of CB position and rotamer for the residue is selected from the results for all positions.
4. The sampling iterates over all requested residues until convergence is achieved in the same way as with conventional sidechain prediction (see above).

5. Once all side-chain sampling has completed, the sidechains and backbone atoms of the predicted residues are minimized.

### With backbone sampling

This method is intended for situations where even larger backbone movements may be required, but where a full loop prediction on the entire region may produce more movement than desired. It is turned on in the input file with the keyword `SAMPLE_BACKBONE`.

For each residue for which side chains are predicted, a backbone loop segment of 3 residues (settable with the input file keyword `BACKBONE_LEN`), centered about that residue is reconstructed using the normal loop prediction algorithm. Because this loop prediction includes the additional residues on either side of each requested residue, you should expect to see the conformations of these residues optimized as well (i.e. the region that is optimized extends somewhat beyond the residues actually selected for side-chain optimization).

Side-chain prediction is not performed for GLY and ALA residues with any of these methods.

## 6.11 Refinement with Nonstandard Residues

If you want to refine a structure with nonstandard residues, you can make changes to the residues in the structures before refinement. Several nonstandard residues are supported in the Build panel: the neutralized forms of ASP, LYS, ARG, and GLU, which are named ASH, LYN, ARN, and GLH; and the tautomers of HIS (HID=HIS and HIE) and its protonated form, HIP.

Ionization (deprotonation) of four other residues are supported, CYS, SER, THR, and TYR. The names of the ionized residues are CYT (thiolate), SRO (alkoxide), THO (alkoxide), and TYO (phenoxide). In addition, the disulfide-bridged CYS is supported as CYX. To change one of these residues, you can either change the name, or change the structure. For example, you can generate an ionized SER by either of the following methods:

- Changing the residue name to SRO. The HG is deleted and a formal negative charge is added to the OG.
- Deleting the HG (if present) and adding a formal charge to the OG. The residue is renamed to SRO.

## 6.12 Output of Refinement Jobs

By default, the refined structures are appended to the Project Table for the project currently open. If you would prefer the output structure to replace the input structure, or would rather not incorporate the output structure into the project at all, select the appropriate option from the Incorporate option menu in the Start dialog box:

- Append new entries as a new group
- Append new entries individually
- Replace existing entries
- Do not incorporate

In addition to returning the structures, the output of a refinement job include several Maestro properties. The main properties are Prime Energy and Prime Energy Difference, which reports the difference in energy between the input and output structures. The energy difference is cumulative: if the property already exists, the difference is incremented in the current job. For loop refinement and side-chain prediction jobs, the lipophilic term is reported separately as Prime Lipo; the sum of the Prime energy and the lipophilic term is used to rank loop conformations.



# Maestro Protein Structure Alignment

It is sometimes useful to be able to align proteins outside the usual Prime workflow. Maestro provides access to the structure alignment facilities of Prime so that you can perform such alignments. Two panels are available from the tools menu: the Protein Structure Alignment panel, and the Align Binding Sites panel.

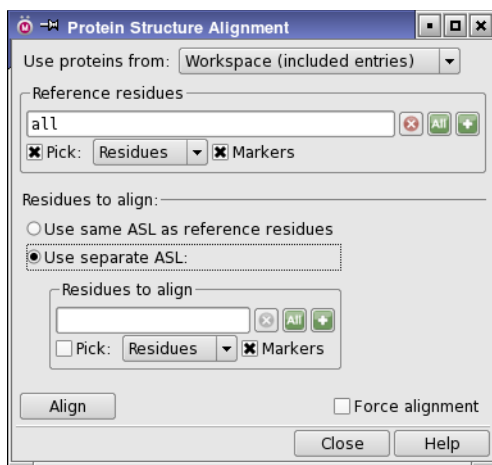
## 7.1 The Protein Structure Alignment Panel

To open the Protein Structure Alignment panel, choose Tools → Protein Structure Alignment.

In the Protein Structure Alignment panel, structure alignment is performed on Project Table entries that have been included in the Workspace or on the selected entries. You can choose which to use from the Use proteins from option menu. If you choose Workspace (included entries) the reference structure, whose frame of reference is used for the alignment, must be the first included entry (the entry with the lowest row number). To make another protein the reference, move that protein above all the other included entries in the Project Table. If you choose Project Table (selected entries), the reference structure must be the only structure included in the Workspace.

By default, the alignment includes all residues. However, it is often useful to align a subset of residues that have greater similarity than the structures as a whole, or are more informative to compare. Because this program uses matching of secondary structure elements, the subset should have enough contiguous residues to include at least one helix or strand. Very small numbers of residues do not produce useful alignments.

When the alignment calculation is complete, the structures are placed in the same frame of reference in the Workspace, and the results of the calculation are listed in the Protein Structure Alignment - Results panel. The aligned residues are listed and an RMSD and Alignment Score are reported. The RMSD and alignment score are also added to the structures as Maestro properties, `StructAlign RMSD (r_psp_StructAlign_RMSD)` and `StructAlign Score (r_psp_StructAlign_Score)`. The RMSD is calculated based only on those residues that are considered to have been successfully aligned, and therefore does not represent an overall comparison of the structures. It is computed from the C-alpha atoms of the aligned residues. An Alignment Score lower than 0.6–0.7 indicates a good alignment. Alignment Scores greater than about 0.7–0.8, or a failure of the structural alignment calculation, indicates there is insufficient structural similarity for a meaningful alignment. For details on the algorithm, see the paper by Honig and Yang (*J. Mol. Biol.* **2000**, 301, 665).



**Figure 7.1. The Protein Structure Alignment panel.**

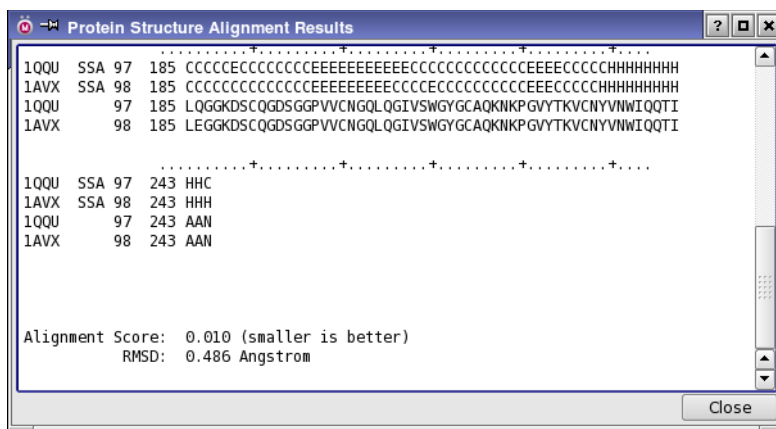
To run a default alignment job, include two or more protein structures in the Workspace, open the Protein Structure Alignment panel, and click Align to start the alignment calculation.

The Protein Structure Alignment panel provides some tools for controlling which parts of the protein are aligned and how they are aligned. These tools are described below.

If you do not want to use all residues in the reference protein for the alignment, you can select the residues you want to use in the Reference residues section, by using the picking tools. The remaining residues are ignored in the alignment.

Alignment can be performed for all residues or for a subset of residues. There are two options:

- **Use same ASL as reference residues**—If you restricted the residues in the reference protein, select this option to use the same expression for the proteins to be aligned as for the reference. This does not necessarily mean the exact same residues (although it can), because the ASL expression is evaluated independently for each protein being aligned.
- **Use separate ASL**—Select this option to specify the subset with a different residue selection (ASL expression) from the reference, and use the picking tools to make the selection. You might need to try different subsets to obtain the most useful alignment. Subsets should include at least one secondary structure element (helix or strand). Clicking the Common objects button then choosing Select opens the Atom Selection dialog box to the Residue tab. Here you can select a range of residue numbers, specify portions of the residue sequence, or click Secondary Structure in the list of options on the left to choose residues based on their location in Helix, Strand, or Loop structures. Subsets can be combined, intersected, or otherwise modified in the Atom Selection dialog box.



**Figure 7.2. The Protein Structure Alignment Results panel.**

When the job starts, the Protein Structure Alignment Results panel opens, displaying the text “Calculating alignment”. When the job finishes, the results are displayed and the structures are aligned.

## 7.2 The Align Binding Sites Panel

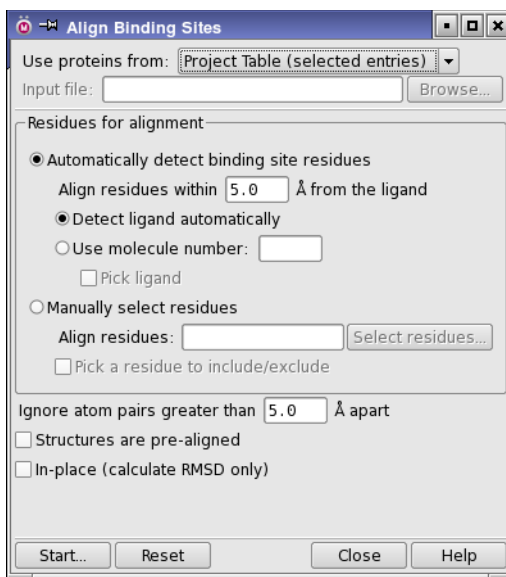
The Align Binding Sites panel allows you to align the binding sites (or other structural features) of members of a family of proteins. The Align Binding Sites job first runs a Protein Structure Alignment to obtain the global alignment and then automatically generates the list of atoms to use in a pairwise alignment of the C-alpha atoms from the residues selected.

To open the Align Binding Sites panel, choose Tools > Align Binding Sites in the main window.

The Use proteins from option menu provides choices of the source of protein structures. If you choose Project Table (selected entries), the entries that are selected in the Project Table are used. If you choose Workspace (included entries), the entries that are include in the Workspace are used. If you choose File, the Input file text box and Browse button become available, and you can browse for the input file. The first entry, either from the Project Table (lowest row index) or from the file, is the reference structure to which the other proteins are aligned.

In the Residues for alignment section you select the residues that are to be used to align the proteins. There are two options, described below.

- Automatically detect binding site residues—Align residues within a specified distance of the ligand. The distance is specified in the Align residues within  $N$  Å from the ligand text box. Two options are available for specifying the ligand:



**Figure 7.3. The Align Binding Sites panel.**

- Detect ligand automatically—Detect the ligand by selecting the molecule that is most ligand-like (number of atoms).
- Use molecule number—Specify the ligand by molecule number. You can enter a number in the text box, or select Pick ligand and pick a ligand atom in the Workspace.
- Manually select residues—Specify the residues to align by using any of the following methods:
  - Enter the residue specification in the Align residues text box, in the format *chain:number*.
  - Click Select residues to open the Atom Selection dialog box and define the residues.
  - Select Pick a residue to include/exclude and pick residues in the Workspace. Picking an already included residue excludes it.

The residues are listed in the Align residues text box, which you can edit to remove or add residues.

The alignment is performed on atom pairs. You can ignore atom pairs in the alignment process if they are further apart than a specified distance, by entering the distance in the Ignore pairs greater than *N* Å apart text box.

If you have proteins that are prealigned, you can select Structures are prealigned to suppress the global alignment of the protein structures, and align only by using the selected residues. If you want to calculate the RMSD between prealigned protein structures, select In place (calculate RMSD only).

When you have finished making settings, click **Start** to start the job.

You can also run a job to align binding sites from the command line—see [Section 11.3 on page 122](#) for more information.



# Docking Covalently Bound Ligands

While Glide is highly efficient at docking ligands that are bound to a receptor through hydrogen bonds or various nonbonded interactions, it is not able to dock ligands that are covalently bound to the receptor. Prime provides a covalent docking facility that allows you to select the attachment point to the receptor and the possible attachment points on the ligand, and run a Prime refinement to find poses for the ligand. Docking can be done to nucleic acids (DNA/RNA) as well as amino acids (proteins).

The covalent docking procedure works by defining the attachment location on the protein and on the ligand, and selecting a set of residues for refinement. The protein backbone is not adjusted during the process, only the side chains.

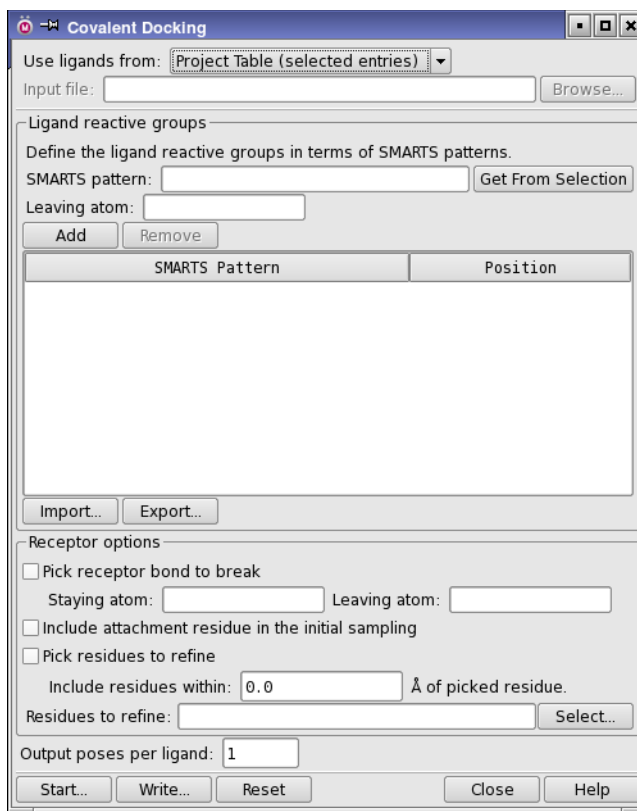
1. The side chains of the residues selected for refinement are temporarily removed.
2. The rotatable bonds of the ligand (and, if the attachment residue is selected for sampling, the rotatable bonds in that residue's side chain) are exhaustively sampled, producing a large number of potential poses.
3. These poses are then clustered into about 50 clusters.
4. For each cluster, the side chains of the residues selected for refinement are restored and predicted.
5. For each cluster, the ligand, the attachment residue (if included), and any residues specified for refinement are minimized.

The results of the minimization are the “poses” of the ligand.

## 8.1 Running Covalent Docking from Maestro

Covalent docking calculations can be set up and run from the Covalent Docking panel. To open the Covalent Docking panel, choose Applications → Prime → Covalent Docking or Tasks → Docking → Covalent Docking in the main window.

The Covalent Docking panel has two main sections. Above these sections, you can specify the source of the ligands. In the Ligand reactive groups section, you can specify the kinds of groups that will react at the receptor attachment site. In the Receptor options section, you can specify the receptor bond that is broken and choose receptor residues to refine.



**Figure 8.1. The Covalent Docking panel.**

### 8.1.1 Selecting the Ligands

The ligands that you dock to the receptor can be taken from the Project Table or from a file.

- To use the structures that are selected in the Project Table, choose Project Table from the Use ligands from option menu.
- To read the ligands from a file, choose File from the Use ligands from option menu. You can then either enter the file name in the Input file text box, or click Browse and navigate to the file in the file selector that opens. The file must be a Maestro file, and can be compressed (.maegz, .mae.gz) or uncompressed (.mae).

You should ensure that the structures are all-atom, 3D structures that are properly prepared, for example by using LigPrep. See the [LigPrep User Manual](#) for more information on ligand preparation.



The poses that are returned include both the ligand and the receptor, so you should consider carefully how many ligands you want to dock.

