

Phase 3.1

User Manual

Phase User Manual Copyright © 2009 Schrödinger, LLC. All rights reserved.

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

Canvas, CombiGlide, ConfGen, Epik, Glide, Impact, Jaguar, Liaison, LigPrep, Maestro, Phase, Prime, PrimeX, QikProp, QikFit, QikSim, QSite, SiteMap, Strike, and WaterMap are trademarks of Schrödinger, LLC. Schrödinger 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. 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 for Third-Party Software in your product installation at `$(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. 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.

June 2009

Contents

Document Conventions	xi
Chapter 1: Introduction to Phase.....	1
1.1 Phase Workflows	1
1.2 Running Schrödinger Software	2
1.3 Citing Phase in Publications	3
Chapter 2: Running Phase from Maestro.....	5
2.1 Developing a Pharmacophore Model	5
2.1.1 General Panel Layout.....	6
2.1.2 The Prepare Ligands Step	7
2.1.3 The Create Sites Step.....	8
2.1.4 The Find Common Pharmacophores Step	8
2.1.5 The Score Hypotheses Step	9
2.1.6 The Build QSAR Model Step	9
2.2 Building or Editing Hypotheses	10
2.3 Preparing a 3D Database for Searching	10
2.4 Finding Matches to a Hypothesis	10
2.5 Running Jobs.....	11
2.6 Preferences	12
Chapter 3: Preparing Ligands for Pharmacophore Model Development	13
3.1 Adding Ligands to a Run	14
3.2 Cleaning Up Ligand Structures	17
3.2.1 Generating Stereoisomers	18
3.2.2 Generating Ionization States.....	19
3.3 Generating Conformers.....	19
3.3.1 Output Options.....	20

3.3.2 Search Method, Sampling, and Minimization Options	20
3.3.2.1 ConfGen Search Method.....	21
3.3.2.2 Mixed MCMM/LMOD Search Method.....	22
3.3.3 MacroModel Options.....	23
3.4 The Ligands Table	24
3.5 Defining the Ligand Set for Model Development.....	25
3.6 Step Summary	26
Chapter 4: Creating Pharmacophore Sites	27
4.1 Viewing Pharmacophore Features	29
4.2 Editing Pharmacophore Features	30
4.2.1 Loading and Storing Feature Sets	31
4.2.2 Adding and Editing Custom Patterns	32
4.2.3 Choosing How Patterns Are Used	34
4.2.4 Viewing Patterns	35
4.2.5 Adding Custom Features	35
4.2.6 Using Projected Points.....	35
4.2.7 Surface Area Calculations for Hydrophobic Features	36
4.3 Defining the Ligand Set for Model Development.....	36
4.4 Creating the Sites	37
4.5 Step Summary	37
Chapter 5: Finding Common Pharmacophores.....	39
5.1 The Search Method	39
5.2 Defining the Scope of the Search.....	40
5.3 Modifying the Search Parameters	43
5.4 Starting the Search	44
5.5 Step Summary	45

Chapter 6: Scoring Hypotheses	47
6.1 The Scoring Process	48
6.2 Scoring the Hypotheses	50
6.2.1 Scoring Method and Filtering.....	50
6.2.2 Survival Score Weighting Factors	52
6.3 Scoring Inactives and Rescoring	53
6.4 Results of Scoring	55
6.5 Examining Hypotheses and Ligand Alignments	56
6.6 Clustering Hypotheses	58
6.7 Adding Excluded Volumes to Hypotheses	59
6.8 Step Summary	61
Chapter 7: Building QSAR Models.....	63
7.1 Phase QSAR Models	63
7.2 Choosing a Training Set and a Test Set	66
7.3 Specifying Options for the QSAR Model	66
7.4 QSAR Model Results	68
7.5 Viewing the QSAR Model	71
7.6 Continuing from the Build QSAR Model Step	74
7.7 Step Summary	75
Chapter 8: Building and Editing Hypotheses.....	77
8.1 The Manage Hypotheses Panel	77
8.2 Creating New Hypotheses	80
8.2.1 Ligand-Based Hypotheses.....	81
8.2.2 Freestyle Hypotheses	82
8.3 Editing Existing Hypotheses	84
8.3.1 Ligand-Based Hypotheses.....	84

8.3.2 Freestyle Hypotheses	85
Chapter 9: Building QSAR Models from Ligands.....	89
9.1 Adding Ligands	90
9.2 Choosing a Training Set and a Test Set	90
9.3 Building and Testing the Model.....	91
9.4 Examining and Using the Model.....	91
Chapter 10: Creating and Updating a 3D Database	93
10.1 Input Structures.....	93
10.2 Preparing the Structures	95
10.3 Filtering the Structures	96
10.4 Specifying the Database	97
10.5 Generating Conformers and Sites	97
Chapter 11: Finding Matches to Hypotheses.....	99
11.1 The Fitness Score	100
11.2 Setting up A Search	101
11.2.1 Selecting a Structure Source	101
11.2.2 Selecting a Hypothesis	103
11.2.3 Selecting the Source of Conformations	103
11.2.4 Setting the Search Mode and Criteria.....	104
11.2.5 Setting Site-Specific Matching Criteria	106
11.2.6 Setting Filtering and Scoring Options	107
11.3 Search Results.....	108
Chapter 12: Pharmacophore Model Development from the Command Line	111
12.1 Workflow Summary.....	111
12.2 Pharmacophore Model Development Utilities	113

12.3 Setting Up a Phase Pharmacophore Model Project	115
12.3.1 pharm_project.....	115
12.3.2 pharm_data.....	117
12.4 Creating Sites	119
12.4.1 pharm_create_sites	119
12.4.2 phase_feature	119
12.5 Finding Common Pharmacophores	121
12.5.1 pharm_find_common	121
12.5.2 phase_partition and phase_multiPartition.....	122
12.6 Scoring Hypotheses	123
12.6.1 pharm_score_actives.....	124
12.6.2 phase_scoring.....	126
12.6.3 pharm_score_inactives	127
12.6.4 phase_inactive	128
12.6.5 pharm_cluster_hypotheses.....	128
12.6.6 phase_hypoCluster	130
12.7 Building QSAR Models	131
12.7.1 pharm_build_qsar	131
12.7.2 phase_multiQsar.....	132
12.7.3 phase_qsar	134
12.7.4 phase_qsar_stats.....	134
12.7.5 qsarVis.....	135
12.8 Adding Excluded Volumes to a Hypothesis	136
12.8.1 create_xvolShell.....	136
12.8.2 create_xvolClash	137
12.8.3 create_xvolReceptor	138
12.9 Other Utilities	140
12.9.1 pharm_archive	140
12.9.2 pharm_align_mol	140
12.9.3 align_hypoPair	142
12.9.4 create_hypoConsensus	144

Chapter 13: Managing and Searching 3D Databases from the Command Line	147
13.1 Managing a 3D Database: phasedb_manage	148
13.2 Generating Conformers and Sites: phasedb_confsites	152
13.3 Searching for Matches in a Database: phasedb_findmatches and phase_dbsearch	154
13.3.1 Searching Database Subsets	159
13.3.2 Examining Search Results.....	159
13.3.3 Applying Feature-Based Cutoffs	160
13.3.4 Applying Excluded Volumes.....	160
13.3.5 Searching for Partial Matches.....	161
13.3.6 Applying Feature-Matching Rules.....	161
13.3.7 Checkpointing and Restarting Searches.....	162
13.4 Creating Database Subsets: phasedb_subset	162
13.5 Exporting Structures from Databases: phasedb_export	164
13.6 Extracting Properties from a Database: phasedb_props	165
13.7 Merging Databases: phasedb_merge	166
13.8 Converting a Database: phasedb_convert	167
13.9 Checking Database Integrity: phasedb_check	168
13.10 Database Backup and Recovery: phasedb_recovery	169
13.11 Compacting Database HDF5 Files: phasedb_compact	170
13.12 Other Utilities	170
13.12.1 phasedb_count_records	170
13.12.2 phasedb_split_records.....	171
13.12.3 phasedb_match_keys.....	171
13.12.4 phasedb_index.....	172
13.13 Running on Multiple Processors	172
13.14 Granting Access to a Database	173
13.15 Checking Job Progress and Completion	173

Chapter 14: Searching Files for Matches from the Command Line	175
14.1 Searching Files with <code>phase_fileSearch</code>	175
14.2 Searching Files with <code>phase_gridSearch</code>	176
Chapter 15: Searching for Molecules by Shape	181
15.1 Running Shape Searches from Maestro	182
15.2 Running Shape Searches from the Command Line	185
15.3 Creating Included Volumes for Shape Queries	189
15.3.1 <code>create_ivolShape</code>	190
15.3.2 <code>convert_ivolToMae</code>	192
Chapter 16: Detecting Multiple Binding Modes	193
Chapter 17: Receptor-Based Hypotheses	197
Appendix A: Phase QSAR Models	201
A.1 The Phase QSAR Methods	201
A.2 Phase Model Validation	204
A.3 Phase QSAR Statistics	205
A.3.1 Training Set and Model	205
A.3.2 Test Set Predictions	206
Appendix B: Phase Input Files	209
B.1 Master Data File	209
B.2 Phase Main Input File	213
B.3 Feature Definition File	218
B.4 Inactives Scoring Input File	220
B.5 Hypothesis Clustering Input File	222
B.6 Multiple QSAR Model Input File	223

B.7 QSAR Model Input File	226
B.8 Feature Frequencies File	228
B.9 Feature-Matching Tolerances File	229
B.10 Hypothesis-Specific Tolerances File	229
B.11 Site Mask File	230
B.12 Hypothesis Rules File	230
B.13 Database Search Input File	232
B.14 Maestro File Search Input File	235
Appendix C: Phase Utilities	239
C.1 combine_hits	239
C.2 combine_matches	239
C.3 convert_hypoDistToXYZ	240
C.4 convert_hypoFeatures	240
C.5 create_hypoSDFFile	241
C.6 create_hypoFiles	241
C.7 phase_volCalc	241
C.8 rmsdcalc	243
C.9 flex_align	244
Getting Help	247
Glossary	251
Index	253

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, and screen 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`.

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 to Phase

Phase is a versatile product for pharmacophore perception, structure alignment, activity prediction, and 3D database searching. Given a set of molecules with high affinity for a particular protein target, Phase uses fine-grained conformational sampling and a range of scoring techniques to identify common pharmacophore hypotheses, which convey characteristics of 3D chemical structures that are purported to be critical for binding. Each hypothesis is accompanied by a set of aligned conformations that suggest the relative manner in which the molecules are likely to bind.

A given hypothesis may be combined with known activity data to create a 3D QSAR model that identifies overall aspects of molecular structure that govern activity. This model may be used in conjunction with the hypothesis to mine a 3D database for molecules that are most likely to exhibit strong activity toward the target.

Phase provides support for lead discovery, SAR development, lead optimization and lead expansion. Phase may also be used as a source of molecular alignments for third-party 3D QSAR programs.

Phase is integrated into Maestro, the graphical user interface (GUI) for all Schrödinger products. An overview of the general capabilities of Maestro is given in the [Maestro Overview](#). For more detailed information on Maestro, see the Maestro online help or the [Maestro User Manual](#). An overview of the Phase interface is given in [Chapter 2](#).

Phase is fully supported on Linux and SGI Irix platforms. On Windows platforms you can set up jobs and submit them to UNIX hosts for execution, with the exception of database jobs.

For a tutorial introduction to Phase, see the [Phase Quick Start Guide](#). For installation instructions, see the [Installation Guide](#).

1.1 Phase Workflows

Phase consists of the following four workflows:

- Building a pharmacophore model (and an optional QSAR models) from a set of ligands
- Building a pharmacophore hypothesis from a single ligand (and editing it)
- Preparing a 3D database that includes pharmacophore information
- Searching the database for matches to a pharmacophore hypothesis

Each of these workflows is supported by a Maestro panel. The first workflow, building a pharmacophore model, involves the following steps:

- Preparing the ligands, including 2D-3D structure conversion and the generation of ligand conformations. This step is described in detail in [Chapter 3](#).
- Defining and identifying the pharmacophore sites in the ligands. This step is described in detail in [Chapter 4](#).
- Creating hypotheses by finding common pharmacophores. This step is described in detail in [Chapter 5](#).
- Scoring the hypotheses, and adding any excluded volumes to the hypotheses. This step is described in detail in [Chapter 6](#).
- Building and examining 3D QSAR models. This step is described in detail in [Chapter 7](#).

Building a pharmacophore hypothesis from one or more ligands manually is an alternative to building a pharmacophore model from a set of ligands by the automated process described above. Details of this task can be found in [Chapter 8](#).

Preparing the 3D database involves the following tasks, which are described in [Chapter 10](#):

- Preparing the molecules, including 2D-3D conversion
- Adding molecules to the database

Depending on how you want to use the database, you can perform the following tasks, which are also described in [Chapter 10](#):

- Generating conformations for each molecule
- Defining and identifying the pharmacophore sites
- Creating subsets

Searching the 3D database for matches to a hypothesis includes various filtering and scoring mechanisms. This workflow is described in [Chapter 11](#).

You can also run these three workflows from the command line, as described in [Chapter 12](#) and [Chapter 13](#). [Chapter 14](#) describes searching a file for matches, rather than a 3D database.

1.2 Running Schrödinger Software

To run any Schrödinger program on a UNIX platform, or start a Schrödinger job on a remote host from a UNIX 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:

cshtcsh: setenv SCHRODINGER *installation-directory*

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

Once you have set the SCHRODINGER environment variable, you can start Maestro 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. For more information on starting Maestro, including starting Maestro on a Windows platform, see [Section 2.1](#) of the *Maestro User Manual*.

1.3 Citing Phase in Publications

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

Phase, version 3.1, Schrödinger, LLC, New York, NY, 2009.

Running Phase from Maestro

Phase consists of the following workflows, each of which is supported by a Maestro panel:

- Building a pharmacophore model and an optional QSAR model
- Preparing a 3D database that includes pharmacophore information
- Building or editing pharmacophore hypotheses
- Searching the database for matches to a pharmacophore hypothesis

An overview of each of these workflows is given in the sections below, along with an overview of the supporting Maestro panel. The stages are described in detail in the following chapters.

The Maestro interface for the first workflow is wizard-like, and takes you through each step of the process. The interface is more flexible than a wizard, however, because it allows you to exit at any step and resume the process later at the same step-or a different step, provided that you have the data to support that step. Default options are provided that should give good results, but you can also choose from a range of other options to suit your purposes. The interface for the remaining three workflows are single panels.

To open any of the panels, Develop Pharmacophore Model, Manage 3D Database, Edit Hypotheses, or Find Matches to Hypotheses, choose the appropriate item from the Phase submenu of the Applications menu in the main window.

All Phase jobs can be started from Maestro. Many of these jobs generate a large amount of data, and most of them can be distributed across multiple processors. When you click a Start button or an action button that starts a job, a panel is displayed that allows you to set job control information.

Phase can also be run from the command line. For information on command-line use, see [Chapter 12](#), [Chapter 13](#), and [Chapter 14](#).

2.1 Developing a Pharmacophore Model

Developing a pharmacophore model from a set of active molecules is the main means in Phase of generating pharmacophore hypotheses, which are subsequently used in database searching.

In Phase, there are five steps in the process of developing a pharmacophore model: preparing the ligands, creating pharmacophore sites from a set of features, finding common pharmacophores, scoring the hypotheses, and building a QSAR model.

The process of developing a pharmacophore model is called a *run*. The data for each run is stored as a separate entity, which you can open from the File menu. When a new run is created, links are made to common data, to avoid duplication. The run is stored as part of a Maestro project.

2.1.1 General Panel Layout

The Develop Pharmacophore Model panel is wizard-like in design, with five steps. Each step occupies the center of the panel, and consists of a title with a brief description of the step at the top, a set of controls and tables for results in the center, and a Back and a Next button at the bottom. The features of each step are described in detail in the online help.

In addition to the step features, the panel contains a menu bar, a toolbar, and an octagon icon at the top; and a step guide (the Guide) at the bottom above the Close and Help buttons. These features are described below.

The File Menu

The File menu allows you to work with the runs that are available in the project.

New	Create a new run
Open	Open an existing run from the submenu. If there are more than 4 runs, choose More to open a dialog box and select a run.
Save As	Save the current run with a new name
Rename	Rename the current run
Delete Run	Delete the current run

The Display Menu and the Toolbar

The Display menu provides options for viewing hypotheses and related attributes in the Workspace. These options are also available as toolbar buttons, and are described below. The items are only available in Step 4 (Score Hypotheses) and Step 5 (Build QSAR Model).



Hypothesis Displays the selected hypothesis as a spatial arrangement of feature symbols. For a description of these symbols, see [Table 4.1 on page 30](#).



Hypothesis Labels Displays feature labels for the selected hypothesis.



Excluded Volumes Displays excluded volumes for the selected hypothesis.

**QSAR Model**

Displays the selected QSAR model for the hypothesis. Only available in the Build QSAR Model step.

**Site Measurements**

Opens the View Site Measurements panel, in which you can select the intersite distances and angles of the hypothesis for display in the Workspace.

The Step Menu

The Step menu contains an item to display or hide the Guide, and items for each of the steps. The current step is marked with a red diamond. If the Guide is displayed, it is marked with a red square. The items for the steps that are not available are dimmed. You can go to any available step by choosing the corresponding menu item.

The Job Status Button

When a job has been launched and is running, the job status button at the upper right of the panel turns green and the icon spins. When the job stops, the button turns pink and the icon stops spinning. It is replaced by an exclamation point if the job is incorporating, and returns to the original when incorporation has finished. To monitor the job using the Monitor panel, click the button. For more information about monitoring jobs, see the *Job Control Guide*.

The Guide

The Guide displays the steps in the model as a set of buttons linked by lines. The buttons for the steps that are not available are dimmed. The current step is highlighted with a white background. You can go to any available step by clicking its button in the Guide. The Guide can be displayed or hidden from the Step menu. If you go back to an earlier step and make changes, you are prompted to save the existing data and create a new run with the changed data.

To navigate the steps, you can click the Back and Next buttons, click the desired step in the Guide, or choose the desired step from the Step menu.

2.1.2 The Prepare Ligands Step

In this step, you add to the run the molecules that you want to use as the basis for the pharmacophore model and any other molecules that you want to use to build or test the QSAR model, and you select both active and inactive molecules for the set that is used for the pharmacophore model. If you include activity data with the ligands when you add them, you can select the active and the inactive sets of ligands either using cutoffs or manually.

To develop a pharmacophore model, you should ensure that you have all-atom 3D structures, and generate different conformations for each structure. If the structures need to be converted from 2D to 3D or otherwise need cleaning up, you can do so in this step. You can also convert the structures to the most probable ionization (protonation) state at a given pH, and generate different chiralities for the structures in this step. The molecules that you add are automatically grouped into conformer sets. You can choose to group stereoisomers in the same set or different sets. If you add only one conformer for a given molecule, you can generate the rest in this step.

Once you have added the molecules, cleaned up the structures, generated conformers, and selected the set of ligands, you can proceed to the next step.

2.1.3 The Create Sites Step

In this step, you use a set of chemical structure patterns to identify pharmacophore features in each ligand. Once a feature has been mapped to a specific location in a conformation, it is referred to as a *pharmacophore site*. The number of occurrences of each feature in each ligand is tabulated, and you can display the locations of the pharmacophore sites in the Workspace.

While the built-in set of features is adequate for many purposes, you might want to add new patterns to the built-in features, ignore patterns in the built-in features, or add custom features. You can add functional groups, defined as SMARTS patterns, to the definition of a feature. You can also designate functional groups that should be excluded from consideration as part of a feature, and you can choose to ignore functional groups.

2.1.4 The Find Common Pharmacophores Step

In this step, you perform a search for common pharmacophores among the set of high-affinity (active) ligands that you chose in the first step. The search spans one or more families of pharmacophores, known as *variants*. You can choose the number of site points in the pharmacophore, filter out variants that have too many or too few of a particular kind of feature, and select a set of variants from the filtered list. You can also set a lower limit on the number of ligands that must match a pharmacophore before it can be considered to be a hypothesis.

The search proceeds by enumerating all pharmacophores of a given variant and partitioning them into successively smaller high-dimensional boxes according to their intersite distances. Each n -point pharmacophore contains $n(n-1)/2$ unique intersite distances, so each box contains $n(n-1)/2$ dimensions. Pharmacophores that are clustered into the same box are considered to be equivalent and therefore common to the ligands from which they arise. The size of the box defines the tolerance on each intersite distance, and therefore how similar common pharmacophores must be. You can set parameters to control the minimum box size and you can exclude pharmacophores for which any intersite distance is below a certain threshold.

Boxes that contain pharmacophores from the minimum required number of ligands are said to *survive* the partitioning process. Each surviving box contains a set of common pharmacophores, one of which is ultimately singled out as a hypothesis.

Once all desired variants have been processed, you can continue to the scoring step.

2.1.5 The Score Hypotheses Step

In this step you apply a scoring function that identifies the best candidate hypothesis from each surviving box, and provides an overall ranking of all the hypotheses. You can finish at this point, or select hypotheses for the generation of QSAR models and continue to the next step, or select hypotheses and proceed to find matches to the hypotheses. You can also add to the hypothesis volumes that should not be occupied by atoms in any active molecule, known as *excluded volumes*.

The scoring algorithm includes contributions from the alignment of site points and vectors, volume overlap, selectivity, number of ligands matched, relative conformational energy, and activity. You can adjust these in the survival score, and you can create a custom score. You can also penalize hypotheses by scoring inactives and subtracting a multiple of this score from the survival score. The scores for each hypothesis are displayed in a table. You can select a hypothesis and view scores for the ligands that match the hypothesis, and the energy of the ligand relative to the lowest conformation.

If you have both active and inactive molecules that match a hypothesis, you can use their structures to define excluded volumes. Any region of space that is occupied by part of an inactive molecule and is not occupied by the active molecules is a good candidate for an excluded volume. The excluded volumes are used to filter out molecules in the database search that are likely to be inactive.

When you have selected one or more hypotheses, you can proceed to the next step.

2.1.6 The Build QSAR Model Step

In this step, you build QSAR models for the selected hypotheses using the activity data for molecules that match at least three points in the hypothesis. You can use molecules with any level of activity, including those that may be inactive due to steric clashes with the target receptor. The QSAR model partitions space into a grid of uniformly sized cubes, and characterizes each molecule by a set of binary-valued independent variables that encode the occupancy of these cubes by six atom classes or a set of pharmacophore feature types. Partial least-squares regression is applied to these variables to build a series of models with successively greater numbers of factors.

You can visualize the QSAR model in the Workspace, and analyze it by atom or feature class and ligand. This can be used to identify ligand features that contribute positively or negatively to the predicted activity.

When you have developed QSAR models, you can continue to the database search and use the QSAR models to predict activities for matches, or return to the previous step to select another set of hypotheses for QSAR model development.

2.2 Building or Editing Hypotheses

As an alternative to building a pharmacophore model, you can build pharmacophore hypotheses directly from known active molecules, in the Edit Hypotheses panel. In this task, Phase identifies the possible pharmacophore sites in the molecule you select, based on a set of pharmacophore feature definitions. You then select the features that you want to include in the hypothesis. No jobs are run in this workflow.

If you create a hypothesis from a known receptor or receptor-ligand complex, you can use the receptor to automatically generate excluded volumes. This task is performed in the Excluded Volume Receptor panel.

2.3 Preparing a 3D Database for Searching

The Manage 3D Database panel provides tools for preparing a structure database that can be searched for matches to a pharmacophore hypothesis. The database must contain all-atom 3D structures that are reasonable representations of the experimental structures. Preparing a database involves adding structures, cleaning the structures if necessary, generating conformers if necessary or desired, creating pharmacophore sites from selected features, and creating subsets of molecules for database searching as desired.

Databases are not connected to Maestro projects.

The Manage 3D Database panel is a single panel, with a menu bar and an octagon button at the top. When a job has been launched and is running, the gray octagon at the upper right of the panel turns green and spins. To monitor the job using the Monitor panel, click the octagon. For more information about monitoring jobs, see the *Job Control Guide*.

2.4 Finding Matches to a Hypothesis

The Find Matches to Hypothesis panel is a single panel, with four sections. In the top two sections, you specify the database to search and the hypothesis to use in the search. In the bottom two sections, you set parameters for the search and for the subsequent display of hits.

The search is performed in two steps: finding matches to the hypothesis, and fetching hits. The second step can be repeated with different processing options without repeating the first step. The processing options include adjusting the fitness score, by which hits are sorted, applying numerical cutoffs on the number of hits, applying excluded volumes to filter hits, and calculating activities using the QSAR model, if one was generated for the hypothesis.

2.5 Running Jobs

When you click an action button that is associated with a job or a Start button, the Start dialog box opens. In this dialog box, you can select the host on which you want to run the job, set the user name on that host, if it differs from the user name on the host on which you are running Maestro, and enter the number of processors to use for the job. The maximum number of processors available on the selected host is displayed in parentheses after the host name. For more information on this dialog box, see [Section 2.2](#) of the *Job Control Guide*.

Phase jobs run under the Schrödinger job control facility. This facility allows you to monitor the progress of jobs within Maestro, both local and remote. It also provides the list of hosts in the Start panel from which you make a selection when you start a job.

The list of hosts is read from the `schrodinger.hosts` file, which is installed in the `$$SCHRODINGER` directory. At installation time, this file should be set up to define the hosts on which Schrödinger software will be run. Instructions for setting up this file are given in the *Installation Guide*. You can copy this file to your home directory to customize it.

The time-consuming parts of Phase can be distributed across multiple processors. You can set up multiprocessor hosts in the `schrodinger.hosts` file, either as hosts on which you run jobs directly or as batch queues, with a specified number of processors. If you run a database search on a multiprocessor host, such as a cluster, the following requirements must be met:

- The database must be located in a directory that is uniformly accessible to all nodes of the cluster on which jobs will be run.
- If the file system where the database is stored is only accessible to the cluster, you must run Maestro on the manager node of the cluster to launch jobs.
- In the `$$SCHRODINGER/schrodinger.hosts` file, each parallel queue that is used for database jobs should have a `tmpdir` entry with a path that is accessible to all nodes. For details of setting up these entries, see the *Installation Guide* or the *Job Control Guide*.

2.6 Preferences

Some options that affect the use of Phase can be set as preferences or with Maestro commands. You can add these options to the `maestro.cmd` file in your user resources directory. See [Chapter 12](#) of the *Maestro User Manual* for more information about this file and the user resources directory.

- **Default custom feature definition file**—To define a default feature definition file other than that in the software distribution, you can set a Maestro preference. The command for setting this preference is as follows:

```
prefer phasedefaultfeaturedefinitions=filename
```

The file name can be either an absolute path or a relative path. If you specify a relative path, Maestro will look for the file at this location relative to its current working directory. You can enter this command into the Commands text box in the main window, and it will be added to your preferences when Maestro exits. Alternatively, you can add this command to `maestro.cmd` in your user resources directory.

- **Color of Phase labels in the Workspace**—You can set the color of Phase labels with a Maestro command, `phasemarkersettings`. This command controls all aspects of Phase markers, including feature colors and sizes, label colors, and so on. For Phase labels, you can use the following command:

```
phasemarkersettings labelred=r labelgreen=g labelblue=b
```

where *r*, *g*, and *b* are the fractions of the red, green, and blue components of the label color, expressed as a real number between 0 and 1. For more information on this command, see the *Maestro Command Reference Manual*. This command is not written as a preference, and must be added to `maestro.cmd`.

Preparing Ligands for Pharmacophore Model Development

The first step in developing a pharmacophore model is to select the molecules that you want to use and to prepare them for use. This step is performed in the Prepare Ligands step of the Develop Pharmacophore Model panel.

The molecules you select should at a minimum include highly active molecules that you want to use as the basis of the pharmacophore model. You can also include inactive or moderately active molecules, which can be used to test pharmacophore hypotheses for specificity, for building and testing a QSAR model, and for the purpose of defining excluded volumes. When you add molecules, you can select an associated activity property to use for activity scoring and for the dependent variable in the QSAR model. This property can also be used to define the active and inactive molecules to use as the basis for the pharmacophore model.

Developing a pharmacophore model requires all-atom 3D structures that are realistic representations of the experimental molecular structure. Most ligands are flexible, so it is important to consider a range of conformations in order to increase the chances of finding something close to the bound structure.

Under some circumstances it can be important to generate variations on the input structures, such as varying the chirality or choosing the most probable protonation state. Varying the chirality of atoms in the molecules can be important if the chiralities are not known. The process of identifying common pharmacophores can then sample the different stereoisomers and locate the one that matches best. Also, if the input structure is not in the most common form at physiological pH values, the identification of common pharmacophores might give incorrect results, because the active form is protonated or deprotonated.

In the Prepare Ligands step, you add the molecules with their activity values to the Phase run. If you have 2D structures or united-atom 3D structures, you can convert them to all-atom 3D structures and locate the minimum energy structure using molecular mechanics. In the process, you can vary the chirality of atoms in the structures and assign the protonation state. Once you have the structures and any variations, you can generate the low-energy conformers for each structure. The conformers are automatically grouped into sets for each molecule. If you already have all-atom, 3D structures with their conformations, you must still pass them through the ligand preparation steps, so that Phase can verify that there are no unusable structures and that the geometry of the structures is sufficiently accurate.

The tasks involved in this step and how to accomplish them are described in detail below.

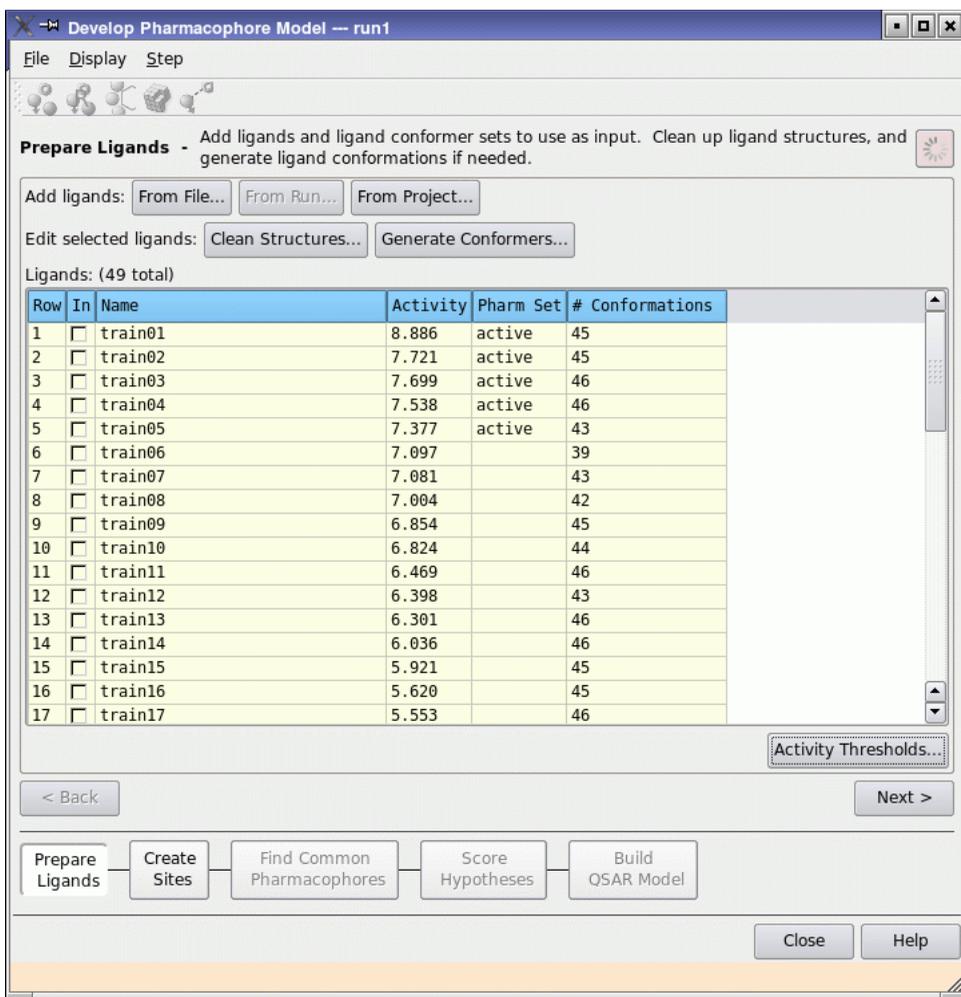


Figure 3.1. The Prepare Ligands step.

3.1 Adding Ligands to a Run

Each “ligand” in Phase is actually a set of conformations of a ligand structure. When you add ligands, they are automatically grouped into conformer sets. Each stereoisomer is considered a separate ligand by default, because the activities of stereoisomers are usually quite different. You can merge or separate ligand conformer sets by stereoisomer using the Ligands table shortcut menu (see Table 3.2). If you add ligands from a previous run, the sets are preserved with all their associated data. If you do not have conformations for the ligands you add, you can generate them in this step.

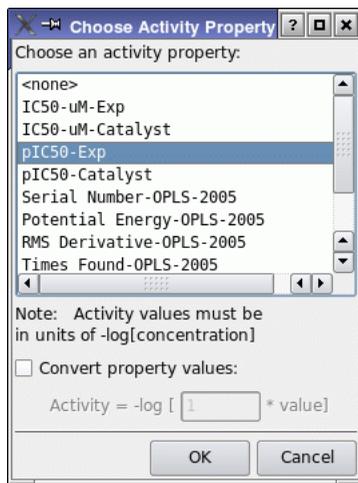


Figure 3.2. The Choose Activity Property dialog box.

You can add ligands to a Phase run from a file, from a previous run, or from a Maestro project. To add ligands, click one of the Add Ligands buttons. Each button opens a dialog box in which you can read or copy the ligands. Activity data can be added with the ligands. Once you have added the ligands, they are displayed in the Ligands table. If the ligands do not have activity data, you can add the data by editing the table cells.

If you want to delete ligands, select them in the table, then right-click in the table and choose Delete from the contextual menu.

Adding Ligands From a File

You can read ligands directly from a file into the Phase run, without importing them into Maestro. To do so, click From File. A file selector opens, so that you can navigate to and select one or more files. You can filter the list of files displayed by choosing Custom File Extension from the Files of type option menu. Only Maestro format is supported, and properties are read with the structures. When you click Open, the Choose Activity Property dialog box opens. This dialog box contains a list of properties, from which you can choose a single property for the activity of the ligands, and convert the activity to a logarithmic scale if necessary. The activity must be a positive quantity that increases with increasing activity.

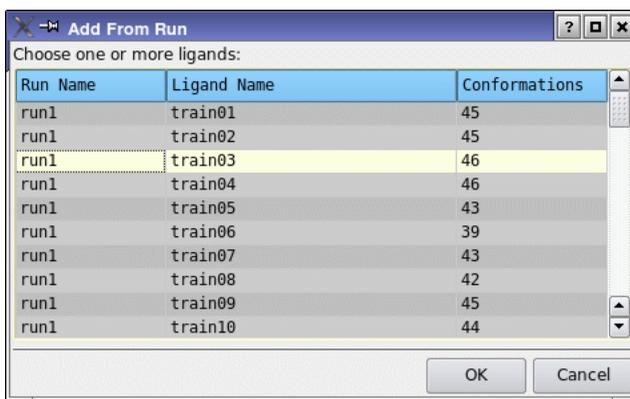


Figure 3.3. The Add From Run dialog box.

Adding Ligands From Another Phase Run

To copy ligands from another Phase run in the current project, click From Run. The Add From Run dialog box is displayed. The dialog box contains a list of all ligands available from all other runs in the current project, with the run name, the ligand name, and the number of conformers. You can choose multiple ligands to add to the current run. The activity values and the membership of the active set are extracted and added with the ligand. If a ligand was used in more than one run, the list of ligands will contain duplicates. If you select duplicates, only one is added, with the activity data from the first run chosen.

Adding Ligands From the Project

If you already have ligands in the Maestro project that you want to use, you can copy them from the project into the Phase run. To do so, click From Project. The Add From Project dialog box is displayed. This dialog box contains two lists: a list of entries, and a list of properties.

You can choose multiple entries to be added to the Ligands table (using shift-click and control-click). The set of entries that is displayed in the entry list in the dialog box is determined by the choice made from the Choose entry from option menu: all entries, selected entries, or included entries. You can sort the list by clicking one of the column headings, or by clicking Sort by Project Table Order. If you want to display some other property in the list, such as an activity property, click Show Property and choose the property from the list in the dialog box that is displayed. You can then sort the entries by the values of this property to aid in your ligand selection. The ligands are copied into the run, so that any changes made by Phase have no effect on the original ligands in the Project Table.

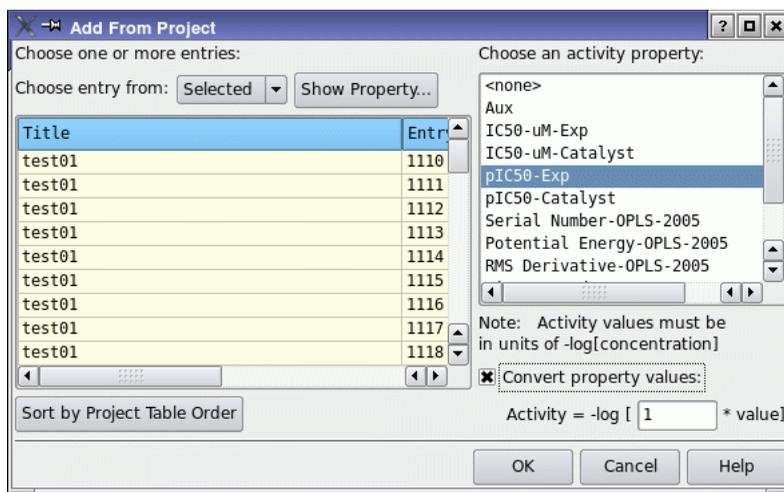


Figure 3.4. The Add From Project dialog box.

You can choose a single property for the activity of the ligands. The ligand activity must be a property that has units of $-\log_{10}[x]$, where x is the ligand. If the property is in units of concentration $[x]$, you can scale the property values and convert them to a logarithmic scale in this dialog box. The converted values are copied to the Ligands table.

3.2 Cleaning Up Ligand Structures

If the ligand structures are two-dimensional, lack hydrogen atoms, or include counter ions or solvent molecules, you must clean them up before proceeding. If the structures do not have the desired chirality or ionization (protonation) state, or if you want structures with different chirality, you can use the Clean Structures facility to generate them. Clean Structures is an interface to LigPrep with a range of options that is most appropriate for Phase. For more detailed information about the process, see the *LigPrep User Manual*.

In the cleanup process, the following actions are performed as necessary or as requested:

- Convert structures from 2D to 3D
- Add hydrogen atoms to ensure that the structure is an all-atom structure
- Remove counter ions and water molecules
- Add or remove protons to produce the most probable ionization state at the target pH
- Generate stereoisomers
- Remove noncompliant structures
- Perform an energy minimization

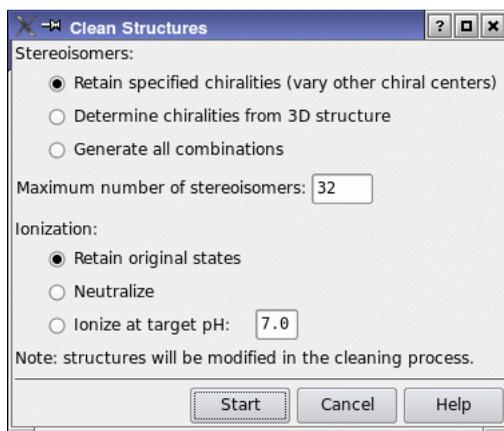


Figure 3.5. The Clean Structures dialog box.

The cleanup process is applied to the ligands that are selected in the Ligands table. You can therefore perform the cleanup with different options for different sets of molecules by making different selections. To clean up the selected structures, click Clean Structures. The Clean Structures dialog box is displayed. In this dialog box, you can set options for generation of stereoisomers and ionization states, then click Start to run the job to perform the cleanup. When you click Start, a dialog box is displayed, in which you can choose the host to run the job. You can distribute this job over multiple hosts.

3.2.1 Generating Stereoisomers

There are three options for generating stereoisomers, described below. For each option, any unspecified chiralities are varied, up to the number given in the Maximum number of stereoisomers text box. When you vary the stereochemistry, the process starts at the configuration with all chiral atoms to be varied set to R, and systematically varies the configuration. If you select fewer stereoisomers than the maximum, there is a chance that you might not generate the most important stereoisomers.

Retain specified chiralities (vary other chiral centers)

If the ligand has chirality information, this information is retained and used to ensure that the chiral atoms all have the correct chiralities. Chirality information includes parities and bond directions from SD files and the chirality property from Maestro files. If the configuration or chirality of one or more chiral centers is not specified, the chiralities for these centers is varied.

Determine chiralities from 3D structures

This option discards any information from the input file and determines the chirality from the 3D geometry. These chiralities are held fixed. For centers whose chirality is indeterminate, the two possible chiralities are generated.

Generate all combinations

This option discards chirality information and generates all possible configurations that result from the combination of chiralities on each chiral center.

3.2.2 Generating Ionization States

In the Ionization section you can choose from three options for generating the appropriate ionization state:

Retain original states

This option bypasses the generation of ionization states. If the ligands all have the correct ionization state for acidic and basic groups, choose this option.

Neutralize

This option converts all acidic and basic groups into their neutral form. For example, zwitterion groups are converted from a carboxylate and an ammonium to a carboxylic acid and an amine.

Ionize at target pH

This option generates the most probable ionization state at the given target pH, for which the default value is 7.

The ionization is performed with the `ionizer`. If you want to use Epik to generate ionization states (or tautomers), you must do so before you add the structures to the Phase run.

3.3 Generating Conformers

Once you have a set of cleaned-up ligands, you can run a conformational search to generate a set of conformers for each ligand. If you already have the conformations you need, you can skip this step.

To set up parameters for the conformational search, click **Generate Conformers**. The **Generate Conformers** dialog box is displayed. The dialog box has options for the search mode and solvation treatment, and allows you to limit the number of conformations generated, either to a specific number or by energy, which is evaluated in aqueous solution with a continuum solva-

tion model. After setting options, click **Start** to run the conformational search job. A dialog box is displayed, in which you can choose the host to run the job. You can distribute this job over multiple processors. When the job finishes, the **Ligands** table displays the number of conformers generated for each ligand.

Some options have a greater impact than others on the outcome of pharmacophore model development. Options with the greatest impact include the maximum number of conformations, maximum relative energy difference, minimum atomic deviation and number of post-minimization iterations. The default settings—rapid search, distance-dependent dielectric solvation model, and no post-minimization iterations—are likely to be adequate for many purposes. However, for consistency, you should use the same options in the pharmacophore model development as you use in the database search.

The options for controlling the conformational search are described below.

3.3.1 Output Options

Current conformers

When you generate conformers, you can discard the existing conformer set or you can keep it. If you keep the existing set, the new conformations are appended to the set. The set might therefore contain redundant conformers.

Maximum number of conformers

This value limits the number of conformers returned from the generation process. If the number of conformers generated is higher than this value, a sample of all the conformers generated is returned.

3.3.2 Search Method, Sampling, and Minimization Options

Conformer generation can be performed with one of two search methods, **ConfGen** or **Mixed MCMMLMOD**. For each method, sampling of conformational space can be done in **Rapid** or **Thorough** mode. Experience to date suggests that the final pharmacophore model is not usually significantly improved by a thorough search.

During the search, hydrogen-bonding interactions are suppressed, because conformations in which the ligand bonds to the receptor are needed in the model, not just conformations with internal hydrogen bonding.

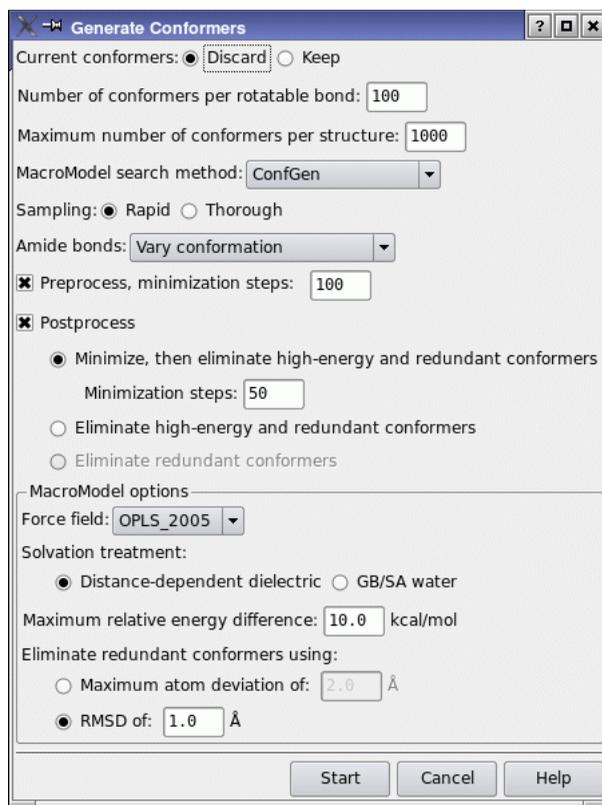


Figure 3.6. The Generate Conformers dialog box.

3.3.2.1 ConfGen Search Method

In a ConfGen search, the molecule is divided into a core and a periphery. The peripheral groups have only one rotatable bond between the terminal groups and the rest of the molecule. All the nonperipheral rotatable bonds are assigned to the core. The conformational search generates all core configurations and then varies the peripheral configurations, either one-by-one or in a complete search. The Sampling options have the following meanings:

- **Rapid**—All the core conformations are generated, and the conformations of the peripheral (rotamer) groups are sampled one by one.
- **Thorough**—A complete set of conformations is generated for both the core and the peripheral groups.

When this option is selected, the Amide bond option menu is displayed below the MacroModel search method option menu. There are three options for treatment of amide bonds in the

search. You can retain the original amide bond conformation in the input structure, you can set the conformation to trans, or you can vary the conformation. Varying the conformation allows the amide dihedral angle to take any value, not just cis or trans.

The ConfGen search produces structures rapidly, and the conformations generated might not be optimal. For this search method, options for preprocessing and post processing are provided. Both the Preprocess and the Postprocess options perform MacroModel tasks, including minimization, using the options in the MacroModel options section. Preprocessing is done on the input structure, and is confined to minimization with a specified number of iterations. Postprocessing is done on the set of conformers generated by the conformational search, and has three options:

- Minimize, then eliminate high-energy and redundant conformers—Minimize the conformers after eliminating high-energy and redundant conformers. This option has a Steps text box, in which you can enter the number of minimization steps. This is the most expensive option.
- Eliminate high-energy and redundant conformers—Eliminate redundant conformers, evaluate the MacroModel energy and eliminate any conformers that exceed the energy threshold given in the MacroModel options section.
- Eliminate redundant conformers—Perform only the redundant conformer elimination step. This is the least expensive option, as it does not require any energy evaluation, and is only available when preprocessing is turned off.

The minimization and the energy calculation are done with MacroModel using the selected force field.

If you do not minimize the energy, the generation of conformers runs much faster. Many of the conformers are rejected because their energy is too high, so the number of conformers is usually smaller than if you do the energy minimization.

3.3.2.2 Mixed MCMM/LMOD Search Method

The alternative search method is a combined Monte-Carlo Multiple Minimum/Low Mode (MCMM/LMOD) search, and is more accurate than the ligand torsional sampling method, but as a consequence takes longer. The difference between Rapid and Thorough sampling is in the number of steps taken per rotatable bond, which is much larger for thorough sampling. There is no need for the amide bond sampling options with this method. Minimization of the conformers generated by a mixed MCMM/LMOD search is recommended. You can specify the number of steps taken in the minimization of each conformer. The minimization is applied to the input structure, which for MCMM/LMOD is treated as the first conformer in the set. For more information on this method, see [Chapter 9](#) of the *MacroModel User Manual*.

3.3.3 MacroModel Options

In this section, you can select the force field and solvation treatment, and set thresholds to limit the number of conformations generated and determine when two conformers are considered to be identical.

Force Field

The default force field is OPLS_2005, but you can also select MMFFs. For details on these force fields, see [Section 2.1](#) of the *MacroModel User Manual*.

Solvation treatment

Two continuum solvation treatments for water are provided.

- Distance-dependent dielectric
- GB/SA water

The distance-dependent dielectric model is somewhat faster than the GB/SA model, and usually produces similar results.

Maximum relative energy difference

This value sets an energy threshold relative to the lowest-energy conformer. Conformers that are higher in energy than this threshold are discarded. The energy is evaluated with MacroModel using the selected force field.

Eliminate redundant conformers using

Two cutoff criteria are available for eliminating redundant conformers:

- Maximum atom deviation of $N \text{ \AA}$ —All distances between pairs of corresponding heavy atoms must be below this cutoff for two conformers to be considered identical.
- RMSD of $N \text{ \AA}$ —The root-mean-square deviation of all pairs of corresponding heavy atoms must be below this cutoff for two conformers to be considered identical.

The cutoff is only applied after the energy difference threshold, and only if the two conformers are within 1 kcal/mol of each other. In addition to the cutoff above, a threshold of 60° is used for torsion angle differences for polar hydrogens. This threshold cannot be changed.

3.4 The Ligands Table

The Ligands table lists the ligands that you added, grouped into conformer sets. You can select table rows in the usual way with click, shift-click and control-click. You can sort the columns by clicking the column header, and you can resize the columns by dragging the column boundary. The table columns are described in [Table 3.1](#).

Table 3.1. Description of the Ligand table columns.

Column	Description
In	Inclusion status of the ligand. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. The molecule that is displayed is the first conformer of the set. To view other conformers, you must export them to the Project Table (right-click menu). This column functions like the In column of the Project Table: click in the diamond to include a ligand and exclude all others, control-click to include or exclude a ligand without affecting the inclusion of the others, and shift-click to include a range of ligands. The included ligands are added as a scratch entry to the Workspace. Inclusion and exclusion of ligands has no effect on the entries in the Project Table.
Name	The name of the ligand. The default name is taken from the Title property of the ligand, if you added it from a project or from a file in which the title is defined. Otherwise a name is created for the ligand. You can edit the name by clicking in the cell, changing the text, then pressing ENTER. The name does not have to be unique.
Activity	Contains the value of the activity you selected when you added the ligands. If you did not select an activity, the table cells are empty. You can edit the activity by clicking in the cell, changing the text, then pressing ENTER.
Pharm Set	Indicates whether a ligand is in the set of active molecules or the set of inactive molecules that will be used to develop the pharmacophore model (the “pharm set”), or is ignored. For these three states the column contains the text <i>active</i> or <i>inactive</i> , or is blank. You can cycle through these states by clicking the table cell. To cycle through the states for all selected rows, control-click any of the selected cells.
# Conformations	The number of conformations stored for the ligand. You will normally want to generate multiple conformers for each ligand, unless, for example, you are developing a pharmacophore model from x-ray structures.

If you right-click in the table, a shortcut menu is displayed, from which you can select an action to be performed on the selected ligands. The actions are described in [Table 3.2](#).

Table 3.2. Ligands table shortcut menu items

Item	Description
Export Table Data	Export the data in the table to a CSV file or an HTML file as a table.
Merge Stereoisomers	Merge the conformers from stereoisomers into a single conformer set for each structure, for the selected ligands.
Separate Stereoisomers	Separate the conformers for different stereoisomers into separate sets, for the selected ligands.
Add Conformers to Project Table	Add the selected ligands to the Project Table. The structures for each ligand are placed in a separate entry group for the ligand if the ligand has multiple conformers.
Export Conformers to File	Export the selected ligands to a file. Opens a file selector, in which you can navigate to the location and name the file.
Select All	Select all ligands
Invert Selection	Invert the selection of ligands: selected ligands are deselected, and unselected ligands are selected.
Delete	Delete the selected ligands from the table.

When you display a ligand in the Workspace, you can also display information about the ligand in the Workspace. You can select the information that is displayed from the Workspace tab of the Preferences panel, by clicking Feedback and making the selection. See [page 231](#) of the *Maestro User Manual* for more information.

3.5 Defining the Ligand Set for Model Development

There are two ways in which you can define the set of ligands (the “pharm set”) that will be used for model development: by setting a threshold, and by manual selection. The ligand set must include some active ligands, and can also include inactive ligands. The ligands marked as active in the Pharm Set column of the Ligands table will be used to develop the model.

To set thresholds for active and inactive ligands, click Activity Thresholds. In the Activity Thresholds dialog box, you can set a threshold for the active ligands and a threshold for the inactive ligands. Ligands with activity greater than or equal to the active threshold are marked as active and included in the pharm set. Ligands with activity less than the inactive threshold are marked as inactive and included in the pharm set. Ligands whose activity lies between the thresholds are excluded from the pharm set.