### **8.1.2 Defining the Ligand Reactive Groups**

In the Ligand reactive groups section, you specify the reactive groups on the ligands, in which a bond is broken to form a bond with the receptor. Each group is specified by a SMARTS pattern and the index of the atom that leaves, with its attachments, when the ligand bond is broken. You can specify multiple reactive groups. For each ligand, all groups are tried for a match to the ligand, and each match is docked.

To define each SMARTS pattern, you can enter the pattern in the SMARTS pattern text box, or you can select atoms on a typical ligand molecule in the Workspace, and click **Get From Selection** to place a pattern in the SMARTS pattern text box. You can then edit the SMARTS pattern in the text box if you want.

When you are satisfied with the SMARTS pattern, you must then designate an atom on the ligand that is removed, along with the atoms attached to it, when the ligand forms a bond with the receptor. The fragment with the smaller number of atoms is the part that is removed. The atom is specified by its index (position) in the SMARTS pattern; the first atom index is 1. This index must be entered into the Leaving atom text box.

Having defined a SMARTS pattern and the leaving group, click **Add**. The SMARTS pattern is added to the table below, and will be used when processing the ligands. You can repeat this process for as many SMARTS patterns as you want to use. If you want to remove a pattern from the table, select it and click **Remove**.

If you want to save the reactive group definitions for later use, you can export the SMARTS patterns and atom indexes to a CSV file, by clicking **Export**. You can then import them for another run by clicking **Import**.

### **8.1.3 Specifying the Receptor Bond to Break**

In the Receptor options section you specify the bond in the receptor that is broken to form a bond with the ligand. The receptor must be displayed in the Workspace, and is written to a file when you start the job.

To define the bond that is broken, select **Pick receptor bond to break**, and pick the desired bond on the receptor in the Workspace. The Staying atom and Leaving atom text boxes are filled with the atom numbers of the atom in the bond that remains and the atom in the bond that is removed when the bond is broken. The bond is marked in the Workspace with yellow markers. If you pick the wrong bond, you can simply pick again, and the new bond replaces the old. Picking of bonds to metals is not permitted.

You can also enter or edit the atom numbers in the Staying atom and Leaving atom text boxes. To ensure that you have the correct atom numbers, you should label the atoms. For more information on labeling atoms, see [Section 8.2](#) of the *Maestro User Manual*.

### 8.1.4 Specifying Receptor Residues to Sample

In the Receptor options section you can also choose residues that are allowed to adjust to the formation of the covalent complex in the side-chain sampling procedure and the refinement.

The first choice is whether to sample the residue that is attached to the ligand. To do so, select Include attachment residue in the initial sampling.

The next step is to pick other residues in the Workspace for refinement. To do so, select Pick residues to refine, and pick residues in the Workspace. You can pick any number of residues. As well as picking, you can select the desired residues in the Atom Selection dialog box, which you open by clicking Select. The residues that you select by either method are listed in the Residues to refine text box.

You can extend your selection to include residues that are within a given distance of the selected residues. To do so, enter a value in the Include residues within N Å of picked residues text box. These residues are also added to the list in the Residues to refine text box.

You can remove residues from the list to refine by deleting them from the Residues to refine text box.

### 8.1.5 Setting the Number of Output Poses

By default, covalent docking only returns a single pose, which is the lowest-energy pose. If you want multiple poses to be returned, you can enter the number of poses you want in the Output poses per ligand text box.

### 8.1.6 Running the Job

When you have finished making settings, click Start to open the Start dialog box. In this dialog box you can make job settings and start the job.

The results are returned in a Maestro file. Each pose includes both the ligand and the receptor, one pose for each ligand attachment point found.

### 8.1.7 Output

The output from a covalent docking job is the same as for a Prime energy calculation, and includes a breakdown of the energies into their contributions. Note that the contribution labeled covalent has nothing to do with covalent docking: it is the covalent contribution to the total energy.

## 8.2 Running Covalent Docking from the Command Line

It is also possible to run covalent docking jobs from the command line. To do so, you should first set up the input file in the Covalent Docking panel, as described in the previous section, and then click **Write**, to write the input file for the job. This button opens a file selector, in which you can browse to the location and choose the file name.

The covalent docking job is run with the same pipelining workflow scripts as other workflows, such as the Virtual Screening Workflow (VSW) and Quantum-Polarized Ligand Docking (QPLD). Covalent docking does not have its own script, so you can use the script for VSW, and run the job as

```
$SCHRODINGER/vsw [job-options] input-file
```

The job options are the usual Job Control options that specify the host and other resources. These options are described in [Section 2.3](#) of the *Job Control Guide*.



# Prime MM-GBSA

The Prime MM-GBSA panel can be used to calculate ligand binding energies and ligand strain energies for a set of ligands and a single receptor, using the MM-GBSA technology available with Prime. You can also run the job from the command line, which offers more flexibility than the panel.

The ligands and the receptor must be properly prepared beforehand, for example, by using LigPrep and the Protein Preparation Wizard. Waters and ions must be removed from the protein, as an implicit solvent model is used. The ligands must be pre-positioned with respect to the receptor, and the receptor must be prepared as for a Prime refinement calculation.

## 9.1 Running Prime MM-GBSA from Maestro

To open the Prime MM-GBSA panel, choose MM-GBSA from the Prime submenu of the Applications menu.

After preparing your structures, you can set up the calculation as follows:

1. Specify the source of the structures.

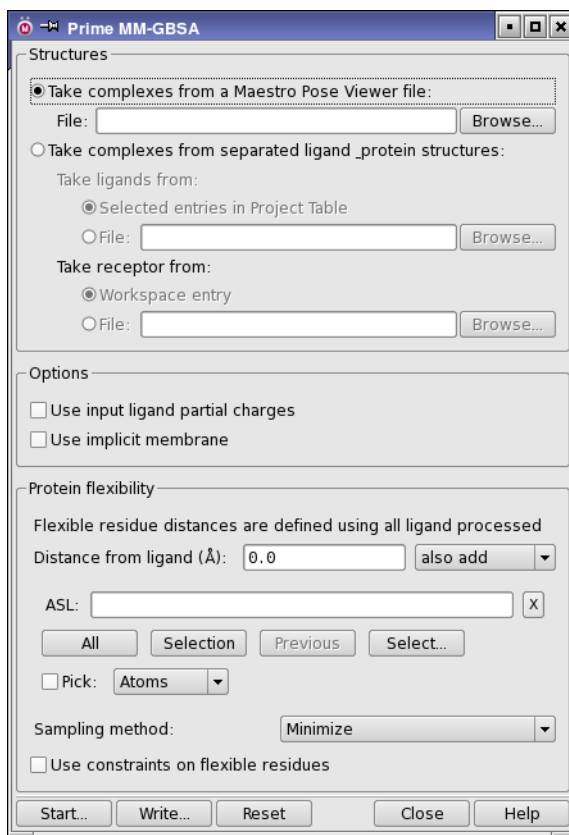
You can take structures from a Pose Viewer file (Glide output), or from separated ligand and protein structures. If you choose the latter option, you must ensure that the ligands and the protein are properly prepared and aligned.

2. (optional) Choose calculation settings.

If you want to evaluate ligand strain energies as well as ligand binding energies, select **Calculate ligand strain energies**. With this option, an extra calculation on the ligand is performed at its geometry in the complex, and this result is combined with the free ligand calculation to calculate the strain energy.

If you want to use the Prime implicit membrane model, select **Use implicit membrane**. The receptor you choose must have been already set up with the membrane, as described in [Section 6.3 on page 52](#).

3. (optional) Define a region within a certain distance of the ligand for which the protein structure will be relaxed in the calculation, in the **Protein flexibility** section.



**Figure 9.1. The Prime MM-GBSA panel.**

Residues that have any atoms within the specified distance of the first ligand processed are included in the flexible region. By default, all protein atoms are frozen, and only the ligand structure is relaxed. The larger the flexible region, the longer the calculation takes.

You can adjust this region by choosing also add, exclude, or intersect with from the option menu to the right of the text box, and specify the residues to add (logical OR), exclude (logical AND NOT), or intersect with (logical AND) using the picking tools below.

You can choose the method used to treat the flexible region from the Sampling method option menu, from Minimize, Minimize side chains only, and Minimize polar hydrogens only.

The movement of the flexible residues can be constrained with a harmonic potential (default strength  $10 \text{ kcal mol}^{-1} \text{ \AA}^{-1}$ ), so that they do not deviate too much from their initial position. To apply the constraints, select Use constraints on flexible residues.

4. Click **Start** to start the job.

When you click **Start**, the **Start** dialog box opens, in which you can choose to incorporate the results into the Maestro project, set the job name, select the host, and set the number of CPUs and number of subjobs, if you are running the job on a multiprocessor host or submitting it to a queuing system. You can divide the ligand set between a specified number of subjobs, to achieve load balancing. The number of subjobs must be no smaller than the number of CPUs, and for optimal load balancing, should be several times the number of CPUs.

Instead of starting the job, you can click **Write** to write the input files (structure input *jobname.maegz* and command input *jobname.inp*) and run the job from the command line

To clear all the settings in the panel, click **Reset**.

## 9.2 Running Prime MM-GBSA from the Command Line

You can run Prime MM-GBSA from the command line, with the input file written from the panel. The syntax is as follows:

```
$SCHRODINGER/prime_mmgsa [options] input-file
```

The input file must be a Maestro file. By default, the receptor must be the first entry, followed by one or more ligands (i.e. Glide pose viewer file format). The file can be compressed (.mae.gz, .maegz) or uncompressed (.mae). If the `-ligand` option is specified, the file must contain ligand-receptor complexes.

To restart a failed job, you can replace the input file with the *jobname.restart* file generated by the failed job.

If you have a command input file, *jobname.inp*, as written from the panel, for example, you can replace the input structure file and the command options with this file. The command input file contains a keyword for the structure input, so it need not be specified. You can also specify command options as well as a command input file, to override values specified in the file.

The standard Job Control options are accepted—see [Section 2.3](#) of the *Job Control Guide*—except that `-HOST` and `-SUBHOST` are used for distributing jobs, not `-DRIVERHOST`. The program options are described in [Table 9.1](#).

The MM-GBSA program has been substantially rewritten in Suite 2012, with new output. If you are relying on output in the earlier format, you can run the old version with the command

```
$SCHRODINGER/run -FROM psp prime_mmgsa2011.py
```

Table 9.1. Options for the *prime\_mmgbsa* command.

Option	Description
<i>Input and output options</i>	
-overwrite	Overwrite any previous instance of this job.
-save_workdir	Keep the working directory, which contains files for intermediate Prime jobs.
-out_type <i>TYPE</i>	Type of Maestro output file to write. The choices are: LIGAND—write a ligand-only file. PV—write the input receptor then the optimized ligand positions. Default for pose-viewer input. COMPLEX—write the optimized ligand and receptor conformations. Default for input of a series of complexes. Of these, only COMPLEX returns the optimized receptor conformation.
-ligand <i>ASL</i>	Specify the ligand with an ASL expression. If this option is present then the input file should be a Maestro file with each entry containing both the ligand and the receptor. The ASL expression provided here is used to determine which part of the complex structure is the ligand. This option is required for trajectory processing.
-thermal	Calculate thermochemical MM-GBSA properties. All structures must come from the same Desmond trajectory of a ligand-receptor complex. This option requires the -ligand option.
-report_top <i>numlig</i>	Report the specified number of top-scoring ligands in the log file. All ligands are reported in the CSV and structure output files.
-report_full	Report simplified results for all ligands in the log file, and full results in the CSV and structure output files.
-strain	Output an estimate of the strain energy for the ligand and the receptor, and create a dG Bind property without strain: Dg_Bind(NS). Receptor strain is only calculated if receptor flexibility is included.
-h[elp]	Print help message and exit
-v[ersion]	Print version information and exit
<i>Prime options</i>	
-use_ligand_charges	Use atomic partial charges from the input Maestro file for all ligand atoms.
-membrane	Use the Prime implicit membrane model in protein and complex simulations. Must be set up from Maestro.
-primeopt <i>list</i>	Set Prime options in the form <i>keyword/value/keyword/value...</i> , with no spaces. This option can be used for any settings you want to make that do not have explicit command options.



Table 9.1. Options for the *prime\_mmgbsa* command. (Continued)

Option	Description
-frozen	Do not treat any part of the ligand or receptor as flexible. This option takes precedence over all other flexible region options.
-ligand_region <i>ASL</i> -lssel <i>ASL</i>	Specify a section of the ligand to be treated as flexible, using an ASL expression. By default the entire ligand is flexible.
-rflexdist <i>value</i>	Specify the distance from the first ligand for defining which protein residues are treated flexibly. All residues that have atoms inside this distance are included in the flexible list. This options takes precedence over -receptor_region. Synonym: -flexdist.
-receptor_region <i>ASL</i> -rsel <i>ASL</i>	Designate a region of the receptor as flexible using an ASL expression. Expressions can refer to atoms in the ligand or the receptor. The selected region is the union of all the regions for each ligand-receptor pair in the input. By default the entire receptor is frozen.
-rflexgroup <i>group</i>	Specify the atom groups in each receptor residue that are to be sampled. Allowed values: residue— sample the entire residue side— sample only the side chains, and fix the backbone. polarh— sample only the polar hydrogens. Default: residue.
-job_type <i>type</i>	Prime job type to use to sample flexible regions. Allowed values are: REAL_MIN (default)—Local minimization. ENERGY—No sampling, just calculate a single-point energy. SIDE_PRED—Optimize side chains using Prime side-chain prediction.
-lcons <i>ASL</i>	Specify a portion of the ligand to harmonically constrain, using an ASL expression. By default no constraints are used.
-rcons <i>ASL</i>	Specify a portion of the receptor to harmonically constrain, using an ASL expression. By default no constraints are used.
-str_cons <i>value</i>	Specify the strength of receptor and ligand constraints in kcal mol <sup>-1</sup> Å <sup>-1</sup> . Default: 10.0.
<i>Job options</i>	
-NJOBS <i>njobs</i>	Divide the overall job into <i>njobs</i> subjobs.
-NSTRUCTS <i>nstructs</i>	Divide the overall job into subjobs with no more than <i>nstructs</i> structures.
-RETRIES <i>n</i>	If a subjob fails for any reason, it will be retried <i>n</i> times. Default: 3.
-NOLAUNCH	Set up subjob input, but don't run the jobs.
-RESTART	Restart a previously failed job. Already completed subjobs are not rerun. Use the <i>jobname.restart</i> file from the failed job as the input file.

### 9.2.1 Command Input File

If you want to use a command input file to store options, you can include any option other than the Job Control options. The input structure should be specified on a line with the following syntax:

```
STRUCT_FILE filename
```

All other options should be placed on separate lines with the preceding dash removed. For example, the following two calculations are equivalent:

```
prime_mmgsa file_pv.mae -job_type ENERGY -lcons SMARTS.C
```

and

```
prime_mmgsa input.inp
```

where the contents of `input.inp` are:

```
STRUCT_FILE file_pv.mae
JOB_TYPE      ENERGY
LCONS         SMARTS.C
```

The standard Job Control options must be set on the command line, because these are parsed by Job Control, not the MM-GBSA program. Other command options can also be specified on the command line, and override those in the file.

### 9.2.2 ASL Expressions

All atoms in the ligand have the `i_psp_Prime_MMGBSA_Ligand` property set to 1, so that ASL expressions can then be used for specifying parts of the structure relative to the ligand. For example,

```
fillres within 5 (atom.i_psp_Prime_MMGBSA_Ligand 1)
```

selects all residues within 5 Å of the ligand.

## 9.3 Prime MM-GBSA Output

The output Maestro structure file, which is compressed by default, includes a number of properties, which are described in [Table 9.2](#). The default file type is a pose-viewer file, except if the receptor is treated flexibly in a run started from Maestro, when the file contains complexes rather than a receptor and ligands.

The binding energy is calculated according to the equation:

$$\text{DG bind} = E_{\text{complex}}(\text{minimized}) - E_{\text{ligand}}(\text{minimized}) - E_{\text{receptor}}(\text{minimized})$$

Table 9.2. Output properties from a Prime MM-GBSA calculation.

Property	Description
Prime Coulomb	Coulomb energy of the complex
Prime Covalent	Covalent binding energy of the complex
Prime vdW	Van der Waals energy of the complex
Prime Solv SA	Nonpolar contribution to the solvation energy due to the surface area
Prime Solv GB	Generalized Born electrostatic solvation energy of the complex
Prime Energy	Prime energy of the complex
Prime MMGBSA Complex Energy	MMGBSA energy of the complex
Prime MMGBSA Ligand Energy	MMGBSA energy of the free ligand
Prime MMGBSA Receptor Energy	MMGBSA energy of the uncomplexed receptor
Prime MMGBSA Ligand in Complex Energy	MMGBSA energy of the ligand in the complex
Prime MMGBSA Ligand Strain Energy	MMGBSA ligand strain energy
Prime MMGBSA DG bind	MMGBSA free energy of binding. Includes ligand strain energy if this was selected for calculation.
Prime MMGBSA DG bind Coulomb	Contribution to the MMGBSA free energy of binding from the Coulomb energy
Prime MMGBSA DG bind Covalent	Contribution to the MMGBSA free energy of binding from covalent binding
Prime MMGBSA DG bind vdW	Contribution to the MMGBSA free energy of binding from the van der Waals energy
Prime MMGBSA DG bind Solv SA	Contribution to the MMGBSA free energy of binding from the nonpolar contribution to the solvation energy due to the surface area
Prime MMGBSA DG bind Solv GB	Contribution to the MMGBSA free energy of binding from the generalized Born electrostatic solvation energy
Prime MMGBSA DG bind no Ligand Strain	MMGBSA free energy of binding without including ligand strain. Only reported if ligand strain energy was selected for calculation.



# Command Syntax

This chapter summarizes the command syntax for various programs that are part of Prime. All the programs are executed by driver scripts, which run under the Schrödinger Job Control facility. For more information on this facility, see the *Job Control Guide*. The conventions used in describing the command syntax are outlined in the *Document Conventions* section at the front of this manual.

Residue specifications are used with a number of the input file keywords for several of the scripts. These have the format *C:N*, where *C* is a one-character chain ID ('\_' if the chain ID is blank) and *N* is the residue number and insertion code (if any). Atom specifications similarly have the format *C:N:A*, where *A* is the 4-character PDB atom name with spaces replaced by underscores, e.g. A:185:\_SG\_.

Most of these scripts accept the standard Schrödinger Job Control command line options and other commonly used options, listed in *Table 10.1*. Exceptions are noted in the program descriptions.

*Table 10.1. Standard Schrödinger job control options.*

Option	Description
-HOST <i>host</i>	run the job on the specified host. The default is the local host.
-LOCAL	Run the job in the local directory rather than the temporary directory
-TMPDIR <i>tmpdir</i>	Run the job in the temporary directory <i>tmpdir</i>
-WAIT	Do not return control to the shell until the job finishes
-INTERVAL <i>n</i>	Update the output every <i>n</i> seconds
-NICE	Run the job at reduced priority
-DEBUG	Turn on debugging for job control as well as the program
-HELP	Show information on command-line usage

## 10.1 blast—Run Blast Searches

The *blast* script provides an interface to the Blast and PSIBlast programs from NCBI (see <http://www.ncbi.nlm.nih.gov/books/NBK21097/> for information), while also supporting standard Schrödinger Job Control options. All options are passed on to the script via an input file.

## Syntax

```
$SCHRODINGER/blast jobname [options]
```

Input is read from the file *jobname.inp*. Output is written by default to *jobname.out*.

## Options

The `blast` script accepts the options listed in [Table 10.1](#).

## Input File

Keywords control the operation of the `blast` script, and are listed in [Table 10.2](#). Most keywords are simply a translation of options available from the `blastall` or `blastpgp` executables distributed by NCBI. For optional keywords, the Blast default values are used. Since these defaults can differ between different version of Blast they are not listed here.

*Table 10.2. Keywords for the blast script*

Keyword	Description
QUERY_FILE <i>filename</i>	Required. Name of the FASTA file containing the input/query sequence
DATABASE { <i>nr</i>   <i>pdb</i> }	Database to be used in the search. In a default installation the two allowed values are <i>nr</i> and <i>pdb</i> .
ALIGNMENT_FORMAT <i>value</i>	Alignment view options (Blast -m option). Allowed values are: <ul style="list-style-type: none"> <li>0 pairwise</li> <li>1 query-anchored showing identities</li> <li>2 query-anchored no identities</li> <li>3 flat query-anchored, show identities</li> <li>4 flat query-anchored, no identities</li> <li>5 query-anchored no identities and blunt ends</li> <li>6 flat query-anchored, no identities and blunt ends</li> <li>7 XML Blast output</li> <li>8 tabular</li> <li>9 tabular with comment lines</li> <li>m2io Maestro format (default)</li> </ul>
EXPECT <i>value</i>	Expectation value (E) (Blast -e option).
FILTER { <i>true</i>   <i>false</i> }	Filter query sequence: DUST with <code>blastn</code> , SEG with others. (Blast -F option).
GAP_OPEN_COST <i>value</i>	Cost to open a gap (Blast -G option)
GAP_EXTEND_COST <i>value</i>	Cost to extend a gap (Blast -E option)

Table 10.2. Keywords for the blast script (Continued)

Keyword	Description
MATRIX <i>type</i>	Sequence similarity matrix to use for the alignments. Allowed values of <i>type</i> are BLOSUM45, BLOSUM62, BLOSUM80, PAM30, PAM70
MAX_ROUNDS <i>n</i>	Maximum number of passes to use in multipass version (PSI-Blast -j option); only applies to runs using PROGRAM blastpgp
E_VALUE_THRESHOLD <i>value</i>	e-value threshold for inclusion in multipass model (PSIBlast -h option).
OUTPUT_FILE <i>filename</i>	Name of the output file. Default: <i>jobname</i> .out.
PROGNAME <i>name</i>	Type of blast search to run. The allowed values are blastp, blastpgp, and bl2seq (for pairwise alignments). Default: blastp.
TEMPLATE_FILE <i>filename</i>	Name of the FASTA file that contains the template sequence (bl2seq -j option). Only applies to pairwise alignments using bl2seq
PSSM_ASCII_OUTFILE <i>filename</i>	ASCII output file for PSI-BLAST matrix (PSIBlast -Q option).
EXPAND_HITS {true   false}	Expand redundant hits, based on information in the NCBI define for a particular hit. Only available when using Maestro format.
SHOW {all   pdb}	Allows the user to restrict the types of hits that are included in the output. all includes all hits; pdb only includes hits from the PDB. Only available when using Maestro format.

## Return Value and Errors

blast returns 0 if successful and a non-zero value in all other cases.

## Examples

The following is an example input file; query.aa is the query sequence in FASTA format.

```

QUERY_FILE      query.aa
PROGNAME        blastp
DATABASE        pdb

```

## Environment Variables

The following environment variables are supported by the blast script:

```

PSP_BLASTDB      directory containing the customized Blast databases
PSP_BLAST_DIR    directory containing a custom Blast installation

```

## 10.2 prime—Build and Refine Protein Structures

The `prime` program can be used to build initial (unrefined) protein models from an alignment to one or more templates, and to refine protein structures, using the Protein Local Optimization Program (PLOP). Structure refinement includes energy minimization, side-chain prediction, helix prediction, and loop prediction, with protocols for extended loop sampling, ultra-extended loop sampling, and cooperative loop pair sampling.

Multithreaded execution (with Open MP) can be enabled by setting the environment variable `OMP_NUM_THREADS` (see [Section 10.2.4 on page 104](#)).

The command syntax is

```
$SCHRODINGER/prime [options] jobname
```

The input is read from the file `jobname.inp` and the structural output is written to the file `jobname-out.maegz`. If the input structure file for a job has alternate coordinates, a refinement is performed on the first set of alternate coordinates. When building models, you can specify multiple input files to build a multimer model. The file `jobname.log` contains text output, and `jobname.err` contains any error messages.

The `prime` script accepts the options listed in [Table 10.1](#), and the `-NOJOBID` option, which runs the job directly rather than under Job Control. In addition, the `-examples` option displays a set of example input files, which are included in [Section 10.2.5 on page 105](#).

### 10.2.1 Command Input File

The command input file contains a list of keywords that control the execution of the program, one per line. The keywords are listed in the tables below. Generally valid keywords are listed in [Table 10.5](#) and protocol-specific keywords are listed in [Table 10.6](#), under the protocol to which they apply. The keywords and values are case-insensitive, except where the case is important, such as for file names.

Table 10.3. General keywords

Keyword syntax	Description
<code>JOB_TYPE job</code>	Type of job to run. The available options are: <code>MODEL</code> Build a structure for a homology model. <code>REFINE</code> Refine a structure.
<code>REDUN_CUTOFF tol</code>	RMSD cutoff for eliminating redundant conformers, where relevant.
<code>REMOVE_REDUNDANT {true false}</code>	Where multiple structures can be generated, remove redundant structures, as determined by <code>REDUN_CUTOFF</code> .



Table 10.4. Keywords for homology model building jobs

Keyword syntax	Description
BUILD_DELETIONS {true   false}	Required. Turn on or off closure of the chain breaks near deletions in the alignment. If not closed, they will remain as chain breaks in the output structure.
BUILD_TRANSITIONS {true   false}	Required. Turn on or off closure of the junctions between templates in multi-template homology modelling jobs. If not closed, they will remain as chain breaks in the output structure.
COMPOSITE_ARRAY <i>string</i>	A string of integers indicating from which template (or, more specifically, from which alignment) coordinates for a given residue should be obtained. Thus, this string of integers must be equal in length to the query sequence being used in the alignment files. COMPOSITE_ARRAY is required for multiple template jobs. It is not required for single template jobs, but if it is present, it is ignored.
FAST_SIDE_OPT {true   false}	Turn on or off fast side chain optimization, in which a set of side chains is selected that has no steric clashes but no energy minimization is performed. Requires SIDE_OPT. Default: false.
KEEP_ROTAMERS {true   false}	Required. Specifies which side chains to optimize. If true, the rotamers for conserved side chains are preserved, and only non-conserved side chains are predicted. If false, all side chains are predicted. This keyword is ignored if SIDE_OPT is false.
KNOWLEDGE_BASED {true   false}	Specify the method used to build the model: knowledge-based (true), which is faster and does not do any minimization, or energy-based (false), which uses the full model-building capabilities. Default: false.
MAX_INSERTION_SIZE <i>n</i>	Specifies the longest insertion in the alignment that will be built. Insertions longer than this will be omitted, and not appear in the output structure. Residue numbering will reflect the full query sequence however. Default: 1000.
MINIMIZE {true   false}	Turn on or off minimization of regions that were not directly copied from the template structure. These regions primarily include portions of the structure involved in closing gaps or building insertions, but also include any side chains optimized due to the SIDE_OPT keyword. Default: true.
MIN_OVERLAP <i>value</i>	Fraction of van der Waals distance used to define a clash. Smaller values allow structures with worse potential atomic overlaps to be generated. Default: 0.70
NUM_OUTPUT_STRUCT <i>n</i>	Number of output model structures to return. Only valid with KNOWLEDGE_BASED true.

Table 10.4. Keywords for homology model building jobs (Continued)

Keyword syntax	Description
<code>SIDE_OPT {true false}</code>	Turn on or off a side chain optimization stage. Default: <code>true</code> .
<code>template_ALIGN_FILE</code> <i>filename</i>	Required. The name of the file containing the alignment between this template and the query. Details on the format are provided in the Examples section below.
<code>template_HETERO_i</code> <i>ligand-spec</i>	Required if a ligand is present. Specifies a ligand to include from this template. The format of the specification is AAA C:### where AAA is the ligand's three letter code, C is its chain ID, ### is its residue number. Multiple ligands from the template are numbered using <i>i</i> as an index, starting from zero.
<code>TEMPLATE_NAME template</code>	Required. A label used to identify a given template in subsequent data fields. For example, if <code>TEMPLATE_NAME</code> is set to <code>1BPJ_A</code> , other template-specific fields are given as <code>1BPJ_A_STRUCT_FILE</code> , <code>1BPJ_A_ALIGN_FILE</code> , and so on. The chain to be used is specified by an underscore and letter suffix: for example, for <code>1BPJ_A</code> , chain A will be used. The chain suffix must be included in the label.
<code>template_NUMBER n</code>	Required. Used to identify a given template in the <code>COMPOSITE_ARRAY</code> (see below). Numbering goes from 1 to 9.
<code>template_STRUCT_FILE</code> <i>filename</i>	Required. The PDB file to be used for this particular template.
<code>TEMPLATE_RESIDUE_NUMBERS</code> <code>{true  false}</code>	Number the residues in the built structure the same as in the template, as far as possible. Does not apply to consensus modeling. When using multiple templates for a single chain, the residue numbers are taken from the first template. Sequential numbering from 1 is used if the attempt to use template numbering fails.
<code>USE_SYMMETRY {true false}</code>	Turn on or off use of the crystallographic symmetry of the template file when constructing the homology model. Default: <code>false</code> .