To add ligands manually to the pharm set, select the ligands (using click, shift-click, or control-click), then control-click the Pharm Set column of the Ligands table. This action changes the status of all selected ligands; a click or a shift-click changes the status of a single ligand.

Note that it is not always necessary to assign every active molecule to the pharm set. If you have groups of highly similar ligands with nearly the same level of activity, you may want to select only one or two ligands from each group. You might also want to reserve some active ligands to test QSAR models.

3.6 Step Summary

To prepare the ligands for pharmacophore model development, follow the steps below.

1. Import the ligands into the Phase run, by clicking From File, From Run, or From Project.
2. Separate stereoisomers if necessary by selecting the relevant ligands in the table and choosing Separate stereoisomers from the shortcut (right-click) menu.
3. If you want to build a QSAR model or perform activity scoring, enter activity data for the ligands if it is not already present.
4. Clean up the ligand structures and generate variations on stereochemistry or ionization state by clicking Clean Ligands.
5. Generate sets of conformers for each ligand by clicking Generate Conformers.
6. Define the pharm set, either by setting the activity thresholds (click Activity Threshold), or by selecting ligands in the Ligands table.
7. Click Next to proceed to the next step.

Creating Pharmacophore Sites

The second step in developing a pharmacophore model is to use a set of pharmacophore features to create pharmacophore sites (site points) for all the ligands. This step is performed in the Create Sites step of the Develop Pharmacophore Model panel.

Phase supplies a built-in set of six pharmacophore features:

- Hydrogen bond acceptor (A)
- Hydrogen bond donor (D)
- Hydrophobic group (H)
- Negatively charged group (N)
- Positively charged group (P)
- Aromatic ring (R)

Each pharmacophore feature is defined by a set of chemical structure patterns. All user-defined patterns are specified as SMARTS queries and assigned one of three possible geometries, which define physical characteristics of the site:

- Point—the site is located on a single atom in the SMARTS query.
- Vector—the site is located on a single atom in the SMARTS query, and it will be assigned directionality according to one or more vectors originating from the atom.
- Group—the site is located at the centroid of a group of atoms in the SMARTS query. For aromatic rings, the site is assigned directionality defined by a vector that is normal to the plane of the ring.

Before proceeding, it is important to point out the difference between a *vector feature* and *vector geometry*. “Vector feature” is a more general term that refers to any pharmacophore feature that contains directionality. This includes hydrogen-bond acceptors, hydrogen-bond donors and aromatic rings. “Vector geometry” is more specific, and refers to the particular types of directionality associated with hydrogen-bond acceptors and donors. Thus vector geometry implies vector feature, but vector feature does not necessarily imply vector geometry.

While the built-in feature definitions are adequate for many purposes, you may find it necessary to expand them to include new patterns. For example, the presence of electron-withdrawing groups may cause an otherwise non-acidic hydrogen to be significantly dissociated at pH 7. If the built-in negative ionic definitions do not cover this case, then you may want to supplement the definitions with the appropriate SMARTS pattern.

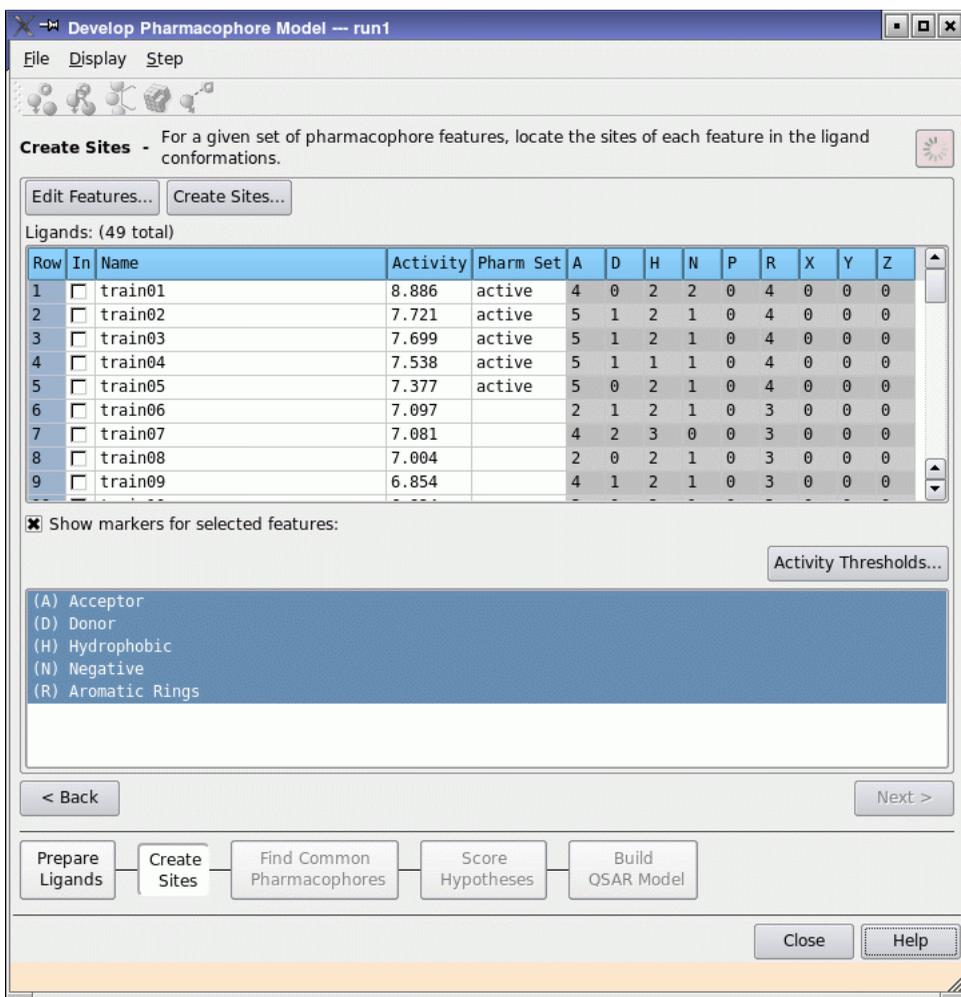


Figure 4.1. The Create Sites step.

In some cases, you may feel that a particular built-in definition should not be used, so you can choose to ignore it. Or there may be instances where a built-in definition matches a functionality that you feel does not qualify. In that case you can add a pattern to exclude the functionality in question.

You may also wish to add your own custom features types (X, Y, Z) to account for chemical functionalities that are not covered by the built-in feature types (A, D, H, N, P, R), or to lend special significance to a particular type of pharmacophoric element. If, for example, you know that all actives must contain a piperidine ring, then you could define a custom feature X with a

corresponding SMARTS pattern to match piperidine. Or, perhaps you want to force the pharmacophore model to map C=O acceptors only to other C=O acceptors. This could be achieved by creating a custom acceptor feature Y that contains only the SMARTS pattern for C=O.

The pharmacophore features can be previewed in the Workspace for any ligand. This allows you to verify that the definitions of the features are what you expect, before proceeding to generate site points for the entire set of ligand conformations.

4.1 Viewing Pharmacophore Features

Before you change the definitions of pharmacophore features, or submit the job to create site points using the pharmacophore features, you might want to view the features for each of the ligands. In this way you can check that the features are correctly identified.

Displaying features requires the creation of site points for one conformation of each ligand. This is done automatically when you enter the Create Sites step. If you change the feature definitions, you can create these site points and view the features by clicking Preview. In either case, a job is run locally to create the sites for the first conformer of each ligand. When the job is done (it should be quick), the feature counts are entered in the columns of the Ligands table, and you can display the features in the Workspace.

The first four columns of the Ligands table are the same as in the Prepare Ligands step, and have the same behavior; the selection behavior of the rows is the same, and the right-click menu is the same—see [Section 3.4 on page 24](#) for a description. In place of the Conformations column is a series of columns, one for each pharmacophore feature. These columns are populated with feature counts (the number of times a feature is present in a ligand).

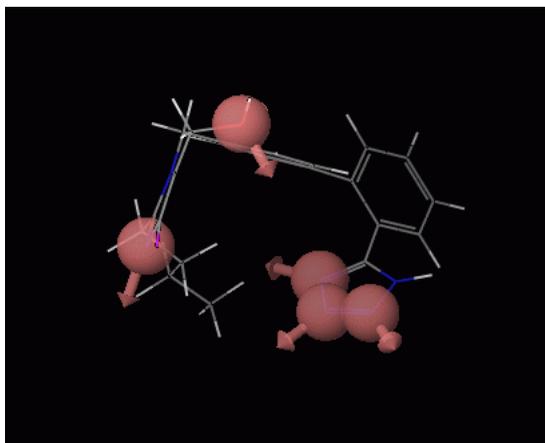


Figure 4.2. Pharmacophore features.

Table 4.1. Visual appearance of pharmacophore features in the Workspace.

Feature	Appearance
Acceptor (A)	Light red sphere centered on the atom with the lone pair, with arrows pointing in the direction of the lone pairs
Donor (D)	Light blue sphere centered on the H atom, with an arrow pointing in the direction of the potential H-bond
Hydrophobic (H)	Green sphere
Negative (N)	Red sphere
Positive (P)	Blue sphere
Aromatic Ring (R)	Orange torus in the plane of the ring
Custom	Colored sphere, with a unique color. Sphere includes arrows if the feature is a vector feature.

To display a ligand and its pharmacophore features in the Workspace, click the **In** column of the Ligands table for the ligand, select **Show markers for selected features**, and choose the feature types from the list below this option. You can select multiple features from the list. The appearance of the features is described in [Table 4.1](#). To view features for a different ligand, include it in the Workspace using the **In** column of the Ligands table.

4.2 Editing Pharmacophore Features

If you want to supplement the built-in features, create custom features, or load features from another location, you can do so in the **Edit Features** dialog box. Features are defined in terms of SMARTS patterns. You can add patterns to both standard features and up to three custom features. You can edit and delete custom patterns, and you can exclude or ignore both standard and custom patterns in a feature.

To open the **Edit Features** dialog box, click **Edit Features**.

The **Edit Features** dialog box, displayed in [Figure 4.3](#), contains a section for loading and storing feature sets, described in the next section, and a section for defining features. The feature definition section has controls for selecting, adding and deleting features, a table listing the SMARTS patterns that define the feature, and controls for adding, deleting, and moving patterns. The **Pattern** list table lists all the patterns that are used to define the pharmacophore feature. You can only select one row at a time in the table, and the text fields are not editable, with the exception of the **Distance** column, which you can edit to set individual distances to projected points. The table columns are described in [Table 4.2](#).

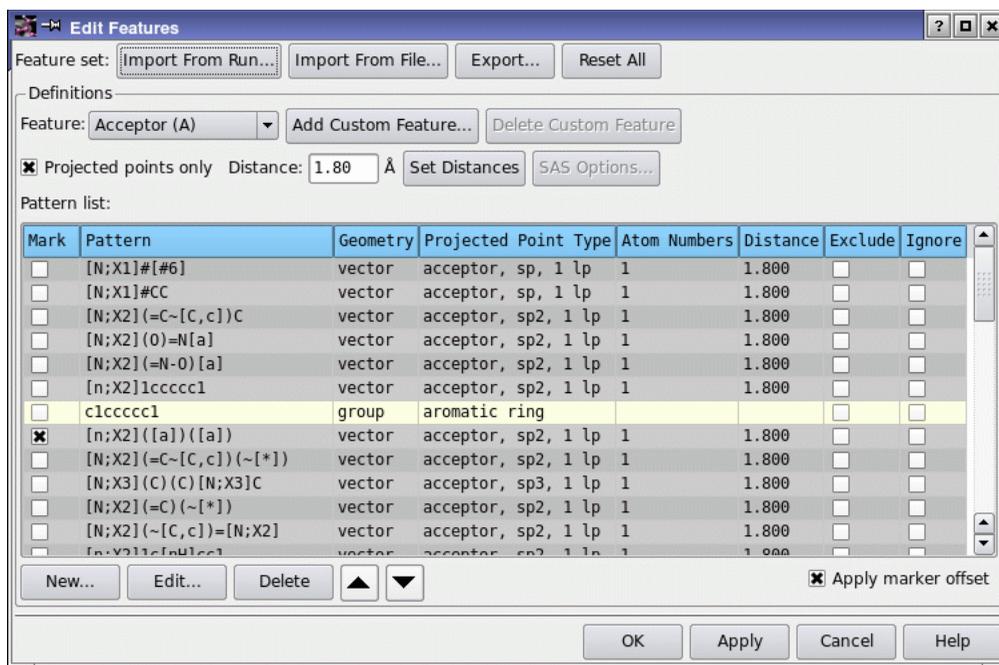


Figure 4.3. The Edit Features dialog box.

4.2.1 Loading and Storing Feature Sets

The built-in feature sets are stored in the Phase product distribution. To reload them, click Reset All. This button also clears the custom sets.

Feature sets are stored with each run. To import a feature set from another run, click Import from Run, and select the desired run in the dialog box that is displayed.

Feature sets can also be stored in a file. As you do not have access to runs from other projects, you must store feature sets that you want to use in other projects in a file. To save a feature set to a file, click Export, and specify the file location in the file chooser that is displayed. To import a feature set from a file, click Import from File, and navigate to the feature file. You can also use a saved feature file as the default—see [Section 2.6 on page 12](#) for instructions.

Table 4.2. Pattern list table columns

Column	Description
Mark	Column of check boxes. Selecting a check box marks the pattern on any ligands that are displayed in the Workspace.
Pattern	Pattern definition. With the exception of default hydrophobic features and aromatic rings, the definitions are all SMARTS strings.
Geometry	Designates physical characteristics of site. Can be point, vector, or group, as described previously.
Projected Point Type	Defines the directionality of vector features. Can be an aromatic ring, a donor, an acceptor with one or more lone pairs, or none (i.e., a nonvector feature).
Atom Numbers	The list of atoms that determine the location of the pharmacophore site, numbered according to the SMARTS string. Point and vector geometries use a single atom, whereas group geometry uses multiple atoms.
Distance	Distance of the projected point from the ligand atom. This column only applies when Projected points only is selected. To change the distance for a pattern, you can edit the value in this column.
Exclude	Column of check boxes. Selecting a check box excludes the atoms in this definition from being mapped by other definitions. This is essentially a NOT operator. Excluded patterns are processed first when searching for features.
Ignore	Column of check boxes. Selecting a check box ignores the pattern when searching for features. Equivalent to deleting the pattern, but keeps the pattern in the table.

4.2.2 Adding and Editing Custom Patterns

If the patterns in a given feature do not cover all the functional groups that you want to include in the feature, you can add extra patterns. To add a new SMARTS pattern to a feature, first choose the feature from the Feature option menu. You then have two options for adding the SMARTS pattern:

- Click the New button below the Pattern list table. The New Pattern dialog box is displayed. In this dialog box, you can enter a SMARTS pattern, define the feature geometry and projected point type and the atoms that represent the feature.
- Right-click on an existing SMARTS pattern that is similar to the one that you want and choose Duplicate Pattern from the shortcut menu. Click Edit to open the Edit Pattern dialog box, in which you can change the SMARTS pattern, the feature geometry and projected point type, and the atoms that represent the feature.

The New Pattern and Edit Pattern dialog boxes have the same controls. When you have made your choices, click OK to add or update the pattern. The choices are described in detail below.

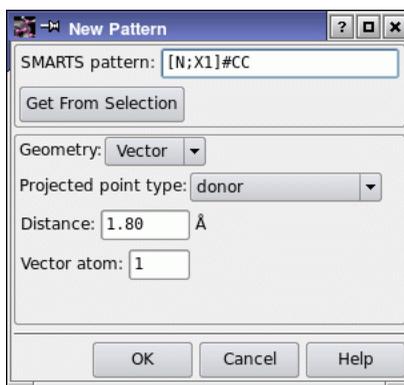


Figure 4.4. The New Pattern dialog box.

To add a pattern to a feature, you must provide the SMARTS string for the desired arrangement of atoms, and define the corresponding pharmacophore site. The pharmacophore site can be a group, such as an aromatic ring; a single point, such as an atom; or a vector, such as a hydrogen bond acceptor or donor.

1. If you are creating a new pattern, type the SMARTS string into the SMARTS pattern text box, or click **Get From Selection** to use the selected atoms in the Workspace to define a SMARTS string.

If you use the Workspace selection to define the SMARTS pattern, or you duplicated an existing pattern, you might want to edit it before proceeding.

2. Choose **Group**, **Point**, or **Vector** from the **Geometry** option menu.

The remaining controls in the dialog box depend on the choice you make from this menu.

- **Group**—The pattern contributes a group of atoms to the pharmacophore feature definition, with the pharmacophore site placed at the centroid. The **Projected point type** menu has only **none** and **aromatic ring** options available, and the **Group atoms** controls are displayed.
- **Point**—The pattern contributes a single atom to the pharmacophore feature definition, with the pharmacophore site placed at that atom. The only available item on the **Projected point type** menu is **none**, and the **Point atom** text box is displayed.
- **Vector**—The pattern contributes an atom with one or more directions to the pharmacophore feature definition, with the pharmacophore site placed at the atom. The **Projected point type** menu has items for **donor** and **acceptor** groups, and the **Vector atom** text box is displayed.

3. Choose the point type from the Projected point type option menu:
 - Group: Choose aromatic ring if the SMARTS pattern defines an aromatic ring, otherwise choose none.
 - Point: none is the only available choice.
 - Vector: Choose donor if the pattern represents a hydrogen bond donor, or choose the acceptor, *spn*, *m lp* item that defines the type of acceptor (hybridization and number of lone pairs at the acceptor) if the pattern represents a hydrogen bond acceptor. If you want to use projected points, enter the desired distance between the projected point and the ligand atom in the Distance text box.
4. Choose the atoms that define the pharmacophore site:
 - Group: Select All if all atoms in the SMARTS pattern are to be used to define the group centroid, or select Numbers and type the atom numbers for the group centroid in the text box, separated by commas.
 - Point: Type the atom number for the pharmacophore site in the Point atom text box.
 - Vector: Type the atom number for the pharmacophore site in the Vector atom text box. This should be the donor or acceptor atom.

The atom numbers refer to the order of the atoms in the SMARTS string.

Once you have added a pattern, you can edit it by clicking Edit. The Edit Pattern dialog box is displayed. This dialog box has the same controls as the New Pattern dialog box. If you no longer need the pattern, you can click Delete to delete it. However, you can also ignore it, if you want to keep it in the definition for other applications, but not use it—see the next section. Both of these buttons are only available when you select a custom pattern. Custom patterns are highlighted in blue in the Pattern list table.

4.2.3 Choosing How Patterns Are Used

Matching of patterns to ligand structures is done in the order specified in the Pattern list table. For example, if the first pattern maps a particular nitrogen in the ligand as an acceptor, that same nitrogen will not be mapped as an acceptor by any subsequent pattern. If you have added custom patterns, you can move them up and down the list with the arrow buttons below the table to set their priority. You cannot change the order of the built-in patterns.

If you want to exclude functional groups represented by a pattern from the feature, you can select the check box in the Exclude column for the pattern. For example, you might want to exclude a carboxylic acid group from being considered as a hydrogen bond donor, because it will be ionized under physiological conditions. Excluded functional groups are processed before included groups, so their position in the table does not matter.

If you want a pattern to be ignored, you can select the check box in the Ignore column. Ignored patterns are equivalent to deleted patterns. If you want to save a custom pattern for later use, but not use it in the current feature, select the check box in the Ignore column.

4.2.4 Viewing Patterns

The patterns that define a feature can be viewed individually in the Workspace for each ligand. To display a pattern for a particular ligand, select the ligand in the Ligands table (in the Define Pharmacophore Model panel), then select the check box in the Mark column of the Pattern list table (in the Edit Features dialog box) for the desired pattern. Any occurrences of the pattern are marked in the ligand structure.

You can display markers for more than one pattern, but the markers do not distinguish between patterns. You can display markers for more than one ligand by including the ligands in the Workspace. To see the atoms and bonds as well as the markers, select Apply marker offset.

4.2.5 Adding Custom Features

Phase allows you to define up to three custom features. By default, these features are listed in the Feature option menu as Custom (X), Custom (Y), and Custom(Z), and have the default aromatic vector and surface and aliphatic surface feature definitions included, but ignored. You can add patterns to these features and set their status as described in the sections above.

If you want to delete a custom feature, click Delete Custom Feature. To add a custom feature back, click Add Custom Feature. The Add Custom Feature dialog box is displayed, in which you can specify a name and choose a code letter for the new custom feature. The custom feature is added to the Feature option menu and selected, and the Pattern list table is populated with the default features as described above. Deleting then adding a custom feature is the only way to rename the feature.

4.2.6 Using Projected Points

By default, donors and acceptors are represented by vectors originating at the donor (hydrogen) or acceptor atom. The alignment of these vectors is used to determine whether ligands share the associated feature. Sometimes, two active ligands can form a hydrogen bond to the same receptor site, but from different directions. The projected point is in the same location but the ligand features are not. With the default representation, these two ligands would not contribute to the same pharmacophore hypothesis.

You can replace the vectors with points at a specified distance from the ligand donor or acceptor atom. These points simulate the corresponding acceptor or donor in the receptor, and are called *projected points*. In the default feature set, the projected points are implicit. To use

projected points, select Projected points only. To use the same distance for all patterns, enter a distance in angstroms in the Distance text box, and click Set Distances. To set a distance for an individual pattern, edit the value in the Distance column of the Pattern list table for the pattern.

With this option set, only the patterns that have a vector geometry and a defined projected point type contribute to the feature. All other patterns that are not excluded are ignored. Vector alignments are not used because the vectors have been replaced by points.

4.2.7 Surface Area Calculations for Hydrophobic Features

For a site to be accepted as hydrophobic, the solvent-accessible surface area of the hydrophobic group must exceed a certain threshold. If the area is too small, the site will be a very weak hydrophobe, if it is a hydrophobe at all. You can adjust several parameters that are used to calculate the surface area in the SAS Options dialog box, which you open by clicking the SAS Options button in the Edit Features panel. This button is only available if you have chosen Hydrophobic from the Feature option menu.

To set the minimum area threshold, enter a value in the Minimum required surface area per site text box. The solvent radius used in the surface area calculations is 1.4 Å, which corresponds to water. You can change this value, if for example you want to use a different solvent. The Surface area resolution element text box provides a way of controlling the accuracy of the calculated surface area. This value is used in the partitioning of the atomic spheres for the surface area calculation.

4.3 Defining the Ligand Set for Model Development

If you have not already done so, you can define the active and inactive ligands that are used to develop the pharmacophore model in this step. The controls for doing so are the same as in the Prepare Ligands step. You can click Activity Thresholds to set thresholds for the activity, or you can select ligands for the active set in the Pharm Set column of the Ligands table. See [Section 3.5 on page 25](#) for more information.

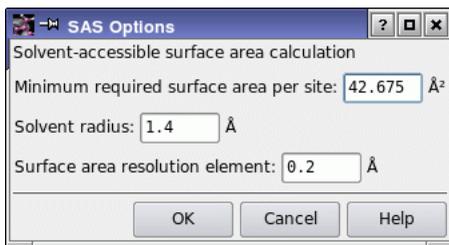


Figure 4.5. The SAS Options dialog box.

4.4 Creating the Sites

Once you are satisfied with the feature set, click **Create Sites** to start the job that creates and stores the site points for each conformer of each ligand. A **Start** dialog box opens, in which you can make settings and start the job. The job status icon turns green and rotates until the job finishes, when it turns red and stops rotating.

If the sites already exist for a conformer set (because you copied them from another run, for example), a link is made to this set instead of running the job.

4.5 Step Summary

To create site points for each ligand:

1. Click **Create Sites**.
2. Click **Start** in the **Start** dialog box to run the job.
3. Click **Next** to proceed to the next step.

Optional tasks:

- Add to the existing features, create custom features, and exclude or ignore patterns by clicking **Edit Features**.
- Select the use of projected points for acceptors and donors rather than treating them as vector features.
- Define the active and inactive ligands by clicking **Activity Thresholds** or clicking in the **Pharm Set** column of the **Ligands** table.

Finding Common Pharmacophores

In the Find Common Pharmacophores step, pharmacophores from all conformations of the ligands in the active set are examined, and those pharmacophores that contain identical sets of features with very similar spatial arrangements are grouped together. If a given group is found to contain at least one pharmacophore from each ligand, then this group gives rise to a *common pharmacophore*. Any single pharmacophore in the group could ultimately become a common pharmacophore *hypothesis*—an explanation of how ligands bind to the receptor.

Common pharmacophores are identified from a set of *variants*. A variant is a set of feature types that define a possible pharmacophore—for example, the variant ADHH contains a hydrogen-bond acceptor, a hydrogen-bond donor, and two hydrophobic groups.

Phase searches for common pharmacophores with a given number of pharmacophore sites. You can specify from 3 to 7 sites: hypotheses with more sites are not likely because each site represents a 2-3 kcal/mol interaction with the receptor. In addition, you can control how many ligands must match to form a valid hypothesis, and how many of each kind of feature must be included in the match. After the search is complete, the variants for which common pharmacophores were found are passed to the next step.

5.1 The Search Method

Common pharmacophores are identified using a tree-based partitioning technique that groups together similar pharmacophores according to their *intersite distances*, i.e., the distances between pairs of sites in the pharmacophore. Each k -point pharmacophore is represented by a vector of n distances, where $n = k \cdot (k-1)/2$. Each intersite distance d is filtered through a binary decision tree, such as in [Figure 5.1](#).

The tree in [Figure 5.1](#) has a depth of four and partitions distances (in angstroms) on the interval (0, 16] into bins that are 2 Å wide. If each of the n distances in a pharmacophore is filtered in this manner, an n -dimensional partitioning of the pharmacophore is created. This representation is referred to as an n -dimensional box, where the sides of the box are equal to the bin width. Thus a pharmacophore is mapped, according to its intersite distances, into a box of finite size. All pharmacophores that are mapped into the same box are considered to be similar enough to facilitate identification of a common pharmacophore. So if each of the minimum required number of active-set ligands contributes at least one pharmacophore to a particular box, then that box represents a common pharmacophore. Such boxes are said to *survive* the partitioning procedure, while all others are eliminated.

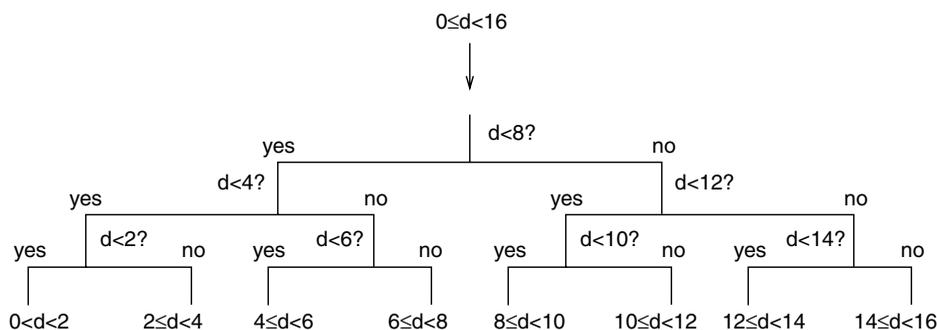


Figure 5.1. Binary decision tree.

5.2 Defining the Scope of the Search

Searching for all possible common pharmacophores could take a long time. From your knowledge of the system of interest, you might not want to search for pharmacophores that have too many or too few site points, or that have too many or too few features of a particular type. Phase provides the means to narrow the search to the variants of interest.

When you enter this step for the first time, a list of all available variants in the set of ligands designated as active is computed, from the number of available sites of each type for each ligand. This list is usually shorter than the theoretical maximum length, because the ligands don't necessarily include all possible variants. The list is filtered with the default settings for the number of sites before it is displayed in the Variant list table. The frequencies of occurrence of the features are used to determine how many occurrences of each feature could be found in a valid hypothesis, given the number of ligands that must be matched. These values are listed in the Available column of the Feature frequencies table, which is described in [Table 5.1](#).

The first task is to decide how many site points to include in the hypothesis. You can choose a maximum number and a minimum number. The default is 5 for both maximum and minimum, but you can choose any number between 3 and 7, inclusive, from the Maximum number of sites and Minimum number of sites option menus. When you make your choice, you should be aware that the likelihood of finding common pharmacophores is decreased as the number of sites is increased. The search then starts with the maximum number of sites, and if it does not find any common pharmacophores, it decreases the number of sites and runs the search again, until it either finds common pharmacophores or passes the minimum number of sites. Thus, the results returned are always for a particular number of site points.

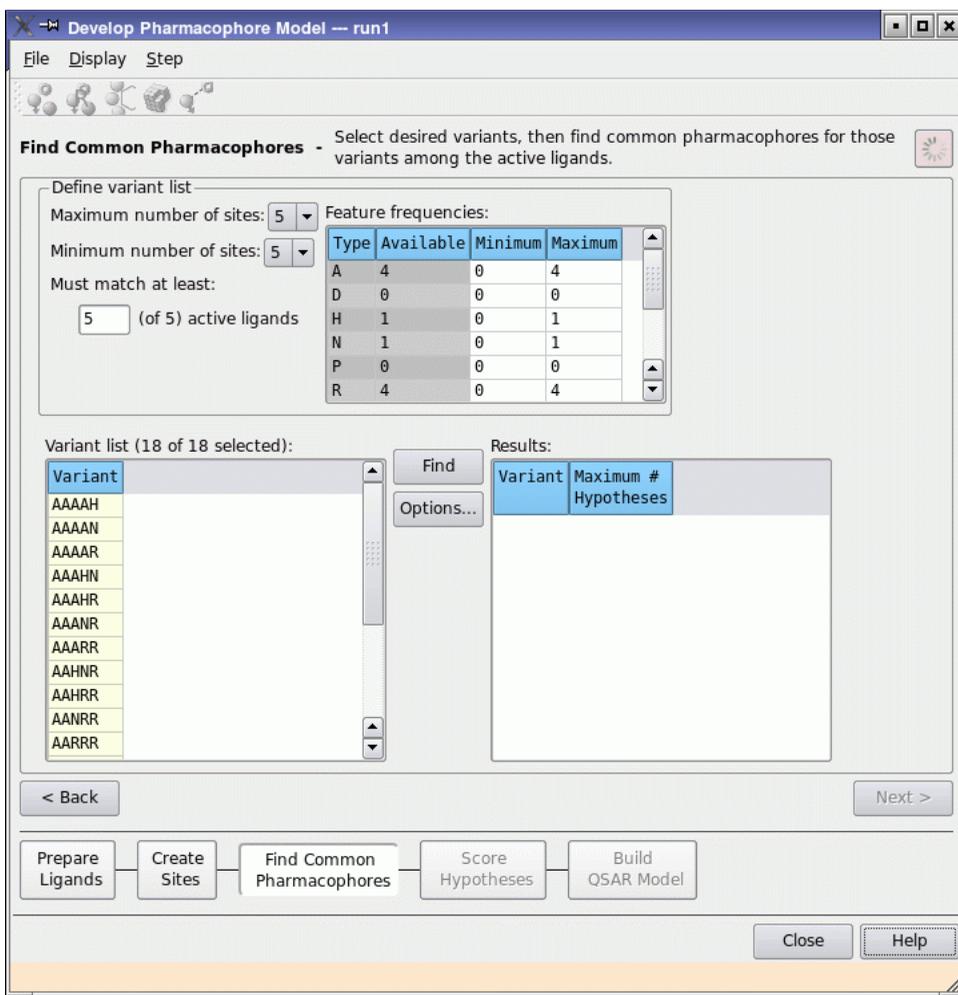


Figure 5.2. The Find Common Pharmacophores step.

If you want to generate and examine hypotheses with different numbers of site points, you can create a new run for each number of points. To create a run that stores the information to date, choose **Save As** from the **File** menu, and name the new run. The original run is preserved, and you are now working in the new run. To revert to the original run, choose the run from the **Open** submenu of the **File** menu.

By default, all of the active-set ligands must contain a given variant for that variant to be listed. However, Phase allows you to relax this criterion so that a common pharmacophore need only match a subset of the chosen actives. This is often a necessity when more than one binding

Table 5.1. Description of Feature frequencies table columns.

Column	Description
Type	Lists the features by code letter. Noneditable.
Available	Number of sites of this type that are available, defined as the largest number of occurrences of this feature for which a match can be found for the number of ligands to be matched. For example, with ten ligands, of which three ligands have 4 Acceptors, six have 5 Acceptors, and one has 6 Acceptors, the number of Acceptors available is 4 if all ten ligands are matched, but is 5 if seven ligands are matched. Noneditable; updated if the number of ligands to match changes.
Minimum	Minimum number of features of the given type allowed in any variant. Editable: you can set the value to restrict the possible variants. The default is zero.
Maximum	Maximum number of features of the given type allowed in any variant. Editable: you can set the value to restrict the possible variants. The default is the maximum possible number, given the number available and the number of sites.

mode is observed among the actives. If you want to widen the search, you can set the number of actives that must contain the variant to a number less than the total, in the **Must match at least** text box. The number must be between 1 and the maximum, inclusive. The maximum number (the number of chosen actives) is displayed to the right of the text box. The fewer ligands you require a match to, the more variants will be listed.

Not all the variants are likely to be useful: for example, a variant with five acceptors might be physically unreasonable, and should be excluded from the search. You can limit the number of occurrences of any of the features by entering a minimum and maximum number in the **Minimum** and **Maximum** columns of the **Feature frequencies** table. For example, you might want variants that have between 1 and 3 acceptors. In this case you would enter 1 in the **Minimum** column of the **A** row, and 3 in the **Maximum** column.

After each change, the variant list is automatically updated.

When the search for common pharmacophores is run, a ligand is considered to match if one of the conformers matches. On occasions, you might want to extend the set of molecules classed as a “ligand” to include other molecules that are structurally related, such as tautomers, stereoisomers, or ionized states. Such extended sets are called **ligand groups**, and can be defined when you run model development from the command line.

If you have set the number of actives that must match to less than the total, you might still want to require certain actives to match. This capability is also available from the command line.

These two features can be used in Maestro by using an override file. For instructions, contact help@schrodinger.com.

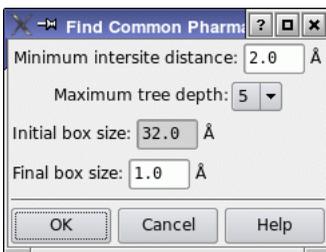


Figure 5.3. The Find Common Pharmacophores - Options dialog box.

5.3 Modifying the Search Parameters

In the Find Common Pharmacophores - Options dialog box, you can specify the parameters that govern the search for common pharmacophores. Specifying the parameters is a balance between the size of the features in the hypotheses, the size of the ligands, and the time taken and storage requirements for the search. To open this dialog box, click Options in the Find Common Pharmacophores step. The text boxes are described below.

Minimum intersite distance

Specifies the minimum distance allowed between two features. If the features in the ligand are closer than this distance, the hypothesis is rejected.

Maximum tree depth

Specifies the number of binary partitioning steps used to sort the pharmacophores into similar groups. This is the maximum recursion level in the partitioning, and the depth of the resulting binary partitioning tree.

Initial box size

Noneditable. Specifies the size of the initial box in the partitioning algorithm, computed from the final box size and the maximum tree depth:

$$\text{Initial box size} = (\text{Final box size}) * 2^{(\text{Maximum tree depth})}$$

This size should be roughly the size of the binding pocket, or of the smallest ligand. You should therefore choose the final box size and the maximum tree depth to ensure that the initial box size is big enough. The default is 32 Å.

Final box size

Specifies the size of the boxes that contain intersite distances that are considered to be equivalent. This option governs the tolerance on matching: the smaller the box size, the more closely

pharmacophores must match. However, the smaller the box size, the longer the search takes. If you choose a smaller final box size, you might have to increase the maximum tree depth so that the initial box size is large enough. If the final box size is too small, the tolerance on matching might be too strict to produce any common pharmacophores.

5.4 Starting the Search

When you have defined the list of variants, you can proceed to the search for common pharmacophores. The search is performed on the variants that are selected in the Variant list table. By default, all variants are selected. You can select variants from the Variant list using the usual combinations of click, control-click, and shift-click. You must have at least one variant selected to run the search.

The common pharmacophores are identified using a binary partitioning algorithm, in which the pharmacophores are split into progressively smaller and more similar groups based on intersite distances. If you want to change the parameters of the search, you can do so by clicking Options and setting the values in the dialog box that is displayed—see [Section 5.3 on page 43](#).

To start the search, click Find. A dialog box is displayed, in which you can select the host, the number of CPUs to use, and the user name on the host. You can distribute this job over multiple processors.

The results of the search can take a large amount of disk space, depending on the number of ligands and their size and flexibility. The search results are kept inside the run, which is stored in the Maestro project. You should make sure that you have adequate disk space: in the temporary storage on the host or hosts on which you run the job, in the Maestro I/O directory, and in the project. Because of the disk space requirements, it is not advisable to run from a scratch project, which is kept by default in `/home/username/.schrodinger`. Instead, you should save the project to a disk that has plenty of free space.

The results of the run are displayed in the Results table. This table shows the maximum number of hypotheses that could be produced by each variant. You can sort the table by clicking the column headers. Some variants may have no common pharmacophores. These variants are not passed to the next step.

5.5 Step Summary

To find common pharmacophores:

1. Choose the number of sites from the Number of sites option menu.
2. Specify the number of actives to match in the Must match section.
3. Set limits on the minimum and maximum number of features of each type in the Feature frequencies table.
4. Select variants from the Variant list.
5. (Optional) Set search parameters by clicking Options and entering values in the Find Common Pharmacophores - Options dialog box.
6. Start the search by clicking Find.
7. Click Next to proceed to the next step.

Scoring Hypotheses

In the Score Hypotheses step, common pharmacophores are examined, and a scoring procedure is applied to identify the pharmacophore from each surviving n -dimensional box that yields the best alignment of the chosen actives. This pharmacophore provides a hypothesis to explain how the active molecules bind to the receptor. There will of course be many hypotheses, because there are many boxes. The scoring procedure provides a ranking of the different hypotheses, allowing you to make rational choices about which hypotheses are most appropriate for further investigation.

Following the scoring of the hypotheses, the remaining molecules can be used to provide extra information in the hypothesis, based on their structure. To make comparisons, Phase uses *partial matching* to obtain alignments for these ligands. If at least three sites in the hypothesis can be matched, an unambiguous alignment is obtained. For each ligand not designated active, Phase searches for matches involving the largest possible number of sites, and identifies the match that yields the highest fitness score.

If the pharmacophore is an adequate hypothesis, it should discriminate between active and inactive molecules. By aligning and scoring known inactives, you can check the validity of the hypotheses that you generated. If inactives score well, the hypothesis could be invalid because it does not discriminate between actives and inactives, and therefore does not explain how active molecules bind but inactives do not. The hypothesis could also be incomplete because it lacks either a critical site that explains the binding or information on what prevents inactives from binding.

The pharmacophore features that were identified are not the only features that may be useful in defining a good hypothesis. Inactive molecules that have the same pharmacophore features could have functional groups in regions of space not occupied by the active molecules. It is reasonable to suppose that these regions are occupied by the receptor. These regions can then be added to the hypothesis as *excluded volumes*, and used in the database search to screen matches to the hypothesis.

Inactive molecules could also have different functional groups in the same location as functional groups in the active molecules, or be missing functional groups that are in the active molecules. Visual inspection of the aligned ligands can help you understand the structural differences. These differences can also be quantified by building a QSAR model (in the next step), which can be used for screening matches in the database search as well as for identifying functional groups that contribute, positively or negatively, to activity.

6.1 The Scoring Process

A surviving box contains a set of very similar pharmacophores culled from conformations of a minimum number of active-set ligands, and certain of these ligands may contribute more than one pharmacophore to a box. Each pharmacophore and its associated ligand are treated temporarily as a *reference* in order to assign a score. This means the other *non-reference* pharmacophores in the box are aligned, one-by-one, to the reference pharmacophore, using a standard least-squares procedure applied to the corresponding pairs of site points.

The quality of each alignment is measured in three ways: (1) the *alignment score*, which is the root-mean-squared deviation (RMSD) in the site-point positions; (2) the *vector score*, which is the average cosine of the angles formed by corresponding pairs of vector features (acceptors, donors, and aromatic rings) in the aligned structures; and (3) a *volume score* based on the overlap of van der Waals models of the non-hydrogen atoms in each pair of structures,

$$S_{\text{vol}}(i) = V_{\text{common}}(i)/V_{\text{total}}(i) \quad (1)$$

$V_{\text{common}}(i)$ is the common or overlapping volume between ligand i and the reference ligand, while $V_{\text{total}}(i)$ is the total volume occupied by both ligands.

In principle, a reference pharmacophore could score well, even though it contains one or two very poor individual alignments. For this reason, user-adjustable cutoffs are applied to the RMSD values and vector cosines of each individual alignment. Any reference pharmacophore that violates a cutoff in any individual alignment is eliminated. A *site score* for each alignment is then computed based on the alignment score $S_{\text{align}}(i)$ and the cutoff C_{align} by

$$S_{\text{site}}(i) = 1 - S_{\text{align}}(i)/C_{\text{align}} \quad (2)$$

This score is always between 0 and 1 because alignments with $S_{\text{align}}(i) > C_{\text{align}}$ are eliminated.

The site score, the vector score, and the volume score are combined with separate weights to yield a combined alignment score for each non-reference pharmacophore that has been aligned to the reference. If a non-reference ligand contributes more than one pharmacophore to the box, the pharmacophore yielding the best alignment to the reference is selected. The overall multi-ligand alignment score for a given reference pharmacophore is the average score from the best individual alignments.

After all pharmacophores in a box have been treated as a reference, the one yielding the highest multi-ligand alignment score is selected as the hypothesis for that box. The ligand that contributes the reference pharmacophore is referred to as the *reference ligand* for that hypothesis. The non-reference information is carried along with each hypothesis so that additional scoring can be performed using the optimal multi-ligand alignment.

Once hypotheses have been identified across all boxes, the lower scoring hypotheses can be eliminated by applying a percentage cutoff to the overall alignment score. In case the percentage filter yields a very small number of hypotheses, a minimum number of hypotheses can be specified.

After this stage of scoring is completed, the ranking of the hypotheses can be refined using volume and selectivity scoring. The overall volume score for a hypothesis is the average obtained by applying the formula given above to all non-reference ligands i . The volume score (S_{vol}) can be added to the overall score with its own user-adjustable weight (W_{vol}).

Selectivity is an empirical estimate of the *rarity* of a hypothesis, i.e., what fraction of molecules are likely to match the hypothesis, regardless of their activity toward the receptor. Selectivity is defined on a logarithmic scale, so a value of 2 means that 1 in 10^2 molecules would be expected to match the hypothesis. Higher selectivity is desirable because it indicates that the hypothesis is more likely to be unique to the active-set ligands. Selectivity is only a rough estimate of the rarity, so you should be careful not to place too much emphasis on it in the overall ranking of hypotheses. As with the other types of scores, the selectivity score (S_{sel}) can be added to the overall score with its own user-adjustable weight (W_{sel}).

If you choose to match less than the total number of chosen actives, you may wish to assign higher scores to hypotheses that match a greater number of the chosen actives. The reward comes in the form of W_{rew}^m , where W_{rew} is user-adjustable (1.0 by default) and m is the number of actives that match the hypothesis minus one. If W_{rew} is increased much above 1.0, care must be taken not to make it too large, or it may completely dominate the scoring function. For example, if you have 10 actives and W_{rew} is 1.4, this contribution to the score could have a value of 32. The other terms have a maximum value of 1.0.

Hypotheses for which the reference ligand has a high energy relative to the lowest-energy conformer for that ligand are less likely to be good models of binding, because of the energetic cost. You can include a penalty for high-energy structures by subtracting a multiple of the relative energy from the final score, $W_E \Delta E$.

Likewise, you can penalize hypotheses for which the reference ligand activity is lower than the highest activity, by adding a multiple of the reference ligand activity to the score, $W_{\text{act}} A$, where A is the activity.

The final scoring function—the *survival score*—has the following form:

$$S = W_{\text{site}} S_{\text{site}} + W_{\text{vec}} S_{\text{vec}} + W_{\text{vol}} S_{\text{vol}} + W_{\text{sel}} S_{\text{sel}} + W_{\text{rew}}^m - W_E \Delta E + W_{\text{act}} A \quad (3)$$

where the W 's are the weights and the S 's the scores.

If the hypothesis itself is a sufficient explanation of binding and activity, hypotheses for which inactives match well are unlikely to be good hypotheses. If the reason that inactives do not bind is steric hindrance rather than the lack of a particular pharmacophore feature, you can use excluded volumes to filter matches—see [Section 6.7 on page 59](#).

You can penalize hypotheses that match inactives by calculating the survival score for the inactives, and subtracting a multiple of this score from the survival score for the actives. To ensure that inactives that do not match all sites in the hypothesis are penalized, their alignment score is adjusted. If a given inactive matches only k out of n sites in a hypothesis, the effective n -point alignment score is computed from the k -point alignment score as follows:

$$S_{\text{align},n} = \sqrt{W_k S_{\text{align},k}^2 + (1 - W_k) C_{\text{align}}^2} \quad (4)$$

where $W_k = k/n$. This score is then used in [Equation \(2\)](#) to calculate the site score. If an inactive fails to match at least 3 sites in the hypothesis, an unambiguous alignment cannot be obtained, and its contribution to the site score is 0. (This follows from setting $k=0$ in [Equation \(4\)](#).)

The inactive scoring function is the same as for actives. In addition to computing the inactive score, an adjusted survival score is calculated in which a multiple of the survival score of the inactives is subtracted from the actives survival score:

$$S_{\text{adj}} = S_{\text{actives}} - W_{\text{inactives}} S_{\text{inactives}} \quad (5)$$

6.2 Scoring the Hypotheses

The first task in the Score Hypotheses step is to align the actives to the hypotheses and calculate the score for the actives. To do this, click **Score Actives**. When you do so, the **Score Actives** dialog box ([Figure 6.1](#)) is displayed, in which you can examine and adjust the weights of the terms in the survival score, the alignment thresholds, and the filters on the number of hypotheses to keep. Details of these parameters are given below. When you have made any changes, click **Start**, set job parameters in the **Start** dialog box, and click **Start** again. When the job finishes, the surviving hypotheses and their scores are displayed in the **Hypotheses** table.

6.2.1 Scoring Method and Filtering

In this section, you can set the thresholds for filtering out hypotheses with low alignment scores and with poor feature matching, and set a limit on the number of hypotheses to keep.

Alignment Scores

Vector and site alignment scores are computed first, and used to filter the hypotheses. You can set the following parameters, all of which are applied to filter the hypotheses:

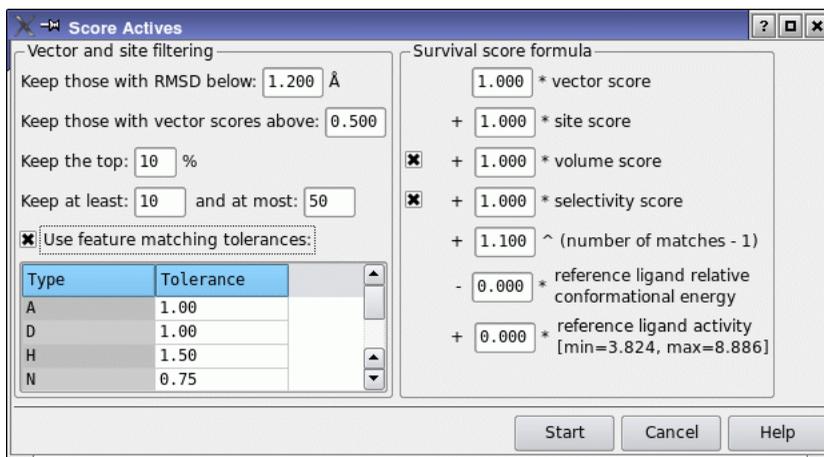


Figure 6.1. The Score Actives dialog box.

Keep those with RMSD below *threshold* Å

Threshold for RMS deviation of the intersite distances of any contributing ligand from those of the reference ligand. The default is 1.2 Å.

Keep those with vector scores above *threshold*

Threshold for the variation in the alignment of vectors between any contributing ligand and the reference ligand. The maximum is 1.0, which corresponds to perfect alignment. The minimum is -1.0, which would keep all hypotheses, regardless of vector alignment.

Numerical Cutoffs

To limit the number of hypotheses, you can set the following cutoffs on the fraction or number of hypotheses to keep.

Keep the top *n* %

Limit on the percentage of hypotheses to keep, in order of combined alignment score.

Keep at least *n* and at most *m*

Lower and upper limits for the number of hypotheses to keep. If the percentage of hypotheses kept is lower or higher than these limits, these limits override the percentage limit.

Feature-Matching Tolerances

In addition to using the RMSD to filter out hypotheses, you can set matching tolerances on individual features. Features are considered to match if the site points are within the specified

tolerance. This feature is useful if the RMSD matching is satisfied, but one or more features do not match well enough.

To apply feature-matching tolerances, select **Use feature matching tolerances**. The tolerances for each feature type are listed in the table below, and can be edited. All tolerances are applied: if you want to disable matching tolerances for a particular feature type, set the tolerance to a large value.

6.2.2 Survival Score Weighting Factors

The Weighting factors section of the Score Actives dialog box defines the survival score of the hypotheses, which is reported in the Hypotheses table of the Score Hypotheses step along with the individual scores that make up the survival score. The possible ranges for each score and weight are given in [Table 6.1](#).

The lower end of the actual range for the vector score is limited by the cutoff specified in the Vector and site filtering section. Similarly, the maximum relative energy is limited by any cutoff you specified when generating conformers, such as in the Generate Conformers dialog box (see [Section 3.3 on page 19](#)).

The selectivity score weight is zero by default because it might eliminate useful hypotheses. Likewise, the energy and activity weights are zero by default.

The weight for the number of matches is raised to the power of the number of matches minus one. A value of 1.0 does not discriminate on the basis of the number of matches. This score can be useful when the required minimum number of actives is smaller than the total number of actives. Adjust this weight with caution: even a value of 2.0 can give large variations in survival scores, and dominate the survival score.

Table 6.1. Maximum score ranges and allowed weight ranges in the survival score

Score	Score Range	Weight Range	Default Weight
vector score	-1.0 to 1.0	0.0 to 1.0	1.0
site score	0.0 to 1.0	0.0 to 1.0	1.0
volume score	0.0 to 1.0	0.0 to 1.0	1.0
selectivity score	0.0 to ∞	0.0 to 1.0	0.0
number of matches	1.0 to ∞	1.0 to ∞	1.0
reference ligand relative conformational energy	0.0 to ∞	0.0 to ∞	0.0
reference ligand activity	Determined by input	0.0 to ∞	0.0

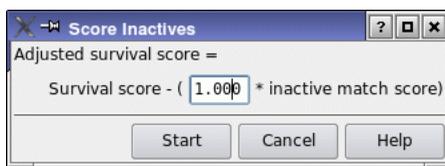


Figure 6.2. The Score Inactives dialog box.

6.3 Scoring Inactives and Rescoring

If the hypothesis is a sufficient explanation of the activity of the ligands, then the inactive ligands can be expected to lack one or more of the features in the hypothesis. However, if it is not a sufficient explanation, the inactives might match all the features in the hypothesis. In that case the hypothesis could for example be missing a feature, or an account of steric clashes. In inactive scoring, survival scores are adjusted to penalize hypotheses that match inactives, assuming that the inactives fail to bind because they do not contain the true pharmacophore. While this condition is rarely satisfied by every inactive in a given set, at least some significant fraction of the inactives must lack the pharmacophore for this technique to be valid.

Once the hypotheses have been scored on the basis of the alignment of the chosen actives, you can calculate an adjusted score based on the alignment of the chosen inactives. The score is adjusted by subtracting a multiple of the survival score of the inactives from the survival score of the actives. To calculate it, click Score Inactives, specify a weight for the inactive score, and click Start, make settings in the Start dialog box and click Start. When the job finishes, the adjusted scores are displayed in the Survival-inactive column of the Hypotheses table.

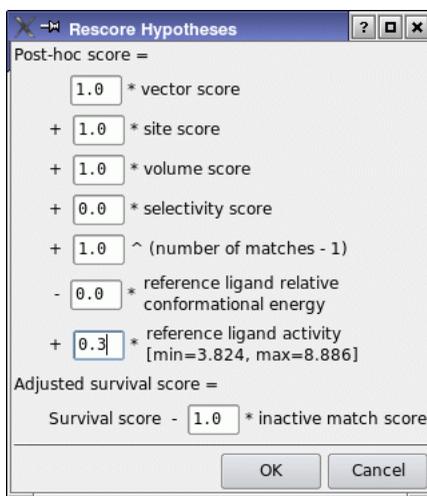


Figure 6.3. The Rescore Hypotheses dialog box.