Table 10.5. Keywords for all refinement jobs

Keyword syntax	Description
CONSTRAINT_ <i>i constraint</i>	Define a constraint on atom positions, distances, angles or dihedral angles. See text for syntax details.
ECUTOFF <i>value</i>	Energy cutoff for return of conformations. Returns all conformations within <i>value</i> kcal/mol of the lowest-energy structure. It can be used in conjunction with NUM_OUTPUT_STRUCT, to impose a maximum on the number of such conformations to return. As with NUM_OUTPUT_STRUCT, it only applies to single loop refinements.
ENTRY <i>list</i>	Comma-separated list of entries in the input structure file to refine. Default: refine all structures.
EXT_DIEL <i>value</i>	Dielectric constant of the continuum solvation model. Default: 80.0. Used with SGB_MOD <i>sgbnp</i> .
INT_DIEL <i>value</i>	Dielectric constant used within the radius of an atom. Default: 1.0.
MIN_OVERLAP <i>value</i>	Fraction of van der Waals distance used to define a clash. Smaller values allow structures with worse potential atomic overlaps to be generated. Valid for LOOP_BLD and HELIX_BLD jobs. Default: 0.70
NUM_OUTPUT_STRUCT <i>n</i>	Number of conformations to return for single loop refinements or cooperative side-chain refinements. This has no effect on helix refinements or jobs composed of more than one refinement (e.g. 2 loops, 1 helix + 1 loop, etc.). All structures are returned in a single structure file in Maestro format. Default: 1.
PAIR_CONSTRAINT_ <i>i atom1, atom2, dist [, force]</i>	Specify a constraint on a pair of atoms. The format of the atom specifications is given at the beginning of this chapter. The value of <i>dist</i> is the target distance between the atoms in angstroms. The value of <i>force</i> is the force constant for the harmonic constraint in kcal mol <sup>-1</sup> Å <sup>-2</sup> which has a default of 350.
PRIME_TYPE <i>type</i>	Type of calculation to perform when running a refinement job (JOB_TYPE REFIN). The available options are: <ul style="list-style-type: none"> <li>SIDE_PRED      Predict side chains (prefix SC)</li> <li>SIDE_COMBI    Predict side chains cooperatively and exhaustively</li> <li>LOOP_BLD      Predict loops and helices</li> <li>EXTENDED      Predict extended loops (6-11 residues)</li> <li>LONG_LOOP_2   Predict ultra-extended loops (10 or more residues)</li> <li>LOOP_PAIR     Cooperative refinement of two loops</li> <li>HELIX_BLD     Rigid-body refinement of helices</li> <li>REAL_MIN      Minimize energy (prefix MINI)</li> <li>SITE_OPT      Active site minimization (including ligand)</li> <li>ENERGY        Energy calculation</li> </ul>

Table 10.5. Keywords for all refinement jobs (Continued)

Keyword syntax	Description
RESIDUE_ <i>i spec</i>	Specify a residue to refine. The numbering <i>i</i> begins at 0, and increases incrementally. The format of the residue specification <i>spec</i> is given at the beginning of this chapter.
RETAIN_CORRECTIONS {yes   no}	Retain corrections made when duplicate residue numbers are found. The duplications detected are between non-protein residues or between protein and non-protein residues. The non-protein residue is assigned a new, unique chain name temporarily. If you set this keyword to yes, the new chain name is written to the output file. Default: no.
SEED <i>n</i>	The integer to use as the seed for the random number generator if USE_RANDOM_SEED is no. If it is yes, SEED is ignored. Default: -1, to generate a random seed rather than use the supplied seed.
SELECT <i>string</i>	Specify the manner for specifying which residues to refine. Available options are: all Refine all residues pick Refine the specific residues, indicated by included RESIDUE_ <i>i</i> parameters (see above). file Read list of residues to refine from a file. The file consists of a series of lines with a residue specification on each line.
SGB_MOD {sgbnp   vdgbnp   vsgb1.0   vsgb2.0}	Specify the implicit solvation model to use. The choices are: sgbnp—use the standard generalized Born model. vdgbnp, vsgb1.0—use version 1 of the the variable-dielectric generalized Born model, which adapts the dielectric constant for residue type. vsgb2.0—use version 2 of the variable-dielectric generalized Born model, which incorporates a wider range of residue-dependent effects. Default: vsgb2.0.
STRUCT_FILE <i>filename</i>	Input structure file in Maestro format, compressed (.mae.gz, .maegz) or uncompressed (.mae). If the file contains multiple structures, each structure is refined, or the structures specified by ENTRY.
USE_CRYSTAL_SYMMETRY {yes   no}	Set to true if the input structure contains unit cell information and crystal symmetric atoms are to be included in calculation. Default: no.
USE_RANDOM_SEED {yes   no}	Indicates whether to use a random seed for the random number generator. This keyword has no effect on Minimization tasks. Default: no.
USE_MAE_CHARGES {yes   no}	Indicates whether to use atomic partial charges from the Maestro input file for untemplated residues (ligands, cofactors). Default: no.
USE_MEMBRANE {yes   no}	Indicates whether to use the implicit membrane model. The model must be set up from the Setup Membrane panel in Maestro. Default: no.
USE_SGB {yes   no}	Use the SGB implicit solvation model. If the model is not used, calculations are run in vacuum with a dielectric of 6.0. Default: no.

Table 10.6. Keywords for loop refinement jobs

Keyword syntax	Description
BURIED_ <i>i spec</i>	Constrain the specified residue to be buried (no more than 20% of surface area exposed to solvent).
CA_CONSTRAINT_ <i>i spec, [x,y,z] value</i>	Specify a constraint on the position of a specific alpha carbon atom. The format of the residue specification <i>spec</i> is given at the beginning of this chapter. The optional coordinates specify the target position; if omitted, the target position is the initial position. The value is the maximum distance that the C-alpha atom can move from the target position, in angstroms. Specifying a target position allows you to move a loop to a desired location.
EXPOSED_ <i>i spec</i>	Constrain the specified residue to be exposed (more than 40% of surface area exposed to solvent).
LOOP_ <i>i_RES_j spec</i>	Specify residue <i>j</i> for defining loop <i>i</i> . Loops are numbered sequentially, beginning at 0. Two occurrences of this keyword are required, with <i>j</i> =0 (the beginning of the loop) and <i>j</i> =1 (the end of the loop). <i>spec</i> is a residue specification as defined above.
HELIX_BUILD <i>spec1/spec2</i>	Build the specified sequence as an alpha helix. The format of the residue specifications <i>spec1</i> and <i>spec2</i> is given at the beginning of this chapter. You should include enough residues on either side of the helical region in the loop prediction to ensure that sufficient flexibility is available to build the loop.
MAX_CA_MOVEMENT <i>value</i>	Specify the maximum distance that any alpha carbon atom in the loop should be allowed to move during refinement. Omitting this parameter indicates that no restriction should be applied.
MEMBRANE inside  outside	Place the C-alpha atoms of the loop either inside or outside the membrane in the building stage. Refinement can move the residues across the membrane boundary.
RES_SPHERE <i>value</i>	All side chains that lie within this distance of the loop will also be optimized during refinement. This allows these nearby side chains to “react” to the loop being predicted. Default 0.0.
LOOP_NCLUST <i>number</i>	Specify the number of clusters to score for loop prediction. More clusters should result in more accurate predictions albeit with greater computational cost.

Table 10.7. Additional keywords for extended loop and multiple loop refinement jobs.

Keyword syntax	Description
MAX_CA_REF1 <i>value</i>	Specify the maximum distance that any alpha carbon atom in the loop should be allowed to move during refinement stage 1 (Ref1). Omitting this parameter indicates that no restriction should be applied.
MAX_CA_REF2 <i>value</i>	Specify the maximum distance that any alpha carbon atom in the loop should be allowed to move during refinement stage 2 (Ref2). Omitting this parameter indicates that no restriction should be applied.
MAX_JOBS <i>n</i>	Maximum number of subjobs that can run at one time. This can be set to a value less than the value of THREADS if sufficient resources are not available. Not available with refinestruct.
THREADS <i>n</i>	Number of simultaneous jobs to be run during selected refinement stages. Determines the sampling level for some protocols. Note: the jobs specified by THREADS do not have to actually run in parallel—see MAX_JOBS.
NUM_FIXED_STAGE <i>n</i>	Specify the number of stages in which backbones of residues at the beginning or end of the loop are fixed. These stages are inserted between the CA refinement stages. Each stage fixes progressively more residues, starting with 1. The fixed residues are distributed in all combinations between the ends of the loop, so, for example, stage 4 has 5 combinations and therefore 5 refinement jobs. Default: 5.
OFAC_INIT_STAGE <i>factors</i>	Run several initial stages with the specified minimum overlap factors (see <a href="#">MIN_OVERLAP</a> ). The factors are specified as a list separated by slashes, with no spaces. One stage is run for each factor. Default: 0.65/0.70/0.75.
PRIME_TYPE <i>n</i>	Type of refinement to be carried out. Can be specified as either a number or text string. Different from refinestruct. Available choices are: 0 or default: Equivalent to running a single loop refinement using refinestruct. Not particularly useful except for completeness and testing purposes. Does not recognize THREADS keyword. 1 or extended: Used for Extended Sampling from the GUI. Does not recognize MIN_OVERLAP and MAX_CA_MOVEMENT keywords, as these are used internally by the protocol. Accepts multiple loops as input which are run sequentially. 2 or long_loop: No longer used. 3 or loop_pair: Used for Cooperative Loop Sampling from the GUI. Exactly two loops must be specified. 4 or long_loop_2: Used for Ultra-extended Sampling from the GUI. Does not recognize the MAX_CA_MOVEMENT keyword, as this is used internally by the protocol. Only one loop can be specified.
STRUCT_FILE	Required. The input structure file in Maestro format, compressed or uncompressed. Same as for refinestruct.

Table 10.7. Additional keywords for extended loop and multiple loop refinement jobs.

Keyword syntax	Description
SEG_LIST <i>filename</i>	Name of a file containing the list of loops to be refined. For the example given for the keyword LOOP_i_RES_j, the file would have a single line: loop A:10 A:20 Not available with refinestruct.

Table 10.8. Keywords for active site optimization (SITE\_OPT) jobs

Keyword syntax	Description
BACKBONE_LEN <i>n</i>	Number of residues to include in the loop defined for backbone sampling in side-chain prediction. Only used with SAMPLE_BACKBONE. Default: 3.
LIGAND <i>string</i>	Residue specification for the ligand.
MAXCONEANG <i>value</i>	Maximum angular displacement of the CA-CB vector from the initial position when sampling CB positions. Only used with SAMPLE_CBETA. Default: 30°.
NITER_SIDE <i>n</i>	Number of iterations of side-chain prediction to perform. Larger numbers generally give better results. If set to zero, only the initial selection of rotamers without clashes is performed. Default: 1.
NPASSES <i>n</i>	Number of passes through side chain optimization of the protein residues followed by minimization of the protein residues (including the backbone). These two steps are repeated the specified number of times before proceeding to optimization of the protein and the ligand. Default: 2. Files written from Maestro have NPASSES set to 1.
SAMPLE_BACKBONE {yes   no}	Sample the backbone by running a loop prediction on a set of residues centered on the residue for which the side chain is being refined. The number of residues in the loop is defined by the BACKBONE_LEN keyword. Default: no.
SAMPLE_CBETA {yes   no}	Sample CB positions in a conical region around the initial position. The maximum displacement of the CA-CB vector in this region is set by the MAXCONEANG keyword. Default: no.

Table 10.9. Keywords for helix rigid refinement (*HELIX\_BLD*) jobs

Keyword syntax	Description
<i>HELIX_0_RES_i spec</i>	Specify helix and region containing helix. Values of <i>i</i> are: 0 start of region containing helix 1 start of helix 2 end of helix 3 end of region containing helix A keyword with each value of <i>i</i> must be included in the input.
<i>MAX_MOVEMENT value</i>	Maximum amount of movement allowed in the C and the N terminus of the helix. Overrides <i>MAX_C_MOVEMENT</i> and <i>MAX_N_MOVEMENT</i> . Default: 5 Å.
<i>MAX_C_MOVEMENT value</i>	Maximum amount of movement allowed in the C terminus of the helix. Default: 5 Å.
<i>MAX_N_MOVEMENT value</i>	Maximum amount of movement allowed in the N terminus of the helix. Default: 5 Å.
<i>ORIGIN_FOR_ROTATION value</i>	Fraction of the distance from the start to the end of the helix that is used as the origin for rotation. Default: 0.5.
<i>ROLL_HI value</i>	Minimum amount of roll sampled, in degrees. Default: 0.
<i>ROLL_LO value</i>	Maximum amount of roll sampled, in degrees. Default: 0.
<i>ROLL_STEP value</i>	Increment in roll between <i>ROLL_LO</i> and <i>ROLL_HI</i> , in degrees.
<i>SIDE_OPT_DISTANCE_CUTOFF</i>	Distance in angstroms to optimize side chains around the helix region.

Table 10.10. Keywords for side-chain prediction (*SIDE\_PRED*) jobs

Keyword syntax	Description
<i>NITER_SIDE n</i>	Number of iterations of side-chain prediction to perform. Larger numbers generally give better results. If set to zero, only the initial selection of rotamers without clashes is performed. Default: 1.



Table 10.11. Keywords for minimization (REAL\_MIN) jobs

Keyword syntax	Description
MINIM_NITER <i>n</i>	Maximum number of minimization cycles for the solvent treatment. Each cycle consists of a minimization to the specified RMSD with a particular set of solvent parameters, after which the solvent parameters are updated. The iterative procedure finishes if the energy changes by less than 1 kcal/mol or the number of cycles is exceeded. Default: 2.
MINIM_RMSG <i>value</i>	RMSD gradient convergence threshold. Default 0.01 kcal mol <sup>-1</sup> Å <sup>-1</sup>

## 10.2.2 Constraint Syntax

The syntax of a general constraint (CONSTRAINT\_*i*) is:

*atom-spec1 ; atom-spec2 ; ... ; strength ; equilibrium-value*

From one to five atom specifications can be given, in the format described at the beginning of this chapter. Positional and distance constraints have other specification options, as described below. A positional constraint requires one atom (or atom group) specification, a distance requires two atom (or atom group) specifications, an angle requires three atom specifications, and a dihedral requires four atom specifications.

*strength* is the interaction strength, in kcal mol<sup>-1</sup> Å<sup>-1</sup> for distances and kcal mol<sup>-1</sup> radian<sup>-1</sup> for angles.

*equilibrium-value* is the equilibrium value of the coordinate, and can be a single value or a range specified by *a+ -b* (meaning from *a-b* to *a+b*), *a+* (greater than *a*) or *a-* (less than *a*). The constraint potential is zero inside the range (flat-bottomed potential) and harmonic outside. The format *a+* is useful for keeping atoms separated; and *a-* is useful for constraining atoms near a starting point.

**Positional constraints:** specify one or more atoms with an atom or residue specification, or an ASL expression in the form *asl= (expression)*. The strength must be given; the equilibrium value is optional. If it is omitted, the atoms are constrained to their starting positions. If it is given, the syntax *a-* specifies the maximum distance from the starting position.

**Distance constraints:** specify two sets of atoms or a set of atoms and a point in space. The center of mass of the sets of atoms is constrained. The atoms can be specified with a atom or residue specification, or an ASL expression in the form *asl= (expression)*. The second specification can also be a set of Cartesian coordinates *x, y, z* or *current*, meaning the current location. For example *A:12; 0.0, 0.0, 1.0; 6.0; 0.5-* constrains the center of mass of residue A:12 to a position within 0.5 Å of the position at 1.0 Å in the *z* direction from the origin.

**Angle and dihedral constraints:** specify three atoms for an angle constraint and four atoms for a dihedral constraint. The strength and equilibrium distance are required.

To specify degrees for any quantity instead, append the letter D.

### 10.2.3 Restarting an Incomplete Loop Refinement Job

If a loop refinement job fails (one with protocol EXTENDED, LOOP\_PAIR, or LONG\_LOOP\_2), you can restart it by submitting it again from the same directory. The *jobname.restart* file and the archives of subjobs must be present in the directory, and the directory must be accessible from the host on which the driver job runs. Also, the job records for the failed jobs must not be purged from the job database, so that the restart mechanism has access to information on the failed subjobs. If the job records are missing, the subjobs are repeated.

Failed jobs that were started from Maestro must be restarted from the command line. Use -PROJ to assign the job to the Maestro project and -DISP to set the incorporation mode.

Once the job finishes successfully, the subjob archives and the restart file are removed.

It is not necessary to use -LOCAL or -SAVE to ensure that a job can be restarted, but there must be sufficient space in the working directory for the subjob archives.

### 10.2.4 Environment Variables

In addition to those used for all Schrödinger software, prime uses the environment variables listed in [Table 10.12](#):

*Table 10.12. Environment variables used by the prime script.*

Variable	Description
OMP_NUM_THREADS	Set the number of threads to use in multithreaded execution (with Open MP). It is recommended to set this environment variable in the <code>schrodinger.hosts</code> file for hosts on which multithreading is supported—see <a href="#">Section 7.1</a> of the <i>Installation Guide</i> .
PSP_OPLS_VERSION	OPLS force field version. The default is 2005. It can be set to 2001 to use the earlier set of force field parameters for ligands and non-standard residues.
PSP_DEBUG	If set, debugging is turned on and the subdirectory containing intermediate files is kept. Equivalent to using the command-line option -DEBUG.
PSP_LOCAL	If set, all calculations run in the local launch directory. Equivalent to using the command-line option -LOCAL.
PSB_SGB_MOD	Specify the solvent model. The value should be set to the same as the SGB_MOD keyword. This environment variable is overridden by a value in the input file.



### 10.2.5.3 Side-Chain Prediction

The following example demonstrates a side chain prediction of residues A:100, A:101, A:102, and A:103.

```
STRUCT_FILE input_structure.maegz
PRIME_TYPE SIDE_PRED
SELECT pick
RESIDUE_0 A:100
RESIDUE_1 A:101
RESIDUE_2 A:102
RESIDUE_3 A:103
USE_CRYSTAL_SYMMETRY no
USE_RANDOM_SEED no
SEED 0
EXT_DIEL 80.00
USE_MEMBRANE no
```

### 10.2.5.4 Minimization

The following example demonstrates the minimization of all hydrogens.

```
STRUCT_FILE input_structure.maegz
PRIME_TYPE REAL_MIN
SELECT asl = (atom.ele H)
USE_CRYSTAL_SYMMETRY no
USE_RANDOM_SEED no
SEED 0
EXT_DIEL 80.00
USE_MEMBRANE no
```

### 10.2.5.5 Loop Prediction

The following example demonstrates loop prediction of loop A:102 to A:110.

```
STRUCT_FILE input_structure.maegz
PRIME_TYPE LOOP_BLD
LOOP_0_RES_0 A:102
LOOP_0_RES_1 A:110
RES_SPHERE 7.50
MIN_OVERLAP 0.70
USE_CRYSTAL_SYMMETRY no
USE_RANDOM_SEED no
SEED 0
EXT_DIEL 80.00
USE_MEMBRANE no
```

## 10.3 ska—Protein Structure Alignment

The SKA program carries out structural alignment of proteins by aligning protein backbones at different levels of detail.

The `ska` script provides an interface to the SKA structural alignment program. As with other computational programs in Prime its operation is controlled through simple input files. The allowed keywords and their values are discussed in more detail below. The `ska` script allows you to run the SKA program in three different modes under Job Control:

- **align mode**—SKA aligns two or more protein structures to one another.
- **dbcreate mode**—SKA creates databases for use with scan mode. You should split PDB files with multiple chains into single-chain files before adding them to the database, as SKA has difficulty aligning to multiple chains.
- **scan mode**—SKA searches a database of protein structures and identifies potential structural analogues. By default a non-redundant database of protein chains from the PDB is searched.

### Syntax

```
$SCHRODINGER/ska [options] jobname
```

By default the input is read from the file `jobname.inp` and the output is written to `jobname.out`.

`ska` does not generate output other than the explicitly defined database and a log file in `dbcreate` mode—see below for details.

### Options

The only options apart from Job Control options accepted by `ska` are `-DEBUG`, which turns on debugging mode for both Job Control and the SKA program, and `-NOJOBID`, which runs the program outside Job Control.

### Command Input File

The input file consists of lines containing keyword-value pairs, one per line. The keywords are described in [Table 10.13](#). The set of allowed and required keywords depends on the mode, which is determined by the `MODE` keyword.

### Return Value and Errors

The script returns 0 for success and 1 for failure.

Table 10.13. Keywords syntax for ska input file

Keyword syntax	Mode	Description
MODE <i>mode</i>	all	The SKA operating mode; allowed values are: *align*: structurally align two or more protein structures to on another. This is the default mode *scan*: given a query structure scan a database of protein structures and identify potential structural homologues *dbcreate*: create a database that can be search in scan mode from a list of PDB files
QUERY_FILE <i>filename</i> [ <i>name</i> ]	*align* *scan*	Required. The input query structure. In *align* mode, the structure against which other proteins are aligned. In *scan* mode, the structure used for scanning the database. The optional <i>name</i> (*align* mode only) sets the name of this structure to a value other than the filename (the default). The name is displayed in the output file (see <a href="#">Examples</a> , below). <i>name</i> cannot contain the file separator character ('/' on UNIX).
TEMPLATE <i>filename</i> [ <i>name</i> ]	*align*	Required. PDB structures that are to be aligned to the query structure, set by QUERY_FILE. The TEMPLATE keyword can appear multiple times in order to be able to carry out multiple structure alignments (aligning more than one template structure to the query structure). The optional <i>name</i> sets the name of this entry to a value other than the filename (the default). The name is displayed in the output file (see <a href="#">Examples</a> , below). <i>name</i> cannot contain the file separator character ('/' on UNIX).
DATABASE <i>filename</i>	*scan*	The absolute path to the database to scan with the query structure. The database can be generated in dbcreate mode. Default: database supplied with Prime.
DBNAME <i>filename</i>	*dbcreate*	The name of the file in which to store the database create by ska. <i>filename</i> must be the file name without any path specification.
FILELIST <i>filename</i>	*dbcreate*	A file containing a list of absolute filenames for structures (in PDB format) that are to be included in a database of templates that can be search with ska in scan mode. Absolute filenames are required to avoid having to copy large numbers of PDB files to the temporary directory in which the job is run.

Table 10.13. Keywords syntax for ska input file (Continued)

Keyword syntax	Mode	Description
GAP_OPEN <i>value</i>	*align* *scan*	Gap initiation penalty.
GAP_DEL <i>value</i>	*align*	Deletion penalty.
OUTPUT_FILE <i>filename</i>	*align* *scan*	File in which to store the structural alignment. Default is <i>jobname</i> .out.
ALISTRUCS_OUTFILE <i>filename</i>	*align*	File in which to store the 3D coordinates of the aligned structures.
PSD_THRESHOLD <i>value</i>	*align* *scan*	PSD threshold for inclusion in the MSA. Overridden by RECKLESS.
SSE_MINSIM <i>value</i>	*align* *scan*	Minimum Secondary Structural Element (SSE) similarity for the alignment.
SSE_MINLEN <i>value</i>	*align* *scan*	Minimum SSE length for the alignment.
SSE_WINLEN <i>value</i>	*align* *scan*	Align only SSE windows of length <i>value</i> .
SWITCH [true false]	*align* *scan*	Carry out global alignment first and then align sub-windows if no similarity was found.
ORDER [seed psd]	*align*	psd: order the alignment by PSD to the seed structure.
CLEAN	*align*	Exclude structures with a PSD < PSD_THRESHOLD from the alignment.
RECKLESS [true false]	*align* *scan*	Force the alignment of structures even if no similarity is found.
USE_AUTOMATIC_SETTINGS	*align*	Automatically run scanning alignment with lengths 8, 9, and 13, and choose the best alignment.

In case of failure, check the log file for details on what might have caused the program to fail. To turn on debugging output use the `-DEBUG` command line option as follows:

```
$SCHRODINGER/ska -DEBUG jobname
```

## Environment Variables

The `ska` program handles all required environment variables internally and requires only that the environment variable `SCHRODINGER` be set. Should the environment variable `TROLLTOP` exist in the current environment it will be overwritten.

## Examples

The following minimal input file (assumed to be called `align.inp`) structurally aligns two protein structures to one another:

```
QUERY_FILE query.pdb
TEMPLATE template.pdb
```

where `query.pdb` is the name of the PDB file containing the query structure (the structure to which we want to align) and `template.pdb` contains the template structure (the structure we want to align to the query).

When run using `$SCHRODINGER/ska align` it should produce the following output:

- `align.out`—a file containing the structural alignment
- `align.log`—a log file that contains log information and diagnostics on the run

Note that the actual 3D coordinates of the superposed structures are not written out by default. You must use `ALISTRUCS_OUTFILE` to write out the structures.

The following input file (`align.inp`):

```
QUERY_FILE query.pdb
TEMPLATE template.pdb
ALISTRUCS_OUTFILE superposition.pdb
```

saves the 3D coordinates of the aligned structures in the file `superposition.pdb`.

The following is a sample input file for database creation in `dbcreate` mode:

```
MODE dbcreate
FILELIST filelist.dat
DBNAME mydatabase.dat
```

The file `filelist.dat` contains the file names for the templates, including the path, as follows:

```
/scr/templates/template-1.pdb
/scr/templates/template-2.pdb
/scr/templates/template-3.pdb
/scr/templates/template-4.pdb
```

A minimal input file for a scan is as follows:

```
MODE scan
QUERY_FILE query.pdb
```



```
QUERY_FILE query.pdb 1ctf
TEMPLATE 1dd3 A.pdb
```

```

      .....+......+.....+.....+.....+.....+.
1ctf    1                                     ceeeeeecc
1dd3_A  1  cchhhhhhhhhhchhhhhhhhhhhhhcccchhhhhhhhhhhhhhhhhhhcceeceeeecc
1ctf    1  -----EFDVILKAA
1dd3_A  1  MTIDEIIEAIEKLTVSELAEVLKKLEDKFGVTAAAPVAVAAAPVAGAAAGAAQEEKTEFDVLKSF

      .....+......+.....+.....+.....+.....+.
1ctf    62 ccchhhhhhhhhhhccchhhhhhhhhhhc c eeeeeccchhhhhhhhhhhhhcceeec
1dd3_A  67  ccchhhhhhhhhhhcchhhhhhhhhhhccccccceccchhhhhhhhhhhhhcceeec
1ctf    62 GANKVAVIKAVRGATGLGLEAKDLVES-P--AALKEGVSKDDAEALKKALEEAGAEEVEVK
1dd3_A  67  GONKIQIVIKVVREITGLGLEAKDLVEKAGSPDAVIKSGVSKEEAEEIKKLEEAGAEEVELK

```

Without the *name* argument for the QUERY FILE keyword, the output would be

[illegible]

Prime 3.1 User Manual | 111

## 10.4 ssp—Secondary Structure Prediction

The `ssp` script provides an interface to the secondary structure prediction programs SSPro, which is bundled with Prime, as well as PSIPRED, which is available from a third party.

### Syntax

```
$SCHRODINGER/ssp [options] jobname
```

By default the input is read from the file `jobname.inp` and the output is written to `jobname.out`.

### Options

The standard Job Control command line options listed in [Table 10.1](#) are accepted. There are no options specific to this program.

### Command Input File

The command input file consists of lines containing keyword-value pairs, one per line. The keywords are described in [Table 10.14](#).

Table 10.14. Keywords for the `ssp` input file.

Keyword syntax	Description
QUERY_FILE <i>filename</i>	Required. Name of the FASTA file containing the query sequence
METHOD <i>program</i>	Required. Name of the secondary structure prediction that is to be run. Allowed values are <code>sspro</code> , <code>psipred</code> , <code>profking</code> , and <code>all</code> . <code>all</code> means run all available programs. If the input file contains multiple METHOD keywords, only the last one is used.
OUTPUT_FILE <i>filename</i>	Optional. Name of the output file. Default is <code>jobname.out</code> .

### Return Value and Errors

`ssp` returns 0 if successful and a non-zero value in all other cases.

### Environment Variables

The environment variables that can be used to control the secondary structure prediction programs are listed in [Table 10.15](#).

### Examples

The following minimal input file, called `sspro.inp`, generates an SSPro secondary structure prediction for the sequence stored in file `query.fasta`:

Table 10.15. Environment variables for secondary structure prediction programs.

Environment Variable	Description
PSP_SSPRO_DB	BLAST database to be searched when assembling the multiple sequence alignment used in the SSPro prediction
PSP_PSIPRED_DB	BLAST database to be searched when assembling the sequence profile used in the PSIPRED prediction

```
QUERY_FILE query.fasta
METHOD      ssp
```

The prediction produced by the back-end is written to the file `sspro.out`.

The following input file, assumed to be called `all-methods.inp`, instructs the `ssp` script to produce secondary structure prediction for all methods available—SSPro and, depending on the local installation, PSIPRED.

```
QUERY_FILE query.fasta
METHOD all
```

## 10.5 sta—Single Template Alignment

The Single Template Alignment (STA) program in PRIME is designed for protein sequences with medium to high sequence identity (>25%). STA differs from standard sequence alignment programs such as BLAST by taking into account secondary structure matching as well as profile-sequence matching. As a result, STA often generates better alignment than BLAST in the regions where sequence conservation is relatively weak. STA can also take in constraints, such as user-selected residue pairs to be aligned together or residues in either query or template that are not to be aligned (gaps).

STA can use templates directly from the PDB, as selected by running the Find Homologs step, and can also use templates from the Prime fold library, as selected in the Fold Recognition step. For the latter, you can export the template from the Fold Recognition table, and save it as a PDB file.

### Syntax

```
$SCHRODINGER/sta [options] jobname
```

The input is read from the file `jobname.inp` and the output is written to `jobname.out`.

### Options

The `sta` script accepts the `-DEBUG` and `-HOST` options, as described in [Table 10.1](#).

## Command Input File

The input file consists of lines containing keyword-value pairs, one per line. The keywords are described in [Table 10.16](#).

Table 10.16. Keywords for the *sta* program.

Keyword syntax	Description
QUERY_NAME <i>name</i>	Name of the query sequence.
QUERY_FILE <i>filename</i>	File name of the query sequence in FASTA format.
FORMAT <i>string</i>	Alignment file format. Default is Maestro. Set to dev for plain text format—see <a href="#">page 115</a> for an example.
BGNRES <i>n</i>	The first residue of the query that is used in alignment. Default: 1.
ENDRES <i>n</i>	The last residue of the query sequence that is used. Default: the last residue of the original query sequence.
PREDSS[ <i>i</i> ] <i>filename</i>	Secondary structure prediction file of the query sequence, in CASP format. One occurrence of this keyword is required for each file, and there must be at least one SSP file specified. If only one prediction is used, the keyword is PREDSS; if more predictions are used, the keyword is suffixed with an index <i>i</i> , starting from zero: PREDSS_0, PREDSS_1, and so on.
TEMPLATE_PDB <i>filename</i>	Filename of the template structure, in PDB format.
MODE <i>string</i>	Optional. Needed in the input file only when there are user-specified constraints. In that case, the value is set to <i>user</i> .
PAIR <i>pair-list</i>	Optional. User-specified residue pairing, used only when MODE is set to <i>user</i> . All the residue pairs can be added after the keyword in one or multiple lines. For example, the following lines pair residue 12, 20, 60 from the query sequence with residue 15, 24, 65 from the template. MODE <i>user</i> PAIR 12 15 20 24 PAIR 60 65
GAP <i>gap-list</i>	Optional. User-specified gaps, used only when MODE is set to <i>user</i> . Can be used with or without PAIR. Gaps are specified by the residues in either query (P) or template (T) that are aligned with them. For example, the following opens gaps against query residue 10, 11, template residue 34 and 35. MODE <i>user</i> GAP P10 P11 T34 T35

## Files

- Input file—named *jobname.inp*. Contains keywords for running the program.
- Query sequence file—File containing the complete sequence of the query, in FASTA format. Specified in the input file.
- Secondary structure prediction files in CASP format for the query sequence.
- Template PDB file—Template structure file, in PDB format. The sequence used for the template is taken from SEQRES lines in the PDB file. If SEQRES is not found in the PDB file, the template sequence is then taken from ATOM records.
- Output alignment file—File named *jobname.out* containing pairwise alignment between the query and the template. By default, the alignment is in Maestro format. To generate plain text format, include `FORMAT dev` in the command input file. In plain text format, aligned residues from the query and the template are placed on top of each other—see below for an example.
- Log file—File named *jobname.log* containing progress of the `sta` program, including warnings and error messages.