If you want to apply a different scoring function to the surviving hypotheses, you can do so by clicking Rescore, and setting values for the coefficients (weights) of the scoring function in the Rescore Hypotheses dialog box (Figure 6.3). This dialog box contains the same controls as in the Score Actives dialog box. At the same time, you can adjust the weight of the inactives in the Survival-inactive score. The results of the rescoring are listed in the Post-hoc column of the Hypotheses table. These results correspond to the Survival score, and do not include any penalty for matching inactives.

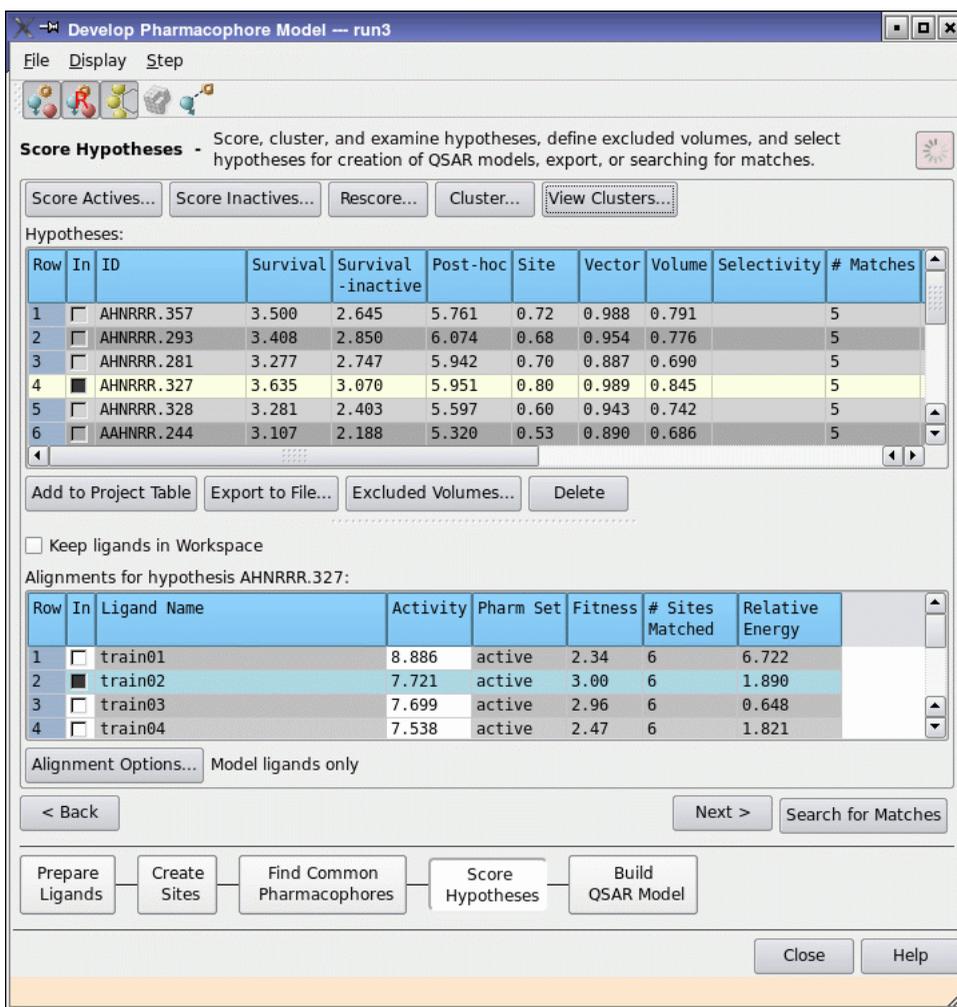


Figure 6.4. The Score Hypotheses step, after scoring.

6.4 Results of Scoring

The Hypotheses table displays the scores for each hypothesis. The In column can be used to display the hypothesis in the Workspace. The hypothesis ID is given in the second column, and consists of the variant name and an index. The remaining columns contain the various scores, whose definitions are given in Table 6.2. Extra columns are added when you cluster the hypotheses. You can sort the table by the values in a column by clicking the column heading. The data in the table is noneditable.

Table 6.2. Description of score columns in the Hypothesis table.

Column	Description
Survival	Weighted combination of the vector, site, volume, and survival scores, and a term for the number of matches. The weights of the volume score and survival score are set to 1.0 and 0.0 by default. The weights can be varied in the Score Actives dialog box. The minimum value of this score is 1.0.
Survival - inactives	Survival score for actives with a multiple of the survival score for inactives subtracted. The weight of the inactive survival score can be set in the Score Inactives dialog box.
Post-hoc	This score is the result of rescoring, and is a weighted combination of the vector, site, volume, and selectivity scores. You can set the weights in the Rescore Hypotheses dialog box, which you open by clicking Rescore .
Site	Site score. This score measures how closely the site points are superimposed in an alignment to the pharmacophore of the structures that contribute to this hypothesis, based on the RMS deviation of the site points of a ligand from those of the reference ligand.
Vector	Vector alignment score. This score measures how well the vectors for acceptors, donors, and aromatic rings are aligned in the structures that contribute to this hypothesis, when the structures themselves are aligned to the pharmacophore.
Volume	Measures how much the volumes of the contributing structures overlap when aligned on the pharmacophore. The volume score is the average of the individual volume scores. The individual volume score is the overlap of the volume of an aligned ligand with that of the reference ligand, divided by the total volume occupied by the two ligands.
Selectivity	Estimate of the rarity of the hypothesis, based on the World Drug Index. The selectivity is the negative logarithm of the fraction of molecules in the Index that match the hypothesis. A selectivity of 2 means that 1 in 100 molecules match. High selectivity means that the hypothesis is more likely to be unique to the actives.
# Matches	Number of actives that match the hypothesis.
Energy	Relative energy of the reference ligand in kcal/mol. This is the energy of the reference conformation relative to the lowest-energy conformation.
Activity	Activity of the reference ligand.
Inactive	Survival score of inactives. The scoring function is the same as for actives.

6.5 Examining Hypotheses and Ligand Alignments

Once you have generated scores for the hypotheses, you can examine the listed hypotheses, one at a time. To examine a hypothesis, select it in the Hypotheses table. All the ligands are listed in the Alignments table. The first four columns contain the same information or controls as the Ligands table in the Prepare Ligands step. The columns of this table are described in Table 6.3.

The columns of this table are noneditable. The row for the reference ligand (the ligand that matches the hypothesis exactly) is colored light blue. Rows for aligned ligands are colored light gray; rows for unaligned ligands are colored dark gray. You can sort the table by the values in a column by clicking the column heading.

Table 6.3. Description of Alignments table.

Column	Description
In	Inclusion status of the ligand. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. You can include and exclude ligands with click, shift-click, and control-click.
Ligand Name	The name of the ligand.
Activity	The ligand's activity. This value is editable.
Pharm Set	Indicates the status of a ligand in the set used to bind the pharmacophore model.
Fitness	Measures how well the conformer matches the hypothesis. The fitness score is a linear combination of the site and vector alignment scores and the volume score, and is related to the default survival score. The reference ligand, which matches exactly, has a perfect fitness score.
# Sites Matched	Number of sites on the ligand that matched the hypothesis.
Relative Energy	Energy of the best matching conformer relative to the lowest conformer. High relative energies indicate that the conformer is strained. A large proportion of high relative energies could indicate a poor hypothesis.

You can view the aligned ligands and information on their alignments in the Workspace by clicking the diamond in the In column of the Alignments table. This column is multiple-select: you can add ligands to the display with shift-click and control-click. You can display and undisplay the hypothesis by clicking the View Hypothesis toolbar button. From the toolbar (or the Display menu) you can also display the distances between the site points and the angles between all sets of three site points.

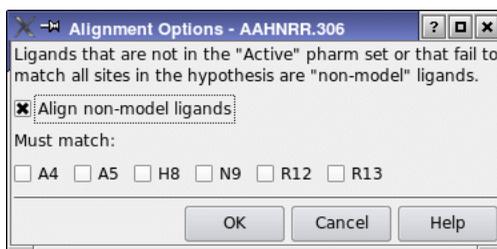


Figure 6.5. The Alignment Options dialog box.

If you want to keep the same set of ligands in the Workspace when you change hypotheses, select **Keep ligands in Workspace**. You can include or exclude ligands, and the new list is used when you change the hypothesis. The exception is that, if any included ligand is not aligned in the new hypothesis, it is not displayed.

It can also be useful to align and display the “non-model” molecules: the inactive molecules from the pharm set, the actives that do not match all site points, and the molecules that are not in the pharm set. To align these ligands, click **Alignment Options**, below the **Alignments** table.

In the **Alignment Options** dialog box (Figure 6.5) select **Align non-model ligands**. By default the molecules can match any three sites, but you can enforce matching at specific sites by selecting the sites under **Must match**. The tolerances for matching these sites are the tolerances specified in the **Score Actives** dialog box. When you have made your selections, click **OK**. The dialog box closes and the alignment is performed. The table is updated with information for all molecules that match three or more sites in the hypothesis. The rows for molecules that were not aligned are colored dark gray and are missing the information coming from the alignment.

If, in your examination of a hypothesis, you decide that the hypothesis is not a good one, you can delete it by clicking **Delete**.

If you have found one or more hypothesis that you want to use outside the **Develop Pharmacophore Model** workflow—for example, to search a database—you must make them available by selecting them in the **Hypotheses** table and performing one of the following actions:

- Add them to the **Project Table**, by clicking **Add to Project Table**. The selected hypotheses are automatically added to the **Project Table** when you click **Search for Matches**.
- Export them to a file by clicking **Export to File**. A file selector opens, in which you can navigate to the desired location and provide the file name. The extension is removed from the name you provide, so you do not need to add it.

Hypotheses are stored in the run as part of the project, but are not available for use in a search until they are either exported to an external file or added to the **Project Table**.

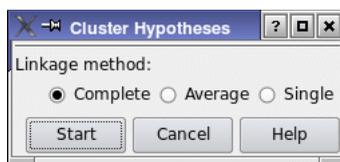


Figure 6.6. The Cluster Hypotheses dialog box.

You can also add aligned ligands to the Project Table or export them to a file. To add aligned ligands to the Project Table, select the ligands in the Alignments table, then right-click in the table and choose Add Alignments to Project Table from the shortcut menu. The ligands are added to the Project Table as an entry group, with all the properties added by Phase. Likewise, to export aligned ligands to a Maestro file, select the ligands in the Alignments table, then right-click in the table and choose Export Alignments to File from the shortcut menu. A file chooser labeled Export Alignments opens, in which you can navigate to the desired location and enter the file name.

6.6 Clustering Hypotheses

Frequently, there are several hypotheses of a given variant that look very much alike and have very similar scores. In such situations, it is useful to cluster these hypotheses, using a suitable clustering algorithm, and showing only a single representative from each cluster. Details of the clustering method used by Phase are given in [Section 12.6.6 on page 130](#).

To start the clustering job, click Cluster. The Cluster Hypotheses dialog box opens, allowing you to set the linkage method and start the job. The linkage method determines what kind of clusters are produced, as follows:

- **Complete**—The distance between clusters is the largest distance between any pair of objects (one object from each cluster). This option produces compact, spherical clusters.
- **Average**—The distance between clusters is the average distance between all pairs of objects in the two clusters.
- **Single**—The distance between clusters is the smallest distance between any pair of objects (one object from each cluster). This option produces diffuse, elongated clusters.

When the job has finished, three columns are added to the Hypotheses table that provide information about the clusters. These columns are described in [Table 6.4](#).

Table 6.4. Description of clustering columns in the Hypothesis table.

Column	Description
Cluster Number	Index of the cluster that the hypothesis belongs to.
Cluster Size	Size of the cluster that the hypothesis belongs to.
Average Similarity	Average similarity of the hypotheses in the cluster.

You can view the cluster representatives by making settings in the View Clusters dialog box, which you open by clicking View Clusters. By default, all hypotheses are shown, but you can choose from the following options:

- All hypotheses—Show all hypotheses in the cluster.
- Cluster representatives: highest average similarity—Show the hypothesis that has the highest average similarity to the other hypotheses in the cluster. All other hypotheses in the cluster are hidden.
- Cluster representatives: highest survival score—Show the hypothesis that has the highest survival score in the cluster. All other hypotheses in the cluster are hidden.

The size of the clusters, and hence the representatives that are displayed, depends on the similarity threshold for the clusters. To change this threshold, enter a value in the Intra-cluster similarity level text box. The default is 0.9. When you have selected an option and set the similarity level, click Apply or OK to view the representatives.

6.7 Adding Excluded Volumes to Hypotheses

If you have included inactive molecules in the Phase run, you can use these molecules to add regions of space to the hypothesis in which there should be no atoms from any active molecule. These excluded volumes are then used when you search for matches in your database to screen out ligands that have atoms in the excluded volumes.

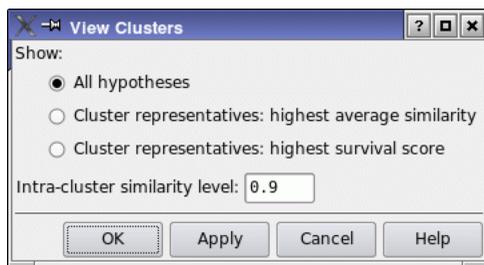


Figure 6.7. The View Clusters dialog box.

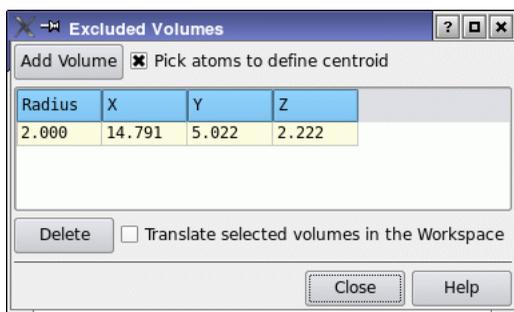


Figure 6.8. The Excluded Volumes dialog box.

To define one or more excluded volumes using the inactive molecules as a guide, first make sure that you have aligned the non-model ligands, then display one or more of the active molecules along with one or more of the inactive molecules. Regions of space in which inactive molecules have atoms but active molecules do not are likely candidates for excluded volumes.

With the ligands displayed, click Excluded Volumes. The Excluded Volumes dialog box is displayed. This dialog box allows you to pick atoms to define spherical excluded volumes.

To define a volume, pick a set of atoms in the Workspace that belongs to the inactive molecule or molecules. After each pick, the centroid of the atom set is calculated and a temporary sphere of radius 1 Å is displayed at the centroid. If you pick an already picked atom, it is removed from the set, the centroid is recomputed, and the sphere is moved.

When you have picked an appropriate set of atoms, click Add Volume to add the sphere to the hypothesis as an excluded volume. Once the sphere is added, it is no longer connected in any way with the atoms that you used to define it. You can subsequently change the radius of the sphere and its location by editing the table cells.

You can also move one or more spheres by selecting them in the table, selecting Translate selected volumes in the Workspace, then dragging the volumes to the new location. Dragging excluded volumes only moves them in the viewing plane. To switch to another plane you can rotate the view, for example with the Rotate around X axis by 90 degrees and Rotate around Y axis by 90 degrees toolbar buttons.



To display excluded volumes, select them in the table. You can select multiple excluded volumes in the table or in the Workspace, with shift-click and control-click. The spheres for the selected rows are highlighted in the Workspace. After you have dismissed the Excluded

Volumes dialog box, you can view the excluded volumes in the Workspace using the toolbar button or from the Display menu.

To delete excluded volumes, select them in the table or the Workspace and click Delete.

You can also add excluded volumes to a hypothesis from the command line. Three utilities are provided that add excluded volumes based on the shape of the reference ligand, on steric clashes from inactive ligands, and on a receptor structure. For more information, see [Section 12.8 on page 136](#). The last of these tasks can also be done from the Receptor-Based Excluded Volumes panel.

6.8 Step Summary

To score hypotheses:

1. Click Score Actives.
2. Set scoring options in the Score Actives dialog box.
3. Score the hypotheses by clicking OK.

Optional tasks:

- Score inactives to generate an adjusted scoring function by clicking Score Inactives.
- Rescore the hypotheses with an adjusted scoring function by clicking Rescore.
- Export the selected hypothesis to a file, by clicking Export.
- Cluster the hypotheses, by clicking Cluster, and restrict the hypotheses shown to a representative of each cluster, by clicking View Clusters.
- Add excluded volumes to the selected hypothesis, by clicking Excluded Volumes.
- View hypotheses and alignments in the Workspace, using the toolbar buttons and the Alignments table.

To proceed to building QSAR models:

1. Select the desired hypotheses in the Hypotheses table.
2. Click Next.

To proceed to searching for matches:

1. Select the desired hypotheses in the Hypotheses table.
2. Click Search for Matches.

Building QSAR Models

Phase provides the means to build 3D QSAR models for a set of ligands that are aligned to a selection of hypotheses, and to visualize these models along with the ligand structures and the hypotheses. The QSAR models are developed from a series of ligands that have a range of activities. The usefulness of the QSAR model depends on how well the activity range is spanned, and how diverse the structures are.

In the Build QSAR Model step, you build QSAR models for the hypotheses selected in the Score Hypotheses step, using the activity data for all the available ligands. You can choose atom-based or pharmacophore-based models, select different training sets and test sets, vary the grid spacing, and visualize the model results. When you have built the models, you can use them to visualize parts of the ligands (atoms or pharmacophores) that contribute positively or negatively to activity, and to predict activities of matches to the hypotheses from a database.

When you have completed this step, you can export the hypotheses used to build the model to an external file for use with other projects, and you can continue directly to a search for matches to the hypotheses. Building QSAR models from aligned ligands without a hypothesis is described in [Chapter 9](#).

7.1 Phase QSAR Models

Phase QSAR models are 3D QSAR models, in which chemical features of ligand structures are mapped to a cubic 3D grid. The ligands are first aligned to the set of pharmacophore features in the selected hypotheses using a standard least-squares procedure, as outlined in [Section 6.1 on page 48](#). A rectangular grid is defined to encompass the space occupied by the aligned ligands. This grid divides the occupied space into N uniformly-sized cubes, typically 1 Å on each side.

The independent variables in the regression are the binary-valued occupancies (“bits”) of the cubes by structural components; the dependent variables are the activities. Because the number of bits is typically much larger than the number of training set molecules, the system is said to be highly *underdetermined*. For this reason, the regression is performed by a partial least squares (PLS) method, in which a series of models is constructed with an increasing number of PLS factors. The accuracy of the models increases with increasing number of PLS factors until over-fitting starts to occur. The independent variables can also be filtered using a t-value filter to eliminate independent variables whose regression coefficients are overly sensitive to small changes in the training set composition.

Phase offers two choices for the structural components that form the basis of the model: atoms and pharmacophore features.

In the atom-based QSAR models, the structural components of the ligands are represented by van der Waals models of the atoms in the ligands. Each atom is treated as a sphere whose radius is the van der Waals radius for the MacroModel atom type. To distinguish different atom types that occupy the same regions of space, atoms are divided into six classes:

- D—Hydrogen-bond donor (hydrogens bonded to N, O, P, S)
- H—Hydrophobic or nonpolar (C, H-C, Cl, Br, F, I)
- N—Negative ionic (formal negative charge)
- P—Positive ionic (formal positive charge)
- W—Electron-withdrawing (N, O; includes hydrogen-bond acceptors)
- X—Miscellaneous (all other types)

These classes have some correspondence to pharmacophore feature types, but atom classes are assigned using fixed internal rules, not the hypothesis feature definitions. The rules are generally consistent with the default pharmacophore feature definitions, but there are some important differences. For example, the pharmacophore feature definitions use complex rules to identify hydrophobic regions, whereas atom-based QSAR does not. Pharmacophore feature definitions can treat a given atom as part of two different pharmacophore sites, e.g., the nitrogen in a pyridine can be both an acceptor and part of an aromatic ring. Atom-based QSAR requires that each atom be assigned to only one category.

A given atom can occupy the space of one or more cubes in the grid. A cube is occupied by an atom of a particular class if the center of that cube falls within the radius of the atom. Each ligand can therefore be represented by a set of bit values (0 or 1) that indicate which cubes are occupied by atoms of each class. The independent variables used in the QSAR model are the $6N$ occupancies of the cubes and atom classes: each variable corresponds to a given cube and a given atom class, and can take the value 0 or 1.

In the pharmacophore-based QSAR models, the structural components of the ligands are represented by pharmacophore features with a specified radius. Only the pharmacophore features that are present in the hypothesis are used in the QSAR model. As for the atom-based models, the independent variables used in the QSAR model are the mN occupancies of the cubes by the m pharmacophore feature types: each variable corresponds to a given cube and a given feature type, and can take the value 0 or 1.

Once the occupancies are determined, a partial least-squares (PLS) regression analysis is applied to these binary-valued variables to obtain the QSAR model. The variables can be filtered. Technical details on the regression analysis and the statistical measures used in the QSAR model are given in [Appendix A](#).

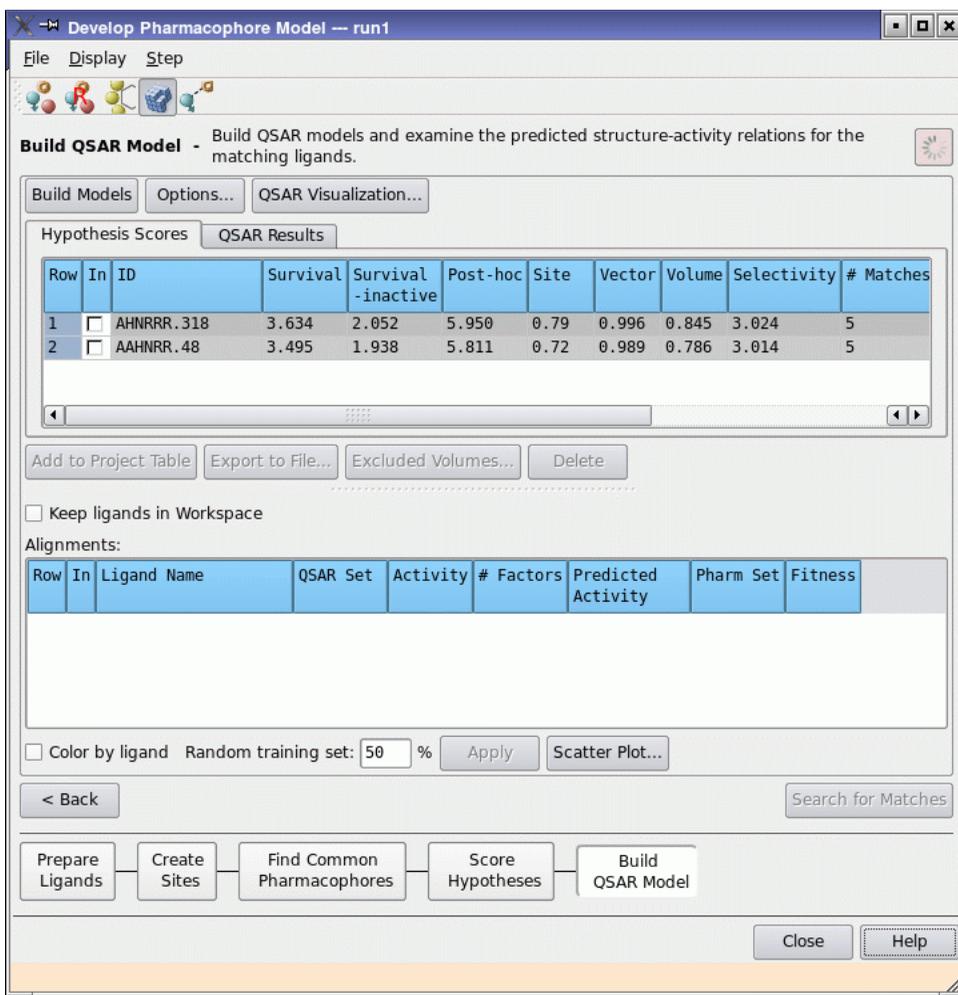


Figure 7.1. The initial view of the Build QSAR Model step.

Atom-based models are useful when features other than the pharmacophores are important to activity, such as steric clashes. However, their performance generally decreases as the diversity of the training set increases. If the structures in the training set contain a relatively small number of rotatable bonds and some common structural framework, then an atom-based model may work quite well.

Pharmacophore-based models assume that the activity is explained entirely by the pharmacophore model itself, and therefore cannot predict activities where other features are important to activity, such as steric clashes. If the structures in the training set are highly flexible or if

they exhibit significant chemical diversity, a pharmacophore-based model may be more appropriate.

If the choice of model is not clear, it is easy to create both types of models and examine the test set statistics to see which approach produces models with the most predictive power.

7.2 Choosing a Training Set and a Test Set

The first task in this step is to choose a training set and a test set, and exclude ligands that you do not want in either set. To display the ligands in the Alignments table, click the In column for any hypothesis in the Hypothesis Scores table. It does not matter which hypothesis you select, because all ligands are listed for all hypotheses. Initially, all ligands are included in the training set, and all rows are colored dark gray, which indicates that there is no corresponding QSAR model. The data columns are empty, and are filled in after the QSAR models are built.

To change the set membership of an individual ligand, click in the QSAR Set column for the ligand. The membership cycles between training, test, and blank, the last of which means that the ligand is excluded from both sets—that is, it is not used. To change the set membership for a group of ligands, select the ligands in the table using shift-click or control-click, then control-click in the QSAR Set column for any of the ligands.

You can select a random fraction of the ligands for the training set by entering a percentage in the Random training set text box and clicking Apply. The specified percentage of ligands is selected at random from the existing training and test sets and assigned to the training set. The remainder are assigned to the test set. Ligands that are in neither set are not used in the selection. The seed for the random selection can be set as an option—see the next section.

7.3 Specifying Options for the QSAR Model

If you want to select the model type, set the grid spacing, choose the maximum number of PLS factors, or specify a seed for random selection of the training set, you can do so in the Build QSAR Model - Options dialog box, which you open by clicking Options.

If you select the training set randomly, you may want to do this in a reproducible way. By default, the random seed changes each time a random training set is selected, so you get a different training set each time you click Apply in the Build QSAR Model step. However, if you change the value in the Random seed text box to any positive integer, you can ensure that the same random training set will be created each time you click Apply. The default value of zero ensures that the assignment is always random.

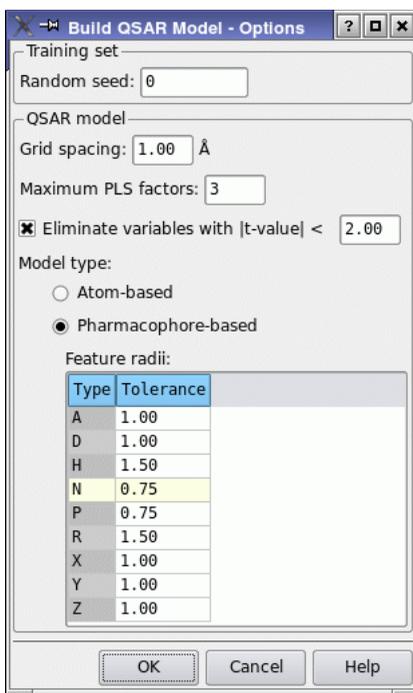


Figure 7.2. The Build QSAR Model - Options dialog box.

The QSAR model partitions the space occupied by the ligands into a cubic grid. Any structural component can occupy part of one or more cubes. A cube is occupied by an atom or a feature if its centroid is within the radius of the atom or feature. You can set the size of the cubes by changing the value in the Grid spacing text box. The allowed range is 0.5 Å to 2.0 Å.

The regression is done by constructing a series of models with an increasing number of PLS factors. The accuracy of the models increases with increasing number of PLS factors until over-fitting starts to occur. The maximum number of PLS factors is $N/5$, where N is the number of ligands. You can adjust this value in the Maximum PLS factors text box. As a general rule, you should stop adding factors when the standard deviation of regression is approximately equal to the experimental error. This point usually occurs at 2 or 3 factors in Phase.

The independent variables can be filtered using a t-value filter. In this scheme, the standard deviation $\sigma(\beta_i)$ of the regression coefficients β_i is estimated from leave- n -out PLS models, and variables whose coefficients have a t-value $\beta_i/\sigma(\beta_i)$ less than a threshold value are eliminated. To apply the filter, select Eliminate variables whose |value| < x , and enter the threshold in the text box. The advantage of the t-value filter is that it eliminates uninformative variables,

reduces model complexity and increases model generality. It does, however, decrease the amount of information. One suggested approach for the use of this filter is to identify the most predictive models without the filter, choose the lowest maximum number of factors that is predictive, then apply a statistically significant but conservative filter. The default value of 2.0 is usually adequate.

To select the type of model, choose Atom-based or Pharmacophore-based. In the pharmacophore-based model, the features are represented by spheres whose radii is given in the Tolerance column of the Feature radii table. The features are those defined in the Create Sites step. You can change the feature radii by editing the values in the Tolerance column.

7.4 QSAR Model Results

After you have selected the test set and the training set and set any options, click **Build Models**. A **Start** dialog box is displayed, in which you can adjust job settings. When you click **Start**, the job is run. The predicted activities are displayed in the Alignments table (see [Table 7.1](#)), and parameters for the quality of the fit are displayed in the QSAR results table (see [Table 7.2](#)). These parameters are defined in [Section A.3 on page 205](#). Each row presents a regression model with a given number of PLS factors.

Table 7.1. Description of the Alignments table columns.

Column	Description
In	Inclusion status of the ligand. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. You can include and exclude ligands with click, shift-click, and control-click.
Ligand Name	The name of the ligand.
QSAR Set	Indicates whether a ligand is in the training set, the test set, or neither (the ligand is ignored). The column is blank if the ligand is ignored. Click the column repeatedly to cycle through the three possible states.
Activity	The ligand's activity. You can alter the activity values by directly editing the table cells.
# Factors	Number of PLS factors used for the QSAR model.
Predicted Activity	Activity predicted by the QSAR model. The number of rows in this column for each ligand is equal to the number of PLS factors specified in the Build QSAR Model - Options dialog box. Each row contains the prediction from a model containing the number of PLS factors indicated in the # Factors column.
Pharm Set	Status of a ligand in the set used to build the pharmacophore model.
Fitness	Fitness score from the scoring step.

Table 7.2. Description of the QSAR results table columns.

Column	Description
In	Inclusion status of the hypothesis. The diamond has a cross in it if the hypothesis is included in the Workspace, and is empty if the hypothesis is excluded. You can include and exclude hypotheses with click, shift-click, and control-click.
# Factors	Number of factors in the partial least squares regression model.
SD	Standard deviation of the regression.
R-squared	Value of R^2 for the regression.
F	Variance ratio. Large values of F indicate a more statistically significant regression
P	Significance level of variance ratio. Smaller values indicate a greater degree of confidence.
Stability	Stability of the model predictions to changes in the training set composition. Maximum value is 1. This statistic can be used to compare models from different hypotheses.
RMSE	Root-mean-square error.
Q-squared	Value of Q^2 for the predicted activities.
Pearson-R	Pearson R value for the correlation between the predicted and observed activity for the test set.

The Alignments table functions the same as for the Score Hypotheses step, and you can select Keep ligands in Workspace to keep the same selection of ligands in the Workspace when you switch hypotheses. You can also select Color by ligand, to color the ligands in the Workspace and the rows in the Alignments table with a unique color. This helps to identify ligands when multiple ligands are displayed.

If you have more than one PLS factor in the model, you should examine the models produced to select the best model. For example, you can examine the predicted activities for the test set, and see at what point they begin to degrade, or you can compare the training set errors with the experimental uncertainty in the data.

It is important to recognize that there is no single statistic that unequivocally determines which model is best. The battery of statistics provided by Phase should be considered in totality, and some measure of common sense must be applied. For example, if the IC_{50} values are accurate to a multiplicative factor of 2, the corresponding $-\log[IC_{50}]$ are only accurate to $\log(2)$. So if the SD statistic is smaller than this experimental uncertainty, then the data are clearly being over-fit, and the model is bound to yield spurious predictions on certain molecules outside the training set, even if the test set predictions appear satisfactory.

Build QSAR Model - Build QSAR models and examine the predicted structure-activity relations for the matching ligands.

Build Models Options... QSAR Visualization...

Hypothesis Scores QSAR Results

Row	In	ID	# Factors	SD	R-squared	F	P	Stability	RM
1	<input checked="" type="checkbox"/>	AHNRRR.318	1	0.6869	0.7319	62.8	5.061e-08	0.603	0.
			2	0.4741	0.8778	79	9.067e-11	0.5949	0.
			3	0.3267	0.9446	119.4	2.36e-13	0.451	0.
2	<input type="checkbox"/>	AAHNRR.48	1	0.6301	0.7744	78.9	6.775e-09	0.6457	1.
			2	0.3642	0.9279	141.6	2.732e-13	0.5418	1.
			3	0.1733	0.9844	442.3	3.949e-19	0.518	0.

Add to Project Table Export to File... Excluded Volumes... Delete

Keep ligands in Workspace

Alignments for hypothesis AHNRRR.318:

Row	In	Ligand Name	QSAR Set	Activity	# Factors	Predicted Activity	Pharm Set	Fitness
1	<input type="checkbox"/>	train01	training	8.886	1	8.13	active	2.34
					2	8.43		
					3	8.74		
2	<input checked="" type="checkbox"/>	train02	training	7.721	1	7.91	active	3.00
					2	7.81		
					3	7.77		

Color by ligand Random training set: 50 % Apply Scatter Plot...

< Back Search for Matches

Prepare Ligands Create Sites Find Common Pharmacophores Score Hypotheses Build QSAR Model

Close Help

Figure 7.3. The Build QSAR Model step showing results of model-building.

One way of assessing the results is to plot the experimental activities against the predicted activities. To do so, click Scatter Plot. The Phase QSAR - Scatter Plot dialog box is displayed, in which you can choose the number of PLS factors for the prediction, which ligands to plot results for, from All, Selected, Training set and Test set, and to draw the 45 degree line of perfect fit. When you click Plot in this dialog box, the scatter plot is displayed in the Plot XY panel (the same panel as used for plotting data from the Project Table), with the R^2 value shown in the plot as well.

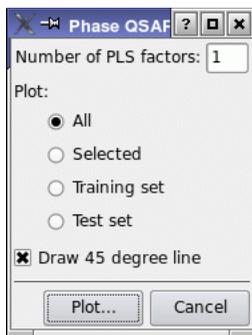


Figure 7.4. The Phase QSAR - Scatter Plot dialog box.

Each time you create a plot, it is added to the Plot XY panel, and the previous plot is hidden. You can redisplay previous plots again with the tools in the panel—see [Chapter 10](#) of the *Maestro User Manual* for more information.

The QSAR models are stored with the run, and can be used in the database search to predict activities for the hits. If you change the training set or the model parameters and build new QSAR models, these models overwrite the previous models. To save QSAR models for later use, you can export them with the hypothesis (click Export). The QSAR models are stored with the same file stem as the hypothesis data. If you intend to export more than one QSAR model for a given hypothesis, you must provide a different name for each copy of the hypothesis, or store each in a different location.

You can also add excluded volumes to the hypothesis before exporting it. The Excluded Volumes button opens the same dialog box as in the previous step. See [Section 6.7 on page 59](#) for more information.

7.5 Viewing the QSAR Model

Once you have a QSAR model for a hypothesis, you can examine its 3D characteristics by displaying the QSAR model, the ligands, and the hypothesis in the Workspace. Each of these can be displayed independently. To display the hypothesis, the excluded volumes, or the QSAR model, click the appropriate toolbar button or choose the appropriate item from the Display menu. The buttons are described below:



View Hypothesis

Displays the selected hypothesis in the Workspace, as a spatial arrangement of feature symbols. For a description of these symbols, see [Table 4.1 on page 30](#).



View Hypothesis Labels

Displays feature labels for the selected hypothesis in the Workspace.



View Excluded Volumes

Displays excluded volumes for the selected hypothesis in the Workspace.



View QSAR Model

Displays the QSAR model for the selected hypothesis.

When you display the QSAR model, the cubes that represent the model are displayed in the Workspace, colored according to the sign of their coefficient values, which by default is blue for positive coefficients and red for negative coefficients. Positive coefficients indicate an increase in activity, negative coefficients a decrease. You can use the visualization of the coefficients to identify characteristics of ligand structures that tend to increase or to decrease activity.

In addition to viewing the model as a whole, you can examine the spatial distribution of contributions to the model by ligand, and by atom class or pharmacophore type either separately or in combination. These capabilities are available in the QSAR Visualization Settings panel, which you open by clicking QSAR Visualization Settings. The visualization tools provided in this panel help you to identify features of ligand structures that are likely to contribute to higher or lower activity.

For example, if you select Workspace ligands under View volume occupied by and choose an atom type from the View effects from list, you can include the ligands in the Workspace one by one (click the In column of the Alignments table) and see which parts of all ligands have a positive or a negative contribution to the activity due to the chosen atom type. This might give a clue to what functional groups are desirable or undesirable at certain positions in a molecule.

The QSAR Visualization Settings panel also has controls for the display of the model. For example, if you want to filter out cubes that have small coefficients, and therefore do not affect the activity much, you can use the sliders for the positive and negative coefficients in the Regression Coefficient Visualization section. You can also change the cube colors, and view effects from atom or pharmacophore type classes individually or in combination.

The three sections of the QSAR Visualization Settings panel are described below.

View Effects From

To view effects from one or more classes of atoms or pharmacophore types, choose the classes from the list. You can select multiple classes with shift-click and control-click. If you select Combine effects, the effects from all the selected classes are visualized; if you deselect this option, the effects from each class are visualized separately.

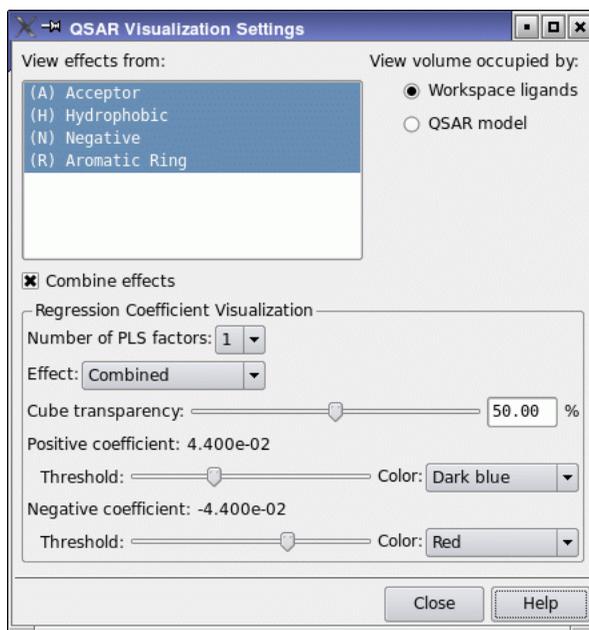


Figure 7.5. The QSAR Visualization Settings panel.

View volume occupied by

The two options under this heading allow you to choose whether to view the volume occupied by the QSAR model or by the ligands that are included in the Workspace.

- **Workspace ligands**— Display the cubes of the QSAR model grid that are occupied by the ligands that are in the Workspace.
- **QSAR Model**— Display all the cubes that are occupied in the QSAR model.

Regression Coefficient Visualization

This section provides controls for the choice of QSAR model and the display of its coefficients. For the coefficient sliders, cubes that have coefficients that are smaller in magnitude than the threshold are not displayed. This means that the coefficients that have the maximum magnitude are always displayed.

- **Number of PLS factors**—Select the number of PLS factors from the list to determine which QSAR model is displayed.
- **Effect**—Choose the effect for which the transparency, thresholds, and colors are to be set. The Combined choice applies when you select the Combine effects option.

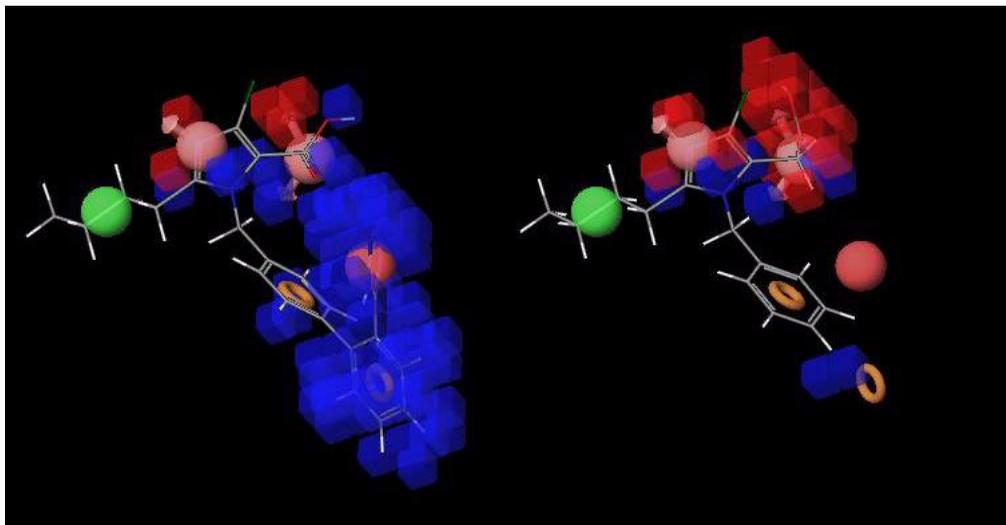


Figure 7.6. QSAR model for an active ligand (left) and an inactive ligand (right).

- Cube transparency—Adjust the transparency of the cubes from 0% (opaque) to 100% (transparent) for the selected effect.
- Positive coefficient—Adjust the threshold for the display of positive regression coefficients with the Threshold slider, and choose the cube color from the Color option menu, for the selected effect.
- Negative coefficient—Adjust the threshold for the display of negative regression coefficients with the Threshold slider, and choose the cube color from the Color option menu, for the selected effect.

7.6 Continuing from the Build QSAR Model Step

When you have finished building QSAR models, you can close the Build Pharmacophore Model panel, export the hypothesis with its QSAR model, add aligned ligands and their properties to the Project Table or export them to a file, continue directly to searching for matches, or return to the previous step and select another set of hypotheses with which to build QSAR models.

To export the hypothesis, click Export. The Export Hypothesis dialog box is displayed, in which you can navigate to the desired location and provide the file name. The extension is removed from the file name, so it is not necessary to provide an extension. This means that you should not provide a name like AAHNRR.26 because the .26 will be removed. If you want to

use a hypothesis name as it appears in the table, you should replace the period with another character, such as an underscore.

To add aligned ligands to the Project Table, select the ligands in the Alignments table, then right-click in the table and choose **Add Alignments to Project Table** from the shortcut menu. The ligands are added to the Project Table as an entry group, with all the properties added by Phase, including the activity predictions from the QSAR models. The entries are selected and the first entry is included in the Workspace.

Likewise, to export aligned ligands to a Maestro file, select the ligands in the Alignments table, then right-click in the table and choose **Export Alignments to File** from the shortcut menu. A file chooser labeled **Export Alignments** opens, in which you can navigate to the desired location and enter the file name.

To search for matches to a hypothesis, click **Search for Matches**. This button opens the **Find Matches to Hypothesis** panel, in which you can start a search for structures that match a hypothesis. All selected hypotheses from the **Build Pharmacophore Model** panel are loaded by default, and the first of these is selected in the **Find Matches to Hypothesis** panel.

To build a QSAR model for another set of hypotheses, click **Back**, or click **Score Hypotheses** in the Guide. You can then select hypotheses, and click **Next** or **Build QSAR Model** in the Guide to return to this step. When you do so, you are prompted to create a new run in which to store the hypotheses and the QSAR models. Each set of QSAR models is stored with its set of hypotheses in a separate run, so you can generate a QSAR model for as many hypotheses as you want. You can only store one set of QSAR models for a given hypothesis in the run, but you can always export a hypothesis (click **Export Hypothesis**) with the current model if you want to store more than one QSAR model for a given hypothesis, or create a new run.

7.7 Step Summary

To build QSAR models:

1. Display the ligands in the Alignments table.
2. Select the training set and the test set.
3. (Optional) Choose a model and set parameters in the **Build QSAR Model - Options** dialog box.
4. Click **Build Models**.

To export alignments:

1. Select the ligands in the Alignments for hypothesis table.
2. Right-click in the table and choose the destination from the shortcut menu.
3. If exporting to file, navigate to the location and name the file.

To proceed to searching for matches:

1. Select the desired hypotheses in the Hypotheses table.
2. Click Search for Matches.

Building and Editing Hypotheses

In the Develop Pharmacophore Model workflow, hypotheses are generated from a set of active molecules, automatically taking into account common features and excluding features that are not common. The process does not directly take into account any explicit knowledge about the binding of a particular molecule to the receptor. (Of course, you can display the possible hypotheses and select the ones that fit with your understanding of the binding.)

Phase provides the means to use knowledge of ligand binding directly in the construction of a hypothesis from a single molecule (the reference ligand). For this molecule, Phase generates all possible pharmacophore sites from a set of pharmacophore features. You then select the sites that are included in the hypothesis. The features are the same as those used in the Develop Pharmacophore Model workflow, and can be supplemented with custom features or custom patterns in the same way. You can also edit hypotheses that were exported from the Develop Pharmacophore Model workflow.

You can add excluded volumes to the hypothesis that you construct. QSAR models, however, require a set of ligands with activities, which are not available in this workflow.

8.1 The Manage Hypotheses Panel

The Manage Hypotheses panel provides controls for creating, editing, deleting, importing, and exporting hypotheses based on an individual structure. To open the Manage Hypotheses panel, choose Manage Hypotheses from the Phase submenu of the Applications menu.

The panel consists of a toolbar, a table of hypotheses, and a set of action buttons. The toolbar contains three buttons, which are the same as in the Develop Pharmacophore Model panel, and allow you to view the excluded volumes and the intersite distances and angles:



View Excluded Volumes

View excluded volumes for the hypothesis in the Workspace as a set of spheres.



View Hypothesis Labels

Displays feature labels for the selected hypothesis in the Workspace.



View Site Measurements

Opens the View Site Measurements panel, in which you can select the intersite distances and angles you want to display in the Workspace.

**View Matching Tolerances**

View feature-matching tolerances as semitransparent spheres whose radius is proportional to the tolerance.

The Hypotheses table lists the hypotheses that are available for editing. These hypotheses are stored as project entries, and can also be viewed in the Project Table. The Hypotheses table displays a filtered view of the Project Table data that includes only the entries from the Project Table that have hypotheses associated with them, and only the properties that are relevant to the hypothesis. (Entries that have hypothesis data are indicated by an H button in the Hyp column of the Project Table.) The structure that is stored in the project entry is the reference ligand for the hypothesis. For hypotheses without a reference ligand, dummy atoms are added to the entry at the site point locations.

You can select a single row of the table, for editing, deleting, adding excluded volumes, or export. The columns of the table are described in [Table 8.1](#). The table is noneditable.

Table 8.1. Description of the Hypotheses table.

Column	Description
In	Inclusion status of the reference ligand and its hypothesis data. The diamond has a cross in it if the ligand is included in the Workspace, and is empty if the ligand is excluded. You can include and exclude ligands with click, shift-click and control-click.
Title	The title of the project entry for the hypothesis. You can edit this column to change the title.
Hypothesis ID	Identifier of the hypothesis. For new hypotheses, the identifier is constructed automatically from the feature letters and the entry ID of the reference ligand. For hypotheses that were exported from a pharmacophore model development run, the identifier is the identifier from the run.
Entry ID	Project entry ID for the hypothesis
Entry Name	Project entry name for the hypothesis.
phase activity	The activity of the reference ligand, as stored in the Phase run from which the hypothesis originated.
QSAR	Indicates whether a hypothesis has an associated QSAR model.
Excluded Volumes	Indicates whether a hypothesis has associated excluded volumes.
Phase Run Name	The name of the Phase run from which the hypothesis originated.
Hypothesis Date	Date when the hypothesis was last modified.
#Sites	Number of sites in the hypothesis

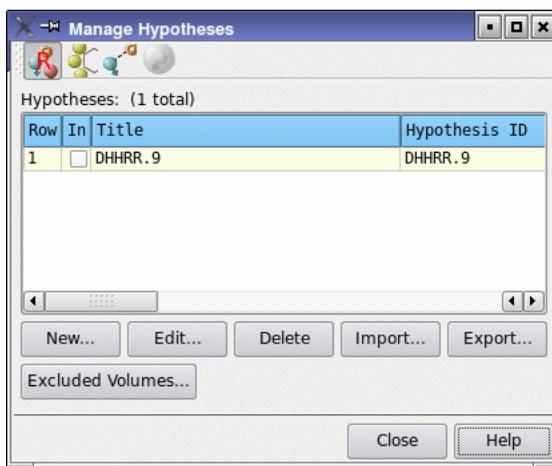


Figure 8.1. The Manage Hypotheses panel.

The Hypotheses table has a shortcut menu, which opens when you right-click in the table. The menu items and their actions are described in [Table 8.2](#).

Table 8.2. Hypothesis table shortcut menu items.

Item	Description
Display	Control the display of the selected hypotheses and their reference ligands. Opens a submenu from which you can choose Hypothesis Only, Atoms Only, or Both.
Color By	Controls the color of the selected hypotheses and reference ligands. Opens a submenu from which you can choose Atom and Site Type, to color the atoms by atom type and the hypotheses by the feature type, as described in Table 4.1 on page 30 ; or Entry, to color the reference ligands and hypotheses with a different uniform color for each entry.
View	Controls the view of excluded volumes and intersite distances and angles. Opens a submenu from which you can choose the objects to display. Same as clicking the toolbar buttons.
Align	Align the selected hypotheses. The stationary ligand in the alignment is the ligand whose hypothesis you right-clicked.
Export	Export the selected hypotheses to external files. Opens a directory chooser in which you can navigate to the desired directory. The files are named with the hypothesis ID as the stem.
Find Matches	Opens the Find Matches to Hypothesis panel, to find matches for the hypothesis you right-clicked.

The action buttons, with the exception of the Delete button, open dialog boxes in which you can make the appropriate choices to perform the action. The buttons are described in [Table 8.3](#).

Table 8.3. Action buttons in the Edit Hypotheses panel

Button	Action
New	Create a new hypothesis. Opens the Choose Reference Ligand dialog box, then the New Hypothesis dialog box.
Edit	Edit the selected hypothesis. Opens the Edit Hypothesis dialog box.
Delete	Delete the selected hypothesis from the project. This action removes the project entry, but does not remove hypotheses that are stored externally or in Phase runs.
Import	Import a hypothesis from disk. Opens the Import Hypothesis dialog box, in which you can navigate to the desired hypothesis. The file you select can be any of the hypothesis-related files.
Export	Export the selected hypothesis to disk. Opens the Export Hypothesis dialog box, in which you can navigate to a location to save the hypothesis.
Excluded Volumes	Add excluded volumes to the selected hypothesis. Opens the Excluded Volumes dialog box. See Section 6.7 on page 59 for more information.

8.2 Creating New Hypotheses

New hypotheses can be created from an existing structure in two ways:

- You can use a set of pharmacophore features that Maestro uses to identify all the possible pharmacophore sites in the reference ligand. You then choose which sites you want to include in the hypothesis. These hypotheses are termed *ligand-based* hypotheses.
- You can place pharmacophore features at will in the Workspace, in relation to a reference ligand. The reference ligand is only there as a guide, and is deleted once the hypothesis is created. These hypotheses are termed *freestyle* hypotheses.

The hypothesis is created from an entry in the Project Table, which is converted into a hypothesis, with the possible loss of information. To ensure that you preserve the original entry, it is advisable to duplicate the entry before creating a hypothesis. This task is included in the procedures given below.

You can also edit the feature definitions used for either kind of hypothesis in the Edit Features dialog box—see [Section 4.2 on page 30](#) for more information.

8.2.1 Ligand-Based Hypotheses

To create a ligand-based hypothesis, follow the steps below.

1. Select the entry you want to use for the reference ligand in the Project Table.
2. From the Entry menu, choose Duplicate, then In Place (or type CTRL+D).

The entry is duplicated and placed below the original. It is also selected and included in the Workspace.

3. Do one of the following:
 - Choose Applications > Phase > Create Hypothesis in the main window.
 - In the Manage Hypotheses panel, click New.

The Choose Reference Ligand entry chooser is displayed. The table in the center lists entries from the Project Table. You can control which entries are displayed by choosing an item from the Choose entry from option menu. You can also sort the table by clicking one of the column headers or by clicking Sort by Project Table order.

4. Select the duplicated entry from the list.

The entry name appears in the Name text box. You should ensure that the structure in the entry is a 3D, all-atom structure. If it is not, the pharmacophore features are likely to be incorrectly assigned.

5. Click Choose.

The Choose Reference Ligand entry chooser closes and the New Hypothesis dialog box opens, with the Ligand-based tab displayed (Figure 8.5).

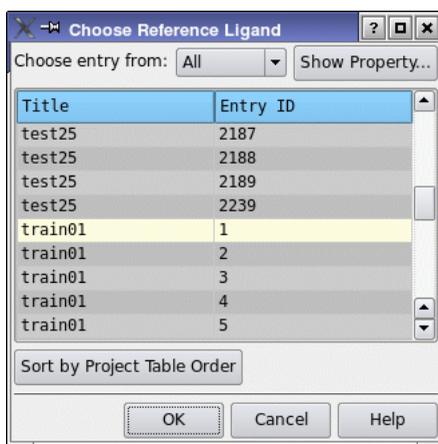


Figure 8.2. The Choose Reference Ligand dialog box.

This tab has two site lists: one of available sites in the reference ligand, and one of sites selected for the hypothesis, which is initially empty. You can choose which sites are displayed in the Workspace by selecting a Mark sites option. By default, all sites are marked.

6. Select the sites you want to include in your hypothesis from the Ligand sites list.

You can select multiple sites with shift-click and control-click. Once you have selected sites, the Add button becomes available.

7. Click Add.

The selected sites are added to the Hypothesis sites list. You can also select sites one by one and add them, and you can remove sites from the list. The Ligand sites list does not change when you add or remove sites.

8. When you have selected the desired sites, click OK.

The New Hypothesis dialog box closes, and the hypothesis is displayed in the Workspace and added to the Hypotheses table of the Manage Hypotheses panel.

8.2.2 Freestyle Hypotheses

To create a new freestyle hypothesis, first follow [Step 1](#) through [Step 5](#) of the procedure above for ligand-based hypotheses. You can select sites in the Ligand-based tab before proceeding to freestyle site addition if you wish.

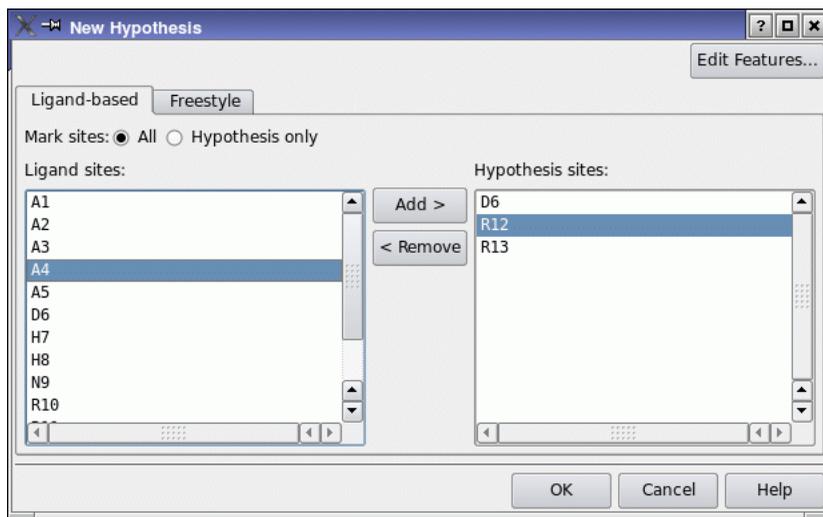


Figure 8.3. The New Hypothesis dialog box, Ligand-based tab.

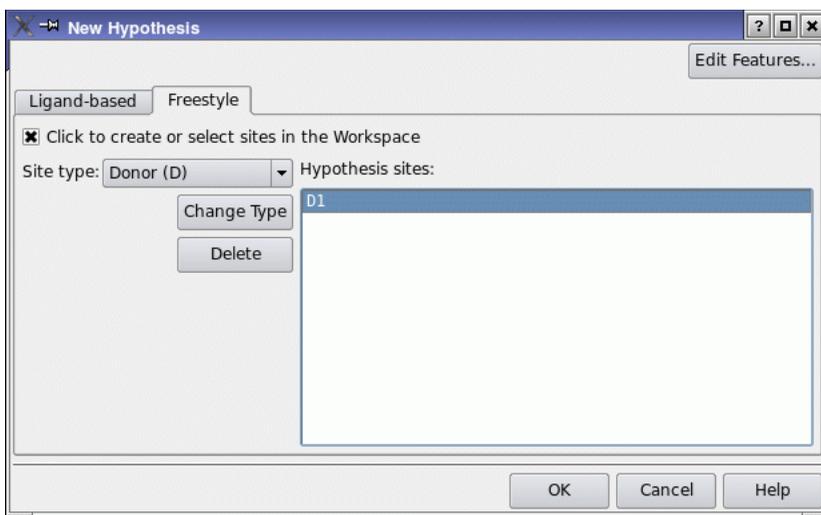


Figure 8.4. The New Hypothesis dialog box, Freestyle tab.

When you are ready to add free sites, click the Freestyle tab. This tab has a list of hypotheses sites, which is initially empty, and controls for choosing and changing the feature type, and placing sites in the Workspace. You can add sites at arbitrary locations, or on atoms.

To add sites in arbitrary locations:

1. Select Click to create or select sites in the Workspace.
2. Select the desired feature type from the Site type option menu.
3. Click in the Workspace where you want to place the feature.

The site is placed in the xy plane (the plane of the screen) at $z=0$. You will probably need to rotate the ligand and move the site to position it precisely. To do so:

- a. Click on the site in the Workspace.

The site turns red to indicate that it is selected.

- b. Drag with the middle mouse button to rotate the structure.

You can also use the toolbar buttons to rotate around the x or y axis by 90° .



- c. Drag with the right mouse button to move the site.

4. Repeat [Step 2](#) and [Step 3](#) for each free site you want to add.

To add sites to atoms:

1. Select Click to attach site to the picked atom.
2. Select the desired feature type from the Site type option menu.
3. Click the atom in the Workspace on which you want to place the feature.
4. Repeat [Step 2](#) and [Step 3](#) for each site you want to add to an atom.

Once you have added sites, you can delete sites or change the site type.

- To delete sites, select them in the Hypothesis sites list and click Delete.
- To change the site type, select the sites in the Hypothesis sites list, choose the new site type from the Site type option menu, and click Change Type.

When you have selected all the sites, click OK. A warning is displayed, that the operation will remove any association with the reference ligand. When you click OK in the warning dialog box, the reference ligand is removed from the entry, which becomes the hypothesis.

8.3 Editing Existing Hypotheses

As well as creating new hypotheses, you can edit hypotheses, including hypotheses that were exported from the Develop Pharmacophore Model panel. You cannot edit hypotheses from a pharmacophore model development run directly: you must export them first, then you can edit the exported version.

To edit an existing hypothesis, select the hypothesis in the Hypotheses table, and click Edit. The Edit Hypothesis dialog box is displayed. This dialog box is identical to the New Hypothesis dialog box. It has two tabs, Ligand-based and Freestyle. In addition to the two tabs, there is an Edit Features button, which opens the Edit Features dialog box. This dialog box allows you to edit the feature definitions, and is described in detail in [Section 4.2 on page 30](#).

The use of this dialog box to edit ligand-based and freestyle hypotheses is described in the next two subsections. You can edit ligand-based hypotheses in the Freestyle tab, but if you do so and save the changes, the reference ligand is discarded, and the hypothesis becomes a freestyle hypothesis.

8.3.1 Ligand-Based Hypotheses

In the Ligand-based hypotheses tab, you can add or remove sites from an existing ligand-based hypothesis. This tab is unavailable if you are editing a freestyle hypothesis.

You can choose which sites are displayed in the Workspace using the Mark sites options. By default, the hypothesis sites are marked.

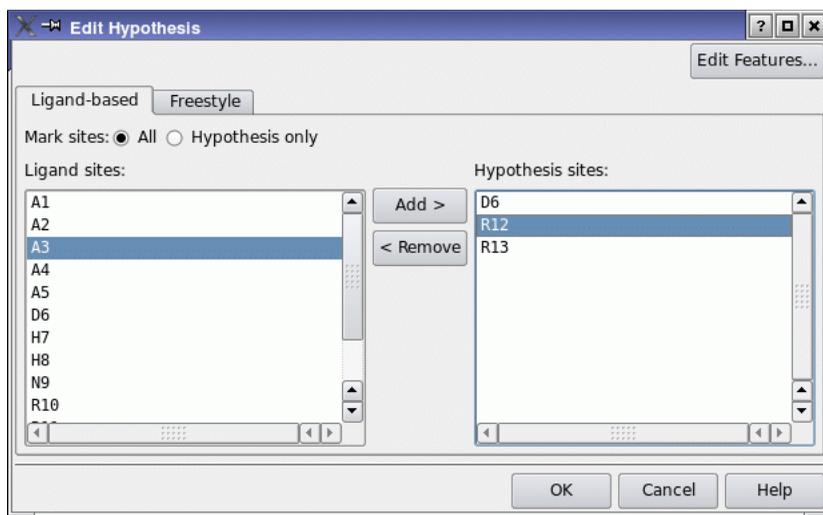


Figure 8.5. The Edit Hypothesis dialog box, Ligand-based tab.

- To add sites to the hypothesis, select the sites you want to add in the Ligand sites list, and click Add.

The selected sites are added to the Hypothesis sites list. You can select multiple sites by shift-clicking and control-clicking. The Add button is only available when you have selected one or more sites in the Ligand sites list.

- To remove sites from the hypotheses, select the sites you want to remove from the Hypothesis sites list, and click Remove.

The selected sites are removed from the Hypothesis sites list. You can select multiple sites by shift-clicking and control-clicking. The Remove button is only available when you have selected one or more sites in the Hypothesis sites list.

When you have made the desired changes, click OK. The Edit Hypothesis dialog box closes, and the changes to the hypothesis are applied.

8.3.2 Freestyle Hypotheses

In the Freestyle tab, you can change the feature type for any site, add a site, move a site, and delete a site. You can select sites either by clicking on them in the Workspace or selecting them in the Hypotheses sites list. Selected sites are colored red in the Workspace.

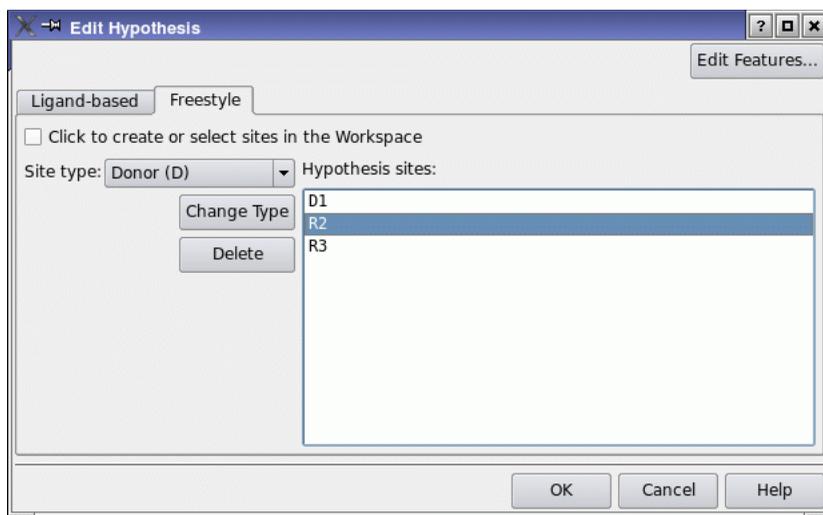


Figure 8.6. The Edit Hypothesis dialog box, Freestyle tab.

To change the feature type for one or more sites:

1. Choose the desired type from the Site type option menu.
2. Select the sites in the Hypothesis sites list or the Workspace.
3. Click Change Type.

To add a site in an arbitrary location:

1. Choose the desired type from the Site type option menu.
2. Select Click to create or select sites in the Workspace.
3. Click in the Workspace where you want to place the feature.

To add a site to an atom:

1. Choose the desired type from the Site type option menu.
2. Select Click to attach site to the picked atom.
3. Click on the atom in the Workspace where you want to place the feature.

To reposition a site:

1. Select Click to create or select sites in the Workspace.
2. Click on the site in the Workspace.

3. Drag the site using the left mouse button.

The site is moved in the *xy* plane (the plane of the screen).

4. As needed, rotate the Workspace contents with the middle mouse button, then drag the site again.

To delete sites:

1. Select the sites in the Hypothesis sites list or the Workspace.
2. Click Delete.

Building QSAR Models from Ligands

If you have a set of aligned ligands, you can build Phase 3D QSAR models for these ligands without having a hypothesis, in the Individual QSAR Model panel. The approach is identical to that described in [Chapter 7](#). Phase QSAR models are described in detail in [Appendix A](#). Having an independent panel enables you to build QSAR models without having to proceed through all the stages of developing a pharmacophore model. You can also use the results to predict activities for other molecules, display a scatter plot of predicted against experimental activities, and add the QSAR model to an existing hypothesis.

To open this panel, from the Applications menu choose Phase, then Individual QSAR Model. The panel has many features in common with the Build QSAR Model step of the Develop Pharmacophore Model panel.

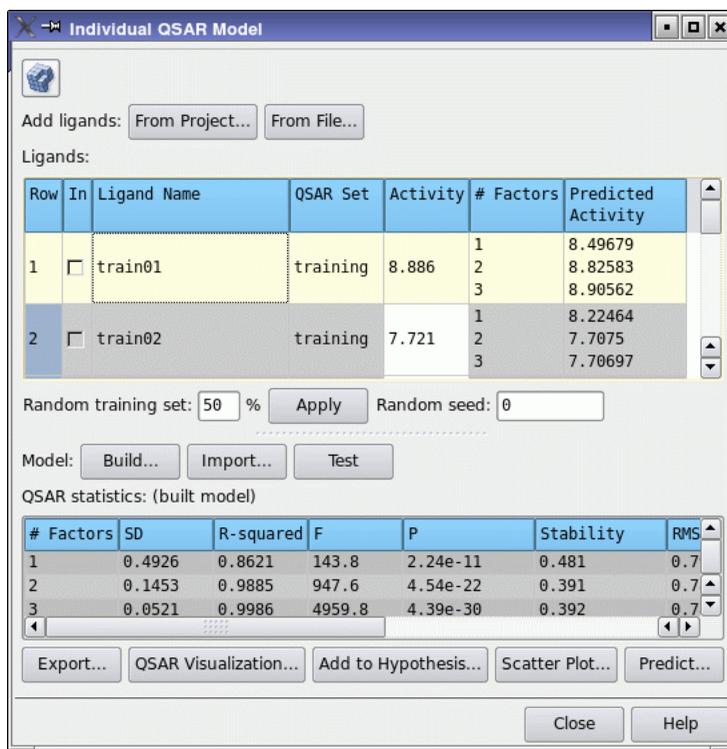


Figure 9.1. The Individual QSAR Model panel.

9.1 Adding Ligands

The first step is to select the ligands to use. You can add ligands to the set to be used for the QSAR model from two sources, by clicking one of the Add ligands buttons:

- From Project—Opens the Add From Project dialog box, in which you can choose a set of entries and select an activity property, converting it into the appropriate units if need be.
- From File—Opens a file selector, in which you can navigate to and select the file. When you click OK, the Choose Activity Property dialog box opens, in which you can select an activity property, converting it into the appropriate units if need be.

These two dialog boxes are the same as in the Prepare Ligands step of the Develop Pharmacophore Model panel—see [Section 3.1 on page 14](#). The ligands you add must, however, be fully prepared 3D structures that are properly aligned. No facility is provided in this panel for preparing the structures or aligning the ligands.

The ligands are displayed in the Ligands table when they are added. This table contains the list of ligands. When the ligands are first read, all ligands are included in the training set, and the # Factors and Predicted Activity columns are empty. These columns are added after the QSAR model is built. The table columns are the same as the Alignments table in the Build QSAR Model step of the Develop Pharmacophore Model panel—see [Table 7.1 on page 68](#).

9.2 Choosing a Training Set and a Test Set

The next task in this step is to choose a training set and a test set, and exclude ligands that you do not want in either set. To display the ligands in the Workspace, click the In column for the ligands in the Ligands table. Initially, all ligands are included in the training set. The data columns are empty, and are filled in after the QSAR models are built.

To change the set membership of an individual ligand, click in the QSAR Set column for the ligand. The membership cycles between training, test, and blank, the last of which means that the ligand is excluded from both sets—that is, it is not used. To change the set membership for a group of ligands, select the ligands in the table using shift-click or control-click, then control-click in the QSAR Set column for any of the ligands.

You can select a random fraction of the ligands for the training set by entering a percentage in the Random training set text box and clicking Apply. The specified percentage of ligands is selected at random from the existing training and test sets and assigned to the training set. The remainder are assigned to the test set. Ligands that are in neither set are not used in the selection.

If you select the training set randomly, you may want to do this in a reproducible way. By default, the random seed changes each time a random training set is selected, so you get a different training set each time you click Apply. If you change the value in the Random seed text box to any positive integer, you can ensure that the same random training set will be created each time you click Apply. The default value of zero ensures that the assignment is always random.

9.3 Building and Testing the Model

Once you have chosen the training and test sets, click Build to build the QSAR models. The Build Model dialog box is displayed, in which you can specify the grid spacing and the maximum number of PLS factors. When you have done so, click OK to build the model. When the results are returned, the # Factors and Predicted Activity columns are filled in for both the training set and the test set, and the QSAR statistics table is filled in. This table is the same as the QSAR results table in the Build QSAR Model step of the Develop Pharmacophore Model panel—see [Table 7.2 on page 69](#).

If you have ligands that you did not include in the test set, you can include them and click Test to calculate the predicted activity and update the QSAR statistics for the test set.

9.4 Examining and Using the Model

There are several ways in which you can assess the accuracy of the model.

- Examine the QSAR statistics, which are described in [Table 7.2 on page 69](#). Definitions of the statistics can be found in [Section A.3 on page 205](#). You can add more ligands to the test set, and update the statistics by clicking Test.
- Create a scatter plot of the experimental data against the predicted data. To do this, click Scatter Plot, which opens the Phase QSAR - Scatter Plot dialog box (see [Figure 7.4 on page 71](#)), in which you can select the number of PLS factors and the ligands to include in the plot, then opens the Plot XY panel to display the plot. See [page 70](#) for more information.
- Visualize the QSAR model in the Workspace. To do so, click the View QSAR Model button at the top of the panel.



You can control what is displayed in the Workspace by using the QSAR Visualization Settings panel. To open this panel, click QSAR Visualization. For information on using this panel, see [Section 7.5 on page 71](#).

Once you are satisfied with the model, you can make use of it in the following ways:

- Export it to an external file. The model can then be used in other projects or applications. To do so, click **Export**, and use the file selector that is displayed to name the file. The model is exported with a `.qsar` extension. Along with it, the ligands are exported to a file with the same base and a `_qsar_pred.mae` extension. Exporting the QSAR model allows you to use it
- Add it to an existing hypothesis in the Project Table. To do so, click **Add to Hypothesis** and select the hypothesis in the entry chooser that is displayed. You can then export the hypothesis from the Project Table for external use.
- Make predictions for other molecules, which must exist as entries in the Project Table. To do so, click **Predict**, and choose the entries in the entry chooser that is displayed. The predicted property for each number of PLS factors is then added to the entries in the Project Table.

You can also import an existing model and examine and use it as described in this section. To import the model, click **Import** and navigate to the desired `.qsar` file.

Creating and Updating a 3D Database

To search for matches to a hypothesis, it is often convenient to store the structures you want to search in a prepared 3D database. When you have prepared the database, the structures will be all-atom structures with reasonable 3D geometries. In addition, you can generate conformers and add site points for a given set of pharmacophore features to the structures.

The creation of a database and addition of structures to the database can be performed in the Generate Phase Database panel. A wider range of 3D database management tasks can be performed from the command line—see [Chapter 13](#).

To open the Generate Phase Database panel, choose Generate Phase Database from the Phase submenu of the Applications menu in the main window.

10.1 Input Structures

When you create a database, you can add structures to the database. You can also add structures to an existing database in this panel. At the top of the panel you specify the source of the structures and set various options for handling these structures.

You can add structures from a single file or from multiple files. The files must be in Maestro or SD format (compressed or uncompressed), or in SMILES format.

- To read from a single file, enter a file name in the Input structure file text box, or click **Browse** to navigate to the file.
- To specify multiple structure files with related names, you can use the wild card characters * and ? in the file name. These characters have their usual Unix file-matching meanings: ? matches a single character, and * matches zero or more characters.
- To specify multiple structure files with unrelated names, you can create a text file that contains a list of structure file names, and specify this text file in the Input structure file text box, or click **Browse** and navigate to it.

If you type in the text box, you must press **ENTER** to ensure that the name is read and the Using property option menu is populated.

The Create a subjob from each input structure file option allows you to run a separate subjob for each structure file. If you do, you cannot retile the structures using the controls described below. In addition, no checking is done for duplicate structures.

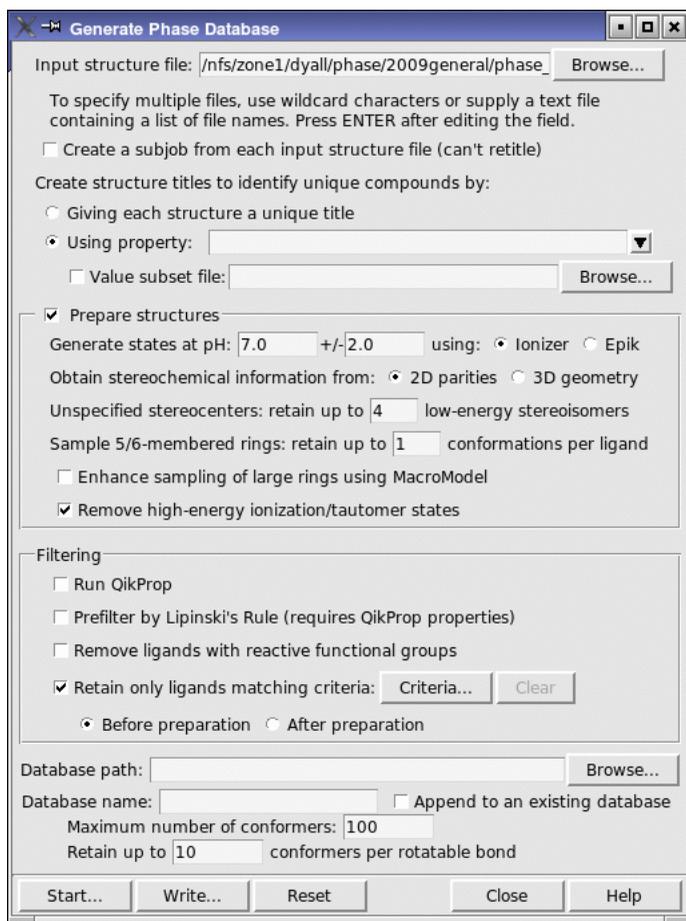


Figure 10.1. The Generate Phase Database panel.

The controls under Create structure titles to identify unique compounds by enable you to select the title for the structures. The title is used to associate structures that originate from the same compound.

- If the structures you have are all unique compounds, you can assign a unique title to each with the first option, Giving each structure a unique title. The title that is assigned is an integer.
- If the structures contain different ionization states or tautomers of the same compound, you can assign a title by selecting Using property and choosing a property from the option menu. The property names are taken from the first structure in each file, and only those properties that exist in each file are presented. You should ensure that the property you

choose exists for each structure in the file, not just the first. The option menu becomes available when a file with valid properties is specified.

When the title is set, a new property is created to store the original title.

10.2 Preparing the Structures

A 3D database should be built from all-atom structures with chemically reasonable 3D geometries. The input structures could be represented in 2D form, without explicit hydrogen atoms, or with counter ions and solvent molecules. In addition, the structures might not have chirality information or be in the appropriate ionization state for physiological conditions. If any of these is the case for your input structures, you must select **Prepare Structures** to obtain structures that are suitable for database searching. If the structures are already all-atom 3D structures, you can deselect this option.

If you need to prepare the structures, you can select from a range of options and settings for ionization, stereochemistry, and ring conformation:

- **Generate states at pH controls**—Generate ionization and tautomeric states that have significant probability in the given pH range. Enter the target pH and the range in the text boxes, and select the tool for generating ionization states. Epik is licensed separately, so you must have an Epik license to use it.
- **Obtain stereochemical information from options**—These options allow you to specify the source of stereochemical information. If you have 3D structures, select 3D geometry to determine the stereochemistry from the geometry. Otherwise, select 2D parities, to use information from the parity property in the input file. Any stereochemical center whose chirality is not determined from this information will have its chirality varied.
- **Unspecified stereocenters: retain up to N low-energy stereoisomers**—Enter the maximum number of stereoisomers to be retained by LigPrep in this text box. LigPrep generates up to 32 stereoisomers, which are then filtered to retain those with the lowest energies. Note that the LigPrep run preserves any existing chirality information in the input file, and selects starting chiralities based on the chemistry of naturally occurring steroids, fused rings and peptides.
- **Sample 5/6-membered rings: retain up to N conformations per ligand text box**—Enter the maximum number of ring conformations for 5- and 6-membered rings to be generated by LigPrep in this text box.
- **Enhance sampling of large rings using MacroModel option**—Sample conformations of 7-membered and larger rings with MacroModel after structure preparation. These ring conformations are not sampled by LigPrep.

- Remove high-energy ionization/tautomer states option—Select this option to remove ionization and tautomeric states that have high energies. These are states that are likely to have low populations at the prevailing conditions.

10.3 Filtering the Structures

If you want to add only those structures to the database that meet certain criteria, such as their suitability as drug candidates, you can apply a filter to the structures before they are added. Three options for filtering are available in the Filtering section of the panel, and a fourth to generate descriptors for filtering.

- Run QikProp—Select this option to run QikProp after the structure preparation is done. You must run QikProp if you want to prefilter the structures using the Lipinski Rule, or use QikProp properties for custom filtering. If your files already have QikProp properties, you do not need to run QikProp again.
- Prefilter by Lipinski's Rule—Prefilter the structures using Lipinski's Rule of 5 before addition to the database. This rule is described in [Table 1.1](#) of the *QikProp User Manual*, under RuleofFive. Structures that do not satisfy this rule are not added. This option requires QikProp properties. If the input structure files do not have QikProp properties, select Run QikProp.
- Remove ligands with reactive functional groups—Prefilter the structures by removing structures that have reactive groups. The filtering is done by `ligfilter` with a pre-defined set of reactive groups, which are:

Acyl halides	Phosphinyl halides	Phosphines
Sulfonyl halides	Phosphonyl halides	Alkyl sulfonates
Sulfinyl halides	Alkali metals	Epoxides
Sulfenyl halides	Alkaline-earth metals	Azides
Alkyl halides without fluorine	Lanthanide series metals	Diazonium compounds
Anhydrides	Actinide series metals	Isonitriles
Perhalomethylketones	Transition metals	Halopyrimidines
Aldehydes	Other metals	1,2-Dicarbonyls
Formates	Toxic nonmetals	Michael acceptors
Peroxides	Noble gases	Beta-heterosubstituted carbonyls
R-S-O-R	Carbodiimides	Diazo compounds
Isothiocyanates	Silyl enol ethers	R-N-S-R
Isocyanates	Nitroalkanes	Disulfides

- Retain only ligands matching criteria—Prefilter the structures using `ligfilter`, retaining only those that match the criteria specified. To set up a custom filter or to read a filter file, click **Criteria**. This button opens the **Filtering Options** dialog box, which is described on [page 11](#) of the *Virtual Screening Workflow* document. To clear the custom filter, click **Clear**. Select **Before preparation** or **After preparation** to determine the point at which the criteria are applied.

For more information on `ligfilter`, see [Section D.2.5](#) of the *Maestro User Manual*.

10.4 Specifying the Database

A Phase 3D database must be stored on a file system that is accessible to all hosts that will read the database. Access to the database is not needed on the host from which you launch database jobs, only on the hosts that run the jobs.

In addition to host access, you should also consider whether other users will need to search or modify a database that you create. If so, you should choose a file system in which you can safely grant other users read and execute permissions to each directory along the path leading to the database. If you want to allow them to modify the database, they must be given write permissions as well as read and execute permissions.

- To specify the location of the database, enter the path to the database in the **Database path** text box, or click **Browse** and navigate to the desired directory.
- To specify the name of the database, enter it in the **Database name** text box. This name is used as the stem for various database-related files.
- If you are adding files to an existing database, select **Append to an existing database**. The database specified by the path and name must exist.

10.5 Generating Conformers and Sites

Conformers are generated using the **ConfGen** facility, and site points are automatically added. This is the same procedure as is used when generating conformers during a search for matches. The **Generate Phase Database** panel offers only limited flexibility in the conformer generation process. You can restrict the number of conformers generated in the **Maximum number of conformers** text box. You can ensure a minimum coverage of conformational space by entering a value in the **Retain up to N conformers per rotatable bond** text box.

If you want to add pregenerated conformer sets or structures without conformers, you can use the command-line tools, which are described in [Chapter 13](#).

Finding Matches to Hypotheses

When you have a pharmacophore model and a prepared 3D database or file of 3D structures, you can proceed to searching a database for structures that match the hypotheses of the model. The search process is normally performed in two steps: *finding* and *fetching*.

In the find step, the database is searched for geometric arrangements of pharmacophore sites that match the site types and intersite distances of the chosen hypothesis. For example, the hypothesis DHRR contains one donor (D), one hydrophobe (H) and two aromatic rings (R1, R2). These four pharmacophore features give rise to six unique intersite distances: dDH, dDR1, dDR2, dHR1, dHR2 and dR1R2. The find step scans the database for occurrences of the four feature types for which the six intersite distances are sufficiently close to those of the hypothesis. When such an occurrence is found, information about the match is written to a *match file*.

In the fetch step, the match file is used as a lookup table to rapidly retrieve the relevant conformers from the database and align them to the hypothesis. We refer to these conformers as *hits*. When hits are fetched, they are ordered and filtered, so that only a fraction of the total number of matches is presented. The hits are ordered first by their fitness score, then filtered by number, and by occupation of excluded volumes if these are defined. Finally, the activity is predicted if there is a QSAR model available. The hits that are fetched are added to the Project Table as an entry group.

The find step is the center of the search, and is the most time-consuming part of the process. Phase separates the two steps so that you do not need to repeat the find step if you only want to apply different filters or a different ordering to the matches before they are retrieved.

The find step offers considerable flexibility in the matching process. A hypothesis might have more features than are actually needed for binding, for example, but there might be some uncertainty about exactly which features are responsible. You can then require that only a certain minimum number of features must match. This is known as *partial matching*. As another example, you might know that a ligand cannot bind to a particular receptor unless it contains a positive site and an aromatic ring. You could then require these features to match. You might also know that a ligand cannot bind if it contains a hydrophobic site in some particular location, and you can require that this feature does not match. Specifying which features must match or must not match is done with a *site mask*. Also, because certain types of ligand-receptor interactions are stronger and more specific, it often makes sense to define different tolerances on matching different types of features. It might also be necessary to

distinguish between different instances of the same feature type. Both types of tolerances can be adjusted when a search is set up.

Searching for structures that match a hypothesis, either in a database or a file, can be done in the Find Matches to Hypotheses panel. This panel provides options for matching and for processing the hits. Scoring of the hits is described in the next section. The subsequent section describes how to set up a search, and the final section discusses the output.

11.1 The Fitness Score

Hits are first fetched in order of decreasing *fitness*. Fitness is a score that measures how well the matching pharmacophore site points align to those of the hypothesis, how well the matching vector features (acceptors, donors, aromatic rings) overlay those of the hypothesis, and how well the matching conformation superimposes, in an overall sense, with the reference ligand conformation. The fitness score is defined by

$$S = W_{\text{site}} (1 - S_{\text{align}}/C_{\text{align}}) + W_{\text{vec}} S_{\text{vec}} + W_{\text{vol}} S_{\text{vol}} \quad (1)$$

The terms in the score are described in Table 11.1. This score is a truncated version of the survival score for hypotheses. See Section 6.1 on page 48 for more information on the various terms in the scoring function and how they are defined.

Table 11.1. Description of parameters in the fitness scoring function.

Parameter	Description
S_{align}	Alignment score: RMS deviation between the site point positions in the matching conformation and the site point positions in the hypothesis.
C_{align}	Alignment cutoff. User-adjustable parameter; default is 1.2.
P_{align}	Alignment penalty for partial matches. User-adjustable parameter; default is 1.2.
W_{site}	Weight of site score. User-adjustable parameter; default is 1.0.
S_{vec}	Vector score: average cosine between vector features in the matching conformation and the vector features in the reference conformation.
W_{vec}	Weight of vector score. User-adjustable parameter; default is 1.0
S_{vol}	Volume score: Ratio of the common volume occupied by the matching conformer and the reference conformer, to the total volume (the volume occupied by both). Volumes are computed using van der Waals models of all non-hydrogen atoms.
W_{vol}	Weight of volume score. User-adjustable parameter; default is 1.0

By adjusting the parameters in the fitness function, you can control the order in which hits are returned. For example, if you want to emphasize the alignment of vector features, you could increase the vector weight. The volume term provides a means of forcing the shape of the hit to resemble that of the reference conformation. If the overall molecular superposition is most important, you could increase the volume weight.

If you choose to match fewer than the number of sites in the hypothesis (“partial matching”), the survival score is modified to penalize the hits that do not match all sites. If there are n sites in the hypothesis, and m sites are matched, the alignment score is modified as follows:

$$S_{\text{new}} = \sqrt{W_{\text{old}} S_{\text{old}}^2 + W_{\text{new}} P_{\text{align}}^2} \quad (2)$$

where $W_{\text{old}} = m/n$, $W_{\text{new}} = (n-m)/n$, and the alignment penalty P_{align} is adjustable, and has the default value 1.2.

11.2 Setting up A Search

Searches are performed from the Find Matches to Hypothesis panel. To open this panel, choose Find Matches to Hypothesis from the Phase submenu of the Applications menu. If you are working in the Score Hypotheses step or the Build QSAR Model step of the Build Pharmacophore Model panel, click Search for Matches.

To perform a search, you must select the source of the structures to search and a hypothesis, set any options for matching, and for filtering and treatment of the hits, then click Start. The options are described below.

If you want to run the job from the command line, or simply to generate the input files, click Write. A dialog box opens in which you can specify a name that is used for the file stem of the input files. When you click Write in this dialog box, the files are written to the current directory.

11.2.1 Selecting a Structure Source

The collection of structures that you use to search for matches can come from one of three sources: a prepared 3D database, an external file, or the Project Table. The structures must be all-atom 3D structures.

- To search a 3D database, choose 3D database from the Search in option menu, then either enter a name in the File name text box, or click Browse and navigate to the desired database. If you want to search a subset of structures from the database, enter the name of the subset in the Subset text box, or click Browse, and select a subset in the Select Subset dialog box. Subsets can be generated in the 3D database management process. By default, the database selected is the last database you used.

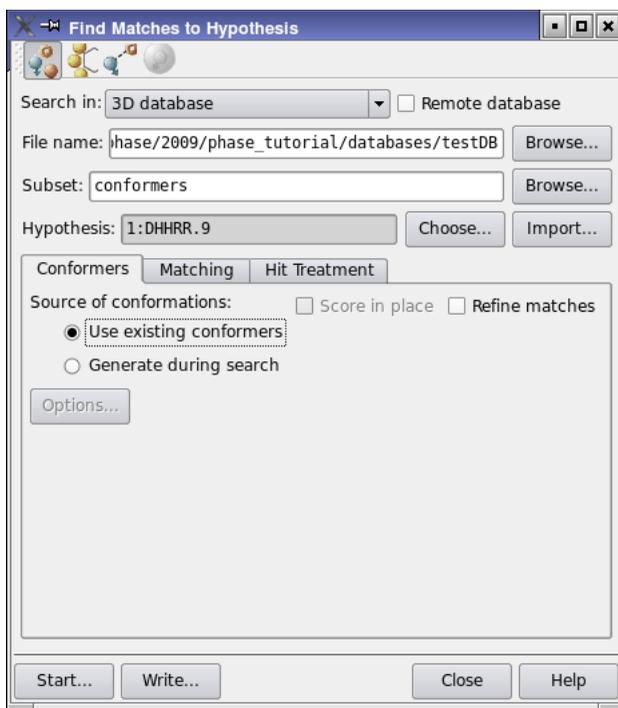


Figure 11.1. The Find Matches to Hypotheses panel.

If the database is located on a remote host on a file system that is not accessible to the local host, select **Remote database**. You must then enter the full path to the database on the remote file system in the **File name** text box. The usual checking done by Maestro is bypassed, and is done instead at the beginning of the search job.

You can specify a subset that is not part of the database, by entering the file name in the **Subset** text box. The name must end in `_phase.inp`. The file need not exist on the local file system: it can be on a remote host, such as the host where the database is stored.

- To search the structures in a file, choose **External file** from the **Search in** option menu, then either enter a file name in the **File name** text box, or click **Browse** and navigate to the desired database. The file must be in Maestro or SD format, and can be compressed with gzip (`.mae.gz`, `.maegz`, or `.sdf.gz`) or uncompressed.
- To search structures from the Project Table, select the entries you want to search in the Project Table, then choose **Project Table (selected entries)** from the **Search in** option menu.

11.2.2 Selecting a Hypothesis

Hypotheses are stored as entries in the Project Table. To choose a hypothesis for the search, click **Choose**. An entry chooser opens, with a list of entries that have hypotheses. Select the hypothesis, and click **OK**. The entry ID and the hypothesis ID are displayed in the Hypothesis text box.

If you open the panel from the **Score Hypotheses** step or the **Build QSAR Model** step of the **Build Pharmacophore Model** panel, the selected hypotheses are added to the Project Table, and the first added hypothesis is chosen as the default hypothesis.

You can add hypotheses to the Project Table by importing them: click **Import**, and navigate to the desired hypothesis in the file chooser that is displayed. However, you can only import hypotheses that were previously exported. You cannot import hypotheses from another project.

You can display the hypothesis, its excluded volumes, its intersite distances and angles, and its feature-matching tolerances in the **Workspace** by clicking the toolbar buttons. The first of these buttons displays the hypothesis; the rest of these buttons are the same as in the **Manage Hypotheses** panel, and are described in [Section 8.1 on page 77](#).

The feature definitions for the hypothesis should match those in the database. If they do not match, you can compute pharmacophore sites as needed using the hypothesis feature definitions, without replacing the existing sites in a database.

If you open this panel from the **Build Pharmacophore Model** panel, the hypotheses that were selected in the table of hypotheses in the **Hypothesis Scores** tab are added to the Project table, and the first of these is selected by default.

11.2.3 Selecting the Source of Conformations

The search for matches to a hypothesis requires conformations of each molecule searched. If the source of structures includes sets of conformers, you can select **Use existing conformers** in the **Conformers** tab. If the source does not include conformers, select **Generate** during search in the **Conformers** tab, to generate them during the search. The conformer generation uses the **ConfGen** method (see [Section 3.3 on page 19](#) for details), with a restricted set of options. You can set the options in the **Generate Conformers** dialog box by clicking **Options**.

Searching a set of molecules that does not include conformers is called “flexible searching”. The storage requirements for flexible searching are much smaller than for standard searching, but the search can take up to 10 times longer. Also, because the matches are not stored on disk, you cannot repeat a flexible search with different criteria for filtering the hits, without repeating the entire find step.

Two other options are available with the Use existing conformers option:

- **Score in place**—Score structures without doing any alignment to the hypothesis. This option is only available when you are searching in an external file.
- **Refine matches**—Generate extra conformers for the top-ranked match from each molecule matched. These conformers are subjected to the hit filters and are returned in the hit file. This procedure improves the fitness score, and can return matches when excluded volumes eliminate every match for some molecules.

11.2.4 Setting the Search Mode and Criteria

The search mode and search criteria are set in the Matching tab.

- If you are starting with a new structure source, select **Find new matches** as the search mode. This is the only mode available if you are not searching a database.
- If you want to apply a different filter to an existing set of matches, or use a different scoring function, select **Use saved matches**. This option is only available for database searches.

The search criteria options are only available if you select **Find new matches**.

- To generate sites during the search, using the hypothesis feature definitions, select **Generate sites during search**. These sites are used only during the search, and are not written out anywhere. This option only applies if you are searching a database, and allows you to use different sites from those in the database without affecting the database.
- To change the tolerance for intersite distances, enter a value in the **Distance matching tolerance** text box. Any intersite distance less than the specified value is considered to match.
- To set site-matching tolerances, require certain sites to match, or allow or forbid sites to match other features. click **Advanced**, and make the desired settings for each site in the **Advanced Matching Options** dialog box. These settings are described in the next section.
- If you want to match fewer sites than there are in the hypothesis (partial matching), enter the minimum number of site points in the **Must match on at least *N* site points** text box. Matches are made on any number of site points from the minimum up to the total number of site points in the hypothesis, but matches that are made on less than the total number of site points are penalized in the survival score—see [Section 11.1 on page 100](#).
- If you are matching fewer than the maximum number of site points and want only the matches that have the greatest number of site points that match, select **Prefer partial matches involving more sites**.

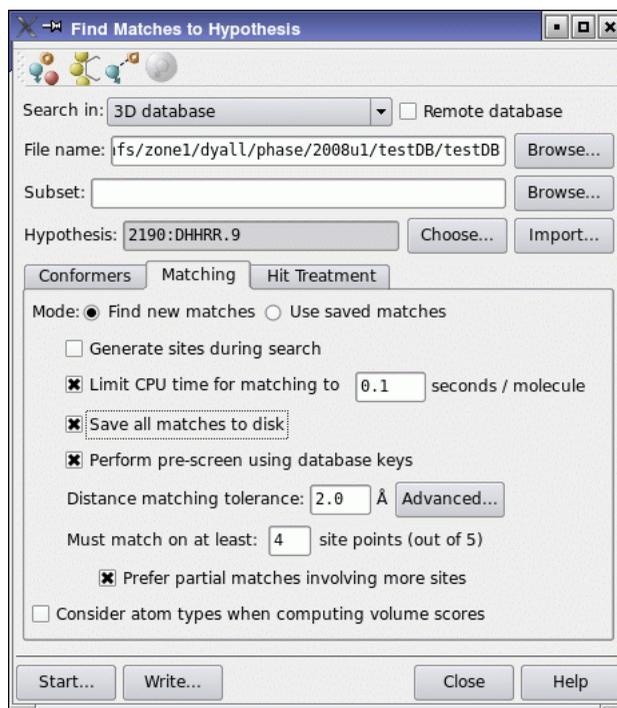


Figure 11.2. The Matching tab of the Find Matches to Hypothesis panel.

- If you want the volume score to reflect the chemistry of the molecules rather than just the shape, select Consider atom types when computing volume scores. Atoms are then considered to overlap only if they have the same MacroModel atom type. This option favors alignments that superimpose chemically similar atoms.
- You can limit the CPU time spent per molecule by selecting Limit CPU time for matching to N seconds/molecule and entering a value in the text box. This option is only available when Find new matches is selected for the mode. Limiting the CPU time is useful when doing partial matching for hypotheses with many sites, as the time spent finding matches to a given molecule may become very large because of combinatorial considerations.
- Matches are saved to disk by default. If you do not want to save the matches (for example, for reasons of space), deselect Save all matches to disk. The matches must be saved to disk in order to reuse them (by selecting Use saved matches). This does not affect the return of hits to the Project Table.

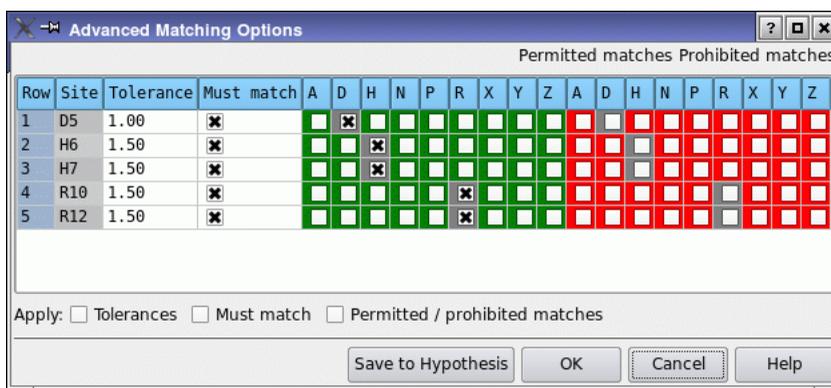


Figure 11.3. The Advanced Matching Options dialog box.

- To speed up the search, you can perform a pre-screening of the database using database keys, by selecting Perform pre-screen using database keys. Use of these 3D keys rapidly filters out the majority of molecules that cannot possibly match the hypothesis. This option is recommended except when searching a small subset of the database or when the search is split across a large number of CPUs.

11.2.5 Setting Site-Specific Matching Criteria

If you want to apply different matching criteria to each site or want to change the default criteria for each site, you can make settings in the Advanced Matching Options dialog box, which you open by clicking the Advanced button in the Matching tab. In this dialog box you can select options to use site-matching tolerances, required site matches (“site mask”), and permitted and prohibited matches (matching rules); and make selections or set values for each of these features. If the hypothesis already has any of these features defined, they are read when the dialog box opens and used to set options.

The dialog box displays a table containing the matching tolerance, site mask, and permitted and prohibited matches. Once you have made settings in the table, you must also select the appropriate Apply option to apply the settings during the search.

- To set tolerances for individual sites, enter the value in the Tolerance column of the table.
- To require matches to certain sites, click in the Must match column for that site.
- To allow sites to match feature types other than that of the site (the “native” feature type), click the relevant feature column under Permitted matches in the table. The native feature type is always selected and cannot be deselected. For example, you might want to allow a hydrophobic site to also match an aromatic ring.

- To prohibit sites from inadvertently matching a particular type of site, click the relevant feature column under Prohibited matches in the table. The native feature type is always unselected and cannot be selected. This capability is useful if you are not matching all sites, to ensure that the sites that are not matched do not accidentally match some other site in the molecule that has an inappropriate type. For example, you might want to prohibit inadvertent matching of an ionic site to a hydrophobic site.

You can save the matching criteria with the hypothesis, by making selections in the table and clicking Save to Hypothesis. The relevant files (.dxyz, .mask, .rules) are then saved in the same location as the hypothesis: in the project, if the hypothesis came from the project, or to external files if the hypothesis was imported. For more information on these files and their use, see [Section B.10](#), [Section B.11](#), and [Section B.12](#).

11.2.6 Setting Filtering and Scoring Options

At the end of the search, the matches are filtered to generate a reduced list of hits, and properties are calculated for these hits. Filters are set in the Hit Treatment tab.

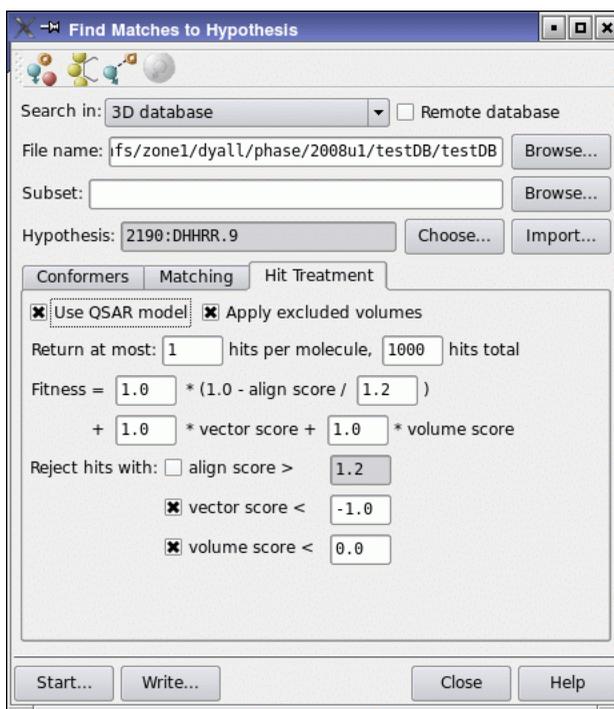


Figure 11.4. The Hit Treatment tab of the Find Matches to Hypothesis panel.

- To calculate activities for the hits based on the QSAR model, if one is available, select **Apply QSAR model**. The hits are not ordered by the activities before they are returned, but you can sort by activity in the Project Table. Activities are calculated for each model, defined by the number of PLS factors in the model. This option is selected by default if the hypothesis has QSAR models, otherwise it is unselected by default.
- To filter out matches that occupy excluded volumes, select **Apply excluded volumes**. This option is selected by default if the hypothesis has excluded volumes. It is unavailable if the hypothesis does not have excluded volumes.
- To limit the number of hits returned, enter values in the **Return at most** text boxes. Some molecules can return more than one hit because different alignments or different conformers might match the hypothesis.
- To change the fitness scoring function, enter new values of the weights in the text boxes.
- To filter out matches that do not satisfy criteria set on the alignment, vector, or volume scores, select the appropriate **Reject hits with** option. For the vector and volume scores, you can change the threshold for rejecting hits by editing the values in the text boxes.

11.3 Search Results

Each time you search the database, the hits are added as an entry group to the Project Table, where you can use the full range of applications and facilities available from Maestro. The fitness score and the activity predicted by the QSAR model (if any) are added as properties, along with a property that indicates which hypothesis was matched. A list of properties is given in [Table 11.2](#).

You can view the hits superimposed on the hypothesis by including them in the Workspace, and displaying the hypothesis from the Find Matches to Hypothesis panel, using the toolbar buttons. If you want to cycle through the hits, you can use the ePlayer: select the hits in the Project Table, then click the Play forward button:



You can change the speed at which the hits are displayed in the ePlayer Options dialog box, which you open from the ePlayer menu.

Table 11.2. Maestro properties generated in a database search

Property	Description
phasedb index	Database record index.
Hit Source	Full path to the database used.
Ligand Name	Name of the ligand.
Conf Index	Index of the matching conformation.
Num Sites Matched	Number of sites matched.
Matched Ligand Sites	String that indicates which sites in the hit matched the hypothesis. The sites are listed in the order of occurrence in the hypothesis. Each site is indicated by the letter for the type of site that matched with the index of the ligand site matched in parentheses. For example D(6) means that site 6 on the ligand matched a donor in the hypothesis. A dash in parenthesis means that the hypothesis site did not match. An example of this property is as follows: D(7) H(-) H(11) R(15) R(16) When using feature-matching rules, it is the ligand feature that matches that is listed rather than the hypothesis feature. For example, allowing aromatic rings to match hydrophobes, you could have the following list of sites for the same hypothesis as in the previous example: D(5) R(15) H(-) R(11) R(12)
Align Score	Alignment score—see Table 11.1 on page 100
Vector Score	Vector score—see Table 11.1 on page 100
Volume Score	Volume score—see Table 11.1 on page 100
Fitness	Fitness score as defined in Equation (1) .
Pred Activity(<i>n</i>)	Predicted activity from the 3D QSAR model with <i>n</i> PLS factors.

Pharmacophore Model Development from the Command Line

To develop a pharmacophore model from a set of ligands, you must prepare the ligands, generate pharmacophore sites for the ligands, find common pharmacophores, then score the resultant hypotheses. You can also build QSAR models for any of the hypotheses. These five steps are referred to by the names used in the Phase panel in Maestro: Prepare Ligands, Create Sites, Find Common Pharmacophores, Score Hypotheses, and Build QSAR Model.

The first step, Prepare Ligands, is the only step that does not have Phase utilities to perform the task. The ligands you provide must be properly prepared and stored in one or more Maestro files prior to starting the workflow. Each molecule should be represented by multiple low-energy 3D structures that provide good coverage of that molecule's conformational space. See [Chapter 9](#) of the *MacroModel User Manual* and the *ConfGen User Manual* for information on creating conformational models, and see the *LigPrep User Manual* for information on 2D-to-3D conversion and structure variation. If you want to develop QSAR models, then each molecule that will be used to train models should contain an activity property, expressed either in concentration units or as $-\log[\text{concentration}]$.

Within each Maestro file, conformers for a single molecule must be stored consecutively. If two consecutive structures differ only in their stereochemistry, they are treated as conformers of a single molecule unless the titles for those two structures are different.

12.1 Workflow Summary

The complete command-line pharmacophore model development workflow is outlined below, in terms of the scripts to run. These scripts are described in detail in the following sections; links to the relevant script are provided in the summary below. The starting point is one or more Maestro files containing multiconformer models for the ligands of interest. A Phase project is created from these ligands, after which a series of steps is followed, directly analogous to the Develop Pharmacophore Model workflow in Maestro.

Each step requires a Phase main input file, and other files that are stored in the current working directory, a subdirectory of this directory, or the Phase distribution. Output files are created in the current working directory, or specified subdirectories of this directory. Most of the required job setup, including creation of the input files, is handled with the `-setup` options of the utilities listed, and cleanup of temporary and intermediate files is done with the `-cleanup` option. For details on the Phase main input file, see [Section B.2 on page 213](#).