## Return Value and Errors

`sta` returns 0 for success and non-zero for failure. In case of failure, check the log file for details, and search for `ERROR`. To further analyze the failure, use the `-DEBUG` option when you run `sta`.

## Examples

Below is an example input file, where `query_seq.fasta` stores the query sequence in FASTA format.

```
QUERY_NAME      query
QUERY_FILE      query_seq.fasta
PREDSS_0        query-ssp1.casp
PREDSS_1        query-ssp2.casp
PREDSS_3        query-ssp3.casp
BGNRES          1
ENDRES          149
TEMPLATE_PDB    pdb1vpe.ent
```

Below is an example of the plain text format for output alignment files. This format is generated when you include `ALIFORMAT dev` in the command input file.

```
probe length=70=====tmplt_T0281_d1dq3a2 length=79 TotalScore= 42 SEQ= -4 SSTR= 102
SEQID= 0.061538 nali= 65
```

```
probe bgnRes=1 endRes=70      template bgnRes=3 endRes=75

ProbeAA: ..MWMPPRPEEVARKLRRLGFVERMAKGGHRLYTHPDGRIVVVPFH...SGELP...KG
ProbeSS:  LLLLLLHHHHHHHHHLLLEEEELLLEEEELLLEEEeLLL  LLLLL  HH
Fold AA:  GNFGPLPLNFNAFKEWASEYGVEFKTNGSQTIAIIND...ERISLGQWHTNRVSKAVLVK
Fold SS:  LLEELLHHHHHHHHHLLLEEEELLLEEEELL  EEEELLHHHLLLEHHHHHHH

ProbeAA: TFKRILRDAGLTEEEFHNL....
ProbeSS: HHHHHHHHhLLLHHHHHL
Fold AA: MLRKLYEATK.DEEVKRMLHLIE
Fold SS: HHHHHHHHL LHHHHHHHHHL
```

The file contains header information at the top, then in the alignment section, the query sequence (ProbeAA), query composite secondary structure (ProbeSS), template sequence (Fold AA) and template secondary structure (Fold SS) are given. Gaps are denoted by periods in the sequences and by spaces in the secondary structures.

## 10.6 tfm—Tertiary Folding Module

The `tfm` program generates one or more complete backbone conformations starting from a primary sequence and secondary structure, along with optional domain boundaries and sheet topologies. In addition, `tfm` can accept several different types of constraint including multiple coordinate templates, multiple types of distance constraints, and dihedral angle constraints. A large number of run-time parameters allow the code to be used in a variety of ways, from minimizing a homology model to full *ab initio* folding.

### Syntax

The `tfm` program has hundreds of user-specified options contained in over a dozen input files. A detailed description of all the options is beyond the scope of this document. The `refine` command can be used as a wrapper to prepare default input files. If suitable input has been prepared, there is also a top-level wrapper for `tfm` that can be used to run the program directly under Schrodinger Job Control. It can be invoked from the command line via:

```
$SCHRODINGER/tfm input-file [options]
```

where *input-file* is typically the job name, with or without a `.inp` extension. The input file must be either the first or last command-line argument, and everything else on the command line is passed as is to Job Control.

### Options

All options are specified in the input file, and there are no command-line options, other than those recognized by Job Control.

## Command Input File

There are only three keywords recognized in the command input file, listed in [Table 10.17](#).

Table 10.17. Keywords for *tfm* program.

Keyword	Description
INPUT_FILE <i>filename</i>	The input file passed to the executable.
OUTPUT_FILE <i>filename</i>	The output produced by the executable.
STRUCT_NAME <i>name</i>	Optional. Prefix for files in the local directory that will be automatically treated as input, such as template or constraint files.

## Files

In addition to the descriptive output in the file specified by OUTPUT\_FILE and the error messages in *name.err*, *tfm* produces the following files, where the first part of the file name is *tfm* by default but can be changed by other input options, and *N* is a serial number.

*tfm.N.pdb*      Output structures, if applicable.

*tfm.N.spair*    Possible strand pair assignments for unpaired strands, if automatic strand-pairing is being used.

## Return Value and Errors

*tfm* returns a value of 0 on successful completion, 1 otherwise. All error messages are printed to the file *name.err*, where *name* is either the STRUCT\_NAME value or the SEQNAME parameter in the file specified by INPUT\_FILE. This file is empty otherwise, and all output from the program itself is written to the file specified by OUTPUT\_FILE. Output from Job Control goes to stdout, and any errors arising before the driver script has started will likewise go to stdout or stderr.

## Examples

Sample input for *tfm* can be generated by running *refine* with the -LOCAL and -DEBUG options. The command

```
refine sample
```

creates a directory *run\_sample* with the input file *sample.1.in* (among other files determined by the refinement type). An input file *sample\_tfm.inp* containing the following lines can be used to run the job from the command line

```
INPUT_FILE sample.1.in
OUTPUT_FILE sample.1.out
```

with the command

```
tfm sample_tfm
```

### Environment Variables

In addition to the standard environment variables tfm recognizes the following environment variables:

TFM_QUIET	These environment variables can be set to decrease or increase the job-
TFM_VERBOSE	related output in the log file. The value to which they are set is irrelevant.



# Command-Line Utilities

In addition to the command-line programs described in [Chapter 10](#), Prime provides a number of utilities, which are described in this chapter. These utilities are installed in the `utilities` directory in your Schrödinger software installation.

To run these utilities:

- On Linux and Mac, open a terminal window, set the `SCHRODINGER` environment variable, and prefix the command with `$SCHRODINGER/utilities/`.
- On Windows, open a command prompt window with `Start > All Programs > Schrodinger-2012 > Schrodinger Command Prompt`, then enter the command without any prefix.

## 11.1 Multiple Template Alignment: `structalign`

If multiple templates are selected in the Find Homologs step, they must be rotated into the same coordinate reference frame before they can be used. The `Align Structures` button is the usual means of performing structural alignment of templates, but the `structalign` utility can be run from the command line, as explained below.

You can specify the input as multiple files in PDB format, or as a Maestro file. The templates should consist of single chains. To use the output files, each must be imported into Find Homologs and selected as one of the templates, replacing the original, unaligned version.

In addition to its use in Prime Structure Prediction, this utility can also be used more generally to structurally align proteins. The syntax is:

```
structalign [options] reference-file input-files
```

The reference file and the input files can be in PDB or Maestro format. The output file name is generated by adding the prefix `rot-` to the input file name.

### Options:

- |                      |   |
|----------------------|---|
| <code>-h</code>      | Print usage message.  |
| <code>-force</code>  | Force alignment even if structures are not sufficiently similar.                      |
| <code>-noauto</code> | Use the structure titles as sequence names instead of the automatic numbering system. |

`-matrix` write out a transformation matrix for each Maestro-format input file instead of the rotated structure.

The following instructions are for use with the Prime Structure Prediction workflow.

### To obtain PDB structures of selected templates:

1. Click the sequence name of a template in the Find Homologs step, to display it in the Workspace.
2. Choose Project > Export Structures in the main window.
3. Select Workspace for the source of structures.
4. Choose PDB for the file format.
5. Name the file with the homolog ID.
6. Repeat the above steps for each template of interest.

### To export templates to a Maestro file:

1. Click the sequence name of a template of interest in the Find Homologs step, to display it in the Workspace.
2. Choose Workspace > Create Project Entry.
3. Save the entry with the ID as the title.
4. Repeat the above steps for each template of interest.
5. Select the templates in the Project Table.
6. Choose Table > Export Structures.
7. Select Selected entries for the source of structures.
8. Export to a single Maestro file.

At the command line, enter the appropriate form of the command:

```
$SCHRODINGER/utilities/structalign template1.pdb template2.pdb ...  
$SCHRODINGER/utilities/structalign templates.mae
```

This command runs the `structalign` utility and returns one aligned file for each template:

```
rot-template1.pdb  
rot-template2.pdb  
...
```

For a Maestro file, the files are named `rot-templates-N.pdb`, where *N* is the entry number in the file.

**Note:** If you encounter a problem involving templates with multiple chains, use the `getpdb` utility (see [Section 2.3](#) of the *General Utilities* manual) to extract each chain into a separate `.pdb` file, then run `structalign` again.

You can then import the templates into the Find Homologs step. Once the files are imported and the aligned templates appear in the Homologs table, select them all using shift-click or control-click. (Do not select the original structures.)

## 11.2 Format Conversion: seqconvert

This utility converts sequences between any permitted input file format and output file format. This may be useful if autoconversion does not proceed as expected.

### Syntax:

```
seqconvert [options] input-format input-filename output-format output-filename
```

One use of the `seqconvert` utility is to convert SSPs produced in Prime to FASTA format with the following command:

```
seqconvert -inative infile.seq -ofasta outfile.fasta
```

Options are described in [Table 11.1](#), including the input and output format options.

*Table 11.1. Keywords for the seqconvert utility*

Option	Description
-start	Read from this sequence in the input file
-end	Read to (and including) this sequence in the input file
-h	Display usage message
-total	Read at most this number of sequences
-ali	Read an alignment in the input file
-pdb_use_atoms	Use ATOM (not SEQRES) records with PDB files
<i>Input File Formats:</i>	
-iany	Autodetect input file format
-inative	Input file in native (Schrödinger) format
-ifasta	Input file in FASTA format

Table 11.1. Keywords for the `seqconvert` utility (Continued)

Option	Description
-ipdb	Input file in PDB format
-inbrf_pir	Input file in NBRF-PIR format
-itext_plain	Input file in TEXT-PLAIN format
-igcg	Input file in GCG format
-iswissprot	Input file in SWISS-PROT format
-igenbank	Input file in GENBANK format
-incbi	Input file in NCBI format
-iphylip	Input file in PHYLIP format
-inexus_paup	Input file in NEXUS-PAUP format
-iselex	Input file in SELEX format
-istaden	Input file in STADEN format
-ipfam_stockholm	Input file in PFAM-STOCKHOLM format
-icasp	Input file in CASP format
<i>Output File Formats:</i>	
-onative	Output file in native (Schrödinger) format
-ofasta	Output file in FASTA format
-onbrf_pir	Output file in NBRF-PIR format
-otext_plain	Output file in TEXT-PLAIN format
-ogcg	Output file in GCG format
-oswissprot	Output file in SWISS-PROT format
-ocasp	Output file in CASP format
-ogenbank	Output file in GENBANK format

### 11.3 Structure Alignment: `align_binding_sites`

This script performs a pairwise superposition of multiple structures using the C-alpha atoms of selected residues. The reference structure for the alignment is the first structure in the input file. The C-alpha atoms used for the alignment come from residues within the specified cutoff distance from the ligand in the reference structure. Alternatively, a list of residues may be given for the alignment, which must correspond to residues in the reference structure. C-alpha

atoms in the structures that are aligned that are greater than the specified distance from any C-alpha in the reference are not considered in the alignment. The RMSD matrix for the alignment is written to the file *jobname-matrix.csv* in three columns: the indexes for the structures, then the RMSD value. The syntax is:

```
align_binding_sites [options] input-files
```

If you specify more than one input file, the reference structure is the first structure in the first file. The options are given in [Table 11.2](#). The Job Control options described in [Table 2.1](#) and [Table 2.2](#) of the *Job Control Guide* are supported, with the exception of `-INTERVAL`.

*Table 11.2. Options for the align\_binding\_sites command.*

Option	Description
<code>-v[ersion]</code>	Show the program version and exit.
<code>-h[elp]</code>	Show usage message and exit.
<code>-l[igmol] molnum</code>	Molecule number of the ligand. Default is to automatically detect the ligand.
<code>-auto</code>	Use optimized settings automatically in SKA.
<code>-color</code>	Color the binding side based on the CA RMSD value. Not compatible with <code>-inplace</code> .
<code>-c[utoff] value</code>	Binding site cutoff distance from the ligand in angstroms. Residues within this distance of the ligand are used in the alignment. Default: 5.
<code>-d[ist] value</code>	Atom pairs greater than this distance are not considered in the alignment (default=5)
<code>-inplace</code>	Do not align the structures, but only report the RMSD between the input structures. Sets the <code>-prealigned</code> option.
<code>-o[utfile] filename</code>	Specify the output file name. The file type is determined by the extension, and can be in Maestro or PDB format. If PDB format is used, only one structure can be aligned.
<code>-p[realigned]</code>	Structures are prealigned. Do not run global protein structure alignment (SKA).
<code>-r[es] list</code>	Residues for the alignment. This option overrides the use of <code>-cutoff</code> . Format for a residue is <i>chain:resnum</i> . If there is no chain name, use an underscore, e.g. <code>_:999</code> . At least three residues are required. The comma-separated list should not have spaces.
<code>-j[obname] jobname</code>	Jobname to use. The default is based on the input file name.

## 11.4 Database Update Utilities

Prime uses two databases, the PDB database and the BLAST database. As these databases are continually being updated, the versions distributed with the software will not have the latest contributions. Utilities have been provided for updating these databases from the web sites. You can run these utilities on Windows from a Schrödinger shell—see [Section 4.2.4](#) of the *Installation Guide* for details. You must have permission to write to the destination in order to run these utilities.

### 11.4.1 update\_BLASTDB

This utility updates the BLAST database from a copy on the Schrödinger web site that is synchronized nightly with the BLAST web site, or via FTP from the NCBI web site. The preformatted database files are downloaded. The syntax is:

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

The options are described in [Table 11.3](#).

*Table 11.3. Options for the update\_BLASTDB utility.*

Option	Description
-b	Run in batch mode
-d <i>blast-dir</i>	Specify the location of the local copy of the BLAST database.
-f	Use ftp to the NCBI web site instead of rsync (default on Windows).
-h	Display usage message and exit.
-p <i>rsync-port</i>	Specify the port to use with rsync. Available ports are: 23, 80 (default), 33444.
-q	Run quietly (if using batch mode)
-r	Use rsync to update from the Schrödinger web site. (default on Linux)

The BLAST database is installed in the first location found in the following list:

- The location specified by the -f option.
- \$PSP\_BLASTDB
- \$SCHRODINGER\_THIRDPARTY/database/blast
- \$SCHRODINGER/thirdparty/database/blast

The environment variables are described in [Appendix B](#).

## 11.4.2 rsync\_pdb

This utility creates or updates a local mirror of the PDB. The syntax is:

```
$SCHRODINGER/utilities/rsync_pdb [-p port|-f] [-h]
```

The options are described in [Table 11.4](#).

*Table 11.4. Options for the rsync\_pdb utility.*

Option	Description
-p <i>port</i>	Specify port for rsync. Available ports: 23 (default), 80, 33444.
-f	Use FTP protocol (wget) by default for download. Default for Windows.
-h	Display usage message and exit.

The PDB is installed in the first location found in the following list:

- \$SCHRODINGER\_PDB
- \$SCHRODINGER\_THIRDPARTY/database/pdb
- \$SCHRODINGER/thirdparty/database/pdb

These environment variables are described in [Appendix B](#).

This utility uses an rsync server on the Schrödinger web site that enables the files to be downloaded. The files are written in the new remediated format. The Schrödinger site is updated nightly from the PDB. On Windows, the utility wget is used to update the local copy.





# Third-Party Programs

## A.1 Location of Third-Party Programs

As well as internal and bundled software and data, Prime uses certain external third-party programs and databases for sequence and homolog searches, protein family searches, and secondary structure prediction. These are described under the headings in this topic.

### A.1.1 BLAST/PSI-BLAST, Pfam, and the PDB

See the home pages for these programs and servers for more information:

- BLAST/PSI-BLAST

<http://www.ncbi.nlm.nih.gov/blast>

The BLAST version distributed with Prime is version 2.2.16. To download the BLAST database, you should use the `update_BLASTDB` utility. This utility uses `rsync` to download or update the database, in the correct format for use with Prime.

- HMMER/Pfam

<http://pfam.sanger.ac.uk>

This site has links to mirror sites that may be more convenient for downloads. The Pfam database is included in the Schrödinger software distribution, so you should not need to download it.

- PDB

<http://www.rcsb.org/pdb/>

Note that the PDB format has changed, as of August 1, 2007. You should use the `rsync_pdb` utility to update your local copy of the PDB for use with Prime. Prime uses the new format, and automatically converts files in the old format.

You can download and install the programs and databases in full, so that you have a local copy. You can also run searches of the PDB and BLAST databases on the web, which does not require a separate installation.

## A.1.2 Secondary Structure Prediction

Two distinct third-party secondary structure prediction programs can be used to increase the accuracy of, and get better results from, Prime. One of these programs (SSpro) is already bundled with Prime. The other (PSIPRED) is highly recommended, and may be manually installed by you, at your election, and subject to applicable license terms, if any. At the third-party web site whose address is given below, you may download the program and then install it. For instructions on how to install third-party programs, see [Section 3.9](#) of the *Installation Guide* (Linux) or [Section 4.2.4](#) of the *Installation Guide* (Windows), or the [Third Party Programs page](#) of our web site.

The PSIPRED home page is located at:

<http://bioinf.cs.ucl.ac.uk/psipred/>

and contains information about the program, referrals to terms of use and/or license terms (in the README file), and a link to download the program. The link to download PSIPRED is contained in the section of the page labeled “Overview of prediction methods.” PSIPRED 2.61 is required for use with Prime 3.1.

**Note:** PSIPRED is not available on Windows.

## A.2 Third-Party Program Use Agreement

By using the links above or the third-party programs, you acknowledge and agree to the following:

PSIPRED is a third party program (“Third Party Program”) and may therefore be subject to third party license agreements and fees. We (i.e., Schrödinger, LLC and its affiliates) have no responsibility or liability, directly or indirectly, for Third Party Programs or for any third party Web sites (“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.

# Environment Variables

The only environment variable that *must* be set before you can run Prime is SCHRODINGER. The following list is presented for your convenience in case you prefer a different location for any Prime-related directories.

## General

SCHRODINGER\_THIRDPARTY: Directory containing third-party software and databases, which can be installed with the installers supplied by Schrödinger. The databases are installed in the database subdirectory; the software in the bin subdirectory. Each third-party product has its own subdirectory under these subdirectories. The default location if this environment variable is not set is \$SCHRODINGER/thirdparty.

## PDB

SCHRODINGER\_PDB: Full path to a user-supplied copy of the PDB database. The default for this directory is \$SCHRODINGER/thirdparty/database/pdb if only SCHRODINGER is set; if SCHRODINGER\_THIRDPARTY is set, the default is \$SCHRODINGER\_THIRDPARTY/database/pdb. The copy of the database must retain the directory hierarchy used by the PDB and must include the following subdirectories to be usable with Prime:

```
data/structures/all/pdb
data/structures/divided/pdb
data/structures/obsolete/pdb
```

On Windows the data/structures/all subdirectory is absent (and is not required).

## Blast

PSP\_BLAST\_DIR: Location of Blast executables. Default is \$SCHRODINGER\_THIRDPARTY/bin/*platform*/blast/bin, with the default for SCHRODINGER\_THIRDPARTY noted above.

PSP\_BLAST\_DATA: Location of Blast matrices. Default is \$SCHRODINGER\_THIRDPARTY/bin/*platform*/blast/data, with the default for SCHRODINGER\_THIRDPARTY noted above.

PSP\_BLASTDB: Location of Blast databases. The BLAST databases distributed with Prime are formatted with version 2.2.16. If your BLAST database does not conform to this format, you might experience problems with Prime. Default is \$SCHRODINGER\_THIRDPARTY/database/blast, with the default for SCHRODINGER\_THIRDPARTY noted above.

### **HMMER/Pfam**

PSP\_HMMER\_DIR: Location of HMMER distribution (the executables are assumed to be in the binaries subdirectory). Default is \$SCHRODINGER\_THIRDPARTY/bin/*platform*/hmmmer, with the default for SCHRODINGER\_THIRDPARTY noted above.

PSP\_HMMERDB: Location of Pfam database. Default is \$SCHRODINGER\_THIRDPARTY/database/pfam, with the default for SCHRODINGER\_THIRDPARTY noted above.

### **PSIPRED**

PSP\_PSIPRED\_DIR: Location of PSIPRED installation (top level directory containing the bin and data directories). Default is \$SCHRODINGER\_THIRDPARTY/bin/*platform*/psipred, with the default for SCHRODINGER\_THIRDPARTY noted above.

PSP\_PSIPRED\_DB: Sequence database to use for assembling the PSI-BLAST profile. Allowed values are nr and pdb. Default: pdb.

### **SSPRO**

PSP\_SSPRO\_DB: Sequence database to use for assembling the SSPRO profile. Allowed values are nr and pdb. Default: pdb.

# File Formats

## C.1 Input File Formats

The input file formats are:

- native (Schrödinger) format
- FASTA format
- PDB format

Sequences are read from the PDB SEQRES records, if they exist. Otherwise they are read from the ATOM records. If SEQRES records do not exist and residues are missing from the ATOM records, the query sequence will of course contain gaps.

- NBRF-PIR format
- TEXT-PLAIN format
- GCG format
- SWISS-PROT format
- GENBANK format
- NCBI format
- PHYLIP format
- NEXUS-PAUP format
- SELEX format
- STADEN format
- PFAM-STOCKHOLM format
- CASP format

## C.2 Output File Formats

The output file formats are:

- native (Schrödinger) format
- FASTA format
- NBRF-PIR format
- TEXT-PLAIN format
- GCG format
- SWISS-PROT format
- CASP format
- GENBANK format



# Error Messages and Warnings

## D.1 Input Sequence

### Invalid File

This message appears if the system fails to extract a valid sequence from the file.

- Check that the sequence is in one of the supported sequence file formats.
- If FASTA format is used, make sure there are no extra spaces between the “>” and the sequence name, and that there are no extra spaces at the end of the line.

### No Sequences Found

This message appears if the system fails to extract any sequences from the workspace.

- Check the Project Table to ensure the correct entries are included in the Workspace.
- Use Open Project in the Maestro Window to open a project.

### Invalid Sequence Characters

This message appears if the sequence contains non-standard one-letter code for amino acids.

- Check that sequence uses the one-letter code and is all uppercase.
- Change/edit any non-standard characters from your sequence using an editor, and then restart job.

## D.2 Find Homologs

### D.2.1 Error Messages

#### Multiple Templates Not Aligned

- If you select multiple templates and proceed to Edit Alignment, you are reminded that the templates must be structurally aligned to one another for composite models.

If one of the following messages appears when you click **Next** to go to the Edit Alignment step, you will not be able to proceed along the Comparative Modeling path until the problem is resolved:

### No Homologs Found

- You may change your search options and continue searching, or proceed to Fold Recognition.

### Invalid File

- Check to ensure the homolog file exists in either of the following locations:  
\$SCHRODINGER/thirdparty/database/pdb/data/structures/divided/pdb/\*  
\$SCHRODINGER\_THIRDPARTY/database/pdb/data/structures/divided/pdb/\*  
and is in the right format (gzip compressed, pdb\*.ent.gz).

## D.2.2 Warnings in Homologs Table

These messages appear in the Warning column of the Homologs table for sequences that are likely to be unusable for model building:

### Many missing residues

- Too many consecutive missing residues to guarantee model building will succeed.

### !UNUSABLE: Alternate residue type

- The residue type was undetermined, causing the file to include the residue twice but with different identity (e.g. ALA20 and VAL20). This is not supported by Prime.

### !UNUSABLE: Bad SEQRES

- General problems with the SEQRES.

### !UNUSABLE: CA only

- File contains only alpha C coordinates.

### !UNUSABLE: Molecule *molname* should be labeled as HETATM

- A non-protein molecule has been listed as ATOM when it should be listed as HETATM. ATOM records are reserved for proteins only.

### !UNUSABLE: SEQRES/ATOM mismatch

- The sequence listed in the SEQRES records does not match the actual sequence in the ATOM records.

### !UNUSABLE: SEQRES is missing residues

- There are residues in the ATOM records that are not listed in the SEQRES records. PDB files require that the ATOM records be a subset of the SEQRES records.



### **!UNUSABLE: SEQRES numbering is wrong**

- The number of residues in a particular chain does not agree with the number of residues shown in columns 14 to 17 of SEQRES.

## **D.3 Edit Alignment**

### **Check Alignment**

When you click **Next** at the end of the Edit Alignment step, the Check Alignment warning may appear. The warning lists possible alignment problems that you may want to fix before continuing to the next step. For example, there may be gaps in secondary structure elements of the template, or gaps in the query that are aligned to secondary structure elements of the template.

If you do not want to make corrections to the alignment, click **Continue** to go to the next step. To make corrections based on the information in the error message, make a note of the gap locations given, click **Cancel** to close the message box while remaining in Edit Alignment, and change the alignment accordingly.

## **D.4 Build Structure**

Error messages and warnings for this generally appear in the log of a running job. The log can be viewed in the job Monitor panel until the job is completed.

## **D.5 Refinement**

Error messages for Refinement generally appear in the log of a running job. The following is an example of a warning in the log of a loop refinement.

### **Invalid Features**

When a loop refinement begins, a validation program checks the loop features of the structure. Apparent errors are reported in a warning dialog box. You may choose to **Run All Features**, **Run Only Valid Features**, or **Cancel**. It is recommended that you make a note of the invalid features, click **Cancel**, and correct the structure if appropriate.



---

# Getting Help

Information about Schrödinger software is available in two main places:

- The `docs` folder (directory) of your software installation, which contains HTML and PDF documentation. Index pages are available in this folder.
- The Schrödinger web site, <http://www.schrodinger.com/>, particularly the Support Center, <http://www.schrodinger.com/supportcenter>, and the Knowledge Base, <http://www.schrodinger.com/kb>.

## Finding Information in Maestro

Maestro provides access to nearly all the information available on Schrödinger software.

### To get information:

- Pause the pointer over a GUI feature (button, menu item, menu, ...). In the main window, information is displayed in the Auto-Help text box, which is located at the foot of the main window, or in a tooltip. In other panels, information is displayed in a tooltip.

If the tooltip does not appear within a second, check that **Show tooltips** is selected under **General → Appearance** in the Preferences panel, which you can open with **CTRL+, (⌘,)**. Not all features have tooltips.

- Click the **Help** button in a panel or press **F1** for information about a panel or the tab that is displayed in a panel. The help topic is displayed in your browser.
- Choose **Help → Online Help** or press **CTRL+H (⌘H)** to open the default help topic in your browser.
- When help is displayed in your browser, use the navigation links or search the help in the side bar.
- Choose **Help → Manuals Index**, to open a PDF file that has links to all the PDF documents. Click a link to open the document.
- Choose **Help → Search Manuals** to search the manuals. The search tab in Adobe Reader opens, and you can search across all the PDF documents. You must have Adobe Reader installed to use this feature.

### For information on:

- Problems and solutions: choose Help → Knowledge Base or Help → Known Issues → *product*.
- Software updates: choose Maestro → Check for Updates.
- New software features: choose Help → New Features.
- Scripts available for download: choose Scripts → Update.
- Python scripting: choose Help → Python Module Overview.
- Utility programs: choose Help → About Utilities.
- Keyboard shortcuts: choose Help → Keyboard Shortcuts.
- Installation and licensing: see the *Installation Guide*.
- Running and managing jobs: see the *Job Control Guide*.
- Using Maestro: see the *Maestro User Manual*.
- Maestro commands: see the *Maestro Command Reference Manual*.

## Contacting Technical Support

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](mailto:help@schrodinger.com)

USPS: Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204

Phone: (503) 299-1150

Fax: (503) 299-4532

WWW: <http://www.schrodinger.com>

FTP: <ftp://ftp.schrodinger.com>

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

- All relevant user input and machine output
- Prime purchaser (company, research institution, or individual)
- Primary Prime user
- Installation, licensing, and machine information as described below.

## Gathering Information for Technical Support

This section describes how to gather the required machine, licensing, and installation information, and any other job-related or failure-related information, to send to technical support.

### For general enquiries or problems:

1. Open the Diagnostics panel.
  - **Maestro:** Help → Diagnostics
  - **Windows:** Start → All Programs → Schrodinger-2012 → Diagnostics
  - **Mac:** Applications → Schrodinger2012 → Diagnostics
  - **Command line:** \$SCHRODINGER/diagnostics
2. When the diagnostics have run, click Technical Support.
 

A dialog box opens, with instructions. You can highlight and copy the name of the file.
3. Attach the file specified in the dialog box to your e-mail message.

### If your job failed:

1. Open the Monitor panel in Maestro.
 

Use Applications → Monitor Jobs or Tasks → Monitor Jobs.
2. Select the failed job in the table, and click Postmortem.
 

The Postmortem panel opens.
3. If your data is not sensitive and you can send it, select Include structures and deselect Automatically obfuscate path names.
4. Click Create.
 

An archive file is created in your working directory, and an information dialog box with the name of the file opens. You can highlight and copy the name of the file.
5. Attach the file specified in the dialog box to your e-mail message.
6. Copy and paste any log messages from the window used to start Maestro (or the job) into the email message, or attach them as a file.
  - **Windows:** Right-click in the window and choose Select All, then press ENTER to copy the text.
  - **Mac:** Start the Console application (Applications → Utilities), filter on the application that you used to start the job (Maestro, BioLuminate, Elements), copy the text.

### If Maestro failed:

1. Open the Diagnostics panel.

- **Windows:** Start → All Programs → Schrodinger-2012 → Diagnostics
- **Mac:** Applications → Schrodinger2012 → Diagnostics
- **Linux/command line:** \$SCHRODINGER/diagnostics

2. When the diagnostics have run, click Technical Support.

A dialog box opens, with instructions. You can highlight and copy the name of the file.

3. Attach the file specified in the dialog box to your e-mail message.

4. Attach the file `maestro_error.txt` to your e-mail message.

This file should be in the following location:

- **Windows:** %LOCALAPPDATA%\Schrodinger\appcrash  
(Choose Start → Run and paste this location into the Open text box.)
- **Mac:** Documents/Schrodinger
- **Linux:** Maestro's working directory specified in the dialog box (the location is given in the terminal window).

5. On Windows, also attach the file `maestro.EXE.dmp`, which is in the same location as `maestro_error.txt`.

# Glossary

**alignment**—The optimal matching of residue positions between sequences, typically a query sequence and one or more template sequences.

**anchor**—A constraint on alignment set at a given residue position. Alignment changes must preserve the query-template pairing at that residue until the anchor is removed.

**ASL**—Atom Specification Language.

**cofactor**—The complement of an enzyme reaction, usually a metal ion or metal complex.

**Comparative Modeling**—Protein structure modeling based on a query-template match with a substantial percentage of identical residues (usually 50% or greater sequence identity).

**constraints**—Tools to keep regions of a sequence (alignment constraints) or structure (during minimization) in a particular configuration.

**deletions**—The residues missing from a query sequence that are present in a template sequence.

**Fold Recognition**—The use of secondary structure matching and profiles generated from multiple sequence/structure alignments to find templates when sequence methods are unsuccessful.