Ligand structures and generated ligand-related information is stored in a subdirectory whose default name is `ligands`; this subdirectory is referred to as the *ligands directory*. Output of the scoring step is stored in a subdirectory whose default name is `result`; this subdirectory is referred to as the *results directory*.

The top-level programs that perform the work accept the standard Job Control options listed in [Table 2.1](#) and [Table 2.2](#) of the *Job Control Guide*, except for those related to Maestro projects. These options are represented by *job-options* in the syntax statements.

Create/Add to a Project:

```
$SCHRODINGER/utilities/pharm_project {-new|-add} [options]
```

Modify Master Data:

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

Create Pharmacophore Sites:

```
$SCHRODINGER/utilities/pharm_create_sites -setup [setup-options]
$SCHRODINGER/phase_feature create_sites [job-options] or
$SCHRODINGER/utilities/pharm_create_sites -cleanup
```

Find Common Pharmacophores:

```
$SCHRODINGER/utilities/pharm_find_common -setup [setup-options]
$SCHRODINGER/phase_partition find_common [job-options] or
$SCHRODINGER/phase_multiPartition find_common [options] [job-options]
$SCHRODINGER/utilities/pharm_find_common -cleanup
```

Score Hypotheses with Respect to Actives:

```
$SCHRODINGER/utilities/pharm_score_actives -setup [setup-options]
$SCHRODINGER/phase_scoring score_actives [job-options]
$SCHRODINGER/utilities/pharm_score_actives -cleanup
```

Score Hypotheses with Respect to Inactives:

```
$SCHRODINGER/utilities/pharm_score_inactives -setup [setup-options]
$SCHRODINGER/phase_inactive score_inactives [job-options]
$SCHRODINGER/utilities/pharm_score_inactives -cleanup
```

Cluster Hypotheses by Geometric Similarity:

```
$SCHRODINGER/utilities/pharm_cluster_hypotheses -setup [setup-options]
$SCHRODINGER/phase_hypoCluster cluster_hypotheses [job-options]
$SCHRODINGER/utilities/pharm_cluster_hypotheses
-cleanup [cleanup-options]
```

Build QSAR Models:

```
$SCHRODINGER/utilities/pharm_build_qsar -setup [setup-options]
$SCHRODINGER/phase_multiQsar build_qsar [job-options]
$SCHRODINGER/utilities/pharm_build_qsar -cleanup
```

Preserve Project Data in a Tar Archive:

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

Once pharmacophore hypotheses and QSAR models have been developed, a number of other command line utilities may be run:

Align Project Ligands or New Molecules to a Pharmacophore Hypothesis:

```
$SCHRODINGER/utilities/pharm_align_mol -setup [setup-options]
    -job jobname
$SCHRODINGER/phase_fileSearch jobname [job-options]
$SCHRODINGER/utilities/pharm_align_mol -cleanup -job jobname
```

Align or Merge a Pair of Hypotheses:

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

Create Excluded Volumes Automatically:

```
$SCHRODINGER/utilities/create_xvolShell [options]
$SCHRODINGER/utilities/create_xvolClash [options]
$SCHRODINGER/utilities/create_xvolReceptor [options]
```

Analyze QSAR Predictions within Hit Files:

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

Visualize QSAR Models:

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

12.2 Pharmacophore Model Development Utilities

The pharmacophore model development utilities are stored in \$SCHRODINGER/utilities. Except where noted, all changes to pharmacophore project files should be done only through the use of the utilities listed below. Brief descriptions of the use of the utilities is given below; full descriptions are given in the following sections.

<code>pharm_help</code>	Prints a help message summarizing the command line pharmacophore model workflow, including all the utilities that follow.
<code>pharm_project</code>	Creates a new command line pharmacophore model project and adds molecules to an existing project.
<code>pharm_data</code>	Performs various operations on the project data.
<code>pharm_create_sites</code>	Does setup/cleanup for the job that creates pharmacophore sites.
<code>pharm_find_common</code>	Does setup/cleanup for the job that identifies common pharmacophores.
<code>pharm_score_actives</code>	Does setup/cleanup for the job that scores hypotheses with respect to actives.
<code>pharm_score_inactives</code>	Does setup/cleanup for the job that scores hypotheses with respect to inactives.
<code>pharm_cluster_hypotheses</code>	Does setup/cleanup for the job that clusters hypotheses by geometric similarity.
<code>pharm_build_qsar</code>	Does setup/cleanup for the job that builds QSAR models.
<code>pharm_archive</code>	Preserves project data in a tar archive.
<code>pharm_align_mol</code>	Does setup/cleanup for the job that aligns project ligands or new molecules to a hypothesis.
<code>align_hypoPair</code>	Aligns/merges a pair of hypotheses.
<code>create_xvolShell</code>	Creates a shell of excluded volume spheres around one or more ligands. Provides a means of defining shape-based queries for database searching.
<code>create_xvolClash</code>	Creates excluded volumes using actives and inactives that have been aligned to a hypothesis. Excluded volumes are placed in locations that would cause steric clashes only for the inactives.
<code>create_xvolReceptor</code>	Creates excluded volumes using a receptor structure or a portion thereof.
<code>phase_qsar_stats</code>	Extracts statistics from a hit file that contains QSAR predictions.
<code>qsarVis</code>	Standalone graphical interface for visualizing QSAR models. Available only on Linux-x86 systems.

In addition to the above utilities, the following programs are in the \$SCHRODINGER directory. These programs accept the standard job options that are described in [Section 12.1](#).

<code>phase_feature</code>	Creates pharmacophore sites.
<code>phase_partition</code>	Identifies common pharmacophores.
<code>phase_scoring</code>	Scores hypotheses with respect to actives.
<code>phase_inactive</code>	Scores hypotheses with respect to inactives.
<code>phase_hypoCluster</code>	Clusters hypotheses by geometric similarity.
<code>phase_multiQsar</code>	Builds 3D QSAR models for a collection of hypotheses, and generates a statistical summary for each model.
<code>phase_qsar</code>	Builds a single 3D QSAR model, and generates detailed output.
<code>phase_fileSearch</code>	Aligns structures in a single file to a hypothesis.

12.3 Setting Up a Phase Pharmacophore Model Project

Phase pharmacophore model projects are a collection of files, managed by a utility called `pharm_project`. These projects are *not* the same as the corresponding Maestro projects, but the results of pharmacophore model development—the hypotheses—can be imported into Maestro. In addition to managing the structures in the project with the utility `pharm_project`, you can add or change certain data associated with the structures with the utility `pharm_data`. These two utilities are described in the next two sections.

12.3.1 `pharm_project`

Creates a new command line pharmacophore model project or adds ligands to an existing project. Conformations must be generated ahead of time and stored in a Maestro file. Consecutive structures with identical titles and connectivities are treated as conformations of a single molecule. If you want to treat stereoisomers as a different molecule, you must use a different title for each stereoisomer. The command syntax is as follows:

```
pharm_project {-new|-add} -mae maefile [-ignoreTitles] [-act actProp]  
              [-conf confProp]
```

The options are described in [Table 12.1](#).

Table 12.1. Options for the `pharm_project` command.

Option	Description
<code>-new</code>	Create a new project in the current directory. Any existing project data will be removed upon confirmation.
<code>-add</code>	Add ligands to an existing project. All project data from Create Sites forward will be removed upon confirmation.
<code>-mae <i>maefile</i></code>	Maestro file containing ligand conformations.
<code>-ignoreTitles</code>	Ignore titles when perceiving conformations. Consecutive structures with identical connectivities are treated as conformations of a single ligand, even if their titles differ.
<code>-act <i>actProp</i></code>	The name of the activity property exactly as it appears in <i>maefile</i> . You must supply this information if you intend to use the ligands to build QSAR models.
<code>-conf <i>confProp</i></code>	The name of the relative conformational energy property exactly as it appears in <i>maefile</i> . You must supply this information if you intend to score hypotheses with respect to relative conformational energy.

Output Files

<code>ligands/</code>	Subdirectory that holds all structural data for the project ligands.
<code>ligands/*.mae</code>	Individual ligand files split out from the input files.
<code>MasterData.tab</code>	A specially formatted text file that holds project data required in various steps of the workflow. Certain modifications are permitted (by hand or through the use of <code>pharm_data</code>).
<code>MasterData.backup</code>	A backup copy of <code>MasterData.tab</code> . Used to revert changes you make to <code>MasterData.tab</code> . Do not modify.
<code>ProjectLigands.inp</code>	Ligand records file. Provides a compact summary of project data, and serves as a template for creating subsets of ligands to align to a hypothesis. While the file can be modified without affecting the integrity of project, it is recommended that you leave it as is, and make a copy of the file if you need to define a subset.
<code>FeatureFreq.tab</code>	Feature frequency file. Sets minimum and maximum allowed feature frequencies for common pharmacophore perception.
<code>FeatureTol.tab</code>	Feature matching tolerances that can be applied when hypotheses are scored with respect to actives.
<code>pharma_feature.ini</code>	Default pharmacophore feature definitions. You may replace this file with customized definitions, but it is strongly recommended that you do the customization with the Phase interface in Maestro.

12.3.2 pharm_data

Performs various operations on `MasterData.tab` and propagates any changes in this file to the rest of the project. This includes changes that may have been made by hand. The syntax of this command is as follows:

```
pharm_data [-log|-exp] [-multiply scale]
           [-active aboveVal] [-inactive belowVal]
           [-train numTrain [-rand seed [-pharm_set] [-sort]]]
           [-conf confProp] [-commit] [-restore]
```

The options are described in [Table 12.2](#).

Table 12.2. Options for the `pharm_data` command.

Option	Description
<code>-log</code>	Perform $-\log_{10}(\text{ACTIVITY})$ conversion on the activity property.
<code>-exp</code>	Perform $10^{-\text{ACTIVITY}}$ conversion on the activity property.
<code>-multiply <i>scale</i></code>	Scale the activity property by <i>scale</i> . Done prior to the $-\log_{10}(\text{ACTIVITY})$ conversion and after the $10^{-\text{ACTIVITY}}$ conversion.
<code>-active <i>aboveVal</i></code>	Use the specified activity threshold to distribute ligands between active and none PHARM_SET categories. All ligands are affected.
<code>-inactive <i>belowVal</i></code>	Use the specified activity threshold to assign ligands to the inactive PHARM_SET category. Only ligands with activities below the threshold are affected. May be used with the <code>-active</code> option.
<code>-train <i>numTrain</i></code>	Distribute ligands between the training set (<code>train</code>) and test set (<code>test</code>) QSAR_SET categories. Only ligands with numeric activities are affected. By default, the first <i>numTrain</i> qualifying ligands are assigned to the <code>train</code> category.
<code>-rand <i>seed</i></code>	Assign ligands to <code>train</code> and <code>test</code> QSAR_SET categories randomly using the supplied random seed integer. Valid only when <code>-train <i>numTrain</i></code> is used.
<code>-pharm_set</code>	Consider the PHARM_SET membership when assigning random QSAR_SET categories. If this option is used, ligands for which PHARM_SET is <code>active</code> or <code>inactive</code> are always assigned to the <code>train</code> QSAR_SET, provided they have numeric activities. Valid only with <code>-rand <i>seed</i></code> .
<code>-sort</code>	Sort by activity and take samples from equal-sized bins so that the training and the test sets cover the activity coordinate as uniformly as possible. Valid only with <code>-rand <i>seed</i></code> .

Table 12.2. Options for the `pharm_data` command. (Continued)

Option	Description
<code>-conf confProp</code>	The name of the relative conformational energy property exactly as it appears in the Maestro files supplied to <code>pharm_project</code> . Use this option if you want to score hypotheses with respect to relative conformational energy, but you did not define the property when <code>pharm_project</code> was run.
<code>-commit</code>	Commit changes in <code>MasterData.tab</code> to project, including ligand Maestro files. Any forward project data affected by these changes will be removed upon confirmation by the user.
<code>-restore</code>	Restore previous <code>MasterData.tab</code> file. You would normally do this when you decide that you do not want to remove forward data, such as when a <code>-commit</code> operation is aborted. May not be used in combination with any other option.

When you make changes to `MasterData.tab`, whether by hand or through operations supported by `pharm_data`, you must use the `-commit` option to update the Maestro files in the `ligands` subdirectory. If you do not update the Maestro files, various Phase modules will not be using the modified values because they read the property data directly from the Maestro files. If you plan to run `pharm_data` multiple times to make a series of changes, then you need only supply the `-commit` flag the final time you run `pharm_data`. You can even run `pharm_data` and supply nothing but the `-commit` flag. This is how you would commit changes made by hand.

If you have completed any forward steps in the project workflow, the results generated in those steps may be invalidated by changes you make to `MasterData.tab`. When you attempt to commit the changes, you will be supplied with a list of files from forward steps that will be invalidated, and you will be given a chance to abort the commit operation. If you choose to abort, you can rerun `pharm_data` with the `-restore` flag to revert to the previous version of `MasterData.tab` (i.e., the data stored in `MasterData.backup`).

If your activities are expressed in concentration units (e.g. K_1 or IC_{50} values) and you intend to create QSAR models, then you must use the `-log` and `-commit` options. You must also perform the `-log` conversion on concentrations if you plan to assign `PHARM_SET` categories using the `-active` or `-inactive` options, because these assignments are based on the assumption that the `ACTIVITY` property increases as potency increases.

12.4 Creating Sites

Pharmacophore sites are created by the `phase_feature` program. A set of pharmacophore feature definitions is applied to each ligand conformation, to identify the positions of all pharmacophore sites in that ligand.

12.4.1 `pharm_create_sites`

Performs pharmacophore site creation setup and cleanup. Requires completion of project setup. The syntax is as follows.

```
pharm_create_sites {-setup [-fd fdFile] |-cleanup}
```

The options are given in [Table 12.3](#).

Table 12.3. Options for the `pharm_create_sites` command.

Option	Definition
<code>-setup</code>	Set up <code>phase_feature</code> job. Any forward project data will be removed upon confirmation.
<code>-fd <i>fdFile</i></code>	Use pharmacophore feature definitions in <i>fdFile</i> . If omitted, the default definitions from the Phase installation will be used. Valid only with <code>-setup</code> .
<code>-cleanup</code>	Clean up after <code>phase_feature</code> job has finished.

Output Files

The files generated by the `-setup` option are:

`create_sites_feature.ini` A copy of the default feature definition file, `pharma_feature.ini`.

`create_sites_phase.inp` Main input file for `phase_feature`.

The file generated by the `-cleanup` option is:

`CreateSitesData.tab` Summary of the pharmacophore feature counts for each ligand.

12.4.2 `phase_feature`

Generates pharmacophore sites for one or more ligands from a set of defined features. The syntax is as follows:

`$SCHRODINGER/phase_feature jobname [job-options]`

If you used `pharm_create_sites` to set up the job, `jobname` is `create_sites`, and the relevant input files are set up automatically.

Input Files

<code>jobname_phase.inp</code>	Phase main input file, which contains options that govern Phase behavior. See Section B.2 on page 213 for details of this file. The list of ligands should be restricted to the ligands to be used in the model development.
<code>jobname_feature.ini</code>	Feature definitions file. This file can be created from the template feature definitions file (<code>pharma_feature.ini</code>) by incorporating any changes to the standard features. The template file is located in <code>\$SCHRODINGER/phase-vversion/data</code> . It is strongly recommended to edit this file using Maestro.
<code>mmphob.ini</code>	File that contains definitions for hydrophobic groups. This file is optional. Unless a local copy is supplied this file will be read from the default location in the mmshare installation.
<code>ligand-name.mae</code>	Files containing ligand structures. These files should be stored in the ligands subdirectory, as specified in the Phase main input file. The default directory name is <code>ligands</code> . Each ligand file is a multi-conformer Maestro file. Ligand names should be listed in the Phase main input file as <code>LIGAND_NAME = ligand-name</code> . You should only list the active ligands to be used in the model.

Output Files

The following output files are created upon successful job completion:

<code>jobname_phase.log</code>	Log information, including the lists of mapped features for each ligand.
<code>ligand-name_sites.phs</code>	Pharmacophore site coordinates for the ligand specified by <code>ligand-name</code> . These files are created in the ligands directory.
<code>ligand-name_xyz.phc</code>	Atom coordinates of each conformer for the ligand specified by <code>ligand-name</code> . These files are created in the ligands directory, and are needed for running Phase scoring jobs. These files have a stripped-down format that allows rapid access to conformer structural data in subsequent steps of the workflow.

12.5 Finding Common Pharmacophores

Common pharmacophores are identified by the `phase_partition` program. All n -point pharmacophores from the `PHARM_SET` ligands are enumerated and filtered into a set of high-dimensional boxes. Pharmacophores in the same box are similar enough to be considered equivalent. Boxes with at least one pharmacophore from a sufficient number of actives are said to “survive” the partitioning process. See [Chapter 5](#) for details on the process. The `phase_multiPartition` program runs `phase_partition` to identify common pharmacophore models with the highest number of n -point pharmacophores.

When setting up a `phase_partition` job, you must decide on the number of sites, how many ligands must match, and whether to restrict the number of features that can occur. The number of sites can range from three to seven, but the most meaningful and useful pharmacophore models typically contain between four and six sites. The recommended approach is to start with five or six sites, and decrease that number only if no common pharmacophores are found. You can do this automatically with `phase_multiPartition`. Likewise it is recommended to first require that all actives match the pharmacophore, and reduce the number only if no common pharmacophores are found. You can require certain actives to match and create ligand groups from structurally-related ligands (such as tautomers or ionization states) by adding data to the ligand blocks in `MasterData.tab`—see [Table B.2 on page 210](#).

12.5.1 `pharm_find_common`

Performs setup and cleanup for common pharmacophore perception. Requires completion of the Create Sites step. The syntax is as follows. The options are given in [Table 12.4](#).

```
pharm_find_common -setup -sites numSites [-match minMatch] [-freq]
pharm_find_common -cleanup
```

Table 12.4. Options for the `pharm_find_common` command.

Option	Description
<code>-setup</code>	Set up <code>phase_partition</code> job. Any forward project data is removed upon confirmation.
<code>-cleanup</code>	Clean up after <code>phase_partition</code> job has finished.
<i>Setup Options:</i>	
<code>-sites <i>numSites</i></code>	The number of sites in each common pharmacophore. Must lie between 3 and 7 inclusive. Required.

Table 12.4. Options for the `pharm_find_common` command. (Continued)

Option	Description
<code>-match minMatch</code>	The minimum number of active <code>PHARM_SET</code> ligands or ligand groups that must match a pharmacophore before it can be considered common. Must lie between 2 and the number of ligands in <code>MasterData.tab</code> for which <code>PHARM_SET = active</code> . If omitted, all active <code>PHARM_SET</code> ligands are required to match.
<code>-freq</code>	Use limits in <code>FeatureFreq.tab</code> to control the pool of variants considered. Individual variants can be removed from the input file once the setup is complete. If this option is omitted, the minimum and maximum frequencies are 0 and 4, respectively, for all types of pharmacophore features.

Output Files

The file generated by the `-setup` option is:

`find_common_phase.inp` Phase main input file for `phase_partition`.

The file generated by the `-cleanup` option is:

`FindCommonPharmData.tab` Summary of the number of boxes for each variant.

12.5.2 phase_partition and phase_multiPartition

The `phase_partition` program is used to find common pharmacophores for a given set of ligands. This step is also known as a partitioning job (from the name of underlying algorithm). This program can be run on multiple processors, which are specified with the `-HOST` option. The `phase_multiPartition` program runs `phase_partition` multiple times, starting with the highest number of site points, and decreasing the number of points successively until common pharmacophores are found. Both programs use the same input file.

Syntax

```
$SCHRODINGER/phase_partition jobname [job-options]
```

```
$SCHRODINGER/phase_multiPartition jobname [-minSites n] [job-options]
```

The `-minSites` option specifies the minimum number of site points to consider.

Input Files

<code>jobname_phase.inp</code>	Phase main input file with options for this type of Phase job.
<code>ligand-name_sites.phs</code>	Site files for each of the ligands in the set, created by a <code>phase_feature</code> run. These files are located in the <code>ligands</code> directory, which is specified in the Phase main input file.
<code>FeatureFreq.tab</code>	Feature frequency file. Used to set minimum and maximum allowed feature frequencies for common pharmacophore perception. See Section B.8 on page 228 for an example.

Output Files

The following intermediate and output files are created by the job in the working directory:

<code>jobname_partition.inp</code>	Partitioning input file, which is generated automatically from the Phase main input file. Used by the computational program, and is useful mainly for troubleshooting purposes.
<code>jobname_partition.out</code>	Output file. Contains some information about the job, but is mainly useful for debugging.
<code>jobname_partition.log</code>	Log file. Contains information on job progress, including boxes generated for each variant and eliminated variants.
<code>jobname_partition_variants.tab</code>	File containing list of variants and number of boxes for each variant.
<code>jobname_boxes.tar</code>	Archive of box file archives generated by the partitioning code. Box files are archived for each variant. This archive is used by the subsequent scoring job.

12.6 Scoring Hypotheses

The Score Hypotheses stage of the workflow involves calculating scores for each possible hypothesis based on ligand alignment, volume overlap, and various properties. Only the highest-scoring hypotheses are kept. For a detailed description of how scoring is done, see [Chapter 6](#). Scoring does not eliminate redundant hypotheses that arise from site permutations, which are treated as distinct by the partitioning algorithm. Redundancies can be identified by applying a clustering technique based on geometric similarity.

Hypotheses are scored with respect to the active `PHARM_SET` ligands by the program `phase_scoring`. This process assigns numerical rankings to the pharmacophores within each surviving box from the Find Common Pharmacophores step. The highest scoring pharmacophore in a given box is designated as a hypothesis, and the ligand giving rise to that pharma-

cophore is known as its reference ligand. The scoring function considers the quality of the alignments afforded by each pharmacophore, along with a number of other user-configurable factors. See [Section 6.1 on page 48](#) for more information on the scoring process.

In addition to scoring hypotheses, it is useful to eliminate hypotheses that are geometrically very similar or identical. It is not uncommon for two or more hypotheses to have very similar or even identical scores and physical characteristics. This is a consequence of the way in which common pharmacophores are perceived. Since the partitioning algorithm operates on an ordered set of intersite distances, it is necessary to consider all permutations among sites of the same type when enumerating pharmacophores. So, for example, the permutations $A_1H_2H_3R_4R_5R_6$ and $A_1H_3H_2R_4R_5R_6$ represent the same pharmacophore, but their 15-dimensional intersite distance vectors would generally not be identical, and they may in fact be dissimilar enough to end up in different boxes. As a result, a given box may have a mirror box that contains many (though not necessarily all) of the same pharmacophores, giving rise to a mirror hypothesis that is indistinguishable from the original, or nearly so. These sorts of redundancies are readily identified, by applying a technique that clusters hypotheses based on geometric similarity.

12.6.1 pharm_score_actives

Performs setup and cleanup for scoring of actives. Requires the completion of the Find Common Pharmacophores step. The syntax is as follows:

```
pharm_score_actives -setup [-tol] [-act weight | -prop weight]
    [-conf weight]
pharm_score_actives -cleanup
```

The options are given in [Table 12.5](#).

Table 12.5. Options for the `pharm_score_actives` command.

Option	Description
-setup	Set up <code>phase_scoring</code> job. Any forward project data is removed upon confirmation.
-cleanup	Clean up after <code>phase_scoring</code> job has finished. Hypothesis files <code>hypoID.def</code> , <code>hypoID.mae</code> , <code>hypoID.tab</code> , and <code>hypoID.xyz</code> are created in the directory hypotheses, and a summary of the active scoring results is written to the files <code>ScoreActivesData.tab</code> and <code>ScoreActivesData.csv</code> .

Setup Options:

-tol Use feature matching tolerances in `FeatureTol.tab` when performing alignments. The default is to apply a single threshold to the overall RMSD.

Table 12.5. Options for the `pharm_score_actives` command.

Option	Description
<code>-act weight</code>	Incorporate reference ligand <code>ACTIVITY</code> into the scoring function, multiplied by the supplied weight. This affects the overall score, and it biases the selection of reference ligands to favor those with higher <code>ACTIVITY</code> .
<code>-prop weight</code>	Incorporate reference ligand <code>1D_VALUE</code> into the scoring function, multiplied by the supplied weight. This affects the overall score, and it biases the selection of reference ligands to favor those with higher <code>1D_VALUE</code> .
<code>-conf weight</code>	Incorporate reference ligand relative conformational energy into the scoring function, multiplied by the supplied weight (positive weights are negated automatically). This affects only the overall score, not the selection of reference ligands. Valid only if <code>-conf confProp</code> was used when <code>pharm_project</code> was run.

If you modify the scoring function with any of these options, you should examine the range of values of the property to choose an appropriate weight. In general, it is advisable to ensure that contributions to the scoring function are in the range 0.0 to 1.0 in magnitude, to prevent the contribution from completely dominating the scoring function.

Output Files

The following files are created with the `-setup` option:

<code>score_actives_phase.inp</code>	Phase main input file.
<code>score_actives_feature.ini</code>	Feature definitions file, as provided to the prior <code>phase_feature</code> job.
<code>score_actives_boxes.tar</code>	Copy of the archive of box files generated by the <code>phase_partition</code> job.

The following files are generated with the `-cleanup` option.

<code>ScoreActivesData.tab</code>	Plain text summary of results in tabular form.
<code>ScoreActivesData.csv</code>	Summary of results in comma-separated value form.
<code>hypoID.def</code>	Feature definitions for the given hypothesis. Stored in the <code>hypotheses</code> subdirectory.

<i>hypoID.mae</i>	Maestro format file containing aligned actives for the given hypothesis. Stored in the <i>hypotheses</i> subdirectory.
<i>hypoID.tab</i>	Primary hypothesis data for the given hypothesis. Stored in the <i>hypotheses</i> subdirectory.
<i>hypoID.xyz</i>	Site coordinates for the given hypothesis. Stored in the <i>hypotheses</i> subdirectory.

12.6.2 phase_scoring

Scores and ranks pharmacophore hypotheses for actives. This program can be run on multiple processors, which are specified with the `-HOST` option. The syntax is as follows:

```
$SCHRODINGER/phase_scoring jobname [job-options]
```

Input Files

<i>jobname_phase.inp</i>	Phase main input file.
<i>jobname_feature.ini</i>	Feature definitions file, as provided to the <code>phase_feature</code> job.
<i>mmphob.ini</i>	Hydrophobic groups definitions file, as provided to the <code>phase_feature</code> job.
<i>jobname_boxes.tar</i>	Archive of box files generated by the <code>phase_partition</code> job. Box files are archived for each variant. This archive is expanded internally during execution.
<i>ligand-name.mae</i>	Files containing ligand structures, in the <i>ligands</i> directory, as provided to the <code>phase_feature</code> job.
<i>ligand-name_sites.phs</i>	Pharmacophore site coordinate files for each ligand, in the <i>ligands</i> directory, as generated by <code>phase_feature</code> .
<i>ligand-name_xyz.phc</i>	Ligand conformation files, in the <i>ligands</i> directory, as generated by <code>phase_feature</code> .
<i>FeatureTol.tab</i>	Feature-matching tolerances file. Optional.

Output Files

The following output files are generated by this job:

<i>jobname_scoring.log</i>	Log file. Contains information on job progress. Stored in the current directory.
<i>jobname_scoring.tar</i>	Archive file that contains all the results of the scoring job. Stored in the current directory.

<code>jobname_variant_hypothesis.tab</code>	File containing a list of hypotheses for the given variant, ordered according to hypothesis rank. Stored in the archive file.
<code>jobname_variant_scores.out</code>	File containing information about hypotheses for each variant. <i>variant</i> is encoded from the variant name as a string of integers; for example AADDH is encoded as 00112. Stored in the archive file.
<code>variant_str_N.mae</code>	Maestro format file containing ligand structures aligned onto the reference ligand for a given hypothesis. <i>N</i> is the unique identifier of the box from which this hypothesis came. Stored in the archive file.
<code>variant_hyp_N.xyz</code>	Site coordinate information file for the given hypothesis. Stored in the archive file.

12.6.3 pharm_score_inactives

Performs setup and cleanup for scoring of inactives. Requires the completion of the `pharm_score_actives` step. The syntax is as follows. Options are given in [Table 12.6](#).

```
pharm_score_inactives {-setup -w weight | -cleanup}
```

Table 12.6. Options for the `pharm_score_inactives` command.

Option	Description
<code>-setup</code>	Set up <code>phase_inactive</code> job. Any forward project data is removed upon confirmation.
<code>-w <i>weight</i></code>	Survival scores from the Score Actives step are adjusted by subtracting this weight multiplied by the average fitness obtained for the molecules in the inactive PHARM_SET.
<code>-cleanup</code>	Cleanup after <code>phase_inactive</code> job has finished. Writes a summary of the inactive scoring results to the files <code>ScoreInactivesData.tab</code> and <code>ScoreInactivesData.csv</code> .

Hypotheses are scored with respect to inactive molecules by the program `phase_inactive`. Survival scores are adjusted to penalize hypotheses that match inactives, on the assumption that the inactives fail to bind because they do not contain the true pharmacophore. See [Section 6.3 on page 53](#) for more information.

Output Files

The following files are created with the `-setup` option:

<code>score_inactives_phase.inp</code>	Phase main input file.
<code>score_inactives_inactive.inp</code>	Input file for <code>phase_inactive</code> (see Section B.4 on page 220).
<code>score_inactives_feature.ini</code>	Feature definitions file, as provided to the prior <code>phase_feature</code> job.
<code>score_inactives_hypoFiles.tar</code>	Archive of hypothesis files.
<code>score_inactives_ligandFiles.tar</code>	Archive of inactive ligand structure files.

12.6.4 phase_inactive

Scores pharmacophore hypotheses for inactives. This program can be run on multiple processors using the `-HOST` option (see [Table 2.1](#) of the *Job Control Guide*). The syntax is as follows:

```
$SCHRODINGER/phase_inactive jobname [job-options]
```

Input Files

Uses the same input files as `phase_scoring`, and the following file in addition.

`jobname_inactive.inp` Phase inactives input file.

The main input file must list only the inactives in the `LIGAND_NAME` records. This is done automatically by `pharm_score_inactives`.

Output Files

The following output files are generated by this job:

<code>jobname_inactive.log</code>	Log file. Contains information on job progress. Stored in the current directory.
<code>ScoreInactivesData.tab</code>	Plain text summary of results in tabular form.
<code>ScoreInactivesData.csv</code>	Summary of results in comma-separated value form.

12.6.5 pharm_cluster_hypotheses

Performs setup and cleanup for the clustering of hypotheses. Clustering is performed by the program `phase_hypoCluster`. Requires completion of the `pharm_score_actives` step. The syntax is as follows. Options are given in [Table 12.7](#).

pharm_cluster_hypotheses {-setup [-link *method*] |-cleanup} -report *level*

Table 12.7. Options for the `pharm_cluster_hypotheses` command.

<code>-setup</code>	Set up <code>phase_hypoCluster</code> job. Forward project data is removed upon confirmation.
<code>-link <i>method</i></code>	Cluster linkage method. Allowed values are as follows: <code>single</code> Use the highest similarity between any two objects from the two clusters. Produces diffuse, elongated clusters. <code>average</code> Use the average similarity between all pairs of objects from the two clusters. <code>complete</code> Use the lowest similarity between any two objects from the two clusters. Produces compact, spherical clusters. The default is <code>complete</code> .
<code>-cleanup</code>	Cleanup after <code>phase_hypoCluster</code> job has finished.
<code>-report <i>level</i></code>	Report results for a particular level of clustering. If <i>level</i> lies between 0 and 1, it is treated as a merge similarity, and the clusters reported will correspond to the point at which the merge similarity becomes less than or equal to <i>level</i> . If <i>level</i> is 2 or greater, it is treated as a cluster count, and results are reported for the formation of <i>level</i> clusters. Different values of <i>level</i> can be tried without rerunning <code>phase_hypoCluster</code> . To determine an appropriate value for <i>level</i> , examine the file <code>cluster_hypothesis_hypoCluster.log</code> .

Output Files

The following files are created with the `-setup` option:

<code>cluster_hypotheses_phase.inp</code>	Phase main input file.
<code>cluster_hypotheses_hypoCluster.inp</code>	Input file for <code>phase_hypoCluster</code> (see Section B.5 on page 222).
<code>cluster_hypotheses_feature.ini</code>	Feature definitions file used to create hypotheses.
<code>cluster_hypotheses_hypoFiles.tar</code>	Archive of hypothesis files.

The following files are generated with the `-cleanup` option.

<code>ClusterHypothesesData.tab</code>	Plain text summary of results in tabular form.
<code>ClusterHypothesesData.csv</code>	Summary of results in comma-separated value form.

12.6.6 phase_hypoCluster

Hierarchical agglomerative clustering of hypotheses is performed by `phase_hypoCluster`, using a geometric similarity computed from the least-squares alignment of each pair of hypotheses i, j :

$$\text{Sim}(i, j) = \frac{\langle i | j \rangle}{\sqrt{\langle i | i \rangle \langle j | j \rangle}}$$

where

$$\langle i | j \rangle = S_{\text{site}}(i, j) W_{\text{align}} + S_{\text{vec}}(i, j) W_{\text{vec}}$$

The site and vector scores are computed just as in the Score Actives step (see [Section 6.1 on page 48](#)). When more than one mapping is possible, the alignment yielding the highest similarity is used. Hypotheses that do not contain the same pharmacophore features (i.e., different variants) are assigned a similarity of zero, because the purpose of the clustering procedure is to distinguish hypotheses that are geometrically equivalent from those that are not.

The syntax of the `phase_hypoCluster` command is as follows:

```
$SCHRODINGER/phase_hypoCluster jobname [job-options]
```

Input Files

`jobname_hypoCluster.inp` Hypothesis clustering input file.

The remainder of the input files are specified in the hypothesis clustering input file—see [Section B.5 on page 222](#).

Output Files

`jobname_hypoCluster.log` Log file. Contains information on job progress and results in readable form. Stored in the current directory.

The name of the output file containing the results of the cluster analysis is specified in the hypothesis clustering input file—see [Section B.5 on page 222](#).

12.7 Building QSAR Models

The Build QSAR Model step develops a QSAR model based on partial least-squares (PLS) analysis for one or more hypotheses. The ligands are aligned to each hypothesis as part of the process. For more information on the QSAR model, see [Section 7.1 on page 63](#).

12.7.1 pharm_build_qsar

Performs setup and cleanup for building a 3D QSAR model. Requires the completion of the `pharm_score_actives` step. A QSAR model is created for each hypothesis by the program `phase_multiQsar`. The syntax of the `pharm_build_qsar` command is as follows. Options are given in [Table 12.8](#).

```
pharm_build_qsar {-setup [options] | -cleanup}
```

Table 12.8. Options for the `pharm_build_qsar` command.

Option	Description
<code>-setup</code>	Set up <code>phase_multiQsar</code> job. Any forward project data is removed upon confirmation by the user.
<code>-cleanup</code>	Clean up after <code>phase_multiQsar</code> job has finished. Writes a summary of the QSAR statistics to the files <code>BuildQsarData.tab</code> and <code>BuildQsarData.csv</code> . Updated hypotheses and input files for <code>phase_qsar</code> jobs are written to the directory <code>BuildQsarResults</code> .
<i>Setup Options:</i>	
<code>-atomTypeVol</code>	Consider only atoms of the same MacroModel type when computing volume score overlaps during alignment. This favors alignments that superimpose chemically similar atoms. The default is to ignore atom types when computing volume scores.
<code>-exclude n</code>	Number of training set molecules to exclude for each leave-n-out model. The default is 10% of the training set.
<code>-factors n</code>	The maximum number of PLS factors in each model. The default and largest legal value is 1/5 the number of training set molecules.
<code>-grid spacing</code>	Grid spacing in angstroms. Must lie between 0.5 and 4.0. The default and recommended value is 1.0.
<code>-model type</code>	Model type. Allowed values are <code>atom</code> and <code>pharm</code> . Atom-based models consider the space occupied by all atoms in each molecule, whereas pharmacophore-based models consider only the space occupied by pharmacophore sites that match the hypothesis. Default: <code>atom</code> .

Table 12.8. Options for the `pharm_build_qsar` command. (Continued)

Option	Description																
<code>-rand seed</code>	Random seed integer for selecting leave-n-out subsets. The default is 0, in which case the seed will be assigned from the local time at which <code>phase_multiQsar</code> is run.																
<code>-tvalue tmin</code>	Eliminate binary-valued variables whose absolute t-values are less than <i>tmin</i> . The t-value is a measure of a variable's statistical significance, and it is defined as b/bse , where <i>b</i> is the regression coefficient associated with that variable, and <i>bse</i> is the standard error in the coefficient (i.e., the uncertainty). For PLS, t-values may be estimated from variations in the coefficients observed across a series of leave-n-out models built on the QSAR training set. Suggested <i>tmin</i> values for different training set sizes are: <table border="1" data-bbox="470 617 604 846"> <thead> <tr> <th>Size</th> <th><i>tmin</i></th> </tr> </thead> <tbody> <tr><td>5</td><td>2.57</td></tr> <tr><td>10</td><td>2.45</td></tr> <tr><td>15</td><td>2.13</td></tr> <tr><td>20</td><td>2.09</td></tr> <tr><td>30</td><td>2.04</td></tr> <tr><td>40</td><td>2.02</td></tr> <tr><td>50</td><td>2.01</td></tr> </tbody> </table> <p>These values are based on the 95% significance level. Larger <i>tmin</i> values will increase the significance level and cause the elimination of more binary-valued variables.</p>	Size	<i>tmin</i>	5	2.57	10	2.45	15	2.13	20	2.09	30	2.04	40	2.02	50	2.01
Size	<i>tmin</i>																
5	2.57																
10	2.45																
15	2.13																
20	2.09																
30	2.04																
40	2.02																
50	2.01																

Output Files

The following files are created with the `-setup` option:

<code>build_qsar_multiQsar.inp</code>	Input file for <code>phase_multiQsar</code> (see Section B.6 on page 223).
<code>build_qsar_hypoFiles.tar</code>	Archive of hypothesis files.
<code>build_qsar_ligandFiles.tar</code>	Archive of inactive ligand structure files.

The `-cleanup` option extracts the results archive from the `phase_multiQsar` job into the directory `BuildQsarResults`. These files are listed in the next section.

12.7.2 phase_multiQsar

This is a driver program that builds QSAR models for multiple hypothesis, by running `phase_qsar` on individual hypotheses and collecting the results. The syntax is as follows:

```
$SCHRODINGER/phase_multiQsar [job-options] jobname
```

Input Files

jobname_multiQsar.inp Multiple QSAR model input file.

The remainder of the input files are specified in the multiple QSAR model input file—see [Section B.7 on page 226](#).

Output Files

<i>jobname_multiQsar.log</i>	Log file. Contains information on job progress.
<i>jobname_multiQsar.tar</i>	Archive file (.tar). Contains files relating to QSAR models.
<i>BuildQsarData.tab</i>	Plain text summary of results in tabular form.
<i>BuildQsarData.csv</i>	Summary of results in comma-separated value form.

Files for each hypothesis are stored in the subdirectory specified by the `resultDir` keyword in the input file, which is set to `BuildQsarResults` by `pharm_build_qsar`, inside the archive file. These files are listed below.

<i>hypoID_align.mae</i>	Aligned structures for training and test set molecules that matched at least 3 sites in the hypothesis. Training set molecules appear first. Stored in archive file.
<i>hypoID.def</i>	Feature definitions (copied from input). Stored in archive file.
<i>hypoID.mae</i>	Reference ligand structure (copied from input). Stored in archive file.
<i>hypoID_order.dat</i>	File that defines the overall order of molecules in <i>hypoID_align.mae</i> .
<i>hypoID_pharm.dat</i>	The <code>pharmFile</code> required by <code>phase_qsar</code> , if a pharmacophore-based model was chosen (see Section B.7 on page 226).
<i>hypoID.qsar</i>	QSAR model file.
<i>hypoID_qsar.inp</i>	Main input file for <code>phase_qsar</code> job.
<i>hypoID.rad</i>	Copy of the feature radius file, if specified in the input file.
<i>hypoID.tab</i>	Primary hypothesis data, with QSAR model flag activated.
<i>hypoID.tol</i>	Copy of the feature cutoff file, if specified in the input file.
<i>hypoID.xyz</i>	Hypothesis site coordinates (copied from input).

12.7.3 phase_qsar

The `phase_qsar` program creates and applies grid-based 3D QSAR models. It makes activity predictions and generates detailed output for individual QSAR models for a single hypothesis. If `pharmFile` is specified in the input file, a feature-based QSAR model is developed or tested rather than an atom-based model. To generate a `pharmFile`, run `phase_fileSearch` on the Maestro file that contains the molecules of interest, with `pharmFile` specified in the input file to `phase_fileSearch`. The command syntax is as follows:

```
$SCHRODINGER/phase_qsar jobname
```

Input Files

`jobname_qsar.inp` QSAR model input file, described in [Section B.7 on page 226](#).

The other input files are specified in the QSAR model input file.

Output Files

`jobname_qsar.log` Log file. Contains information on job progress.

`jobname_qsar.out` Output file. Contains complete model statistics, Cartesian coordinates and regression coefficient for each bit in the model, training and test set predictions, and actual bit values for each molecule.

The other output files are specified in the QSAR model input file.

12.7.4 phase_qsar_stats

Extract statistics from Phase QSAR models and from hit files that contain QSAR predictions. The syntax is as follows. Options are given in [Table 12.9](#).

```
phase_qsar_stats -hypo hypoID [-hits hitFile [-act actProp] [-plot csvFile]]
                   [-out outFile]
```

Table 12.9. Options for the `phase_qsar_stats` command.

Option	Description
-act <i>actProp</i>	Experimental activity property name exactly as it appears in <i>hitFile</i> .
-hits <i>hitFile</i>	Maestro file containing hits that match the hypothesis from which the QSAR model was derived.
-hypo <i>hypoID</i>	Prefix that identifies the hypothesis from which the QSAR model was derived. The file <i>hypoID</i> .qsar must be present.

Table 12.9. Options for the `phase_qsar_stats` command. (Continued)

Option	Description
<code>-out outFile</code>	File for program output. If omitted, standard output is used.
<code>-plot csvFile</code>	Comma-separated value file for output of experimental and predicted activities.

12.7.5 qsarVis

This utility allows you to visualize QSAR models that you create in command line projects. It launches a graphical interface entitled Visualization Toolkit – OpenGL, with an interactive 3D image of the ligand, hypothesis, and QSAR model. The QSAR model is displayed in a similar way to the Maestro interface—see [Section 7.5 on page 71](#) for more information. You can rotate, translate, and zoom in using the mouse, but the controls are different from those in Maestro. To rotate the image, drag with the left mouse button; to translate, drag with the middle mouse button; to zoom, drag with the right mouse button.

To change the visualization settings, you must start a new instance of `qsarVis`. However, if you place each instance in the background, you can display and compare QSAR models with various settings.

Note: This utility is only available on Linux-x86 platforms.

The syntax of the command is as follows. Options are given in [Table 12.10](#).

```
qsarVis -hyp hypoID -mol molname [options]
```

Table 12.10. Options for the `qsarVis` command.

Option	Description
<code>-hyp hypoID</code>	Hypothesis ID. Required.
<code>-mol molname</code>	Molecule name (for example, <code>mol_5</code>). Required.
<code>-volume_qsar</code>	View volume bits in QSAR model.
<code>-class name</code>	View effects from a single atom/feature class. Allowed values are D, H, N, P, W, X.
<code>-pc posThresh</code>	Threshold for display of positive values. Default: 0.02.
<code>-nc negThresh</code>	Threshold for display of negative values. Default: -0.02
<code>-trans value</code>	Transparency value. Allowed values are between 0.0 and 1.0. Default: 0.5.
<code>-npls plsFactors</code>	Number of PLS factors. Default: 1.

12.8 Adding Excluded Volumes to a Hypothesis

A molecule may satisfy a pharmacophore model, but fail to bind to the associated receptor due to steric clashes. These clashes can be included by defining excluded volumes, which are used to filter out matches that have any atoms inside these volumes. The Phase distribution contains three utilities, described in the following sections, that allow you to create excluded volumes in an automated fashion, using varying amounts of ligand and receptor information.

12.8.1 create_xvolShell

This utility creates a shell of excluded volume spheres to surround the supplied molecules. This shell defines the outer boundary of a shape-based constraint that can be applied when searching for matches to the hypothesis. The assumption is that the supplied molecules define the binding pocket, and molecules that do not fit in this shell will not fit into the receptor.

The syntax is as follows. Options are given in [Table 12.11](#).

```
create_xvolShell -hypo hypoID [options]
```

Table 12.11. Options for the `create_xvolShell` command.

Option	Description
-append	Append to existing excluded volumes. If this option is used, excluded volumes are added to existing volumes stored in either <i>hypoID</i> .ev, or <i>hypoID</i> .xvol. If both files exist, <i>hypoID</i> .ev will be used.
-buff <i>dist</i>	Buffer distance in angstroms between the excluded volume surface and the van der Waals surface of the reference ligand. Default: 1.0.
-cut	Create excluded volumes with a cutaway view. Since the shell of spheres typically obscures the view of the reference ligand, this option is provided to allow creation of only half the shell. The hypothesis and cutaway excluded volumes may then be imported into the Phase GUI to confirm that the shell is surrounding the ligand as expected. If everything is satisfactory, this program should be rerun without the -cut option.
-grid <i>spacing</i>	The size in angstroms of the grid used to assign excluded volume sphere positions. Also determines the sphere radii. Default: 1.0.

Table 12.11. Options for the `create_xvolShell` command. (Continued)

Option	Description
<code>-hydrogens</code>	Consider hydrogens when creating the shell and when checking for excluded volume violations. If <code>-hydrogens</code> is used, then the file <code>hypoID.ev</code> is created with excluded volume data in a format that is recognized only by Phase computational programs. This file contains a special flag that signals the programs to consider hydrogens when checking for excluded volume violations. The Phase GUI does not support the use of this option, so the search must be set up and run through the command line using <code>hypoID.ev</code> as the source of excluded volumes. Default: consider only non-hydrogen atoms.
<code>-hypo hypoID</code>	File prefix for hypothesis. Required. The files <code>hypoID.mae</code> and <code>hypoID.tab</code> must be present (unless <code>-ref maeFile</code> is used). The file <code>hypoID.xvol</code> is created with the excluded volume data in a format that is recognized by both the Phase GUI and Phase programs.
<code>-mask dist</code>	Eliminate excluded volume spheres whose surfaces are within the specified distance of the van der Waals surface of any masked atom. Valid only with <code>-partial</code> . Default: 1.0.
<code>-partial atoms</code>	Build a partial shell to surround only the specified subset of atoms in the reference structure. <code>atoms</code> is a comma-delimited list of atom numbers and ranges. A range is defined by <code>n:m</code> , where <code>n</code> and <code>m</code> are the first and last atoms in the range, and their defaults are the first and last atoms in the structure. Thus <code>15:</code> means atoms 15 through the last atom. All other atoms in the reference structure are masked out, so as to create an opening in the shell. Cannot be used in combination with multiple reference structures.
<code>-ref maeFile</code>	Build the shell around the structures in <code>maeFile</code> . By default, the reference conformer in <code>hypoID.mae</code> is used.

12.8.2 create_xvolClash

Creates excluded volumes using actives and inactives that have been prealigned to a pharmacophore hypothesis. Excluded volumes are placed in locations that would cause steric clashes only for the inactives. The syntax is as follows. Options are given in [Table 12.12](#).

```
create_xvolClash -hypo hypoID -pos maeFilePos -neg maeFileNeg [options]
```

Table 12.12. Options for the `create_xvolClash` command.

Option	Description
<code>-hypo hypoID</code>	File prefix for hypothesis. Required. The file <code>hypoID.xvol</code> is created with the excluded volume data in a format that is recognized by both the Phase GUI and Phase programs.
<code>-neg maeFileNeg</code>	Maestro file containing the inactives, aligned to the hypothesis. Required.
<code>-pos maeFilePos</code>	Maestro file containing the actives, aligned to the hypothesis. Required.
<code>-append</code>	Append to existing excluded volumes. If this option is used, excluded volumes are added to existing volumes stored in either <code>hypoID.ev</code> , or <code>hypoID.xvol</code> . If both files exist, <code>hypoID.ev</code> will be used.
<code>-buff dist</code>	Buffer distance in angstroms between the excluded volume surface and the van der Waals surface of the reference ligand. Default: 1.0.
<code>-cut</code>	Create excluded volumes with a cutaway view. Since the shell of spheres typically obscures the view of the reference ligand, this option is provided to allow creation of only half the shell. The hypothesis and cutaway excluded volumes may then be imported into the Phase GUI to confirm that the shell is surrounding the ligand as expected. If everything is satisfactory, this program should be rerun without the <code>-cut</code> option.
<code>-freq minClash</code>	The minimum number of inactives that must experience a clash before creating an excluded volume sphere. Default: 1.
<code>-grid spacing</code>	The size in angstroms of the grid used to assign excluded volume sphere positions. Also determines the sphere radii. Default: 1.0.
<code>-hydrogens</code>	Consider hydrogens when creating the shell and when checking for excluded volume violations. If <code>-hydrogens</code> is used, then the file <code>hypoID.ev</code> is created with excluded volume data in a format that is recognized only by Phase computational programs. This file contains a special flag that signals the programs to consider hydrogens when checking for excluded volume violations. The Phase GUI does not support the use of this option, so the search must be set up and run through the command line using <code>hypoID.ev</code> as the source of excluded volumes. Default: consider only non-hydrogen atoms.

12.8.3 create_xvolReceptor

Creates excluded volumes from a receptor structure or a portion of a receptor structure. An excluded volume sphere is created for each atom in the receptor structure that satisfies the minimum and maximum distance criteria. The syntax is as follows.

```
create_xvolReceptor -hypo hypoID -receptor maeFile [-ligand maeFileLig]
  [-radius r|-rprop rpropName] [-scale s|-sprop spropName]
  [-buff dmin] [-limit dmax] [-hydrogens] [-append]
```

Options are given in [Table 12.13](#).

Table 12.13. Options for the `create_xvolReceptor` command.

Option	Description
<code>-hypo hypoID</code>	File prefix for hypothesis. Required. The file <code>hypoID.xvol</code> is created with the excluded volume data in a format that is recognized by both the Phase GUI and Phase computational programs.
<code>-receptor maeFile</code>	Maestro file containing the receptor structure. Required.
<code>-append</code>	Append to existing excluded volumes. If this option is used, excluded volumes are added to existing volumes stored in either <code>hypoID.ev</code> , or <code>hypoID.xvol</code> . If both files exist, <code>hypoID.ev</code> will be used.
<code>-buff dmin</code>	Buffer distance in angstroms between the excluded volume surface and the van der Waals surface of the reference ligand. Default: 1.0.
<code>-hydrogens</code>	Consider hydrogens when creating the shell and when checking for excluded volume violations. If <code>-hydrogens</code> is used, then the file <code>hypoID.ev</code> is created with excluded volume data in a format that is recognized only by Phase computational programs. This file contains a special flag that signals the programs to consider hydrogens when checking for excluded volume violations. The Phase GUI does not support the use of this option, so the search must be set up and run through the command line using <code>hypoID.ev</code> as the source of excluded volumes. Default: consider only non-hydrogen atoms.
<code>-ligand maeFileLig</code>	Specify a Maestro file containing one or more structures that are used instead of the hypothesis reference structure (<code>hypoID.mae</code>) for filtering out excluded volume spheres that are too close to the structures. The structures must be aligned to the receptor.
<code>-limit dmax</code>	Limit the thickness of the shell created by ignoring receptor atoms that are more than a distance <code>dmax</code> from the reference ligand. By default, no limit is applied.
<code>-radius r</code>	Radius for excluded volume spheres. Default: use the van der Waals radius of each receptor atom.
<code>-rprop rpropName</code>	Use radii from the values of the atom-level property <code>rpropName</code> in <code>maeFile</code> . The property <code>rpropName</code> must correspond to a real-valued property, and therefore must begin with <code>r_</code> . Atoms with a zero or unspecified value of this property will be skipped, i.e. no excluded volume spheres will be created.

Table 12.13. Options for the `create_xvolReceptor` command.

Option	Description
<code>-scale s</code>	Multiply all excluded volume radii by <i>s</i> .
<code>-sprop spropName</code>	Scale radii by the factor specified by the values of the atom-level property <i>spropName</i> in <i>maeFile</i> . The property <i>spropName</i> must correspond to a real-valued property, and therefore must begin with <i>r_</i> . Atoms with a zero or unspecified value of this property will be skipped, i.e. the radii will not be scaled.