**gaps**—The spaces in an alignment resulting from insertions and deletions.

**HETATOMs**—The atoms of residues, including amino acids, that are not one of the standard 20 amino acids. In PDB files, HETATM.

**homolog**—A sequence/structure related to the query sequence; i.e., a sequence with many of the same residues in the same patterns as the query sequence. Usually these sequences are derived from the same family and may have similar function.

**insertions**—The extra residues found in a query sequence that are not found in a template sequence.

**ligand**—A small molecule that binds to a protein; e.g., an antigen, hormone, neurotransmitter, substrate, inhibitor, and so on.

**loop**—A region of undefined secondary structure.

**project**—A collection of related data, such as structures with their associated properties. In Prime a project comprises one or more *runs* (executions of the Prime workflow). The project may include data that does not appear in the *project table*.

**Project Table**—The Maestro panel associated with a project, featuring a table with rows of entries and columns of properties. In Prime, project table entries are usually model structures.

**query sequence**—A sequence of unknown structure or fold.

**refinement**—An improvement of a model structure through energy-based optimization of selected regions.

**run**—A single execution of the Prime workflow using a particular set of choices (of templates, of paths, and of settings). Each run belongs to a *project*. Runs cannot be saved without saving the project to which they belong.

**SSA**—Secondary structure assignment.

**SSP**—Secondary structure prediction.

**template sequence**—A sequence of known structure and fold used as a basis for building a model of the query.

**Workspace**—The open area in the center of the Maestro window in which structures are displayed.

**Z-Score**—Measures the compatibility of the query sequence with the model structure, relative to the compatibility of randomly shuffled sequences of the same composition.



## Symbols

- (hyphen)
  - for locked gap ..... 15, 37
  - for loop ..... 14, 34
- ~ (tilde), for unlocked gap ..... 15, 37

## A

- Add Anchors ..... 15, 37
- alignment ..... 141
  - editing ..... 15, 37
  - exporting ..... 32
  - GPCRs ..... 35
  - importing ..... 32
  - methods ..... 33
  - with ClustalW ..... 33
  - with Prime STA ..... 33
- alignment score ..... 35
  - interpretation of values ..... 71
- alpha carbons
  - aligning structures by ..... 122
  - constraining specific ..... 99
  - maximum movement ..... 62, 99, 100
- anchors ..... 141
  - alignment constraints ..... 15, 37
  - removing ..... 15, 37
  - setting ..... 15, 37
- ASL ..... 141
- atom names, correcting ..... 50
- ATOM records ..... 131

## B

- backbone sampling, in side-chain prediction... 60, 101
- Balloon Help ..... 9
- binding sites, aligning ..... 73
- bit score ..... 26
- BLAST
  - database format ..... 129
  - expectation value ..... 26
  - percentage gaps ..... 26
  - updating database ..... 124
  - web-based searching ..... 24
- BLAST/PSI-BLAST ..... 127
- blocks, sliding ..... 15, 37
- BLOSUM62 similarity matrix ..... 26
- bond orders, assigning ..... 49

- bonds
  - disulfide ..... 42
  - to metals ..... 48
- Build Structure step ..... 39

## C

- chain lengths ..... 22
- chain name ..... 26
- chains, multiple ..... 21, 107
- charged residues ..... 37
- code, sequence ..... 21
- cofactors ..... 42, 141
- Color by Sequence ..... 12
- Color Scheme ..... 11, 15
  - Prime Display Menu ..... 12
  - Prime toolbar ..... 12
- color schemes ..... 12–13
  - built model ..... 44
- columns, sorting ..... 9, 27
- Comparative Modeling ..... 141
- composite SSP ..... 36
- Compound (PDB COMPND) ..... 26
- consensus sequence ..... 25
- conserved residues ..... 25
- constraints ..... 141
  - alpha carbon ..... 64, 99
  - atom pair ..... 42, 63, 97
  - helical ..... 64
  - loop placement in membrane ..... 64
  - membrane ..... 99
  - proximity ..... 42
  - solvent exposure ..... 99
  - spatial ..... 64
- continuum solvation model ..... 45, 98
- conventions, document ..... ix
- cooperative refinement of loops ..... 56
- Covalent Docking panel ..... 78
- covalently bound ligands ..... 47
- crystal symmetry ..... 59
- C-terminal gaps ..... 15, 37

## D

- deleting runs ..... 11
- deleting SSP ..... 17, 34
- deletions ..... 47, 141
- dielectric, variable ..... 98

directory  
     installation ..... 4  
     Maestro working..... 4  
 Display menu (Prime)..... 11, 14  
 disulfide bond..... 42  
 domains ..... 23

**E**

E (for beta strand) ..... 14, 34  
 Edit Alignment step ..... 31  
 Edit menu (Prime)..... 14  
 Edit Query Sequence ..... 14  
 Edit SSP ..... 14, 15  
 editing alignment ..... 37  
 editing sequences ..... 11  
 EMBL ..... 21  
 energy minimization ..... 57  
 entries, Project Table..... 19  
 environment variable  
     SCHRODINGER ..... 4  
     SCHRODINGER\_PDB..... 5, 129  
     SCHRODINGER\_THIRDPARTY..... 129  
 environment variables ..... 129  
 error messages..... 23, 133, 135  
 Expect column ..... 26  
 expectation value..... 26  
 Export Sequences..... 11, 17  
 Export SSP..... 17, 33  
 extracting sequence names..... 22

**F**

FASTA  
     file format ..... 21, 33  
     file header ..... 22  
 file formats ..... 21  
 File menu ..... 8  
     Job Options ..... 18  
 Find Homologs step ..... 21, 23  
 Fold Recognition..... 141  
 Font Size, in Sequence Viewer ..... 13  
 force field  
     Build Structure ..... 45  
     energy minimization ..... 57  
     nonstandard residues ..... 48  
     refinement tasks ..... 51  
     single-point energy ..... 58

formal charges, correcting ..... 49

**G**

gaps ..... 141  
     C-terminal..... 15, 37  
     locking ..... 15, 37  
     N-terminal ..... 15, 37  
     percentage..... 26  
     in query ..... 26  
     in secondary structure..... 37, 135  
     in template ..... 26  
     unlocking ..... 15, 37  
 GENBANK ..... 21  
 getpdb utility ..... 28  
 GPCRs  
     aligning ..... 35  
     SSPs for ..... 33  
 Guide..... 8, 9, 14

**H**

H (for alpha helix) ..... 14, 34  
 header  
     FASTA file..... 22  
     PDB ..... 22  
 helices  
     building..... 64, 99  
     placing membrane on ..... 53  
 HETATMs ..... 141  
 HETATOMs ..... 141  
 heteromultimer  
     saving runs for ..... 30  
     template alignment ..... 42  
 hidden sequences ..... 25  
 HMMER/Pfam..... 24, 31, 127  
 homologs..... 141  
     importing ..... 23  
     searching for ..... 23  
 Homologs table ..... 26  
 host machine ..... 18  
 hydrogens, adding ..... 48  
 hydrophobic residues ..... 37  
 hyphen symbol  
     for locked gap..... 15, 37  
     for loop ..... 14, 34

**I**

ID, PDB chain .....	26
Identities column .....	26
Import SSP .....	33
importing structurally aligned templates .....	23
Include ligands and cofactors window .....	42
Input Sequence step .....	21
insertions .....	37, 47, 141

**J**

Job Monitor panel .....	19
job options .....	18
Job Options dialog box .....	11, 18
job status icon .....	9, 19
jobs	
killing .....	19
monitoring .....	19

**K**

killing jobs .....	19
--------------------	----

**L**

leaving group	
ligand .....	79
receptor .....	79
libraries	
backbone dihedral .....	45
side-chain rotamer .....	45
ligands .....	42, 141
covalently bound .....	47, 48, 77
lipophilic term .....	70
locking gaps .....	15, 37
loop refinement .....	47
alpha-carbon constraints .....	64
constraining to helix .....	64
cooperative .....	56
defining region .....	55
first residue .....	54
invalid features check .....	56
last residue .....	54
minimum interatomic distance .....	63
options .....	61
order of .....	56
restriction with respect to membrane ...	64, 99

solvent exposure constraints .....	99
time required .....	55
loops .....	141

**M**

maximum sequence length .....	23
membrane, implicit .....	52
adding to other structures .....	54
adjusting .....	53
constraining loops with respect to .....	64
excluding solvent pockets .....	54
from OPM database .....	53
thickness .....	53
using .....	51
menu bar (Prime) .....	8, 10
menus	
Display (Prime) .....	11, 14
Edit (Prime) .....	14
File (Prime) .....	8, 18
merging projects .....	7
metals, bonds to .....	48
minimization .....	57
missing residues .....	131
model-building method .....	44
Monitor panel .....	9
monitoring jobs .....	19
mouse functions .....	18
multimer	
input for .....	22
multiple chains	
with same sequence .....	21
with SKA .....	107
multiple sequences .....	22
multiple templates .....	40

**N**

names	
chains .....	26
extracting sequence .....	22
non-unique .....	22
sequences .....	22, 26
native Schrödinger format .....	33
New command .....	10
None color scheme .....	12
N-terminal gaps .....	15, 37

**O**

Open command .....	11
OPLS_2005 force field .....	45, 48, 57, 58
optimizing side chains .....	43
options	
job .....	11, 18
in panel layout .....	8

**P**

panel layout .....	8
panels	
Job Options .....	18
Monitor .....	9
PDB .....	127
ATOM records .....	131
chain ID .....	26
header .....	22
ID code .....	26
SEQRES .....	131
updating database .....	125
PDB atom names, correcting for receptor-ligand complex .....	50
Pfam .....	24, 31, 127
PIR .....	21
polar residues .....	37
position-specific substitutional matrix (PSSM) .....	36
Positives column .....	26
Preferences .....	14
command .....	13
Prime Guide .....	8, 14
Prime menu bar .....	8, 10
Prime MM-GBSA panel .....	84
Prime runs .....	7
Prime Sequence Viewer .....	8
Prime toolbar .....	8, 14
Prime workflow .....	9
Prime–Refinement .....	2, 47
Prime–Structure Prediction .....	1, 8, 21
processors, maximum number .....	62
product installation .....	138
programs, third-party .....	127
Project Table .....	19, 142
projects .....	142
Protein Structure Alignment panel .....	72
Protein Structure Alignment Results panel .....	73
proximity .....	13

Proximity color scheme .....	13
proximity cutoff .....	13, 14
PSIPRED .....	33, 128

**Q**

query sequence .....	142
editing .....	22
query, maximum length .....	23

**R**

random seed, for side-chain prediction .....	59
Ranking Score .....	142
read sequence .....	21
records	
ATOM .....	131
SEQRES .....	131
Refine Structure .....	19
Refinement .....	2
refinement .....	142
cooperative .....	56
protocols .....	47
Refinement panel .....	47
regions of multiple templates .....	40
Rename command .....	11
Res1 .....	54
Res2 .....	54
Residue Matching color scheme .....	12
residue names, correcting for receptor-ligand complex .....	50
Residue Property color scheme .....	12
Residue Type color scheme .....	12
residues	
charged .....	37
conserved .....	25
hydrophobic .....	37
incomplete .....	56
missing .....	131
nonstandard .....	48
polar .....	37
specification syntax .....	91
standard, list of .....	47
rotamers .....	43
ruler .....	16
runs .....	7, 142
deleting .....	11
for heteromultimer .....	30

new .....	10	including in loop refinement .....	62
opening .....	11	missing atoms .....	56
renaming .....	11	number of conformations returned .....	97
saving .....	11, 30	optimizing .....	43
		rotamers .....	43
<b>S</b>		similarity, BLOSUM62 matrix .....	26
sampling accuracy, loop refinement .....	61	Slide As Block .....	15, 37
Save As .....	11	Slide Freely .....	15, 37
saving runs .....	11	solvation .....	45
Schrödinger contact information .....	138	solvent pockets	
schrodinger.hosts file .....	18	excluding from membrane .....	54
score		sorting tables, by column heading .....	9, 27
alignment .....	35	SSA .....	142
BLAST bit .....	26	SSpro .....	33, 128
scratch projects .....	7	SSPs .....	142
Search program		command menu for .....	18
in Find Homologs .....	23	composite .....	36
secondary structure assignment .....	15, 36	deleting .....	17, 34
Secondary Structure color scheme .....	12	editing .....	14
secondary structure prediction programs .....	128	exporting .....	17, 33
<i>See also</i> SSPs		importing .....	33
Select toolbar command .....	14	retrieving unmodified .....	35
seqconvert utility .....	33, 121	in Sequence Viewer .....	15
SEQRES records .....	131	standard residues .....	47
sequence identity .....	26	Step View .....	8
sequence positives .....	26	steps	
Sequence Viewer (Prime) .....	8	action .....	8
sequences		available and unavailable .....	10
coloring .....	15	Build Structure .....	39
command menu for .....	15	changing data .....	8, 10
cropping .....	23	Edit Alignment .....	31
editing .....	11	Find Homologs .....	21, 23
editing query .....	22	Input Sequence .....	21
exporting .....	11	name .....	8
family consensus .....	25	panel layout .....	8
maximum length .....	23	structalign utility .....	119
multiple .....	22	structurally aligned templates .....	29, 42
names .....	22, 26	structure refinement, loop .....	47
non-unique .....	22	structures, coloring .....	15
Pfam .....	25	Surface Generalized Born (SGB) .....	45
selecting .....	22		
showing hidden .....	25	<b>T</b>	
standard code .....	21	tables .....	8
unusable .....	27	cells .....	27
side chains		sorting .....	9
backbone sampling when predicting ..	60, 101	in Step View .....	9
cooperative prediction .....	97	Template ID color scheme .....	13

template sequence ..... 142  
 templates  
   homomultimer ..... 41  
   multiple ..... 40, 41  
   regions ..... 40  
   structural alignment ..... 29, 42  
 third-party programs ..... 127, 128  
 tilde symbol, for unlocked gap ..... 15, 37  
 Title, PDB header ..... 26  
 toolbar, Prime ..... 8, 14  
 tooltips ..... 9, 27  
   for buttons ..... 14  
   for table entries ..... 9  
 truncated-Newton ..... 57

## U

unlocking gaps ..... 15, 37  
 unusable sequence ..... 27

utilities  
   getpdb ..... 28  
   seqconvert ..... 33, 121  
   structalign ..... 119

## V

View SSA ..... 15  
 View Structure ..... 13, 15

## W

warnings ..... 23, 27, 133–135  
 web-based BLAST searches ..... 24  
 workflow ..... 9, 21  
 Workspace ..... 11, 15, 142

## Z

Z-Score ..... 142



120 West 45th Street  
17th Floor  
New York, NY 10036

155 Gibbs St  
Suite 430  
Rockville, MD 20850-0353

Quatro House  
Frimley Road  
Camberley GU16 7ER  
United Kingdom

101 SW Main Street  
Suite 1300  
Portland, OR 97204

Dynamostraße 13  
D-68165 Mannheim  
Germany

8F Pacific Century Place  
1-11-1 Marunouchi  
Chiyoda-ku, Tokyo 100-6208  
Japan

245 First Street  
Riverview II, 18th Floor  
Cambridge, MA 02142

Zeppelinstraße 73  
D-81669 München  
Germany

No. 102, 4th Block  
3rd Main Road, 3rd Stage  
Sharada Colony  
Basaveshwaranagar  
Bangalore 560079, India

8910 University Center Lane  
Suite 270  
San Diego, CA 92122

Potsdamer Platz 11  
D-10785 Berlin  
Germany

**SCHRÖDINGER®**