12.9 Other Utilities

12.9.1 pharm_archive

Archives forward steps in a Phase pharmacophore model project using `tar` and `gzip`, allowing data to be preserved before it is overwritten when a step is rerun. The syntax is as follows. Options are given in Table 12.14.

```
pharm_archive -step stepName -tar tarFile [-gzip]
```

Table 12.14. Options for the `pharm_archive` command.

Option	Description																					
<code>-step stepName</code>	The step at which to begin archiving. Allowed values of <i>stepName</i> are listed below, with the step number and the steps archived: <table border="0" style="margin-left: 20px;"> <tr> <td>1</td> <td>project</td> <td>Entire project</td> </tr> <tr> <td>2</td> <td>create_sites</td> <td>2-7</td> </tr> <tr> <td>3</td> <td>find_common</td> <td>3-7</td> </tr> <tr> <td>4</td> <td>score_actives</td> <td>4-7</td> </tr> <tr> <td>5</td> <td>score_inactives</td> <td>5</td> </tr> <tr> <td>6</td> <td>cluster_hypotheses</td> <td>6</td> </tr> <tr> <td>7</td> <td>build_qsar</td> <td>7</td> </tr> </table>	1	project	Entire project	2	create_sites	2-7	3	find_common	3-7	4	score_actives	4-7	5	score_inactives	5	6	cluster_hypotheses	6	7	build_qsar	7
1	project	Entire project																				
2	create_sites	2-7																				
3	find_common	3-7																				
4	score_actives	4-7																				
5	score_inactives	5																				
6	cluster_hypotheses	6																				
7	build_qsar	7																				
<code>-tar tarFile</code>	Name of <code>tar</code> archive to be created.																					
<code>-gzip</code>	After creating archive, compress using <code>gzip</code> .																					

12.9.2 pharm_align_mol

Performs setup and cleanup for aligning molecules to a pharmacophore hypothesis. Supports alignment of command-line project ligands and structures stored in a Maestro or SD file. Also performs setup and cleanup for searching a file for matches with `phase_fileSearch`.

Alignments are generated by the program `phase_fileSearch`, which can operate on existing conformers, or generate them during the search. See [Section 14.1 on page 175](#) for information on this program.

The command syntax is as follows.

```
pharm_align_mol -setup jobName [options] [search-options] | -cleanup jobName
```

Options specific to `pharm_align_mol` are given in [Table 12.15](#). Search options, which are common to `pharm_align_mol` and `phase_gridSearch` are given in [Table 14.2 on page 177](#).

Table 12.15. Options for the `pharm_align_mol` command

Option	Description
<code>-setup <i>jobName</i></code>	Set up <code>phase_fileSearch</code> job. Existing job files that would be overwritten are removed upon confirmation. <i>jobName</i> is the <code>phase_fileSearch</code> job name.
<code>-cleanup <i>jobName</i></code>	Clean up after <code>phase_fileSearch</code> job has finished. <i>jobName</i> is the <code>phase_fileSearch</code> job name.
<i>Setup Options:</i>	
<code>-hypo <i>hypoID</i></code>	Hypothesis ID. This is the prefix for the associated hypothesis files (<i>hypoID</i> .tab, <i>hypoID</i> .xyz, etc.) Include path if these files are not in the current directory.
<code>-lig <i>recordFile</i></code>	File containing LIGAND_NAME records for the subset of project ligands to be aligned. Use <code>ProjectLigands.inp</code> if you wish to align all ligands. Otherwise, copy <code>ProjectLigands.inp</code> to <i>recordFile</i> , and delete or comment out LIGAND_NAME records you wish to skip. You must run the job from the project directory when using this option.
<code>-mol <i>structFile</i></code>	Maestro or SD file containing cleaned 3D structures to be aligned. Can be compressed. Successive structures with the same title and connectivity are treated as conformers of a single molecule unless <code>-flex</code> is used.
<code>-phase <i>inpFile</i></code>	Phase-style input file from a completed project step involving ligand alignments. The files for each step are listed below: Score Actives <code>score_actives_phase.inp</code> Score Inactives <code>score_inactives_phase.inp</code> Cluster Hypotheses <code>cluster_hypotheses_phase.inp</code> Build QSAR <code>build_qsar_phase.inp</code> If this option is used, the <code>phase_fileSearch</code> alignment options will be consistent with those used in the applicable project step. If this option is omitted, default alignment options are written to the <code>phase_fileSearch</code> input file.

12.9.3 align_hypoPair

Aligns one hypothesis onto another. Alignment is done using least-squares fitting of the matching site points in the two hypotheses, considering all possible mappings. Alignments are summarized to standard output in order of increasing RMSD. By default, only the best alignment is saved as a new hypothesis, but this can be overridden. The syntax is as follows. Options are given in [Table 12.16](#).

```
align_hypoPair -fixed fixedHypoID -free freeHypoID -new newHypoID
  [-dtol deltaDist] [-match minSites] [-mix] [-equiv equivFile]
  [-merge method] [-keep maxAlign] [-rmsd rmsdFile] [-sim [simFile]]
  [-rmsdMax rmsdMax]]
```

Table 12.16. Options for the `align_hypoPair` command.

Option	Description
<code>-fixed <i>fixedHypoID</i></code>	File prefix for the hypothesis that remains fixed when the alignment is performed. At a minimum, the files <code><i>fixedHypoID</i>.xyz</code> and <code><i>fixedHypoID</i>.def</code> must be present. Matching filters are applied if the associated files are present: <code><i>fixedHypoID</i>.tol</code> for feature-matching tolerances, <code><i>fixedHypoID</i>.dxyz</code> for hypothesis-specific tolerances, <code><i>fixedHypoID</i>.mask</code> for site masks.
<code>-free <i>freeHypoID</i></code>	File prefix for the hypothesis that will be aligned to the fixed hypothesis (the “free” hypothesis). At a minimum, the files <code><i>freeHypoID</i>.xyz</code> and <code><i>freeHypoID</i>.def</code> must be present. The feature definitions in <code><i>freeHypoID</i>.def</code> must be identical to those in <code><i>fixedHypoID</i>.def</code> .
<code>-new <i>newHypoID</i></code>	File prefix for the aligned version of the free hypothesis. The files <code><i>newHypoID</i>.def</code> and <code><i>newHypoID</i>.xyz</code> are created automatically. The files <code><i>newHypoID</i>.mae</code> and <code><i>newHypoID</i>.tab</code> are created if <code><i>freeHypoID</i>.mae</code> and <code><i>freeHypoID</i>.tab</code> are present, except when <code>-merge</code> is used.
<code>-dtol <i>deltaDist</i></code>	Intersite distance matching tolerance. Default: 2.0 Å.
<code>-equiv <i>equivFile</i></code>	File that defines the allowed mappings between the sites in the fixed and free hypotheses. This file consists of two lines: the first for the fixed hypothesis and the second for the free hypothesis. Each line contains a string of arbitrary non-blank characters; each string contains one character for each site. The mapping is defined by choosing the same character in each string for sites that match. For example, if the fixed hypothesis contains 4 sites and the free hypothesis contains 5 sites, the following strings could be used to define the mappings: <pre>abbc ababc</pre>

Table 12.16. Options for the `align_hypoPair` command. (Continued)

Option	Description										
	<p>In this mapping, the first site in the fixed hypothesis, denoted as a, can be matched to either the first or third site of the free hypothesis, because these are also denoted as a. The second and third sites of the fixed hypothesis, denoted as b, can be matched to the second or fourth sites of the free hypothesis. The fourth site of the fixed hypothesis, denoted as c, can be matched to only the fifth site of the free hypothesis.</p> <p>If this option is omitted, the variants for the two hypotheses are used to define the allowed mappings, hence only sites of the same type can be matched to each other.</p>										
<code>-keep maxAlign</code>	Maximum number of aligned hypotheses to keep. By default, only a single set of hypothesis files is created with the prefix <code>newHypoID</code> , corresponding to the smallest RMSD. However, if <code>maxAlign</code> is greater than 1 and multiple alignments are possible, a series of hypotheses <code>newHypoID_1</code> , <code>newHypoID_2</code> , ... is created, with progressively larger RMSD values.										
<code>-match minSites</code>	Minimum number of sites that must match. Default: 3.										
<code>-merge method</code>	<p>Merging method. If this option is used, the new hypothesis is a union of the sites in the fixed and aligned free hypotheses, with matching sites merged or replaced as follows:</p> <table border="0"> <tr> <td><code>method</code></td> <td>Merged Site</td> </tr> <tr> <td>1</td> <td>Fixed site type; fixed site coordinates</td> </tr> <tr> <td>2</td> <td>Free site type; free site coordinates</td> </tr> <tr> <td>3</td> <td>Fixed site type; average coordinates</td> </tr> <tr> <td>4</td> <td>Free site type; average coordinates</td> </tr> </table> <p>When merging is done, the new hypothesis has no real reference ligand, so the files <code>newHypoID.mae</code> and <code>newHypoID.tab</code> are not created. The RMSD value is computed based on the positions of the aligned free sites, not the positions of the merged sites.</p>	<code>method</code>	Merged Site	1	Fixed site type; fixed site coordinates	2	Free site type; free site coordinates	3	Fixed site type; average coordinates	4	Free site type; average coordinates
<code>method</code>	Merged Site										
1	Fixed site type; fixed site coordinates										
2	Free site type; free site coordinates										
3	Fixed site type; average coordinates										
4	Free site type; average coordinates										
<code>-mix</code>	Consider alignments involving different numbers of matching sites. By default, only <i>n</i> -point matches are retained and ranked by RMSD, where <i>n</i> is the greatest number of sites that could be matched. If <code>-mix</code> is specified, matches involving fewer sites will be mixed with the <i>n</i> -point matches. RMSD values are generally smaller for matches with fewer sites, so using the <code>-mix</code> option favors those alignments.										
<code>-rmsd rmsdFile</code>	File to which RMSD values should be written, one per line. The file contains one value for each hypothesis created, up to <code>maxAlign</code> . By default, no RMSD file is written.										
<code>-rmsdMax rmsdMax</code>	Similarity parameter. The default is 1.2. Requires <code>-sim</code> .										

Table 12.16. Options for the `align_hypoPair` command. (Continued)

Option	Description
<code>-sim [simFile]</code>	Use RMSD values to compute a similarity for each alignment, according to the following formula: $\text{Sim} = \max(0, (m/n) * (1 - \text{RMSD}/\text{rmsdMax}))$ where <i>m</i> is the number of sites matched, and <i>n</i> is (number of sites fixed + number of sites free)/2. If <i>simFile</i> is supplied, the similarities are written to that file, one per line, for each hypothesis created (up to <i>maxAlign</i>). RMSD is used to rank the alignments, even if similarity is calculated.

12.9.4 create_hypoConsensus

This utility creates a consensus hypothesis from a set of pre-aligned ligands. The approach involves first performing complete linkage hierarchical clustering on each type of site: cluster all the acceptors, cluster all the donors, and so on. The clusters that are retained are those in which all the sites in a given cluster are within a user-specified distance. Then a representative site is chosen from each cluster. By default, the representative is the one closest to the centroid of the cluster. You can also control the number of ligands that must be represented in a cluster (default is all), and constrain the consensus hypothesis to contain only sites that are matched by an existing hypothesis. The consensus hypothesis that is created has no reference ligand, so the output consists only of the `.xyz` and `.def` files.

The syntax is as follows. Options are given in [Table 12.17](#).

```
create_hypoConsensus -mae maeFile -tol d -new newHypoID [-min minLig]
                    [-def fdFile] [-ref refHypoID] [-centroid]
```

Table 12.17. Options for the `create_hypoConsensus` command

Option	Description
<code>-mae maeFile</code>	Maestro file containing pre-aligned structures. Required
<code>-tol d</code>	Clustering tolerance, in angstroms. All sites of the same type that are within a distance <i>d</i> are placed in the same cluster. Required.
<code>-new newHypoID</code>	Hypothesis ID for consensus hypothesis. The files <code>newHypoID.xyz</code> and <code>newHypoID.def</code> are created. The consensus hypothesis has no reference ligand. Required.

Table 12.17. Options for the `create_hypoConsensus` command (Continued)

Option	Description
<code>-min minLig</code>	Minimum number of ligands that must be represented in a cluster before a site from that cluster is included in the consensus hypothesis. By default, all ligands must be represented.
<code>-def fdFile</code>	Use feature definitions in <code>fdFile</code> . If omitted, the default feature definitions in the Phase installation will be applied, unless <code>-ref refHypoID</code> is specified.
<code>-ref refHypoID</code>	Limit the pool of sites to an existing hypothesis. Any cluster whose members are not within the distance d specified by <code>-tol</code> of a site in the supplied hypothesis is discarded. The feature definitions in <code>refHypoID.def</code> are used.
<code>-centroid</code>	Use the actual cluster centroid coordinates for each site, rather than the coordinates of the site that is closest to the centroid.

Managing and Searching 3D Databases from the Command Line

Phase provides two ways of searching a set of structures for matches to a hypothesis: searching a plain Maestro file, or creating a database for searching. In both cases, the process involves generating conformers of each structure and creating the site points. For a search on a Maestro file, the conformers and sites are generated during the search. When a database is created, the conformers and the sites can be stored in the database. In this case, the search step only involves matching the sites to the hypothesis, which is much quicker than the conformational search. So if you intend to search a set of structures more than once with the same set of features, you should consider creating a database and storing conformers in the database. Another advantage of creating a database is that you can define database subsets, which can be searched.

The structures that are used for searching must be all-atom, 3D structures in the correct ionization state. If the structures are not in this form—for example, if they are 2D structures—they must be prepared in the correct form first, which you can do with LigPrep. See the *LigPrep User Manual* for details. If you already have a database from some other source that matches these requirements, you can use it for Phase searches by exporting it to an SD file.

The structures in the database are stored in HDF5 format. The input files can be in Maestro or SD format. Previously, the structures in the database were also stored either in Maestro format or in SD format. You can convert a database in the old format with the utility `phasedb_convert`. If you need to convert between SD and Maestro format, you can do so with the utility `sdconvert` (see [Section D.1.5](#) of the *Maestro User Manual*).

The database must be stored on a file system that is accessible to all hosts that will be used to create, modify, or search the database. Thus if a job is launched from `host1` and run on `host2`, both of these hosts must have access to the database using the same absolute path.

Phase provides a set of tools for managing and searching 3D databases from the command line. These tools are stored in `$(SCHRODINGER)/utilities`, and are prefixed with `phasedb_`. Each section in this chapter describes the use of one of these tools. Some of the tools perform setup and cleanup for Phase computational programs, which are stored in `$(SCHRODINGER)`. Tutorial examples are given in [Chapter 6](#) of the *Phase Quick Start Guide*. Information on searching a plain Maestro file from the command line is given in [Chapter 14](#).

All of the tools can be run as a job under Job Control, and each tool accepts the job options listed in [Table 13.1](#).

Table 13.1. Job options for 3D database jobs

Option	Description
-JOB <i>jobname</i>	Job name. If omitted, no other job control options are permitted.
-HOST <i>host</i>	Run job on the specified host. To use multiple CPUs on a given host, append a colon and the number of processors after the host name: for example -HOST <i>myhost:4</i> . To use multiple hosts, enclose the blank-separated list of hosts in quotes: for example, -HOST " <i>cluster:8 myhost yourhost</i> ". Not all 3D database jobs accept multiple hosts.
-LOCAL	Store temporary job files in current directory.
-TMPDIR <i>dir</i>	Store temporary job files in <i>dir</i> .
-WAIT	Do not return control to the shell until job finishes.
-INTERVAL <i>n</i>	Update log file every <i>n</i> seconds.
-NICE	Run job at reduced priority.

13.1 Managing a 3D Database: `phasedb_manage`

Once you have a set of structures that satisfy the requirements listed above, you can set up the Phase database, add ligands to the database, and remove ligands from the database. If your set of structures exists as several files, you can create the Phase database with the first file, then add the others. Structures are stored in the database in HDF5 format.

When you create a database, the feature definition file used to create sites for the molecules is copied into the database as well as the structures. This is normally done when the database is created, but you can subsequently copy a new feature definition file into the database with `phasedb_manage`. However, if you do so after creating sites with `phasedb_confsites`, you should ensure that you create sites again for all molecules in the database to ensure that the site definitions are consistent.

The feature definition file does not have to correspond to any particular hypothesis. In general, you should create the database using the most universal set of feature definitions that you have, so that you do not have to re-create the sites in the database for different hypotheses. However, if you want to search the database with a hypothesis that was created using feature definitions that differ from those used to create the database, the sites can be created as needed when the database is searched.

If you want to ensure that the database does not contain duplicate structures, you can use the `-unique` option when you create the database to specify a property that is unique for each structure. Duplicates are determined by comparing the value of this property for each added structure with the values in the database. The property you choose must guarantee the unique-

ness of the structure: for example, a molecular registration ID. One way of generating the property is to create a unique SMILES string for each structure, which you can do by running the `uniquesmiles` utility on the structure files before you add them to the database.

The values of the property used to check for duplicates are stored in an SQLite database, which is named `dbname_unique`. While this file exists, checking for duplicates is performed. You can move the file to another location to turn off checking, and move it back to turn on checking again with the values that were last written to the file. You can also delete the file and create a new file based on a different property.

The `phasedb_manage` utility is the primary tool for creating and modifying Phase 3D databases. It may be run as a regular foreground process, or as a single-CPU job on any host that has access to the database directory. The general syntax of the command is as follows:

```
$SCHRODINGER/utilities/phasedb_manage action [job-options] [database-options]
```

The job options are described in [Table 13.1](#), and the database options are described in [Table 13.2](#). The three possible actions are `-new`, `-add`, and `-delete`. The detailed syntax is best described via usage cases, since only certain combinations of options are permitted. Job options should be added to any of these usage cases.

To create a new database:

```
phasedb_manage -new [-fd fdFile] -db dbname {-mae maeFile|-sd sdFile}  
  [-title propName] -confs multiConfs [-ignoreTitles] [-stereo]  
  [-unique propName [-keepMissing]] [-blimit maxMol]
```

If you want to use nonstandard or custom feature definitions, specify the feature definition file with the `-fd` option.

To add molecules to an existing database:

```
phasedb_manage -add -db dbname {-mae maeFile|-sd sdFile}  
  [-title propName] -confs {true|false} [-ignoreTitles] [-stereo]  
  [-unique propName [-keepMissing]] [-blimit maxMol]
```

To delete molecules from a database:

```
phasedb_manage -delete -db dbname -records recordFile
```

To restart a failed database creation or molecule addition:

```
phasedb_manage -db dbname {-mae maeFile|-sd sdFile} [-fd fdFile]  
  -RESTART
```

To completely remove a database:

```
phasedb_manage -delete -db dbname -all
```

The other database options do not need to be supplied when restarting a job, because they are read from the restart file. If you restart the job from a different directory, you will have to ensure that the paths to the database, the structure file and the feature definition file are correct.

For a tutorial example of using this utility, see [Section 6.2](#), [Section 6.3](#), and [Section 6.5](#) of the *Phase Quick Start Guide*.

Table 13.2. Database options for *phasedb_manage*

Option	Description
<code>-blimit <i>maxMol</i></code>	Specify maximum number of molecules per block in the database. Each block is stored in a separate directory. The default and largest allowed value is 5000. The starting molecule ID in each block is always $5000*(i-1)+1$, where i is the block number, so if <i>maxMol</i> is less than 5000, there are unused IDs. In distributed jobs, each subjob operates on a full block, so decreasing the block size increases the number of subjobs that can be run.
<code>-confs [true false]</code>	Option indicating whether or not the supplied structure file contains multiple conformations per molecule. If <code>true</code> , consecutive structures with identical titles and connectivities are treated as conformations of a single molecule. If <code>false</code> , each structure is treated as a separate molecule.
<code>-db <i>dbname</i></code>	Database name, including absolute path. Required.
<code>-fd <i>fdFile</i></code>	Use pharmacophore feature definitions in <i>fdFile</i> . If this option is omitted, the default definitions in the Phase installation are used.
<code>-ignoreTitles</code>	Ignore titles when perceiving conformations. With this option, consecutive structures with identical connectivities are treated as conformations of a single molecule, even if their titles differ. Valid only with <code>-confs true</code> .
<code>-keepMissing</code>	Import structures for which <i>propName</i> is missing or blank. By default, these structures are automatically skipped. Valid only with <code>-unique propName</code> .
<code>-mae <i>maeFile</i></code>	Maestro file containing cleaned structures to store in the database. If compressed, the extension must be <code>.mae.gz</code> or <code>.maegz</code> . If running as a job, use the absolute path to avoid file copy.
<code>-records <i>recordFile</i></code>	Text file containing list of records to delete. Each line in the file should have the form: <pre>LIGAND_NAME = block_<i>i</i>/mol_<i>j</i> # optional comments</pre> <p>where i is the block number (starting from 1) and j is the molecule number (starting from $5000*(i-1)+1$ for each block). If present, the pound sign # and any text following it is ignored.</p>

Table 13.2. Database options for *phasedb_manage*

Option	Description
-RESTART	Restart a database creation or molecule addition job that failed. Before using -RESTART, ensure there are no jobs or subjobs currently running on the database, and verify that the restart directory <i>dbname_manage_restart</i> exists. The values of the database options used for the failed job are preserved. If you supply new values for these options, the new values are ignored. A complete set of instructions on restarting a job is written to the file <i>dbname_manage_restart/README</i> .
-sd <i>sdFile</i>	SD file containing cleaned structures to store in database. If compressed, the extension must be <i>.sdf.gz</i> or <i>.sd.gz</i> . If running as a job, use the absolute path to avoid file copy.
-stereo	Consider annotated stereochemistry when perceiving conformations. With this option, consecutive structures with the same connectivity are treated as conformations of a single molecule only if their stereochemical properties (<i>s_st_EZ*</i> , <i>s_st_Chirality*</i>) are identical. Valid only with <i>-confs true</i> and when importing structures from a Maestro file.
-title <i>propName</i>	Specify the property name from the input file to be used for the structure title. Any existing title property is replaced by the specified property. If the property does not exist for a given structure, the resulting title is blank for that structure.
-unique <i>propName</i>	Specify the property name to be used to ensure that only unique structures are added to the database. Structures are not added if they have the same property value as any structure in the database. The value of <i>propName</i> must be given as it appears in the input structure file. If the property name contains spaces, it must be quoted or the spaces escaped with a backslash, e.g. "my name" or <i>my\ name</i> . The values <i>name</i> and <i>title</i> can be used to select the title (<i>s_m_title</i> Maestro property). This option is only valid with <i>-new</i> and <i>-add</i> , and should only be used once. If used with <i>-add</i> , <i>LIGAND_NAME</i> records of duplicates already in the database are written to <i>dbname_duplicates_phase.inp</i> , which can be used to delete them.

13.2 Generating Conformers and Sites: phasedb_confsites

The `phasedb_confsites` utility creates conformations and pharmacophore sites in a Phase 3D database. Sites are always created, but conformation creation is optional.

Conformer generation is not exhaustive, and therefore depends to some extent on the input structures. However, the conformers generated should represent a reasonable sample, and the results of a search should not depend much on the input structures. If you want to be sure that you have a complete set of conformers, you should run a conformational search beforehand (with MacroModel, for example) and import the conformer sets.

The feature definitions used for the sites are taken from the definition file copied into the database at the point it was created. If you want to use custom feature definitions, you can copy the feature definition file into the database by running `phasedb_manage` with the `-fd` option. You should do this when you create the database or before you run `phasedb_confsites` for the first time. If you change the feature definitions after creating sites, you risk embedding inconsistencies in the database, and you should run `phasedb_confsites` on all molecules in the database to ensure that there are no inconsistencies.

By default, sites are created for all molecules. If you have already created sites for some of the molecules in the database and want to create sites only for those molecules that do not have them, you should create a subset by running `phasedb_subset` with the `-sites false` option (see [Section 13.4 on page 162](#)), and use this subset to restrict the range of molecules for site creation.

The conformers and pharmacophore sites are stored in HDF5 files in the subdirectory `dbName_ligands`.

The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_confsites -db fullname -JOB jobname  
  [job-options] [-RESTART|database-options]
```

The job options are described in [Table 13.1](#), and the database options are described in [Table 13.3](#).

You can run `phasedb_confsites` on multiple CPUs or on a single CPU. The job is divided into subjobs, regardless of the number of CPUs. Subjobs are automatically restarted for certain kinds of failures. If a subjob fails during conformer or site generation for a particular structure, the subjob is automatically restarted at that structure; if it fails again, it is restarted at the next structure. If a subjob fails while writing the HDF5 file, the entire subjob is retried up to 5 times (or up to the number set by the environment variable `SCHRODINGER_PHASE_MAX_RETRY`). For other failures, you can use the `-RESTART` option.

Table 13.3. Database options for `phasedb_confsites`

Option	Description
<code>-amide option</code>	Torsional angle treatment for amides that are not in rings. Allowed values are: <code>vary</code> —Allow angles to vary <code>orig</code> —Retain original torsional angles <code>trans</code> —Set all torsional angles to trans Default: <code>vary</code> .
<code>-bf numPerBond</code>	Maximum number of conformations per rotatable bond to generate for each molecule. Default: 10.
<code>-BLOCK m</code>	Process molecules in blocks of size <i>m</i> . Forces memory cleanup every <i>m</i> molecules, and controls the frequency at which HDF5 files are updated. In a distributed job, each subjob runs multiple blocks sequentially. (This option does not control the number of subjobs, which is determined by the number of CPUs.) Default: 1000.
<code>-confs mode</code>	Generate conformations using torsional sampling. Allowed values are: <code>all</code> —Generate conformers for all molecules. <code>auto</code> —Generate conformers only for molecules that do not already have them. If this option is omitted, the database structures will not be modified. This option must be used if any other conformational options (<code>-sample</code> , <code>-max</code> , <code>-ewin</code>) are used.
<code>-db fullname</code>	Database name, including absolute path. The format of <i>fullname</i> should be <code>dbPath/dbName</code> , where <i>dbPath</i> is the path to the database (the database directory), and <i>dbName</i> is the name of the database. Required.
<code>-ewin deltaE</code>	Conformational energy window in kcal/mol. Conformations that are higher in energy than the lowest-energy conformation by this amount are discarded. Requires <code>-confs</code> option. Default: 10.
<code>-max maxConfs</code>	The maximum number of conformations to generate for each molecule. Requires <code>-confs</code> option. Default: 100.
<code>-RESTART</code>	Restart a previous job that died, was killed, or had failed subjobs. Before using <code>-RESTART</code> , ensure there are no <code>phasedb_confsites</code> jobs or subjobs currently running on the database, and verify that the restart directory <code>fullname_confsites_restart</code> exists. If the original job died or was killed, there may still be a write lock file in the database directory that you will have to remove. This file is named <code>phaseDBAccessWrite_dbName.tmpN</code> . The values of the <code>-confs</code> , <code>-sample</code> , <code>-max</code> , <code>-ewin</code> , and <code>-sub</code> options are preserved. If you supply new values for these options, the new values are ignored. A complete set of instructions on restarting a job is written to the file <code>fullname_confsites_restart/README</code> .

Table 13.3. Database options for `phasedb_confsites` (Continued)

Option	Description
<code>-sample method</code>	Conformational sampling method. Requires <code>-confs</code> option. Allowed values of method are <code>rapid</code> and <code>thorough</code> . See Section 3.3.2 on page 20 for a definition of these methods. Default: <code>rapid</code> .
<code>-sub dbSubset</code>	Operate on only a subset of the database. The file <code>dbSubset_phase.inp</code> must contain the applicable <code>LIGAND_NAME</code> records with the following format: <code>LIGAND_NAME = block_i/mol_j # optional comments</code> where <i>i</i> is the block number (starting from 1) and <i>j</i> is the molecule number (starting from $5000*(i-1)+1$ for each block). If present, the pound sign # and any text following it is ignored. You can create subsets with <code>phasedb_subset</code> —see Section 13.4 on page 162 .

For a tutorial example of using this utility, see [Section 6.4](#) of the *Phase Quick Start Guide*.

13.3 Searching for Matches in a Database: `phasedb_findmatches` and `phase_dbsearch`

Database searching takes place in two steps, finding matches and fetching hits. In the find step, the database is searched for geometric arrangements of pharmacophore sites that match the feature types and intersite distances of the hypothesis, and a file containing all the matches is written out. The find step is the most expensive, and need only be run once. In the fetch step, the match file is used as a lookup table to rapidly retrieve conformers from the database and align them to the hypothesis. The fetch step can be used to filter these hits using excluded volumes, a scoring function, and numerical limits. For more information on searching the database, see [Chapter 11](#). Tutorial examples are given in [Chapter 6](#) of the *Phase Quick Start Guide*.

The utility `phasedb_findmatches` is used to set up and clean up database search jobs, and is run in the foreground. The database search is performed by the program `phase_dbsearch`. The setup mode of `phasedb_findmatches` sets up the input file for `phase_dbsearch`; the cleanup mode removes intermediate files generated during the search. The input file is named `jobname_dbsearch.inp`, and is described in detail in [Section B.13 on page 232](#). The syntax for performing the three steps is as follows:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup jobname -db dbName
    -hypo hypoID -mode runMode [options]

$SCHRODINGER/phase_dbsearch [job-options] jobname

$SCHRODINGER/utilities/phasedb_findmatches -cleanup jobname
```

The options for `phasedb_findmatches` are given in Table 13.4. The general job options for `phase_dbsearch` are given in Table 13.1. There are some specific job options for `phase_dbsearch`, which are listed in Table 13.5. In particular, you must specify a check-pointing option or the restart option.

In addition to setting options to control the search and filter the hits, you can include certain files for the hypothesis in the directory that contains the hypothesis to perform filtering tasks. When these files are detected by `phasedb_findmatches`, the relevant file keyword is added by default to the database search input file. You can override the default action by setting the appropriate option. The files and the associated actions are described in Table 13.6. See Table B.13 for information on the input file.

Table 13.4. Options for `phasedb_findmatches`

Option	Description
<code>-setup jobname</code>	Job name that will be used when launching <code>phase_dbsearch</code> . This determines the names of various input and output files.
<code>-db dbName</code>	Database name, including absolute path. The format of <i>fullname</i> should be <i>dbPath/dbName</i> , where <i>dbPath</i> is the path to the database, and <i>dbName</i> is the name of the database.
<code>-hypo hypoID</code>	Prefix for hypothesis files. At minimum, the files <i>hypoID.xyz</i> and <i>hypoID.def</i> must be present. To use a reference ligand, <i>hypoID.mae</i> and <i>hypoID.tab</i> must also be present.
<code>-mode runMode</code>	Database searching mode. The allowed values are listed below: <code>find+fetch</code> —Search a database of precomputed conformations for geometric matches to the hypothesis, all of which are written to a match file. Aligned conformations for the best matches are written to a hit file in Maestro format. <code>find+fetch+flex</code> —Perform a <code>find+fetch</code> search and refine matches by generating additional conformers. Only the original matches are written to the match file. <code>fetch</code> —Use matches from a previous <code>find+fetch</code> job to create the hit file. This allows different hit criteria to be applied without having to search the database again. You must supply the name of the match file using the <code>-matchFile</code> option. <code>fetch+flex</code> —Run a <code>fetch</code> job and refine the matches by generating additional conformers. You must supply the name of the match file using the <code>-matchFile</code> option. <code>flex</code> —Conformations and pharmacophore sites are generated as the database is searched. These data are never written to disk, and no match file is produced, only a hit file.

Table 13.4. Options for `phasedb_findmatches` (Continued)

Option	Description
<code>-sub dbSubset</code>	Search a database subset. The file <code>dbSubset_phase.inp</code> should contain <code>LIGAND_NAME</code> records for some subset of the database molecules. Include the path in <code>dbSubset</code> if the file is not in the current working directory. If this option is omitted, all molecules in the database are searched. Irrelevant when <code>runMode</code> includes <code>fetch</code> .
<code>-flexSearchMethod method</code>	Flexible conformational search method. Allowed values are <code>rapid</code> and <code>thorough</code> . The default is <code>rapid</code> . Valid only when <code>runMode</code> includes <code>flex</code> .
<code>-flexMaxConfs maxConfs</code>	Maximum number of flexible conformers per molecule. Default: 100. Valid only when <code>runMode</code> includes <code>flex</code> .
<code>-flexConfsPerBond maxPerBond</code>	Maximum number of conformations per rotatable bond to generate for each molecule. Default: 10. Valid only when <code>runMode</code> includes <code>flex</code> .
<code>-flexMaxRelEnergy energy</code>	Flexible conformational energy window in kJ/mol. Default: 41.84, i.e., 10.0 kcal/mol. Valid only when <code>runMode</code> includes <code>flex</code> .
<code>-flexAmideOption option</code>	Flexible amide torsional angle sampling option. Legal values are <code>vary</code> (vary angles), <code>orig</code> (keep original angles), and <code>trans</code> (make angles trans). The default is <code>vary</code> . Valid only when <code>runMode</code> includes <code>flex</code> .
<code>-deltaDist deltaDist</code>	Intersite distance matching tolerance. Default: 2.0 Å.
<code>-minSites minSites</code>	Minimum number of hypothesis sites to match. Default: all sites must match.
<code>-preferBigMatches true false</code>	Option that governs whether partial matches containing a greater number of sites should be favored. If set to <code>true</code> , matches involving fewer than n sites will not be sought if there are any matches with n sites. The default is <code>true</code> . Irrelevant when <code>runMode</code> includes <code>fetch</code> .
<code>-timeLimit timeLimit</code>	CPU time limit in seconds for finding matches for each molecule. CPU usage is checked before each conformer from a given molecule is searched, and matching is terminated if <code>timeLimit</code> is exceeded at that point. The time limit does not apply to the process that generates conformers during the search, so if any of the <code>flex</code> options are specified and a molecule has a large number of rotatable bonds, the overall CPU time may significantly exceed the imposed limit. The default is unlimited CPU time. Irrelevant when <code>runMode</code> is <code>fetch</code> .
<code>-atomTypeVol true false</code>	Option for computing volume scores using overlap only between atoms of the same MacroModel atom type. This favors alignments that superimpose chemically similar atoms. The default is <code>false</code> .

Table 13.4. Options for `phasedb_findmatches` (Continued)

Option	Description
<code>-useDbKeys</code> <code>true false</code>	Option to pre-screen the database using 3D keys, which rapidly filter out the majority of molecules that cannot possibly match the hypothesis. Applicable only when <code>runMode</code> is <code>find+fetch[+flex]</code> . The pre-screening is recommended except when a very small subset of the database is being searched (e.g., < 1%) or when the <code>phase_dbsearch</code> job will be split across a large number of CPUs (e.g., 100).
<code>-useExistingSites</code> <code>true false</code>	Option for searching against existing database sites. Set to <code>false</code> if the hypothesis and database feature definitions differ. If the database is local (i.e., <code>-remote</code> is not used), then the feature definitions will be compared by this utility and the option will be set to the appropriate value. Valid only when <code>runMode</code> is <code>find+fetch[+flex]</code> . Note that database keys can be used only when this flag is <code>true</code> .
<code>-useDeltaHypo</code> <code>true false</code>	Option for applying hypothesis-specific matching tolerances. The default is <code>true</code> if <code>hypoID.dxyz</code> exists and <code>false</code> if it does not. Irrelevant when <code>runMode</code> includes <code>fetch</code> .
<code>-useFeatureCutoffs</code> <code>true false</code>	Option for applying feature-specific matching tolerances. The default is <code>true</code> if <code>hypoID.tol</code> exists and <code>false</code> if it does not. Irrelevant when <code>runMode</code> includes <code>fetch</code> .
<code>-useFeatureRules</code> <code>true false</code>	Option for applying feature-matching rules, which associate permitted and prohibited features with each site in the hypothesis. The default is <code>true</code> if <code>hypoID.rules</code> exists and <code>false</code> if it does not. If the feature rules permit any site to be matched to more than one type of feature, vector scoring is turned off. Irrelevant when <code>runMode</code> includes <code>fetch</code> .
<code>-useSiteMask</code> <code>true false</code>	Option for applying a site mask to partial matches. The default is <code>true</code> if <code>hypoID.mask</code> exists and <code>false</code> if it does not. Irrelevant when <code>runMode</code> includes <code>fetch</code> .
<code>-useExclVol</code> <code>true false</code>	Option for applying excluded volumes to hits. The default is <code>true</code> if <code>hypoID.xvol</code> or <code>hypoID.ev</code> exists and <code>false</code> if it does not. If both files exist, <code>hypoID.ev</code> is used.
<code>-useQSARModel</code> <code>true false</code>	lag for applying a QSAR model to hits. The default is <code>true</code> if <code>hypoID.qsar</code> exists and <code>false</code> if it does not.
<code>-useRefLigand</code> <code>true false</code>	Option for using a reference ligand. The default is <code>true</code> if <code>hypoID.mae</code> and <code>hypoID.tab</code> exist, and <code>false</code> if they do not. If <code>false</code> , vector and volume scoring will not be done. Irrelevant when <code>runMode</code> includes <code>fetch</code> .
<code>-alignWeight</code> <i>alignWeight</i>	Alignment score weight. Must be nonnegative. Default: 1.0.
<code>-alignCutoff</code> <i>alignCutoff</i>	Alignment score cutoff. Must be positive. Default: 1.2.

Table 13.4. Options for `phasedb_findmatches` (Continued)

Option	Description
<code>-hardAlignCutoff</code> <code>true false</code>	Option for applying <code>alignCutoff</code> as a hit filter. By default, <code>alignCutoff</code> is used only to compute fitness, not to eliminate hits. If this option is set to <code>true</code> , hits with <code>alignScore > alignCutoff</code> are rejected.
<code>-alignPenalty</code> <code>alignPenalty</code>	Partial matching alignment penalty. Must be nonnegative. Default: 1.0.
<code>-vectorWeight</code> <code>vectorWeight</code>	Vector score weight. Must be nonnegative. Relevant only when the hypothesis has a reference ligand. The default is 1.0.
<code>-vectorCutoff</code> <code>vectorCutoff</code>	Vector score cutoff. Must lie on the interval $[-1, 1]$. Hits whose vector score is lower than this value are discarded. Relevant only when the hypothesis has a reference ligand. Default: -1.0 .
<code>-volumeWeight</code> <code>volumeWeight</code>	Volume score weight. Must be nonnegative. Relevant only when the hypothesis has a reference ligand. Default: 1.0.
<code>-volumeCutoff</code> <code>volumeCutoff</code>	Volume score cutoff for filtering matches. Must lie on the interval $[0, 1]$. Hits whose volume score is lower than this value are discarded. Relevant only when the hypothesis has a reference ligand. Default: 0.0.
<code>-matchFile</code> <i>matchFile</i>	Name of match file from a previous <code>find+fetch</code> job. Valid and required only when <code>runMode</code> is <code>fetch[+flex]</code> .
<code>-writeMatchFile</code> <code>true false</code>	Option for writing matches to the file <code>jobname-matches.out</code> . The default is <code>true</code> when <code>runMode</code> is <code>find+fetch</code> or <code>find+fetch+flex</code> . Not valid with other <code>runMode</code> values.
<code>-maxHits</code> <i>maxHits</i>	Maximum number of hits to return in the file <code>jobname-hits.mae</code> . Hits with the highest fitness score are retained. Default: 1000.
<code>-maxHitsPerMol</code> <i>maxHitsPerMol</i>	Maximum number of hits per molecule. Default: 1.

Table 13.5. Job options for `phase_dbsearch`.

Option	Description
<code>-BLOCK</code> <i>m</i>	Process molecules in blocks of size <i>m</i> . This option forces memory cleanup every <i>m</i> molecules. The default is 5000 for standard searching and 1000 for flexible searching. In a distributed job, each subjob runs multiple blocks sequentially. (This option does not control the number of subjobs, which is determined by the number of CPUs.)
<code>-CHECKPOINT</code> <i>path</i>	Create checkpoint files for progress of search at the given absolute path.

Table 13.5. Job options for `phase_dbsearch`.

Option	Description
<code>-NO_CHECKPOINT</code>	Do not create checkpoint files.
<code>-RESTART path</code>	Use checkpoint files at the given absolute path to restart a search job. If a sub-job fails, you can restart it before the original master job finishes. Results from restarted jobs are merged.

Table 13.6. Optional files and their associated actions

File	Action performed
<code>hypoID.dxyz</code>	Apply hypothesis-specific tolerances when matching.
<code>hypoID.mask</code>	Apply site mask when matching.
<code>hypoID.qsar</code>	Calculate predicted activities for the provided QSAR models.
<code>hypoID.tol</code>	Apply feature-based cutoffs
<code>hypoID.rules</code>	Apply feature-based matching rules
<code>hypoID.xvol,</code> <code>hypoID.ev</code>	Apply excluded volumes to filter matches

The level of log file output from `phase_dbsearch` can be controlled by setting the environment variable `SCHRODINGER_PHASE_VERBOSITY`. A value of 0 produces minimal output; a value of 1 produces verbose output.

13.3.1 Searching Database Subsets

You can search database subsets by creating a subset with `phasedb_subset` (see [Section 13.4 on page 162](#)), and using the `-sub` option to `phasedb_findmatches`. For a tutorial example of searching using a subset, see [Section 6.15](#).

13.3.2 Examining Search Results

To view the hits in Maestro, import the file `jobname-hits.mae`. The following properties from the database search are added to the Project Table:

Ligand Name	Matched Ligand Sites	Volume Score
Conf Index	Align Score	Fitness
Num Sites Matched	Vector Score	Pred Activity(<i>n</i>)

The Pred Activity score is included only if a QSAR model was used. There will be as many predicted activity properties as there were PLS factors in the QSAR model, numbered according to the number of PLS factors in the model.

13.3.3 Applying Feature-Based Cutoffs

When you search for matches, the alignment cutoff factor specified by `-deltaDist deltaDist` is used to eliminate molecules whose intersite distances are too large. This filtering procedure does not distinguish between different kinds of features. You might want to apply a tighter cutoff for a given feature type. These cutoffs can be specified by creating a feature-matching tolerances file and saving it as `hypoID.tol` in the directory that contains the hypothesis. The format of this file is described in [Section B.9 on page 229](#). When you run `phasedb_findmatches`, the keyword `useFeatureCutoffs` is added to the database search input file with the value `true`.

The alignment cutoff factor, *deltaDist*, is still applied to identify all the initial matches, so the value of this parameter may be adjusted independently of the positional tolerances. While the mathematical relationship between *deltaDist* and positional tolerances is quite complex, choosing a value of *deltaDist* that is twice the largest positional tolerance will ensure that *deltaDist* is no more stringent than the positional tolerances. However, this usually requires assigning a fairly large value to *deltaDist* (3 or 4 Å), so the number of initial matches can be quite large, and the computational cost of searching can increase significantly. In most cases, increasing *deltaDist* is not worth the additional cost because the vast majority of the additional matches provide poor overall fits to the hypothesis, even if all of the positional tolerances are satisfied.

Cutoffs based on feature types might be too restrictive in some cases. You can specify cutoffs for each feature in a hypothesis, by creating a hypothesis-specific tolerances file and saving it as `hypoID.dxyz` in the directory that contains the hypothesis. The format of this file is described in [Section B.10 on page 229](#). When you run `phasedb_findmatches`, the keyword `useFeatureCutoffs` is set to `true` in the database search input file.

In the search, each match is checked to see whether any aligned site deviates from the hypothesis by more than the cutoff associated with that feature type or feature. The match is eliminated if a cutoff is exceeded.

13.3.4 Applying Excluded Volumes

Matches can also be filtered based on regions of space that should not be occupied by any part of the ligand, known as excluded volumes. You can create excluded volumes as part of the hypothesis generation—see [Section 12.8 on page 136](#). To apply these excluded volumes to filter the matches, copy the excluded volume file `hypoID.xvol` or `hypoID.ev` to the directory

that contains the hypothesis before running `phasedb_findmatches`. The `useExclVol` keyword is set to `true` in the database search input file.

13.3.5 Searching for Partial Matches

You can search for partial matches to a hypothesis by setting the value of the `-minSites` option to a value less than the number of sites in the hypothesis. The hits that are returned can match any of the hypothesis features. If you want to require certain features to be included in the match, you can create a file that sets a mask for matching, and use it in the search by naming the file `hypoID.mask` and saving it in the directory that contains the hypothesis before you run `phasedb_findmatches`. The keyword `useSiteMask` is then set to `true` in the database search input file. See [Section B.11 on page 230](#) for information on site mask files.

For example, suppose you had the following hypothesis coordinate file (`.xyz` file):

```
2 A 7.714 -2.723 0.686
0 D 4.737 0.34 1.532
4 P 6.512 -0.62 3.508
8 R 0.852 0.512 1.119
7 R 2.102 1.913 -0.489
```

If you set partial matching with `minSites=3`, and you required the acceptor site and the positive site to be matched, the corresponding site mask file would be:

```
2 A 1
0 D 0
4 P 1
8 R 0
7 R 0
```

13.3.6 Applying Feature-Matching Rules

You can apply rules for matching each site in the hypothesis: for example, an aromatic ring can be matched as a hydrophobe as well as a ring, or an acceptor site should never be matched as an ionic site. To set up feature rules, create a file named `hypoID.rules` according to the specifications given in [Section B.12](#) and copy it to the directory that contains the hypothesis before you run `phasedb_findmatches`. The `useFeatureRules` keyword is then set to `true` in the database search input file.

When a given site in the hypothesis is allowed to match more than one type of site in the database, this condition is described as using *mixed permitted features*. One important consequence of doing this is that all vector scoring will be turned off, so that matches between vector and non-vector features (e.g., H and R) are not automatically scored lower than matches between two vector features. Another consequence is that if you are doing a `find+fetch` search, the prescreening using 3D keys is not performed because it may require a large number of separate

SQL queries that consider all possible combinations of permitted features. This could become quite expensive, especially when partial matching is used, because a separate SQL query is already performed for each unique subset of *minSites* sites in the hypothesis.

Site masks can be used in conjunction with feature-matching rules. However, you must be careful to avoid inconsistencies, such as requiring a match to a site that has prohibited features in the rules file. In this case, the prohibited features would be irrelevant.

13.3.7 Checkpointing and Restarting Searches

If you are searching a large database, it is strongly recommended that you use the `-CHECKPOINT path` option when you launch the job, so that if the job fails to finish for any reason (such as a disk going offline or a power failure), you will be able to restart it by using `-RESTART path`. The path is the absolute path to a directory that will be used to store checkpoint files that are updated throughout the course of the search. This directory must be empty when you start the job. You must have write permissions to this directory and it must be accessible to the host on which the job is run. (It need not be accessible to the host from which you start the job.)

13.4 Creating Database Subsets: `phasedb_subset`

The `phasedb_subset` utility is used for creating and manipulating Phase database subset files. The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_subset -db database {input-options}
    -out subset
```

Table 13.7. Options for `phasedb_subset`

Option	Description
<code>-JOB jobname</code>	Job name. If omitted, the job is run in the foreground, and no Job Control options are permitted. Required if the database is on a remote host.
<code>-db database</code>	Database name, including absolute path.
<code>-hits hitfile</code>	Hit file created from a previous search of the database.
<code>-out subOut</code>	Name of subset to create. The appropriate <code>LIGAND_NAME</code> records are written to the file <code>subOut_phase.inp</code> .
<code>-in1 subset1</code>	First subset. <code>LIGAND_NAME</code> records must be stored in the file <code>subset1_phase.inp</code> .
<code>-in2 subset2</code>	Second subset. <code>LIGAND_NAME</code> records must be stored in the file <code>sub2_phase.inp</code> .

Table 13.7. Options for `phasedb_subset` (Continued)

Option	Description
<code>-logic operator</code>	Binary operator that specifies how to combine subsets. Allowed values of <i>operator</i> are AND, OR, and NOT: AND Records that are in both <i>sub1</i> and <i>sub2</i> . OR Records that are in either <i>sub1</i> or <i>sub2</i> . NOT Records that are in <i>sub1</i> but not <i>sub2</i> .
<code>-confs {true false}</code>	Create subset of molecules with multiple conformers (<code>true</code>) or molecules with only a single conformer (<code>false</code>).
<code>-sites {true false}</code>	Create a subset of molecules with pharmacophore sites (<code>true</code>) or molecules without pharmacophore sites (<code>false</code>).
<code>-titles titlefile</code>	Create a subset from the list of titles in the specified file. There must be one title per line in the file. All molecules that match the titles are added to the subset; if several molecules have the same title, they are all added.

The syntax of the input options is given in the following three usage scenarios. The descriptions of the options are given in [Table 13.7](#).

To create a subset from the structures in a hit file:

```
phasedb_subset -db database -hits hitfile -out subset
```

To create a subset from a logical operation on two existing subsets:

```
phasedb_subset -db database -in1 subset1 -logic operator -in2 subset2  
-out subset
```

To create a subset from a list of titles:

```
phasedb_subset -db database -titles titlefile -out subset
```

To create a subset from a query of the database:

```
phasedb_subset -db database {-confs|-sites} {true|false} -out subset
```

This usage allows you to select molecules for which conformers have or have not been generated or for which sites have or have not been created.

For a tutorial example of the use of this utility, see [Section 6.15](#) of the *Phase Quick Start Guide*.

13.5 Exporting Structures from Databases: phasedb_export

This utility allows you to export structures, including conformer sets, from a database to one or more structure files. May be run as a regular foreground process, or as a single-CPU job on any host that has access to the database directory. The syntax is as follows:

```

$SCHRODINGER/utilities/phasedb_export -db dbName -ofmt baseName
  [job-options] [options]

```

The options are described in Table 13.8. The job options listed in Table 13.1 are supported.

Table 13.8. Options for *phasedb_export*.

Option	Description
-JOB <i>jobname</i>	Job name. If omitted, the job is run in the foreground, and no Job Control options are permitted.
-db <i>dbName</i>	Database name, including absolute path. Required.
-ofmt <i>baseName</i>	Output file specification. <i>fmt</i> must be <i>mae</i> or <i>sd</i> . Output files are named <i>baseName_1.ext</i> , <i>baseName_2.ext</i> , and so on, where <i>ext</i> is <i>mae</i> or <i>sd</i> . If running as a job, <i>baseName</i> must be an absolute path. Required.
-sub <i>dbSubset</i>	Export only a subset of the database. The file <i>dbSubset_phase.inp</i> must contain the applicable LIGAND_NAME records. The default is to export all records.
-get <i>maxConfs</i>	Maximum number of conformations to extract for each database molecule. By default, all conformations are extracted.
-limit <i>maxStruct</i>	Maximum number of structures per output file. The default is 100,000. Note that conformations of a given molecule will <i>not</i> be split across multiple output files.
-quota <i>Gbytes</i>	Enforce a quota (in gigabytes) on the total disk space consumed by the output files. Disk space usage is checked at intervals of 1000 output structures (plus any overrun that occurs when multiple conformations are exported for a given molecule). The program will abort if the quota is exceeded at any checkpoint. By default, no quota is enforced.
-gz	Compress each output file using <i>gzip</i> . If <i>-quota</i> is used, the quota is enforced before compressing the latest output file (i.e., peak disk space usage).
-ext <i>ext</i>	Extension for compressed files. The default for Maestro files is <i>maegz</i> , and the default for SD files is <i>sd.gz</i> . The extension cannot be <i>mae</i> or <i>sd</i> .

13.6 Extracting Properties from a Database: phasedb_props

Properties stored in a phase database are kept in the HDF5 files, and are not generally accessible. You can use this utility to extract properties, query the database using the properties, and create a subset file using the query. May be run as a single-CPU job on any host that has access to the database directory. The syntax for the two tasks (extract and query) is as follows:

```
$SCHRODINGER/utilities/phasedb_props [job-options] -extract dbName
    -props propsFile [-csv csvFile]
```

```
$SCHRODINGER/utilities/phasedb_props [job-options] -query string
    -props propsFile [-sub subsetName] [-csv csvFile]
```

The options are described in [Table 13.9](#). The job options listed in [Table 13.1](#) are supported.

Table 13.9. Options for *phasedb_props*.

Option	Description
-extract <i>dbName</i>	Extract properties from an existing Phase database with the specified name, including the full path. Not valid with -query.
-query <i>string</i>	Perform SQL query against extracted properties. The supplied string should be enclosed in quotes and it should conform to the syntax of a SQL WHERE clause, e.g., "mol_id < 1000". Not valid with -extract.
-sub <i>subsetName</i>	Create the subset file <i>subsetName_phase.inp</i> from the records that match the query. Valid only with -query.
-props <i>propsFile</i>	Properties file that will be created or queried. An existing file will not be overwritten. If running as a job, use absolute path to avoid file copy. Required.
-csv <i>csvFile</i>	Comma-separated file to be created. If -extract is used, this file contains the names of all the properties written to <i>propsFile</i> . If -query is used, this file contains the full set of properties for each record in <i>propsFile</i> that matches the query.

The extracted property values come only from the first conformer stored for each molecule, so you should avoid queries on properties that depend on the 3D structure, such as *r_mmod_Potential_Energy-MMFF94s*, because the reported value for a given record is not generally representative of the entire conformational ensemble.

For a tutorial example of the use of this utility, see [Section 6.16](#) of the *Phase Quick Start Guide*. Some examples of queries are given below, assuming that the property `r_user_Activity` is present:

1. Match values of `r_user_Activity` greater than 7.5:

```
-query "r_user_Activity > 7.5"
```

2. Match values of `r_user_Activity` in the range 7.5 to 8.0:

```
-query "r_user_Activity BETWEEN 7.5 AND 8.0"
```

3. Match if title is equal to `endo-8` (note that extra quotes are needed in the query expression):

```
-query "s_m_title = 'endo-8'"
```

4. Match titles that start with `endo`:

```
-query "s_m_title LIKE 'endo%'"
```

5. Match titles that don't start with `endo`:

```
-query "s_m_title NOT LIKE 'endo%'"
```

6. Match if title starts with `endo` and `r_user_Activity` is greater than 8.0:

```
-query "s_m_title LIKE 'endo%' AND r_user_activity > 8.0"
```

7. Match if title is equal to any of the titles in a list:

```
-query "s_m_title IN ('endo-1', 'endo-3', 'endo-5')"
```

13.7 Merging Databases: `phasedb_merge`

This utility allows you to merge two databases. The main purpose of this utility is to facilitate the creation of several smaller databases independently and then perform a merge operation to create a single large database. For creation of a large database, this strategy makes it easier to recover and rerun subjobs in case of failures. It is assumed that databases that are merged were created using the same feature definitions: no checking is done to verify that this is so. However, you can use the `phasedb_check` utility ([Section 13.9](#)) after a merge operation for verification.

The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_merge arguments
```

The arguments are required and may be given in any order. They are described in [Table 13.10](#).

Table 13.10. Required arguments for `phasedb_merge`

Argument	Description
<code>-dest target-name</code>	Target database name, not including path. This is the database into which data from the source database will be merged.
<code>-source source-name</code>	Source database name, including path. This is the database from which data will be merged into the target database.

13.8 Converting a Database: `phasedb_convert`

The utility `phasedb_convert` is a tool for reformatting Phase 3D databases. It supports forward conversion of databases from previous versions of Phase, and importing databases into a target database using the current version of Phase. The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_convert -source dbSource -target dbTarget
[job-options] [conversion-options]
```

The job control options listed in Table 13.1 are supported. The `-JOB jobname` job option must be supplied, since this utility is run as a job. The job can be distributed over multiple processors. The source and target databases do not have to be accessible to the host from which the job is started, but they must be accessible at the path provided on the execution host.

Conversion options are listed in Table 13.11.

Table 13.11. Conversion options for `phasedb_convert`

Option	Description
<code>-source dbSource</code>	The database containing the records to be converted. <i>dbSource</i> should be of the form <i>dbPath/dbName</i> . If the database was created using the Phase 1.0 GUI, <i>dbName</i> should always be <code>phasedb</code> . Required.
<code>-target dbTarget</code>	The database that will receive the converted records. <i>dbTarget</i> should also be of the form <i>dbPath/dbName</i> , but it cannot be the same as <i>dbSource</i> . If the database was created using the Phase 1.0 GUI, a different directory <i>dbPath</i> must be used for source and target. Required.
<code>-fd fdFile</code>	Use pharmacophore feature definitions in <i>fdFile</i> when creating a new target database. If this option is omitted, the default definitions in the Phase installation are used. This option is illegal if the target database already exists.
<code>-new</code>	Create a new target database in Phase 3.1 (HDF5) format. This option is illegal if the target database already exists.

Table 13.11. Conversion options for `phasedb_convert` (Continued)

Option	Description
<code>-records recordFile</code>	File containing the source records to convert. <i>recordFile</i> should contain a series of lines of the form LIGAND_NAME = <i>sourceLigandName</i> If omitted, the entire database is converted.
<code>-RESTART</code>	Restart a failed database conversion. For complete instructions on restarting, see the file <i>dbName_convert_restart/README</i> , where <i>dbName</i> is the target database name, including absolute path.
<code>-blimit maxMol</code>	Maximum number of molecules per target database block. The default and largest allowed value is 5000. Note that the starting molecule IDs in each target block will always be 1, 5001, etc., so if <i>maxMol</i> is less than 5000, there will be unused IDs.
<code>-sites</code>	Specify how sites are handled. Allowed values are: new Create new sites as records are added to the target database. copy Copy existing source sites to the target database. Valid only when the source database was created using the current version of Phase and when the source and target feature definitions are identical. skip Do not create sites in the target database. If this option is omitted, it is automatically set to <code>new</code> or <code>copy</code> , depending on the characteristics of the source database. If the source database was created prior to Phase 2.5, new sites are always created unless you specify <code>-sites skip</code> .

13.9 Checking Database Integrity: `phasedb_check`

The purpose of this utility is to check database integrity and identify possible inconsistencies. Some of the tests performed include: checking the database access file, checking that records stored in the SQLite file corresponds to data stored in the HDF5 file, checking the feature definition file, testing database blocks. This utility should be run whenever a database undergoes extensive modifications or whenever there is a suspicion that a database creation operation has failed. This utility should help to identify parts of the database that have been corrupted so that if the job needs to be rerun it only needs to be done on a subset of the database.

The command syntax is as follows:

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

The options for this command are listed in [Table 13.12](#).

Table 13.12. Options for `phasedb_check`

Option	Description
<code>-db dbName</code>	Database name, including path. Required.
<code>-blocks_file filename</code>	Optional file containing a list of database blocks to be checked. Each line has the format BLOCK_ID <i>n</i> where <i>n</i> is the integer block ID.

13.10 Database Backup and Recovery: `phasedb_recovery`

This utility is used to back up Phase database data and recover it in cases when a database becomes corrupted. When the backup operation is performed, only the first conformation is stored, and the site data is discarded. This utility can be run on an entire database or on a selection of database blocks. Thus, if only a few blocks were corrupted, the restoration can be limited to the affected blocks. The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_recovery {-backup|-restore} [options]
```

The options for this command are listed in [Table 13.13](#).

Table 13.13. Options for `phasedb_recovery`.

Option	Description
<code>-db db-name</code>	Database name including path. Required.
<code>-db_backup backup-name</code>	Backup database name, including path. Required.
<code>-backup</code>	Back up the database.
<code>-restore</code>	Restore the database from the specified backup.
<code>-add_to_dest</code>	Append data to the backup database when backing up. By default, existing blocks in the backup database are overwritten.
<code>-blocks_file filename</code>	Optional file containing a list of database blocks to be backed up or recovered. Each line has the format BLOCK_ID <i>n</i> where <i>n</i> is the integer block ID.

13.11 Compacting Database HDF5 Files: phasedb_compact

This utility reduces the size of Phase database HDF5 files, by copying all data into a new HDF5 file, which replaces the original one if the copy operation succeeds. This is done to overcome a known limitation in the HDF5 library, in which disk space is not released when records are deleted from HDF5 files. Thus, it is only necessary to run this utility when a large number of molecules or molecule conformations are deleted from the database. This operation can be expensive when performed on a very large database.

The command syntax is as follows:

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

The options for this command are listed in [Table 13.14](#).

Table 13.14. Options for *phasedb_compact*

Option	Description
-db <i>db-name</i>	Database name, including path. Required.
-blocks_file <i>block-file</i>	Optional file containing a list of database blocks to be checked. Each line has the format BLOCK_ID <i>n</i> where <i>n</i> is the integer block ID.

13.12 Other Utilities

Several other utilities that may be useful in scripting are provided, as described below.

13.12.1 phasedb_count_records

This is a utility program to count the number of LIGAND_NAME records in a Phase database subset file. The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_count_records -rec recordsFile
  [-count countFile]
```

recordsFile is the name of the file containing the LIGAND_NAME records, and *countFile* is the file to which the number of records is written. If -count is omitted, the count is written to standard output.

13.12.2 phasedb_split_records

This is a utility program to split a ligand records file for subjob processing. The files created are named *jobname_sub_0_phase.inp*, *jobname_sub_1_phase.inp*, and so on. The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_split_records -rec recordsFile
      -job jobname -nsub nsub [-minRec minRec]
```

The options are described in [Table 13.15](#).

Table 13.15. Options for *phasedb_split_records*

Option	Description
-rec <i>recordsFile</i>	File to be split.
-job <i>jobName</i>	Job name.
-nsub <i>nsub</i>	Number of subjobs.
-minRec <i>minRec</i>	Minimum number of records per subjob. The number of subjob files created is reduced as necessary.

13.12.3 phasedb_match_keys

This utility program can be used to pre-screen a Phase database using 3D keys. The information in the keys is the set of distance ranges for pairs of site points of each type, taken across all conformers. This prescreening is done automatically in database searching, so this utility may be useful in a scripting context. The prescreening results in a superset of hits.

The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_match_keys -db dbName -hypo hypoID
      [-dtol deltaDist] [-minSites minSites] [-sub dbSubsetFile] -rec recordsFile
```

The options are described in [Table 13.16](#).

Table 13.16. Options for *phasedb_match_keys*

Option	Description
-db <i>dbName</i>	Absolute path and name of database.
-hypo <i>hypoID</i>	Hypothesis ID. The file <i>hypoID.xyz</i> must exist.
-dtol <i>deltaDist</i>	Intersite distance matching tolerance. The default is 2.0.

Table 13.16. Options for `phasedb_match_keys` (Continued)

Option	Description
<code>-minSites <i>minSites</i></code>	Minimum number of hypothesis sites to match. The default is to match all sites.
<code>-sub <i>dbSubsetFile</i></code>	Perform logical AND of matched records with the records in <i>dbSubsetFile</i> .
<code>-rec <i>recordsFile</i></code>	File to which matched records should be written. This file contains the <code>LIGAND_NAME</code> records for the matches.

13.12.4 `phasedb_index`

This utility creates indices for 2D and 3D database keys in a Phase database. Creating the index increases the speed of record deletion and also key matching in a database search. This utility is automatically run for databases created with Schrödinger Suite 2007 Update 2. You need only run this utility once to create the index for existing databases.

The command syntax is as follows:

```
$SCHRODINGER/utilities/phasedb_index -db dbName
```

13.13 Running on Multiple Processors

Site creation jobs and database search jobs can be split across multiple processors on an appropriately configured cluster. The basic requirements for database creation and searching are as follows:

- The database must be located in a directory that is uniformly accessible to all nodes of the cluster on which jobs will be run.
- If the file system on which the database is stored is only accessible to the cluster, you must be logged into the manager node of the cluster to start jobs.
- In the `$SCHRODINGER/schrodinger.hosts` file, each parallel queue that is used for database jobs should have a `tmpdir` entry with a path that is accessible to all nodes. See the *Installation Guide* for details of setting a `tmpdir` entry.

When you start a `phase_dbcreate` or `phase_dbsearch` job, the `-HOST` option should specify the processors to use. If the processors are on a single host, you can add the number of processors after the host name and a colon—for example, `-HOST cluster:4`. One subjob is started on each processor. The molecules in the database are distributed to the subjobs, one at a time, until the list is exhausted. For this example, the first subjob would process molecules 1, 5, 9, and so on; the second subjob would process molecule 2, 6, 10, and so on. This scheme

provides near optimal load balancing as it is very unlikely that any one processor would have to process a disproportionate number of expensive molecules.

13.14 Granting Access to a Database

When you create a database, you are the owner of all the files and directories associated with that database. As such, you will normally have read and write permissions to the files, and read, write, and execute permissions to the directories. You should therefore be able to modify or search a database that you create yourself. However, you may or may not want other users to be granted those same privileges.

If you want to be certain that you are the only person who will ever modify or search a particular database, then you should remove write permissions for all other users throughout the database tree, and remove read permission from the database. By default, the files and directories you create will normally not carry write permissions for other users, unless you change the default `umask`.

If you want to allow other users to search the database, but not to modify it, you must grant those users read and execute permissions to all directories throughout the database tree. The same read and execute permissions must also exist upward from the database directory to the file system mount point.

To remove all permissions for other users:

```
chmod -R g-wx,o-wx dbpath
chmod g-rwx,o-rwx dbpath/dbname_ligands
```

To grant permission to search but not to modify the database:

```
chmod g+rx dbpath dbname_ligands dbname_ligands/block*
```

This command gives permissions only to members of your group, and assumes that they have execute permission to all directories above *dbpath*, but do not have write permission to the database. To grant permission to everyone, you could add the `o+rx` permission code.

13.15 Checking Job Progress and Completion

The progress of jobs can be monitored in the Monitor panel, or using the `jobcontrol` command. Job Control reports on the status of the job when it finishes. You can also check on the progress of the job in the log file, which is named *jobname_process.log* for the processes of database management (`manage`), database conversion (`convert`), conformer and site generation (`confsites`) and database searching (`dbsearch`). This file is displayed in the

Monitor panel and is updated periodically. For more information on job monitoring, see [Chapter 3](#) of the *Job Control Guide*.

In addition to the log file, a file named `jobname_process.okay` is written when the job finishes successfully. The presence of this file can be used to verify that the job has finished—for example, in a script that is used to run one or more jobs.

A progress file is written by the programs `phase_dbsearch`, `phasedb_confsites`, and `phasedb_convert` for each job or subjob, with the suffix `_progress.txt`. By default, the file is written to the job or subjob directory, but you can specify the location for these files by setting the environment variable `SCHRODINGER_PHASE_PROGRESS_DIR` to a directory that is accessible from the execution hosts.

Searching Files for Matches from the Command Line

In addition to searching a previously prepared database for matches, Phase provides two programs for searching files for matches. The first, `phase_fileSearch`, can be run only on a single processor. The second, `phase_gridSearch`, can be run on multiple processors.

The same requirements on the structures apply to searching a file as for adding structures to a database: the structures must be all-atom, 3D structures. If the structures do not meet these requirements, you should convert them using `LigPrep`, for example (see the *LigPrep User Manual*).

14.1 Searching Files with `phase_fileSearch`

If you have a relatively small number of structures to search for matches, you can run `phase_fileSearch`. This program only runs on a single processor, so is not suited for large numbers of structures. If you want to distribute a large set of structures across multiple processors, consider using `phase_gridSearch` or creating a database and using `phase_dbsearch`.

Before you use `phase_fileSearch`, you must set up the input file for the search. You can do this with the utility `pharm_align_mol`. See [Section 12.9.2 on page 140](#) for a description of this utility, and see [page 113](#) for a summary of the syntax of using `pharm_align_mol` with `phase_fileSearch`.

Syntax

```
phase_fileSearch [job-options] jobname
```

The file `jobname_fileSearch.inp` must contain the keyword=value pairs required to run this job. This file can be set up with the utility `pharm_align_mol`. Not all of the settings can be made with `pharm_align_mol`, so you might need to edit the input file. The syntax of this file is described in [Section B.14 on page 235](#). The standard job control options listed in [Table 13.1](#) can be used with this program.

14.2 Searching Files with `phase_gridSearch`

For large sets of structures that are stored in SD format, you can use `phase_gridSearch` to search for matches. This program can be run on multiple CPUs. If you choose only a single processor for this job, the job runs on the local host regardless of which host you selected. This program is not intended for single-processor use, so you should always ensure that multiple processors are specified with the `-HOST` option.

When you run `phase_gridSearch`, the search input file is prepared automatically using the options that you specify. However, you must also prepare a file that contains a list of files to be searched, one file name per line (even if you only want to search one file). The file names should include the absolute path to the file. All files should be either standard SD files or compressed SD files. You cannot have both standard and compressed files in the same run.

The structures that are searched are distributed across the available processors by adding structures one at a time to temporary files, one file per processor. When a predetermined number of structures has been added to a temporary file, the file is closed and compressed with `gzip`, and a new temporary file is opened for that processor. If you are using a large number of processors, you may want to set the number of structures per temporary file to a small value to ensure that these files do not take up too much disk space. The distribution process is run on the local host and the temporary files are stored in the current working directory, so you should ensure that you have enough disk space in this directory to store all the temporary files.

Syntax

```
phase_gridSearch -structFileList fileName -hypoID hypoID -maxHits maxHits  
[options] [search-options] [job-options] jobname
```

The arguments and options specific to `phase_gridSearch` are described in [Table 14.1](#). Search options are described in [Table 14.2](#). These options correspond to keywords in the database search input file, which is described in detail in [Section B.13 on page 232](#). The standard job control options listed in [Table 13.1](#) can be used with this program. Two additional job options are listed at the end of [Table 14.1](#).

Table 14.1. Options for `phase_gridSearch`

Option	Description
<code>-structFileList</code> <i>fileName</i>	Name of the file containing the list of SD files to search. The SD files may be all in standard form, or all compressed via <code>gzip</code> , but mixing of standard and compressed files is not permitted. Compressed file names should have a <code>.gz</code> extension.
<code>-hypoID</code> <i>hypoID</i>	Prefix used to name all hypothesis files. The files <i>hypoID</i> . <code>def</code> and <i>hypoID</i> . <code>xyz</code> must always be present. Other files are optional.
<code>-maxHits</code> <i>maxHits</i>	Maximum total number of hits that will be returned in the file <i>jobname</i> - <code>hits.mae</code> . If the maximum limit is reached, only the best hits are kept.
<code>-MAXSTRUCT</code> <i>n</i>	Maximum number of structures written to each temporary input file before it is compressed. The default is 1000. Use a smaller value to reduce the amount of uncompressed data on disk at any given time.
<code>-SPLIT</code> <code>true false</code>	Option to split input SD files into a series of temporary SD files of equal size so that computations are divided equally among all CPUs. Splitting is done by default if multiple CPUs are requested. If splitting is turned off, the number of CPUs requested cannot exceed the total number of input SD files.

Table 14.2. Search options for the `phase_gridSearch` and `pharm_align_mol` commands

Option	Description
<code>-flexSearchMethod</code> <i>method</i>	Conformational sampling method. Allowed values are <code>rapid</code> and <code>thorough</code> . Default: <code>rapid</code> .
<code>-flexMaxConfs</code> <i>maxConfs</i>	Maximum number of conformations/molecule to generate. If zero, no conformations will be generated, and the supplied structures will be searched directly. Default: 100.
<code>-flexConfsPerBond</code> <i>maxPerBond</i>	Maximum number of conformations per rotatable bond to generate for each molecule. Default: 10.
<code>-flexMaxRelEnergy</code> <i>energy</i>	Conformational energy window in kJ/mol. Default: 41.84 kJ/mol (10 Kcal/mol).
<code>-flexAmideOption</code> <i>option</i>	Flexible amide torsion sampling option. Allowed values are <code>vary</code> , <code>orig</code> , and <code>trans</code> . Default: <code>vary</code> .
<code>-deltaDist</code> <i>deltaDist</i>	Intersite distance matching tolerance in angstroms. Default: 2.0.
<code>-minSites</code> <i>minSites</i>	The minimum number of hypothesis sites that must be matched. Must be 3 or greater. The default is to match all sites.

Table 14.2. Search options for the *phase_gridSearch* and *pharm_align_mol* commands

Option	Description
<code>-preferBigMatches</code> <code>true false</code>	Option that governs whether partial matches containing a greater number of sites should be favored. If set to <code>true</code> , matches involving fewer than <i>n</i> sites will not be sought if there are any matches with <i>n</i> sites. The default is <code>true</code> .
<code>-scoreInPlace</code> <code>true false</code>	Option that governs whether matches should be scored in place. If set to <code>true</code> , no conformations are generated, and fitness is computed directly from the supplied poses, without aligning to the hypothesis. The default is <code>false</code> .
<code>-timeLimit</code> <i>timeLimit</i>	CPU time limit in seconds for finding matches for each molecule. CPU usage is checked before each conformer from a given molecule is searched, and matching is terminated if <i>timeLimit</i> is exceeded at that point. The time limit does not apply to the process that generates conformers during the search, so if any of the <code>flex</code> options are specified and a molecule has a large number of rotatable bonds, the overall CPU time may significantly exceed the imposed limit. The default is unlimited CPU time.
<code>-atomTypeVol</code> <code>true false</code>	Option for computing volume scores using overlap only between atoms of the same MacroModel atom type. This favors alignments that superimpose chemically similar atoms. The default is <code>false</code> .
<code>-alignWeight</code> <i>alignWeight</i>	Alignment score weight. Must be non-negative. Default: 1.0.
<code>-alignCutoff</code> <i>alignCutoff</i>	Alignment score cutoff. Must be greater than zero. Default: 1.2.
<code>-hardAlignCutoff</code> <code>true false</code>	Option for applying <i>alignCutoff</i> as a hit filter. By default, <i>alignCutoff</i> is used only to compute fitness, not to eliminate hits. If this option is set to <code>true</code> , hits with <code>alignScore > alignCutoff</code> are rejected.
<code>-alignPenalty</code> <i>alignPenalty</i>	Partial matching alignment penalty. Must be non-negative. Default: 1.2.
<code>-vectorWeight</code> <i>vectorWeight</i>	Vector score weight. Must be non-negative. Relevant only when the hypothesis has a reference ligand. Default: 1.0.
<code>-vectorCutoff</code> <i>vectorCutoff</i>	Vector score cutoff. Must be on the interval [-1, 1]. Relevant only when the hypothesis has a reference ligand. Default: -1.0.
<code>-volumeWeight</code> <i>volumeWeight</i>	Volume score weight. Must be non-negative. Relevant only when the hypothesis has a reference ligand. Default: 1.0.
<code>-volumeCutoff</code> <i>volumeCutoff</i>	Volume score cutoff. Must be on the interval [0, 1]. Relevant only when the hypothesis has a reference ligand. Default: 0.0.

Table 14.2. Search options for the `phase_gridSearch` and `pharm_align_mol` commands

Option	Description
<code>-maxHitsPerMol</code> <i>maxPerMol</i>	Maximum number of hits per molecule. Default: 1.
<code>-useRefLigand</code> {true false}	Use reference ligand information. If this option is set to <code>true</code> , the files <code>hypoID.mae</code> and <code>hypoID.tab</code> must be present, or the job will fail. Default: use a reference ligand if the files <code>hypoID.mae</code> and <code>hypoID.tab</code> are present.
<code>-useFeatureRules</code> {true false}	Apply feature-matching rules, which associate permitted and prohibited features with each site in the hypothesis. If the feature rules permit any site to be matched to more than one type of feature, vector scoring is turned off. If this option is set to <code>true</code> , the file <code>hypoID.rules</code> must be present, or the job will fail. Default: apply the rules if the file <code>hypoID.rules</code> is present.
<code>-useSiteMask</code> {true false}	Apply a site mask in partial matching (i.e., requiring certain sites to match). If this option is set to <code>true</code> , the file <code>hypoID.mask</code> must be present, or the job will fail. Default: apply a site mask if the file <code>hypoID.mask</code> is present.
<code>-useFeatureCutoffs</code> {true false}	Apply feature-based matching tolerances. If this option is set to <code>true</code> , the file <code>hypoID.tol</code> must be present, or the job will fail. Default: apply these tolerances if the file <code>hypoID.tol</code> is present.
<code>-useDeltaHypo</code> {true false}	Apply hypothesis-specific matching tolerances. If this option is set to <code>true</code> , the file <code>hypoID.dxyz</code> must be present, or the job will fail. Default: apply these tolerances if the file <code>hypoID.dxyz</code> is present.
<code>-useQSARModel</code> {true false}	Apply a QSAR model to the hits. If this option is set to <code>true</code> , the file <code>hypoID.qsar</code> must be present, or the job will fail. Default: apply a QSAR model if the file <code>hypoID.qsar</code> is present.
<code>-useExclVol</code> {true false}	Apply excluded volumes to filter the hits. If this option is set to <code>true</code> , the file <code>hypoID.xvol</code> must be present, or the job will fail. Default: apply excluded volumes if the file <code>hypoID.xvol</code> is present.

Searching for Molecules by Shape

There are occasions on which the shape of the molecule is its most important feature, and a search for molecules that are most similar in shape is needed. Phase provides this capability with the `phase_shape` program.

The `phase_shape` program can be used to screen one or more files or a Phase database against a shape query. Each conformer from a given molecule is aligned to the query in various ways, and a similarity is computed based on overlapping hard-sphere volumes. The conformer and alignment yielding the highest similarity for each molecule is written to a Maestro file, along with the similarity property `r_phase_Shape_Sim`, which appears in the Project Table as Shape Sim.

The shape query can be a single template molecule, or it can be a set of three or more spheres. The latter form provides a high degree of flexibility in designing the shape. For each molecule searched, `phase_shape` returns an aligned structure that provides the best overlap with the shape query. If you are searching a set of homologous structures and the shape query is a member of the same series, then `phase_shape` should return the most sensible alignment among all those available. If you are searching against structures that are not necessarily related to the shape query, `phase_shape` should return something that looks most like the query in an overall sense.

The shape search can treat all atoms as equivalent, or it can incorporate information on atom types as part of the search. Searching on atom types favors alignments that superimpose atoms of the same type. There are three possibilities for atom typing in a shape search: use of MacroModel types, typing by element, and use of Phase QSAR types. MacroModel atom types will impose the most stringent conditions on the matching, and Phase QSAR types will impose the most general conditions.

Volume scoring as part of a search for matches to a hypothesis also uses molecular shape, but there are some important differences between volume scoring and shape-based queries. In volume scoring the molecules are aligned to the hypothesis, and the volume overlap is then computed on the basis of that alignment. The molecular shape alignment might not be optimal in this case. Shape queries investigate a much greater variety of alignments than a typical search for matches to a hypothesis. A second difference is in the algorithm used to evaluate the volume scores on the one hand, and the shape similarities on the other.

The search by shape can be set up and run from Maestro or run from the command line.

15.1 Running Shape Searches from Maestro

To run a shape search from Maestro, you use the Shape Screening panel, which you open from the Phase submenu of the Applications menu.

To set up a search by shape:

1. Select the source of the shape query from the Use shape query from option menu.

The options are Workspace, Project Table (selected entries) and File.

If you selected File, enter the file name in the Shape query file text box or click Browse to browse to the file. You can read Maestro files (.mae), SD files (.sdf), or Phase included volume files (.ivol).

If you selected either of the other two options and there are multiple entries selected in the Project Table, a query is constructed for each entry.

2. Enter the file name or database name in the Screen file or Phase DB text box, or click Browse to browse to the file or database.

The file can be a Maestro file or an SD file, and can be compressed or uncompressed.

3. (Optional) Enter the name of a Phase database subset to search, or click Browse to browse to the database subset file.

Database subset files end in `_phase.inp`.

4. (Optional) Enter the name of an excluded volumes (.xvol or .ev) file in the Excluded volumes text box, or click Browse to browse to the file.

This file is optional. If it is specified, the excluded volumes in the file are applied in the search.

5. Choose an atom type to use in volume scoring. The volume overlaps used to compute the similarity are only calculated for atoms of the same type. The choices are:

- None—Do not distinguish different types of atoms when calculating volume overlaps: all atoms are treated the same.
- MacroModel—Calculate volume overlaps only between atoms that have the same MacroModel atom type.
- Element—Calculate volume overlaps only between atoms of the same element.
- Pharmacophore—Calculate volume overlaps between atoms that have the same pharmacophore type (Acceptor, Donor, etc.) as defined for Phase QSAR models.

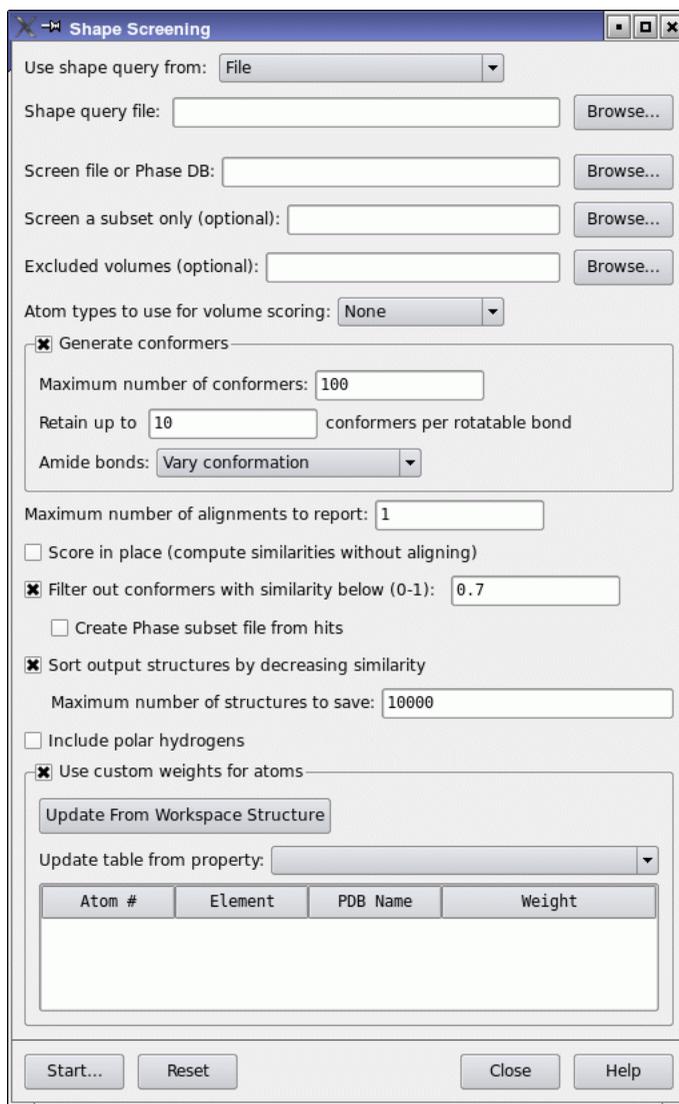


Figure 15.1. The Shape-Based Screening panel.

6. If the structures to be searched do not include conformer sets, you can choose to generate conformers, by selecting Generate conformers.

If you want to limit the number of conformers generated per input structure, enter a value in the Maximum number of conformers text box, or enter a value in the Retain up to N conformers per rotatable bond text box, or both.

If you are generating conformers, choose an amide bond treatment from the Amide bonds optim menu. The choices are:

- Vary conformation—allow the conformation around the amide bond to vary freely.
 - Retain original conformation—do not vary the conformation around the amide bond, but keep the original conformation.
 - Set to trans—Set the conformation around the amide bond to trans (180°).
7. (Optional) If you want more than one alignment of a given molecule to a given query, enter the number of alignments in the Maximum number of alignments to report text box. By default, only one alignment is returned.
 8. (Optional) To compute similarities without aligning the molecules, select Score in place.
 9. (Optional) If you want to filter out conformers whose similarity to the query is less than a certain value, select Filter out conformers with similarity below and enter a value in the text box. The value must be in the range 0–1.

If you are searching a Phase database and want to create a subset file based on the filter, select Create Phase subset file from hits.

10. (Optional) If you want to sort the results by similarity, select Sort output structures by decreasing similarity, and enter a limit on the number of structures in the Maximum number of structures to save text box.
11. (Optional) If you want polar hydrogens (which are almost always donors) to be considered in the shape search, select Include polar hydrogens.
12. (Optional) If you want to provide weights for particular atoms in the queries, select Use custom weights for atoms, choose an atom-level property from the Update table from property option menu, and click Update.

The table is filled in with the list of atoms in the query with the defined weight. The weights are used to scale the atomic volumes used to compute the overlaps. The values must be between 0 and 1. You should choose or create a property that scales down the volume of the less-important atoms.

13. Click Start to enter job settings in the Start dialog box and start the job.

When the job finishes, the results are appended to the Project Table if you chose to incorporate them, with the property Phase Sim added.

15.2 Running Shape Searches from the Command Line

The command syntax for a shape search is as follows:

```
$SCHRODINGER/phase_shape -JOB jobname
    {CHECKPOINT | -RESTART path | -NO_CHECKPOINT}
    [program-options] [job-options]
```

The job options are the same as for other Phase programs, and are described in [Table 13.1 on page 148](#). The program options and the required arguments are described in [Table 15.1](#). When you run the job, you must specify whether you want to create checkpoint files or not. If you do not create checkpoint files, the job cannot be restarted. To restart a job, you can use the syntax:

```
$SCHRODINGER/phase_shape -JOB jobname -RESTART path [-osub dbSubOut]
    [job-options]
```

Table 15.1. Options and arguments for *phase_shape*.

Option	Description
-atomTypes <i>atomType</i>	Consider atom types when computing similarities. If this option is used, overlapping volumes are computed only between atoms of the same type, so that alignments favor superposition of chemically similar atoms. The supported atom typing schemes are: mmod Use MacroModel atom types element Type by element pharm Use generalized pharmacophore types, as defined for Phase QSAR models: D—H-bond donor hydrogen H—hydrophobic/non-polar N—negative ionic P—positive ionic W—electron-withdrawing X—other Not valid if the query shape is composed of included volumes.
-atomWeights <i>propName</i>	Use the real atom-level property <i>propName</i> in <i>shapeFile</i> to weight the overlap with the query atoms. This is achieved by scaling the radius of each atom by the cube root of its weight. Values in the range 0 to 1 are recommended. Valid only when <i>shapeFile</i> is a Maestro file.
-best	For each screened molecule, report results only for the query that yields the highest similarity. Valid only when multiple queries are supplied.

Table 15.1. Options and arguments for *phase_shape*. (Continued)

Option	Description
-CHECKPOINT <i>path</i>	Create checkpoint files in the directory <i>path</i> . If this option is not used on the initial submission, it will <i>not</i> be possible to restart the job if it fails. The specified path must be absolute and it must be a single, empty directory that is mounted uniformly on the chosen host and all compute nodes on which subjobs will run. It must <i>not</i> be the current working directory. The file <i>path</i> /README will contain instructions for restarting the job if it terminates abnormally.
-distinct	When screening one or more files, treat each structure as a distinct molecule (i.e., one conformer only). Not valid with -flex options.
-dual	Report similarities with and without atom typing, from the alignment obtained using atom typing. The Maestro property Shape Sim Pure (<i>r_phase_shape_sim_pure</i>) is added to the output Maestro structure file, <i>jobname_align.maegz</i> for each aligned structure. This property is the similarity calculated without atom typing. Valid only with -atomTypes.
-filter <i>minSim</i>	Filter out molecules whose similarities fall below <i>minSim</i> . Default: do not apply the filter.
-flex	Generate conformers during the search. Default: Use existing conformers.
-flexAmideOption <i>option</i>	Flexible amide torsion sampling option. Allowed values are vary, orig, and trans. Default: vary.
-flexMaxConfs <i>maxConfs</i>	Maximum number of conformations/molecule to generate. If zero, no conformations will be generated, and the supplied structures will be searched directly. Default: 100.
-flexConfsPerBond <i>numPerBond</i>	Maximum number of conformations per rotatable bond to generate for each molecule. The total number of conformers is bounded by the product of <i>numPerBond</i> and the number of rotatable bonds. If <i>maxConfs</i> is increased, <i>numPerBond</i> may have to be increased as well in order to retain additional conformers for more rigid structures. Default: 10.
-flexMaxRelEnergy <i>energy</i>	Conformational energy window in kJ/mol. Default: 41.84 kJ/mol (10 Kcal/mol).
-flexSearchMethod <i>method</i>	Conformational sampling method. Allowed values are rapid and thorough. Default: rapid.
-hydrogens	Consider hydrogens attached to non-carbon atoms when computing shape similarity. Hydrogens attached to carbon atoms are always ignored. Default: ignore all hydrogens.

Table 15.1. Options and arguments for *phase_shape*. (Continued)

Option	Description
-isub <i>dbSubIn</i>	Screen a subset of a Phase database. The file <i>dbSubIn_phase.inp</i> must contain the applicable LIGAND_NAME records. Not valid unless screening a database.
-JOB <i>jobname</i>	Job name, used by Job Control. Aligned structures are written to the file <i>jobname_shape.maegz</i> . If the job finishes successfully, the file <i>jobname_shape.okay</i> is created.
-keep <i>maxKeep</i>	Maximum number of structures to keep when sorting. This option has the effect of limiting the memory required. The top <i>maxKeep</i> structures are written to <i>jobName_shape.maegz</i> ; the rest are discarded. Valid only with -sort. Default: keep all structures.
-limit <i>numConfs</i>	Screen no more than the first <i>numConfs</i> conformers provided or generated for each molecule in <i>screenSource</i> . This option may be specified in any scenario, but it is most useful for benchmarking tests on a Phase database of pre-computed conformers. By choosing different values for <i>numConfs</i> , you can search a single database while simulating the effect of searching databases with different numbers of pre-computed conformers.
-NO_CHECKPOINT	Do not create checkpoint files. Valid only on the initial job submission. If this option is used, it will <i>not</i> be possible to restart the job.
-NOJOBID	Do not run under Job Control. This option is intended only for very short runs on a single processor, or when you are using a CPU that is not shared with others. All other job control options are ignored.
-norm {1 2 3 4}	<p>Similarity normalization scheme. The similarity between the shape query A and a screening molecule B is a function of the overlap $O(A,B)$ between the two, and the self-overlaps $O(A,A)$ and $O(B,B)$:</p> $\text{Sim}(A,B) = O(A,B)/f(O(A,A), O(B,B))$ <p>The normalization scheme determines the form of the function f:</p> <ol style="list-style-type: none"> 1 $f = \max\{O(A,A), O(B,B)\}$ (default) 2 $f = \min\{O(A,A), O(B,B)\}$ 3 $f = O(A,A)$ 4 $f = O(B,B)$ <p>Because <i>phase_shape</i> computes molecular overlaps as a sum of pairwise atomic overlaps, using any scheme other than the default can yield similarities larger than 1 in certain cases, such as when B is a substructure of A.</p>
-osub <i>dbSubOut</i>	Create subset file <i>dbSubOut_phase.inp</i> with LIGAND_NAME records for the structures written to <i>jobname_shape.maegz</i> . Not valid unless screening a database.

Table 15.1. Options and arguments for *phase_shape*. (Continued)

Option	Description
-redun <i>tol</i>	Tolerance for redundant alignments. An alignment for a given conformer is considered to be redundant if all of its atoms are within a distance <i>tol</i> of the corresponding atoms in another alignment of the same conformer. When two or more alignments are redundant, only the one with the highest similarity is retained. Valid only with <code>-report</code> . Default: 0.5 Å.
-report <i>n</i>	Report up to <i>n</i> aligned structures for each molecule that is screened, sorted by decreasing similarity. Alignments for a given molecule are always grouped, with the order of different groups determined by the highest similarity within each group. If a limit on the number of structures to keep is given, a group at the end of the sorted list will be retained or eliminated in its entirety. If multiple templates are used, up to <i>n</i> aligned structures are reported for each template. Default: report only the most similar alignment.
-RESTART <i>path</i>	Restart a job using the checkpoint files in the directory <i>path</i> . The file <i>path</i> /README contains instructions about restarting the job. When a job is restarted, the checkpoint files continue to be updated, so that multiple restarts are possible.
-scoreInPlace	Compute similarities without aligning.
-screen <i>screenSource</i>	Required. Structures to screen. <i>screenSource</i> may be any one of the following: <ol style="list-style-type: none"> 1. Maestro file. Recognized extensions are <code>.mae</code>, <code>.mae.gz</code>, <code>.maegz</code>. 2. SD file. Recognized extensions are <code>.sdf</code>, <code>.sd</code>, <code>.sdf.gz</code>, <code>.sd.gz</code>. 3. List file. This is a text file that contains the names of one or more Maestro or SD files, with one name per line. The list file extension must be <code>.list</code>. 4. Phase database. <i>screenSource</i> must be of the form <i>dbPath/dbName</i>. If existing conformers are used, consecutive structures with identical titles and connectivities are treated as conformers of a single molecule.
-shape <i>shapeFile</i>	Required. File that defines the shape query or queries. This file may be a Maestro or SD file, or it may be an included volumes file (<code>.ivol</code>) that contains <i>x</i> , <i>y</i> , and <i>z</i> coordinates and radii for three or more spheres that define the desired shape. The format of an included volumes file is identical to that of an excluded volumes file (<code>.xvol</code>). If the Maestro or SD file contains multiple structures, each structure is used as a query. The titles of the structures must be unique, because they are used for identifying the queries.

Table 15.1. Options and arguments for *phase_shape*. (Continued)

Option	Description
-sort	Sort output structures by decreasing similarity. This requires a sorted list of structures to be held in memory, which consumes about 2 kB memory per structure. Default: write out structures in the order in which they were read.
-table	Create the file <i>jobname_sim.csv</i> with a table of comma-separated similarities. These correspond to the values of the Maestro property Shape Sim (<i>r_phase_shape_sim</i>) reported in <i>jobname_align.maegz</i> . For each molecule, the rows are ordered by similarity if multiple alignments are requested. There is one column for each query.
-title <i>propName</i>	Use property <i>propName</i> as the source of the titles of the structures. Valid only when screening existing conformers in one or more files of the same type (i.e., all Maestro or all SD). If screening Maestro files, <i>propName</i> must begin with <i>s_</i> or <i>i_</i> (string or integer).
-v	Verbose log file output. This is the default. Can be overridden by setting the environment variable <code>SCHRODINGER_PHASE_VERBOSITY</code> to 0.
-nv	Non-verbose log file output. Can be overridden by setting the environment variable <code>SCHRODINGER_PHASE_VERBOSITY</code> to 1.

The aligned structures are written to the file *jobname_shape.maegz*. For each molecule, the alignments for a given query are stored consecutively, ordered by similarity. If sorting is requested, the blocks of alignments for each query are ordered by similarity, and the molecules are ordered by the maximum similarity for any query or alignment for that molecule. If sorting is not requested, the blocks of alignments for each query are in the order that the queries appear in the query input, and the molecules are in the same order as the molecules in the input.

15.3 Creating Included Volumes for Shape Queries

The simplest way to create an included volume file that can be used as a shape query is to rename an `.xvol` excluded volume file to have the extension `.ivol`. To create an excluded volume, use the Hypothesis Table panel in Maestro to create a hypothesis, add the desired included volumes as excluded volumes to this hypothesis, then export the hypothesis. You can then take the exported `.xvol` file and rename it for use with *phase_shape*.

You can also use following simplified format (which is the same format as the `.ev` excluded volume file):

```
NumSpheres
x1 y1 z1 r1
```

```
x2 y2 z2 r2
...
```

where *NumSpheres* is the number of spheres, and the *x*, *y*, *z*, and *r* values on any line are the coordinates of a sphere center and its radius.

A utility is available for creating included volumes, `create_ivolShape`, which is described in the next subsection. For converting included volumes files to Maestro files, you can use the utility `convert_ivolToMae`, which is described in the following subsection.

15.3.1 create_ivolShape

This utility creates an included volumes file to represent the “positive image” of a ligand or the “negative image” of a receptor, in terms of a set of spheres. For a receptor, you must define the binding pocket using a structure file that contains one or more ligands, or a box file that contains the limits of the box.

The syntax is as follows:

```
$SCHRODINGER/utilities/create_ivolShape -in maeFile
    {-pos positive-options | -neg negative-options}
    -out ivolFile [-append [-avoid dmin]]
```

The options are described in Table 15.2. You can only create a positive image or a negative image in a single run, but you can combine a positive image and a negative image with the `-append` option.

Table 15.2. Options for the `create_ivolShape` command.

Option	Description
<code>-in <i>maeFile</i></code>	Maestro file containing a single ligand structure or a receptor structure.
<code>-out <i>ivolFile</i></code>	Included volumes file to be created. The extension must be <code>.ivol</code> .
<code>-append</code>	Append to an existing included volumes file.
<code>-avoid <i>dmin</i></code>	When appending, do not create a sphere if its center will lie within a distance <code><dmin></code> of the center of any sphere already in <code><ivolFile></code> . This allows one to create an exact positive image of a single ligand, then surround it with spheres from a negative image.
<i>Positive image options</i>	
<code>-pos</code>	Create a positive image from a ligand. By default, an included volume sphere is created for each heavy atom, with the associated van der Waals radius. Use <code>-hydrogens</code> to include spheres for hydrogens attached to non-carbon atoms.
<code>-radius <i>r</i></code>	Use a radius of <i>r</i> for all included volume spheres.

Table 15.2. Options for the `create_ivolShape` command.

Option	Description
<code>-scale <s></code>	Multiply each radius by <code><s></code> .
<code>-sprop propName</code>	Multiply each radius by an atom-level property in <code>macFile</code> . <code><propName></code> must begin with "r_".
<code>-hydrogens</code>	Create spheres for hydrogens attached to non-carbon atoms. Hydrogens attached to carbons are never included in a positive image.
<i>Negative image options</i>	
<code>-neg</code>	Create a negative image from a receptor, i.e., an image of the binding pocket. By default, spheres of radius 1.5 Å. are placed at positions on a rectangular grid of spacing of 1.5 Å. Use <code>-grid</code> to change the grid and sphere sizes. The extent of the binding pocket is controlled by a set of reference points, which are presumed to be located inside the binding pocket (see below).
<code>-pocket {structFile boxFile}</code>	<p>If a structure file is given, derive the binding pocket from the structures in the specified Maestro or SD file. By default, each heavy atom in <code>structFile</code> is a binding pocket reference point. Use <code>-hydrogens</code> to include reference points from hydrogens attached to non-carbon atoms.</p> <p>If a box file is given, derive the binding pocket from the rectangular region given in the specified file. <code>boxFile</code> must have the extension <code>.box</code>, and must contain two lines that define the limits of the rectangle:</p> <pre>xmin ymin zmin xmax ymax zmax</pre> <p>Every position in this region is a reference point in the binding pocket.</p>
<code>-grid dgrid</code>	Spacing between grid points, and the radius of each sphere created. The default is 1.5 Å.
<code>-ext dext</code>	Only create spheres whose centers are within a distance <code>dext</code> of any reference point. The default is 1.5 Å.
<code>-probe dprobe</code>	Reject a sphere if its center is within a distance <code>dprobe</code> of the van der Waals surface of any receptor atom. If a ligand is used to define the binding pocket, this test is applied only when the sphere would lie more than a distance <code>dgrid</code> from the nearest reference point. The default probe distance is 1.5 Å.
<code>-limit dlimit</code>	Only create spheres whose centers are within a distance <code>dlimit</code> of the center of some receptor atom. This prevents spheres from protruding into the solvent when a large <code>dext</code> value is used. The default is 5 Å.
<code>-hydrogens</code>	When the binding pocket is derived from one or more structures, include reference points from hydrogens attached to non-carbons. By default, only heavy atoms are considered.

15.3.2 convert_ivolToMae

This utility converts an included volumes file to a structure in a Maestro file. This allows included volumes to be imported and visualized in Maestro without having to associate them with a Phase hypothesis. It also allows multiple sets of included volumes to be stored in a single Maestro file and supplied to `phase_shape` as multiple shape queries.

The utility works by creating a carbon atom for each included volume sphere. The positions of the spheres will be correct, but the radii will always be 1.7 angstroms (the van der Waals radius of carbon). If you use the Maestro file as a shape query, you should run `phase_shape` with the option `-atomWeights r_m_shape_weight` to scale the carbon van der Waals radii to the values that were present in the original included volumes file.

The command syntax is as follows:

```
$SCHRODINGER/utilities/convert_ivolToMae -in ivolFile -out maeFile
    [-append] [-title title]
```

The options are described in [Table 15.3](#).

Table 15.3. Options for the `convert_ivolToMae` command.

Option	Description
<code>-in <i>ivolFile</i></code>	Input included volumes file. The extension must be <code>.ivol</code> .
<code>-out <i>maeFile</i></code>	Output Maestro file. A carbon atom is created for each sphere in <i>ivolFile</i> , with two atom-level properties that relate the carbon radius to the radii of the original included volumes spheres: <code>r_m_ivol_scale</code> —The factor by which the carbon atom radius should be multiplied to obtain the radius of the original included volume sphere. <code>r_m_shape_weight</code> —The atomic weight property that should be supplied to <code>phase_shape</code> when using <i>maeFile</i> as a shape query, i.e., <code>-atomWeights r_m_shape_weight</code> .
<code>-append</code>	Append the CT block to an existing Maestro file.
<code>-title <i>title</i></code>	Use the supplied title when creating the structure. The default is to use the base name of <i>ivolFile</i> . If you are using this utility to create multiple shape queries for <code>phase_shape</code> , each title in <i>maeFile</i> must be unique.

Detecting Multiple Binding Modes

Some receptor sites permit ligand binding in distinct binding modes, rather than a single binding mode. Phase includes an automated, scripted procedure to help you identify subsets of ligands that bind in distinct modes. This is done by applying a clustering algorithm to a large number and variety of common pharmacophore hypotheses derived from the ligands of interest. Clustering is based on a bit string similarity metric, in which each ligand is assigned a bit, and the bits set for a given hypothesis correspond to the ligands that contain (i.e. match) that hypothesis. Thus the similarity between hypotheses is measured to be high if they have many bits (i.e., many ligands) in common.

As an example, suppose you have 30 ligands and this procedure has been applied to identify two highly distinct clusters. Suppose further that hypotheses A and B are representatives from each cluster, and that 13 of the ligands match hypothesis A, while the remaining 17 match hypothesis B. In this case, perfect separation is achieved, and hypotheses A and B provide a model to explain the multiple binding modes.

In practice, the separation may not be perfect, and a given cluster may contain very different families of pharmacophores, so selecting a representative may not always be straightforward. Because of this, the critical goal is identifying the most probable subsets of ligands, rather than specific pharmacophore models. Once the ligand subsets are known, each can be pursued independently to fine-tune the associated pharmacophore models.

To facilitate identification of subsets, visualization tools are employed. A 2-dimensional “heat map” is provided to illustrate the association of hypotheses, ligands and clusters. If large, distinct clusters are apparent, then evidence exists for multiple binding modes and the appropriate ligand subsets can be pursued.

When generating common pharmacophore models, there are a few parameters that must be assigned using a reasonable amount of discretion. For example, minimum and maximum limits must be set on the number of sites in the pharmacophores to be generated. The software supports a range of 3 to 7 sites, but in most cases it is sufficient to consider only 4- and 5-point pharmacophores.

The minimum number of ligands that must match each pharmacophore must also be chosen, and hence the minimum number of ligands that is needed to establish a binding mode. In theory, requiring a minimum of 2 ligands per binding mode would provide exhaustive identification of potential binding modes, but only at tremendous computational expense. It is strongly recommended that this value be no smaller than it needs to be. For example, if you have 30

ligands, and you suspect that they bind in 2 equally probable modes, then it is reasonable to require that at least 10 ligands match each pharmacophore.

Once common pharmacophores are generated, the usual scoring procedure is performed to rank hypotheses and eliminate those that provide less satisfactory alignments. If you wish to consider pharmacophore models with different numbers of sites (e.g., 4-point and 5-point pharmacophores), then separate Phase runs should be created within the same project, branched at the common pharmacophore step. Note that the exact same ligands should be used in all runs, and the same minimum number of matching ligands should be required.

After generating scoring results for one or more runs, you can run the clustering calculation and visualize the clusters in the Phase Cluster Visualization panel, which you can open from the Phase submenu of the Applications menu. This panel allows you to select one or more runs, the number of clusters, and clustering options.

The number of clusters should be equal to the number of binding modes that are believed to exist. This does not affect how the clustering is performed, it merely changes the location of the vertical yellow lines that are drawn on the heat map, which suggest the most probable splitting points between ligand clusters.

To perform a clustering calculation:

1. Select the runs in the Project Data section.
2. Select options from the option menus in the Clustering Options section.

These option menus are described below.

3. Click Calculate.

Option menus

Number of Clusters option menu

Select the number that corresponds to the number of likely binding modes.

Metric option menu

Select the method used to compute distances between bit strings representations.

- Euclidean—Sum of the squares of the differences in the bit string values. This is just the number of positions at which two bit strings differ.
- Tanimoto—Tanimoto metric, $1 - N(\text{common})/N(\text{total})$, where $N(\text{common})$ is the number of 'on' bits shared by the two strings (bit intersection), and $N(\text{total})$ is the total number of 'on' bits in either string (bit union).

- Cosine—Cosine of the angle between the two bit string vectors.

Linkage option menu

Select the method used to compute distances between clusters.

- Single—The distance between clusters is the smallest distance between any pair of objects (one object from each cluster). This option produces diffuse, elongated clusters.
- Average—The distance between clusters is the average distance between all pairs of objects in the two clusters.
- Complete—The distance between clusters is the largest distance between any pair of objects (one object from each cluster). This option produces compact, spherical clusters.

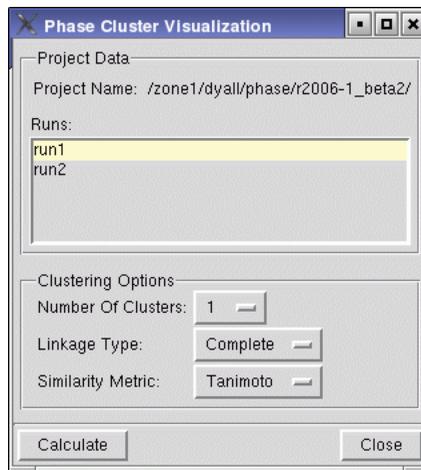


Figure 16.1. Cluster Visualization Panel

Receptor-Based Hypotheses

Although Phase does not have a direct facility for creating hypotheses from receptors, you can make use of receptor information when creating hypotheses. If you have a ligand that is docked to a receptor, you can create a hypothesis manually from the ligand sites, with the receptor displayed in the Workspace. The procedure for manually creating a hypothesis is described in [Chapter 8](#).

When you create the hypothesis, you might want to display hydrogen bonds between the receptor and the ligand in the Workspace, to help you choose the important sites. You might also want to run a Glide XP docking job, and use the Glide XP Visualizer to display representations of the various contributions to the XP GlideScore in the Workspace, to help visualize important hydrophobic or pi-pi stacking interactions. Note that the hypothesis must be created using the ligand only, and that the ligand and receptor must be in separate entries.

Once you have created the hypothesis, you can use the receptor to create excluded volumes. You can do this from the command line, as described in [Section 12.3.1 on page 115](#), or you can use the Receptor-Based Excluded Volumes panel to set up and run the job, as described here. The spheres are added to each selected receptor atom.

To open the Receptor-Based Excluded Volumes panel, choose Receptor-Based Excluded Volumes from the Phase submenu of the Applications menu.

The steps in setting up the job to create an excluded volume for a receptor or part of a receptor are as follows:

1. Choose a hypothesis, by entering the file name of the hypothesis in the text box, including the path, or clicking **Browse** and navigating to the hypothesis file (.tab).

The hypothesis reference ligand must be stored in the corresponding .mae file and be properly aligned to the receptor.

2. Select atoms to define the part of the receptor that you want to create excluded volumes for, using one or both of the following methods:
 - a. Select **Using panel(ASL)**, then click **Select** to open the Atom Selection dialog box and select the atoms. The ASL expression is displayed in the text box.
 - b. Select **From workspace**, and select the atoms in the Workspace using the Workspace selection tool, then click **Finish** when you have finished selecting atoms. The atoms are listed in the text box by atom number.

3. Set the radii of the excluded volume spheres.

The options in the Radii sizes and Radii scaling factors sections allow you to set the sphere size at the level of individual atoms, if desired. There are three options for setting sphere size:

- Van der Waals radii of receptor atoms—Use the van der Waals radii of the receptor atoms for the radii of the spheres.
- Fixed radius—Set the radii of the excluded volume spheres to the value supplied in the text box, in angstroms.
- Atom-level Maestro property—Set the radii of the spheres to the value of the atom-level Maestro property chosen from the option menu. Atoms with a zero or unspecified value of this property are skipped.

In addition, you can specify a scaling factor, which is applied to the spheres. This is useful if, for example, you want to scale the van der Waals radii. There are two options for scaling:

- Fixed scaling factor—Set the scaling factor for the excluded volume spheres to the value supplied in the text box, in angstroms. The default is 1.0.
- Atom-level Maestro property—Set the scaling factor for the spheres to the value of the atom-level Maestro property chosen from the option menu. Atoms with a zero or unspecified value of this property do not have their excluded volume spheres scaled.

4. Filter out any unwanted excluded volume spheres.

Spheres that are too close to the reference ligand should not be created, and spheres that are too far away from the reference ligand will not have any influence and will only slow the filtering of matches. The two options for filtering the spheres are:

- Ignore receptor atoms whose surfaces are within $N \text{ \AA}$ of ligand surface—Excluded volume spheres are not created for receptor atoms whose surfaces are within the specified distance of the reference ligand van der Waals surface.
- Limit excluded volume shell thickness to $N \text{ \AA}$ —Spheres located more than the specified distance from the reference ligand are not included.

5. Choose whether to append to or overwrite existing excluded volumes.

6. Click Create Excluded Volumes.

The job that runs `create_xvolReceptor` to create the excluded volumes is started.

If you make a mistake and want to start over, click **Reset to Defaults** to restore the default values in the panel.

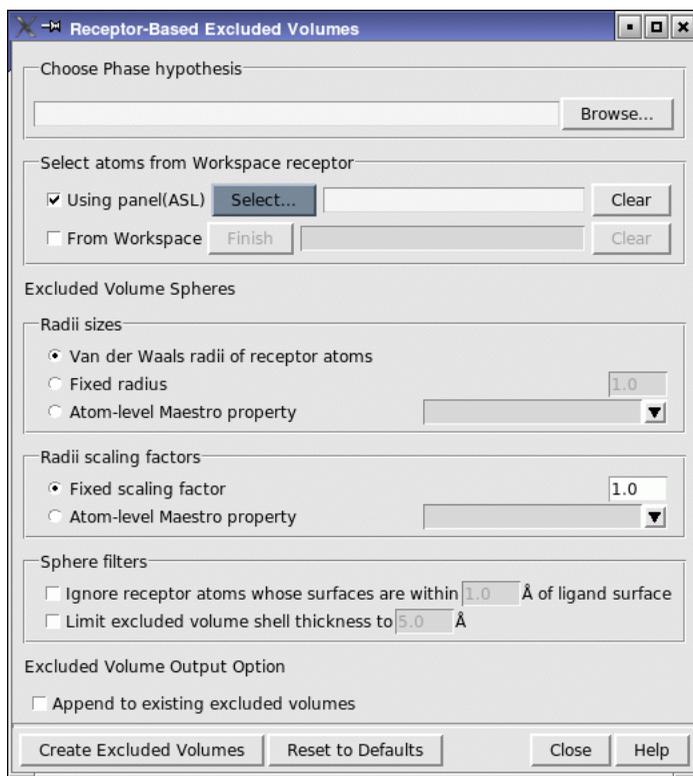


Figure 17.1. The Receptor-Based Excluded Volumes panel

Phase QSAR Models

A.1 The Phase QSAR Methods

Phase QSAR models are developed from a series of molecules, of varying activity, that have all been aligned to a common pharmacophore hypothesis that is associated with a single reference ligand. QSAR models may be atom-based or pharmacophore-based: the former takes all atoms into account; the latter uses the pharmacophore sites that can be matched to the hypothesis.

A rectangular grid is defined to encompass the space occupied by a training set of aligned molecules. This grid divides space into uniformly-sized cubes, typically one angstrom on each side. In atom-based models, the grid is populated by van der Waals spheres, with radii that depend on the atom type. In pharmacophore-based models, the grid is populated by the pharmacophore sites that match the hypothesis, with each site represented by a sphere with a user-definable radius.

A given atom or pharmacophore site will occupy the space of one or more cubes in the grid. Occupation of a cube is deemed to occur if the center of that cube falls within the radius of the atom or site. A given cube may be occupied by more than one atom or site, and that occupation may come from the same molecule or from different molecules.

Each occupied cube gives rise to one or more volume bits. A volume bit is allocated for each different class of atom or site that occupies a cube.

In pharmacophore-based models, sites are assigned to classes that are determined by the feature definitions used to create the hypothesis (e.g., A, D, H, N, P, R). In atom-based models, there are 6 distinct atom classes that have some correspondence or similarity with pharmacophore feature types, but atom classes are assigned using fixed internal rules, not the hypothesis feature definitions:

- D – Hydrogen bond donor (hydrogens bonded to N, O, P, S)
- H – Hydrophobic/non-polar (C, H–C, Cl, Br, F, I)
- N – Negative ionic (formal negative charge)
- P – Positive ionic (formal positive charge)
- W – Electron-withdrawing (N, O)
- X – Miscellaneous (all other types of atoms)

So, if a particular cube is occupied by a “D” from molecule 1, an “H” from molecule 5, and a “P” from molecule 9, that cube would be allocated three volume bits. If a cube is never occu-

plied by any molecule in the training set, no volume bits would be allocated. Hence, each volume bit must be set by at least one molecule in the training set.

The pool of volume bits provides a means of characterizing the molecules. In atom-based models, the pattern of volume bits that are set by a molecule encodes the size, shape, and chemical characteristics of that molecule. In pharmacophore-based models, the pattern of set bits determines which subset of critical pharmacophore features that molecule contains, and the positions of those features in relation to other molecules.

If a binary scheme (0/1) is used to denote which bits are set by each molecule, a table of bit values may be assembled:

<i>molecule</i> ₁	0	1	1	0	0	1	1	0	0	1	0	1	0	.	.	.
<i>molecule</i> ₂	0	1	1	0	0	0	1	0	0	1	0	1	0	.	.	.
<i>molecule</i> ₃	0	0	0	1	0	1	0	0	1	0	1	0	1	.	.	.
	.															
	.															
	.															

For atom-based models, there are usually several hundred or more volume bits for each series of aligned molecules. For pharmacophore-based models, that number is much smaller, usually only a few dozen. The number of bits increases as the grid spacing becomes finer, and, in the case of atom-based models, as the molecules become larger.

To generate a QSAR model, the 0/1 bit values are treated as independent variables in partial least-squares (PLS) regression analysis. This involves finding a linear least-squares relationship between the activity data and a special set of orthogonal factors that are linear combinations of the bit value variables.

More precisely, if there are *n* molecules in the training set and *v* volume bits, let the *n*×*v* matrix **X** represent the table of volume bits, and let the *n*×1 vector **y** represent the activity values for the training set molecules. Creation of the PLS regression model proceeds as follows:

Center each column of **X**:

for *i* = 1, ..., *v*

$$\mu_i^x = \frac{1}{n} \sum_{k=1}^n X_{k,i}$$

for *k* = 1, ..., *n*

$$X_{k,i} \rightarrow X_{k,i} - \mu_i^x$$

next k

next i

Center \mathbf{y} :

$$\mu^y = \frac{1}{n} \sum_{k=1}^n y_k$$

for $k=1, \dots, n$

$$y_k \rightarrow y_k - \mu^y$$

next k

Determine PLS factors and regression coefficients for up to m PLS factors ($m \leq v$):

$\mathbf{X}_1 = \mathbf{X}$

for $i = 1, \dots, m$

Compute the vector of weights that define PLS factor i :

$$\mathbf{w}_i = \mathbf{X}_i^T \mathbf{y} / |\mathbf{X}_i^T \mathbf{y}|, \quad \mathbf{w}_i \in \mathbf{R}^{v \times 1}$$

Project the rows of \mathbf{X}_i onto factor i :

$$\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i, \quad \mathbf{t}_i \in \mathbf{R}^{n \times 1}$$

Project \mathbf{t}_i onto each column of \mathbf{X}_i :

$$\mathbf{p}_i = \mathbf{X}_i^T \mathbf{t}_i / |\mathbf{t}_i^T \mathbf{t}_i|, \quad \mathbf{p}_i \in \mathbf{R}^{v \times 1}$$

Compute the i th PLS regression coefficient by projecting \mathbf{t}_i onto \mathbf{y} :

$$\mathbf{b}_i = \mathbf{t}_i^T \mathbf{y} / |\mathbf{t}_i^T \mathbf{t}_i|, \quad \mathbf{b}_i \in \mathbf{R}^{m \times 1}$$

Orthogonalize \mathbf{X}_i with respect to PLS factor i :

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$$

next i

For a regression with m PLS factors, the fitted activities are then given by:

$$\hat{\mathbf{y}} = \mu^y + \sum_{i=1}^m b_i \mathbf{t}_i$$

To apply the m -factor PLS model to a new set of n_T ligands with bit value matrix $\tilde{\mathbf{X}}$, the regression coefficients \mathbf{b} must first be translated back to the space of the original \mathbf{X} variables:

Define

$$\mathbf{W} \equiv [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m], \quad \mathbf{W} \in \mathbf{R}^{v \times m}$$

$$\mathbf{P} \equiv [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m], \quad \mathbf{P} \in \mathbf{R}^{v \times m}$$

Then

$$\mathbf{b}^x \equiv \mathbf{W}(\mathbf{P}^T \mathbf{W})^{-1} \mathbf{b} \quad \mathbf{b}^x \in \mathbf{R}^{v \times 1}$$

The coefficients \mathbf{b}^x may then be used to predict activities for the new ligands:

$$\hat{y}_k = \mu^y + \sum_{i=1}^m (\tilde{X}_{k,i} - \mu_i^x) b_i^x \quad k = 1, \dots, n_T$$

A.2 Phase Model Validation

Phase QSAR models do not use internal cross-validation techniques, but rather use distinct training and test sets. The use of a true test set is far superior to internal cross-validation techniques such as leave- n -out, where small subsets of the training set are temporarily held out and predicted using models built from the remainder of the training set. Leave- n -out, as it is usually applied, is not an unbiased validation technique because the activity data being predicted typically have some role in selecting or constructing the variables used in the series of models being built. This is especially true of partial least-squares (PLS) regression, the multivariate method that is used to develop Phase QSAR models. Statistics from PLS leave- n -out predictions will almost always be overly optimistic, because the latent variables included in each model are constructed from the full training set, so they correlate with all activities, even those of the molecules being predicted. Further, leave- n -out predictions are frequently used to arrive at an *optimal* number of PLS factors, but in fact, internal cross-validated statistics cannot provide a meaningful measure of how the model will actually perform when applied to new

molecules. As a result, the optimal number of PLS factors arrived at using this technique may very well correspond to a situation wherein the activity data have been seriously over-fit. For these reasons, Phase supports only the use of true, external test sets.

However, the use of leave-*n*-out techniques are useful for assessing the stability of the model to changes in the training set. In Phase QSAR models, leave-*n*-out models are built, and the R^2 value is computed between the leave-*n*-out predictions and the predictions from the model built on the full training set. This value is reported as the stability value, and has a maximum value of 1. If the stability value is high, the model built from the full training set is fairly insensitive to changes in that training set, i.e., the predicted values don't change much. Models with high stability are preferred because they are not overly dependent on the idiosyncracies of any particular training set. Stability should not be used to decide on an optimum number of PLS factors, but it can be helpful in choosing between models from different hypotheses whose other statistics are essentially the same.

A.3 Phase QSAR Statistics

This section defines the various statistical measures that are used in Phase QSAR models.

A.3.1 Training Set and Model

Statistical quantities describing the training set and the QSAR model are defined below.

m	number of PLS factors in the model
n	number of molecules in the training set
$df_1 = m + 1$	degrees of freedom in model
$df_2 = n - m - 2$	degrees of freedom in data
y_i	observed activity for training set molecule i
\hat{y}_i	predicted activity for training set molecule i
$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$	mean observed activity
$\sigma_y^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$	variance in observed activities

$sse = \sum_{i=1}^n (\hat{y}_i - y_i)^2$	sum of squared errors
$\sigma_{\text{err}}^2 = \frac{sse}{n}$	variance in errors
$ssy = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$	variance in model
$SD = \sqrt{sse/df_2}$	standard deviation of regression
$R^2 = 1 - \frac{\sigma_{\text{err}}^2}{\sigma_y^2}$	R-squared; coefficient of determination
$F = \frac{ssy/df_1}{sse/df_2}$	F statistic; overall significance of model
$P = B(df_1, df_2, \frac{df_2}{df_2 + Fdf_1})$	statistical significance; probability that correlation could occur by chance. The beta function $B(a, b, x)$ is defined by
	$B(a, b, x) = \int_0^x t^{a-1} (1-t)^{b-1} dt$

Note that R^2 can never be negative, because the regression coefficients are optimized to minimize sse . The worst-case scenario is when the independent variables have absolutely no statistical relationship with activity. Under those circumstances, the regression coefficients will all be zero, and the model will contain only an intercept parameter, the value of which will be \bar{y} . Thus every predicted activity will be \bar{y} , and σ_{err}^2 will be equal to σ_y^2 , yielding $R^2 = 0$.

A.3.2 Test Set Predictions

Statistical quantities describing the test set predictions are described below.

T	The test set of molecules
n_T	number of molecules in T
y_j	observed activity for molecule $j \in T$
\hat{y}_j	predicted activity for molecule $j \in T$
$RMSE = \sqrt{\frac{1}{n_T} \sum_{j \in T} (\hat{y}_j - y_j)^2}$	root-mean-squared error

$Q^2 = R^2(T)$ Q-squared

$$r = \frac{\sum_{j \in T} (y_j - \bar{y}_T)(y_j - \hat{y}_T)}{\sqrt{\sum_{j \in T} (y_j - \bar{y}_T)^2 (y_j - \hat{y}_T)^2}}$$

Pearson r value, Pearson correlation coefficient

The formulas for R^2 and Q^2 are equivalent, with the only difference being that Q^2 is computed using the observed and predicted activities for the test set. However, Q^2 can take on negative values. This happens whenever the variance in the test set errors is larger than the variance in the observed test set activities. Often, the test set does not have as large a range of activity values as the training set (so the variance in y is smaller), and the errors for the test set tend to be larger than those for the training set (so the variance in the errors is larger). It is therefore not uncommon to see negative Q^2 values from time to time.

Because all values are shifted by the sample means, the Pearson correlation coefficient is insensitive to systematic errors in the predictions, whereas Q^2 is not. So if the rank order of the activity predictions is basically correct, but there's a significant constant shift in the values compared to the observed activities, the Pearson correlation coefficient may still be quite high, even if Q^2 is small or negative.

Phase Input Files

In addition to structure files, Phase uses a variety of data input files. These files are described in the following sections. Normally you would not need to edit most of these files, as they are set up using the command-line utilities or from Maestro. However, some of the files used when searching for matches must be created by hand.

In this appendix, references to *utilities* are to programs or scripts in the `$SCHRODINGER/utilities` directory.

B.1 Master Data File

This file stores various pieces of ligand data that are used throughout the pharmacophore model development workflow. It is named `MasterData.tab`, and can be updated by the utility `pharm_data` or by hand. At the top of the file is a description of the data it contains, and a number of rules regarding how the data may be modified. The body of the file contains first a ligand property name block, followed by a set of ligand blocks, one per ligand in the project.

If you make any changes to `MasterData.tab`, whether by hand or through operations supported by the `pharm_data` utility, you must run `pharm_data` with the `-commit` option to update the Maestro files in the `ligands` subdirectory. If you do not update the Maestro files, various Phase modules will not be using the modified values because they read the property data directly from the Maestro files.

If you have completed any forward steps in the project workflow, the results generated in those steps may be invalidated by changes you make to `MasterData.tab`. When you attempt to commit the changes, you will be supplied with a list of forward files that will be invalidated, and you will be given a chance to abort the commit operation. If you choose to abort, you can rerun `pharm_data` with the `-restore` flag to revert to the previous version of the file, which is stored in `MasterData.backup`.

The ligand property name block keywords are described in [Table B.1](#). These properties are set by `pharm_project` and must not be altered by hand. This block defines the names of certain properties that are relevant to the pharmacophore model development. The Maestro files in the `ligands` subdirectory contain these named properties.

Table B.1. Ligand property block keywords in the Master Data file.

Keyword	Description
LIGAND_NAME_PROPERTY	Name of Maestro property that defines the ligand name. Set automatically by <code>pharm_project</code> to <code>s_phase_Ligand_Name</code> .
PHARM_SET_PROPERTY	Name of Maestro property that defines the pharm set membership. Set automatically by <code>pharm_project</code> to <code>s_phase_Pharm_Set</code> .
QSAR_SET_PROPERTY	Name of Maestro property that defines the QSAR set membership. Set automatically by <code>pharm_project</code> to <code>s_phase_QSAR_Set</code> .
ACT_PROPERTY	Name of the property that stores the ligand activity values. Set by the <code>-act</code> option of <code>pharm_project</code> .
1D_PROPERTY	Name of Maestro property that defines the <code>1D_VALUE</code> used to control which ligands can give rise to hypotheses. Set automatically by <code>pharm_project</code> to <code>r_phase_Ligand_1D_Property</code> .
CONF_PROPERTY	Name of Maestro property that defines the conformation-dependent quantity used in scoring. This is normally the relative conformational energy. Set by the <code>-conf</code> option of <code>pharm_project</code> .

The keywords that define the ligand data values in the ligand blocks are described in [Table B.2](#). These data values are stored in the ligand Maestro files in addition to the master data file. Only the conformation-independent properties are kept in these blocks: the conformation-dependent property defined by `CONF_PROPERTY` is not stored here, but only in the ligand Maestro file.

Table B.2. Ligand block keywords in the Master Data file.

Keyword	Description
LIGAND_NAME	Ligand name, in the project. This is not the same as the title or the Maestro entry name, but is a unique identifier used in the pharmacophore model project. Do not modify.
TITLE	Title of the ligand. Taken from the Title property in the Maestro file. Do not modify.
PHARM_SET	Pharm set membership of the ligand. Can be modified by hand or using the <code>-active</code> and <code>-inactive</code> options of the <code>pharm_data</code> utility. Allowed values are <code>active</code> , <code>inactive</code> , and <code>none</code> . <ul style="list-style-type: none"> <code>active</code> Ligand is used to identify common pharmacophores and to score hypotheses. There must be at least two ligands with <code>PHARM_SET = active</code> <code>inactive</code> Ligand is used to measure the degree to which hypotheses discriminate actives from inactives by inactive scoring. <code>none</code> Ligand is not used in pharmacophore model development, but may be used in QSAR model development.

Table B.2. Ligand block keywords in the Master Data file.

Keyword	Description
QSAR_SET	<p>QSAR set membership of the ligand. Affected by the <code>-train</code>, <code>-rand</code>, and <code>-pharm_set</code> options of the <code>pharm_data</code> utility. Can be modified by hand. Allowed values are <code>train</code>, <code>test</code>, and <code>none</code>.</p> <p><code>train</code> Ligand is used to develop QSAR models. You should use at least five training set ligands for each PLS factor.</p> <p><code>test</code> Ligand is used to test QSAR models.</p> <p><code>none</code> QSAR models are not applied to this ligand.</p>
ACTIVITY	Ligand activity. Affected by the <code>-log</code> , <code>-exp</code> , and <code>-multiply</code> options of the <code>pharm_data</code> utility. Can be modified by hand. Values should increase as potency increases, as for example in <code>-log K_i</code> or <code>-log IC₅₀</code> . If activity is unknown, the value should be set to missing.
1D_VALUE	A conformationally independent numerical property that may be used during hypothesis scoring to influence or control the selection of reference ligands. This property is added to the actives score when the <code>-prop</code> option is used with <code>pharm_score_actives</code> . If you assign a non-zero <code>1D_VALUE</code> for certain actives, and a zero value for the remaining actives, you can force hypotheses to come from only those actives with a non-zero <code>1D_VALUE</code> . Must be set by hand.
LIGAND_GROUP	Numerical identifier (group number) of ligand group. Ligands that have the same ligand group number belong to the same group. Ligands in the same group are treated as equivalent when finding common pharmacophores: to match the pharmacophore, only one member of the group has to match. Defining groups is useful for making tautomers, stereoisomers, and ions equivalent. By default, all ligands are in a separate group.
MUST_MATCH	Require this ligand to match when finding common pharmacophores. Allowed values are <code>true</code> and <code>false</code> . This property can be set to <code>true</code> only when <code>PHARM_SET</code> is active. All ligands in the same group must have the same value of this property. Default: <code>false</code> .

An example of the top of a Phase Master Data file is given below. This excerpt includes the header, the ligand property name block, and a few ligand blocks.

```
#####
#
# Phase Master Data File
#
# You may change PHARM_SET, QSAR_SET, ACTIVITY and 1D_VALUE. To propagate
# these changes to the project, use 'pharm_data -commit'. To revert to the
# most recently committed version of the file, use 'pharm_data -restore'.
#
# PHARM_SET: Allowed values are "active", "inactive" and "none".
# active - Used to identify common pharmacophores and to score
```

Appendix B: Phase Input Files

```
#             hypotheses.  There must be at least two ligands      #
#             with PHARM_SET = active.                              #
#             inactive - to measure the degree to which hypotheses  #
#             discriminate actives from inactives.                 #
#             none      - Not used in pharmacophore model development. #
#                                                                 #
# QSAR_SET:  Allowed values are "train", "test" and "none".       #
#             train - Used to develop QSAR models.  Recommend at least five #
#             training set ligands for each PLS factor.           #
#             test  - Used to test QSAR models.                   #
#             none  - QSAR models not applied to these ligands.   #
#                                                                 #
# ACTIVITY:  Ligand activity.  Values should increase as potency increases, #
#             for example, -logKi or -logIC50.  If activity is unknown, the #
#             value should be "missing".                            #
#                                                                 #
# 1D_VALUE:  A conformationally-independent numerical property that may be #
#             used during hypothesis scoring to influence or control the #
#             selection of reference ligands.                       #
#                                                                 #
#####
LIGAND_NAME_PROPERTY = s_phase_Ligand_Name
PHARM_SET_PROPERTY = s_phase_Pharm_Set
QSAR_SET_PROPERTY = s_phase_QSAR_Set
ACT_PROPERTY = r_phase_Ligand_Activity
1D_PROPERTY = r_phase_Ligand_1D_Property
CONF_PROPERTY = r_mmod_Relative_Potential_Energy-MMFF94s
#####
LIGAND_NAME = mol_1
TITLE = "endo-1"
PHARM_SET = active
QSAR_SET = train
ACTIVITY = 5.509
1D_VALUE = 0.0
LIGAND_GROUP = 1
MUST_MATCH = false
#####
LIGAND_NAME = mol_2
TITLE = "endo-2"
PHARM_SET = active
QSAR_SET = train
ACTIVITY = 5.456
1D_VALUE = 0.0
LIGAND_GROUP = 2
MUST_MATCH = false
```

B.2 Phase Main Input File

The Phase main input file contains information that is used across the entire Develop Pharmacophore Model workflow. It contains sections relevant to all programs in the workflow and sections that are used only by specific programs. This file is created by the model development utilities. If you edit this file, the order of ligands must remain unchanged for an entire run. When you run Phase from Maestro, Maestro creates and updates a main input file for each run.

Each line of the Phase main input file contains a keyword-value pair separated by an equals sign ("="), as follows:

keyword=value

Extra spaces are ignored, but blank lines are not permitted. Keywords can have string, integer, or real types. These types are enforced.

Keywords for the job name, file names, and directories are given in [Table B.3](#). In the tables, the same text is used for the value as for the keyword, but it is set in a different font: for example, *ligand-name* represents the value of the keyword LIGAND_NAME.

Optional keywords for the Find Common Pharmacophores step are described in [Table B.4](#). Optional keywords for the Score Hypotheses step are given in [Table B.5](#). Each Score Hypotheses keyword has a default value, so it is not necessary to include any of them in the input file.

Changes to this file should be made with the utility programs described in [Chapter 12](#).

Table B.3. Name and directory keywords.

Keyword	Description
BOXES_DIR	Name of directory where box files are stored. Box files are generated during the Find Common Pharmacophore step. These files contain data that is used as input for the scoring step. This keyword is used for both the Find Common Pharmacophore and Score Hypothesis steps. Optional keyword; the default is <code>boxes</code> .
JOB_NAME	Name given to each Phase job run with this input file. This name is used as a base for input and output files for the run. The name must match the name of the input file, <code>jobname_phase.inp</code> .
LIGAND_DIR	Directory where all ligand-related files are stored (the <i>ligands directory</i>). Should be a relative path. The ligand input files are multi-conformer Maestro files named <code>ligand-name.mae</code> . The output files include the conformer coordinate files, named <code>ligand-name_xyz.phc</code> , and the ligand sites files, named <code>ligand-name_sites.phs</code> . Optional keyword; the default is <code>ligands</code> .
LIGAND_NAME	Name of a ligand. This name is used to construct file names for ligand-related files. The ligand structure is contained in the file <code>ligand-name.mae</code> . For example, if the input file contains the line <code>LIGAND_NAME=aspirin</code> , there should be a Maestro file named <code>aspirin.mae</code> in the ligands directory. The input file should contain multiple lines of this kind, one for each ligand in the set. The order of these lines should <i>not</i> be changed during a run. For pharmacophore model development, the set should include only the active ligands on which the model is to be based. Do not change by hand.
RESULT_DIR	Name of the directory where results of the Score Hypothesis step are stored. Optional keyword; the default is <code>result</code> .

Table B.4. Optional Keywords for the Find Common Pharmacophores step.

Keyword	Description
NUM_SITES	Total number of sites in pharmacophore hypothesis (integer). Default is 5.
MIN_INTERSITE_DIST	Minimum distance between pharmacophore sites in angstroms (real). May be used to reject pharmacophores that contain, for example, an acceptor site and a donor site from the same oxygen. Default is 2.0 Å.
MAXIMUM_DEPTH	Number of times each side of the "box" is divided (integer). Default is 5.
INITIAL_BOX_SIZE	Initial box size in angstroms (real). This option should not appear in the Phase main input file. Set automatically using values for FINAL_BOX_SIZE and MAXIMUM_DEPTH as described below.
FINAL_BOX_SIZE	Final box size in angstroms (real). Default is 2.0 Å.
MIN_NUM_LIGANDS_PER_BOX	Minimum number of ligands or ligand groups that must be matched (integer).
MIN_MAX_SITES	Minimum and maximum number of sites for a feature type. Value is a string that contains 3 integers separated by commas with no spaces: <i>n1,n2,n3</i> . The first integer (<i>n1</i>) is the numerical code for the site type (see VARIANT_NAMES, below). The second integer (<i>n2</i>) is the minimum feature frequency and third integer (<i>n3</i>) is the maximum feature frequency. The maximum value of <i>n3</i> is 4. By default, the values of <i>n2</i> and <i>n3</i> are set to 0 and 4 for the standard features (A, D, H, N, P, and R), and to 0 and 0 for the custom features. If you change the defaults, the input file should contain multiple lines of this kind, one for each feature type.
VARIANT_NAMES	Comma-separated list of variants for which common pharmacophores are to be identified. These names are used to construct file names for variant-related files. Multiple lines of this kind can be used to specify the variants. By default, all variants are used. Each variant is a string of single-digit numbers in ascending order. The numbers encode the feature types, as follows: 0 Hydrogen-bond acceptor (A) 1 Hydrogen-bond donor (D) 2 Hydrophobic group (H) 3 Negatively-charged atom (N) 4 Positively-charged atom (P) 5 Projected point (Q)—not used 6 Aromatic ring (R) 7 Custom (X) 8 Custom (Y) 9 Custom (Z)

Table B.5. Optional keywords for the Score Hypotheses step.

Keyword	Description
ALIGN_CUTOFF	Maximum RMS deviation in angstroms of aligned site points from two ligands, in angstroms (real). Default is 1.2 Å.
ALIGN_WEIGHT	Weighting factor of the site alignment term in the survival score (real). See Section 6.2.2 on page 52 for definitions. Default is 1.0.
BOXES_TO_KEEP	Percentage of top-scoring boxes to be retained for volume scoring after the first pass (integer). Default is 10.
CONFORMATION_PROPERTY	Name and weight of a conformation-dependent property to use in property scoring. Value contains property name and weight, separated by a comma. Multiple conformation property entries can be specified in the input file. For example, to use MMFF relative conformation energies and weight -0.1 the value of CONFORMATION_PROPERTY is <code>r_mmod_Relative_Potential_Energy-MMFF94s,-0.1</code> .
FEATURE_ALIGN_CUTOFF_FILE	Name of the file that contains feature-matching tolerances. See Section B.9 on page 229 for the format of this file.
MAX_BOXES	Maximum number of boxes to be scored. Default is 50. Overrides percentage specified by BOXES_TO_KEEP.
MIN_BOXES	Minimum number of boxes to be scored, equivalent to the minimum number of returned hypotheses per variant. Overrides percentage specified by BOXES_TO_KEEP. Default is 10.
PENALTY_CONST	Used to penalize hypotheses that do not have matches for all ligands. Default is 1.1. A value of 1.0 means that no penalty is applied.
PROPERTY_NAME	Name of the property to use in property scoring, e.g. <code>r_m_phase_activity</code> for the activity value. Only one property can be specified, and the property must be conformation-independent.
PROPERTY_WEIGHT	Weighting factor for the property (activity) term in the survival score. See Section 6.2.2 on page 52 for definitions. Default is 0.0.
SELECTIVITY_WEIGHT	Weighting factor for the selectivity term in the survival score. See Section 6.2.2 on page 52 for definitions. Default is 0.0.
USE_PROPERTY	Calculate property scores. Value can be <code>true</code> or <code>false</code> . Default is <code>false</code> .
USE_SELECTIVITY	Calculate selectivity score. Value can be <code>true</code> or <code>false</code> . Default is <code>false</code> .
USE_VOLUME	Calculate volume scores. Value can be <code>true</code> or <code>false</code> . Default is <code>true</code> .

Table B.5. Optional keywords for the Score Hypotheses step. (Continued)

Keyword	Description
VECTOR_CUTOFF	Minimum vector score value needed to keep the hypothesis. Default is 0.5.
VECTOR_WEIGHT	Weighting factor of the vector term in the survival score (real). See Section 6.2.2 on page 52 for definitions. Default is 1.0.
VOLUME_WEIGHT	Weighting factor of the volume term in the survival score (real). See Section 6.2.2 on page 52 for definitions. Default is 1.0.

An example of a Phase main input file is shown below. This example includes options for which defaults exist.

```
JOB_NAME=index_26
MIN_INTERSITE_DIST=2
NUM_SITES=5
FINAL_BOX_SIZE=2
MAXIMUM_DEPTH=5
MIN_NUM_LIGANDS_PER_BOX=5
MIN_MAX_SITES=0,0,5
MIN_MAX_SITES=1,0,5
MIN_MAX_SITES=2,0,5
MIN_MAX_SITES=3,0,5
MIN_MAX_SITES=4,0,5
MIN_MAX_SITES=6,0,5
ALIGN_CUTOFF=1.2
ALIGN_WEIGHT=1
VECTOR_CUTOFF=0.5
VECTOR_WEIGHT=1
VOLUME_WEIGHT=1
SELECTIVITY_WEIGHT=1
BOXES_TO_KEEP=100
PENALTY_CONST=1
MIN_BOXES=10
MAX_BOXES=50
LIGAND_DIR=ligands
BOXES_DIR=boxes
RESULT_DIR=results
LIGAND_NAME=120_ligand
LIGAND_NAME=121_ligand
LIGAND_NAME=130_ligand
LIGAND_NAME=132_ligand
LIGAND_NAME=BAM_ligand
LIGAND_NAME=BMZ_ligand
```

B.3 Feature Definition File

This file contains definitions used to specify pharmacophore features. The default feature definition file, `phase_feature.ini`, is provided in `$SCHRODINGER/phase-vversion/data`. This file contains commonly used definitions for the six basic feature types. You can create your own feature definition file for a particular phase run. The file should be stored in the working directory for the run, and should be named `jobname_feature.ini`, where *jobname* is the name of the current job as specified in the Phase main input file.

Feature definition files contain blocks of data for each feature type. Feature types can be either the default types, such as acceptor, donor, or hydrophobic, or custom features. Each feature has a geometry, which can be one of `point`, `group`, or `vector`, and a projected point type, which depends on the geometry. Projected point types include donor and a range of acceptor types for vector geometries, and aromatic ring for group geometries.

Each block of data for a feature has the following format:

```
#FEATURE           Beginning of a new feature type block
#IDENTIFIER char  Single character feature identifier
#COMMENT string   Feature comments
#INCLUDE           Beginning of include block
pattern1         Pattern to include
pattern2         Pattern to include
...
#EXCLUDE           Beginning of exclude block
pattern1         Pattern to exclude
pattern2         Pattern to exclude
...
```

The `INCLUDE` block must contain at least one pattern; the `EXCLUDE` block can be empty. The identifier character must be one of the standard set, A, D, H, N, P, R, X, Y, or Z.

Individual patterns have the following format:

```
string1 string2 int1 int2 int3 int4 int5 [# string3]
```

The components of the patterns are described in [Table B.6](#).

Table B.6. Feature definition pattern components.

Component	Description
<i>string1</i>	SMARTS pattern. For hydrophobic or aromatic features this string may be default, indicating that the default mechanism that calls underlying libraries should be used instead of pattern matching.
<i>string2</i>	Geometry definition. The allowed values are <code>point</code> , <code>vector</code> , and <code>group</code> . The <code>point</code> and <code>vector</code> strings may be followed by the index of an atom in the SMARTS pattern, in parentheses: for example, <code>point(2)</code> . This index defines the point or vector atom, and by default is the first atom in the SMARTS pattern. The <code>group</code> string may be followed by a comma-separated list of atom indices, in parentheses, which define the group atoms. The default is all atoms.
<i>int1</i>	Reserved for future use. Set it to 1.
<i>int2</i>	Reserved for future use. Set it to 1.
<i>int3</i>	Projected point type, which can be one of the following: <ul style="list-style-type: none"> 0 no projected points -1 donor -2 acceptor, sp³, 1 lone pair (lp) -3 acceptor, sp², 1 lone pair -4 acceptor, sp, 1 lone pair -5 acceptor, sp³, 2 lone pairs -6 acceptor, sp², 2 lone pairs -7 acceptor, sp, 3 lone pairs -8 aromatic ring -9 acceptor, planar, 3 lone pairs
<i>int4</i>	Indicates whether this pattern is used (0) or ignored (1).
<i>int5</i>	Indicates whether this pattern is a default pattern (1) or not (0).
<i>string3</i>	Optional comments.

An example of a feature definition file is shown below. This file contains definitions of 3 types: acceptor, donor and hydrophobic.

```
#FEATURE
#IDENTIFIER D
#COMMENT donor: hydrogen atom attached to oxygen, nitrogen, sulfur or carbon
#INCLUDE
[#1][O;X2]          vector(1)  0 1 -1  0 1 # OH
[#1]S[#6]           vector(1)  0 1 -1  0 1 # SH
[#1][#7]            vector(1)  0 1 -1  0 1 # any NH
#EXCLUDE
[#1]OC(=O)          point(1)   0 1  0  0 1 # exclude carboxyl group
[#1]O[S;X3]=O       point(1)   0 1  0  0 1 #
```

```
#FEATURE
#IDENTIFIER A
#COMMENT acceptor: oxygen, nitrogen or sulfur with at least one lone pair
#INCLUDE
n1c[nH]cc1          vector(1) 0 1 -3 0 1 # his
O=[C,c]             vector(1) 0 1 -6 0 1 # carbonyl oxygen
[O;X2] (~[A,a])C    vector(1) 0 1 -5 0 1 # oxygen with two lone pairs
#EXCLUDE
O=C[O-,OH]          point      0 1 0 0 1 #
[#7;X3] [*]=[O,S]   point      0 1 0 0 1 # general amide
[N;X3] (C) (C) [C;X3] point      0 1 0 0 1 #
[N;X3] [a]           point      0 1 0 0 1 # planar N bonded to aring
#FEATURE
#IDENTIFIER H
#COMMENT hydrophobic feature
#INCLUDE
default             point      1 1 0 0 1 # default calls mmphob library
#EXCLUDE
```

In addition to features, projected point features can be included in the feature definition file. These features are defined by a point at a specified distance along the vector from a donor or acceptor atom. The format of a projected feature block is the same as for a feature, except that the initial keyword is `#PROJECTED_FEATURE`, and there is an additional `#EXTEND_DISTANCE` keyword that defines the distance of the projected point site from the donor or acceptor atom. An example of a projected point feature for a donor is given below.

```
#PROJECTED_FEATURE
#EXTEND_DISTANCE 1.8
#IDENTIFIER D
#COMMENT donor: hydrogen atom attached to oxygen, nitrogen, sulfur or carbon
#INCLUDE
[#1][O;X2]          vector(1) 0 1 -1 0 1 # OH
[#1]S[#6]           vector(1) 0 1 -1 0 1 # SH
[#1][#7]            vector(1) 0 1 -1 0 1 # any NH
#EXCLUDE
[#1]OC(=O)          point(1) 0 1 0 0 1 # exclude carboxyl group
[#1]O[S;X3]=O
```

B.4 Inactives Scoring Input File

The input file for inactives scoring (`phase_inactive`) contains *keyword=value* strings that provide instructions for scoring hypotheses with respect to inactives. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in [Table B.7](#). This file is automatically generated by the utility `pharm_score_inactives`. A sample input file is shown below.

Table B.7. Keywords for inactives scoring.

Keyword	Description
<code>inactiveWeight</code>	Required. Weight of the inactives score in the final score.
<code>phaseOptionsFile</code>	Required. Phase main input file for scoring inactives. The following keywords must be set in this file: <code>FINAL_BOX_SIZE</code> From <code>phase_partition</code> job. <code>USE_VOLUME</code> From <code>phase_scoring</code> job. <code>ALIGN_CUTOFF</code> From <code>phase_scoring</code> job. <code>ALIGN_WEIGHT</code> From <code>phase_scoring</code> job. <code>VECTOR_WEIGHT</code> From <code>phase_scoring</code> job. <code>VOLUME_WEIGHT</code> From <code>phase_scoring</code> job. <code>LIGAND_NAME</code> For each inactive molecule.
<code>ligandArchive</code>	Required. Name of the archive file (<code>.tar</code>) containing the ligand multi-conformer Maestro files for each ligand specified in the main input file. Must be stored in the current directory.
<code>ligandDir</code>	Required. Name of the directory used to store the ligands when they were archived, and therefore of the temporary directory that they will be extracted to.
<code>hypoArchive</code>	Required. Name of the archive file (<code>.tar</code>) containing the hypotheses. For each hypothesis, the files <code>hypoDir/hypoID.mae</code> , <code>hypoDir/hypoID.tab</code> and <code>hypoDir/hypoID.xyz</code> must be present in the archive. Must be stored in the current directory.
<code>hypoDir</code>	Required. Name of the directory used to store the hypotheses when they were archived, and therefore of the temporary directory that they will be extracted to.
<code>survivalScore(hypoID)</code>	Required. Survival score from actives scoring for the given hypothesis. There should be one record containing a survival score for each hypothesis for which inactive scores are wanted. Inactive scores are calculated only for hypotheses whose survival score is listed. Records may be deleted for hypotheses whose inactive score is not required.
<code>featureFile</code>	Required. Name of the feature definition file that was used to create the hypotheses.
<code>tableFile</code>	Required. Name of plain text file containing results.
<code>csvFile</code>	Name of CSV file containing results.

```
inactiveWeight=1
phaseOptionsFile=score_inactives_phase.inp
ligandArchive=score_inactives_ligandFiles.tar
ligandDir=.ligands.tmp
hypoArchive=score_inactives_hypoFiles.tar
hypoDir=.hypotheses.tmp
```

```

survivalScore(DHRRR_37)=14.9862
survivalScore(DHRRR_40)=14.8790
...
survivalScore(AAADHH_16)=13.4861
survivalScore(AAADHH_12)=13.4861
featureFile=score_inactives_feature.ini
tableFile=ScoreInactivesData.tab
csvFile=ScoreInactivesData.csv

```

B.5 Hypothesis Clustering Input File

The input file for clustering of hypotheses (`phase_hypoCluster`) contains *keyword=value* strings that provide instructions for clustering hypotheses according to their geometric similarity. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in [Table B.7](#). This file is automatically generated by the utility `pharm_cluster_hypotheses`. A sample input file is shown below.

Table B.8. Keywords for hypothesis clustering.

Keyword	Description
<code>phaseOptionsFile</code>	Required. Phase main input file with combined options from <code>phase_partition</code> and <code>phase_scoring</code> . The following options must be set in this file: <code>FINAL_BOX_SIZE</code> From <code>phase_partition</code> job. <code>ALIGN_CUTOFF</code> From <code>phase_scoring</code> job. <code>ALIGN_WEIGHT</code> From <code>phase_scoring</code> job. <code>VECTOR_WEIGHT</code> From <code>phase_scoring</code> job.
<code>hypoArchive</code>	Required. Name of the archive file (<code>.tar</code>) containing the hypotheses. For each hypothesis, the files <code>hypoDir/hypoID.mae</code> , <code>hypoDir/hypoID.tab</code> and <code>hypoDir/hypoID.xyz</code> must be present in the archive. Must be stored in the current directory.
<code>hypoDir</code>	Required. Name of the directory used to store the hypotheses when they were archived, and therefore of the temporary directory that they will be extracted to.
<code>featureDefFile</code>	Required. Name of the feature definition file used to create the hypotheses.
<code>featureTolFile</code>	Name of the feature-matching tolerances file. Required only if feature-matching tolerances were used to create the hypotheses
<code>clusterFile</code>	Required. Name of the output file containing results of the cluster analysis. Usually named <code>jobname_hypoCluster.clu</code> .

Table B.8. Keywords for hypothesis clustering.

Keyword	Description
linkageMethod	Method to be used for linking clusters. Allowed values are single, average, and complete. single Use the highest similarity between any two objects from the two clusters. Produces diffuse, elongated clusters. average Use the average similarity between all pairs of objects from the two clusters. complete Use the lowest similarity between any two objects from the two clusters. Produces compact, spherical clusters.
survivalScore(<i>hypoID</i>)	Required. Survival score from scoring of actives for the given hypothesis. There should be one record containing a survival score for each hypothesis that is to be clustered. Records may be deleted for hypotheses that you do not wish to cluster.

```

phaseOptionsFile=cluster_hypotheses_phase.inp
hypoArchive=cluster_hypotheses_hypoFiles.tar
hypoDir=.hypotheses.tmp
featureDefFile=cluster_hypotheses_feature.ini
linkageMethod=complete
clusterFile=cluster_hypotheses_hypoCluster.clu
survivalScore(DHRRRR_37)=14.9862
survivalScore(DHRRRR_40)=14.8790
survivalScore(DHRRRR_43)=14.8454
...

```

B.6 Multiple QSAR Model Input File

The input file for `phase_multiQsar` contains *keyword=value* strings that provide instructions for creation and use of the QSAR models. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in Table B.10. A sample input file is shown below.

Table B.9. Keywords for building multiple QSAR models with `phase_multiQsar`.

Keyword	Description
modelType	Type of model to build. Allowed values are atom and pharm. Default: atom.
numTrain	Required. Number of molecules in the training set. The first <i>numTrain</i> ligands from <i>phaseOptionsFile</i> (see below) are assigned to the training set, and the rest are assigned to the test set.

Table B.9. Keywords for building multiple QSAR models with `phase_multiQsar`.

Keyword	Description
<code>gridSpacing</code>	Grid spacing, in angstroms. Must lie between 0.5 and 4.0. The recommended and default value is 1.0.
<code>actProperty</code>	Required. The name of the activity property exactly as it appears in the ligand Maestro files. Missing activities are set to zero.
<code>maxFactors</code>	Required. Maximum number of PLS factors to include in the QSAR model. Models are created for the full sequence of models with the number of factors set to 1,...,maxFactors.
<code>useVolumeGroups</code>	Consider only atoms of the same MacroModel type when computing volume score overlaps. This favors alignments that superimpose chemically similar atoms. Allowed values are <code>true</code> and <code>false</code> . Default: <code>false</code> .
<code>phaseOptionsFile</code>	Required. Phase main input file with combined options for <code>phase_partition</code> and <code>phase_scoring</code> . The following keywords must be set in this file: FINAL_BOX_SIZE From <code>phase_partition</code> job. USE_VOLUME From <code>phase_scoring</code> job. ALIGN_CUTOFF From <code>phase_scoring</code> job. ALIGN_WEIGHT From <code>phase_scoring</code> job. VECTOR_WEIGHT From <code>phase_scoring</code> job. VOLUME_WEIGHT From <code>phase_scoring</code> job. LIGAND_NAME For each ligand in the training and test sets. LIGAND_NAME records for the training set should come first.
<code>ligandArchive</code>	Required. Name of archive file (<code>.tar</code>) containing ligand multiconformer Maestro files for each ligand specified in <code>phaseOptionsFile</code> .
<code>ligandDir</code>	Required. Name of the directory used to store the ligands when they were archived, and therefore of the temporary directory that they will be extracted to.
<code>hypoArchive</code>	Required. Name of archive file (<code>.tar</code>) containing hypothesis files for each hypotheses for which <code>hypoID</code> is specified. For each hypothesis, the files <code>hypoDir/hypoID.mae</code> , <code>hypoDir/hypoID.tab</code> and <code>hypoDir/hypoID.xyz</code> must be present in the archive.
<code>hypoDir</code>	Required. Name of the directory used to store the hypotheses when they were archived, and therefore of the temporary directory that they will be extracted to.

Table B.9. Keywords for building multiple QSAR models with `phase_multiQsar`.

Keyword	Description
<code>hypoID</code>	Required. Hypothesis for which QSAR model is to be built. There should be one record containing this keyword for each hypothesis for which you want a QSAR model. This list need not contain every hypothesis in the hypotheses archive.
<code>featureFile</code>	Required. Name of feature definition file used to create hypotheses.
<code>featureCutoffFile</code>	Name of file that defines positional tolerances for matching different types of pharmacophore features during ligand alignment. See Section B.9 on page 229 for a description of the format. Positional tolerances are enforced only after applying a tolerance to the intersite distances. If omitted, matching is done purely on intersite distances.
<code>featureRadiusFile</code>	Required if <code>modelType=pharm</code> . Name of file that defines feature radii. The format of this file is identical to <code>featureCutoffFile</code> , but the distances need not be the same.
<code>tableFile</code>	Required. Name of plain text file containing a summary of each QSAR model.
<code>csvFile</code>	Name of CSV file containing a summary of statistics for each QSAR model.
<code>resultArchive</code>	Required. Name of archive file (<code>.tar</code>) containing QSAR model results.
<code>resultDir</code>	Required. Name of directory for storing QSAR model results, relative to the current working directory.

```

modelType=atom
numTrain=27
gridSpacing=1
maxFactors=3
actProperty=r_phase_Ligand_Activity
useVolumeGroups=false
phaseOptionsFile=build_qsar_phase.inp
ligandArchive=build_qsar_ligandFiles.tar
ligandDir=.ligands.tmp
hypoArchive=build_qsar_hypoFiles.tar
hypoDir=.hypotheses.tmp
hypoID=DHRRR_37
hypoID=DHRRR_40
hypoID=DHRRR_43
...
hypoID=AAARRR_22
hypoID=AAADHH_16
hypoID=AAADHH_12

```

```
featureFile=build_qsar_feature.ini  
tableFile=BuildQsarData.tab  
csvFile=BuildQsarData.csv  
resultDir=BuildQsarResults  
resultArchive=build_qsar_results.tar
```

B.7 QSAR Model Input File

The input file for the QSAR module contains *keyword=value* strings that provide instructions for creation and use of the QSAR model. An exclamation point "!" may be used to add comments to input file. The allowed keywords and their values are given in [Table B.10](#). A sample input file is shown below.

Table B.10. Keywords for the QSAR input file

Keyword	Description
runMode	Legal values are train and test. Indicates whether a new model will be created (train) or whether an existing model will be imported (test). Required.
maeFile	Maestro file containing the molecules of interest. Required if runMode=train.
actFile	Text file containing activity data for the structures in maeFile. There should be one activity value per line, with no extraneous data or characters. Either actFile or actProperty (but not both) must be specified if runMode=train.
actProperty	The name of the activity property exactly as it appears in maeFile. Missing activities are set to zero. Either actFile or actProperty (but not both) must be specified if runMode=train.
pharmFile	Used for creating or testing a pharmacophore-based model. This file contains coordinates of site points that have been aligned to a particular hypothesis, one set of points for each molecule in maeFile. The file may be obtained by running phase_fileSearch on a Maestro file containing the molecules of interest. Valid when runMode=train or runMode=test. If runMode=test and modelFile contains a pharmacophore-based model, then pharmFile must be specified.
featureRadiusFile	File that defines the size of pharmacophore features. Each line in the file should contain a 1-character feature type, followed by a radius in angstroms. These radii are used only in the model creation process. Valid only when runMode=train and pharmFile has been specified.
modelFile	QSAR model file. The model is exported if runMode=train; the model is imported if runMode=test. Required if runMode=test.

Table B.10. Keywords for the QSAR input file

Keyword	Description
outputFile	File for ordinary output. The default is to write to standard output.
numTrain	The number of molecules in maeFile that will be used to train the model. The default is all molecules. Use the duplex option to control which molecules are assigned to the training set. Valid only when runMode=train.
duplex	A non-negative integer value that controls how the training set molecules are selected. If duplex=0, the first numTrain molecules in maeFile are used. If duplex>0, the value is treated as a random seed to sample numTrain molecules from maeFile. The default is duplex=0. Valid only when runMode=train.
gridSpacing	The distance in angstroms between neighboring points in the 3D grid. The default is 1.0. Values may range from 0.5 to 4.0. Valid only when runMode=train.
maxFactors	The maximum number of PLS factors to include in the QSAR model. Statistics and predictions are ultimately accessible for the full sequence of models with the number of factors set to 1,...,maxFactors. Valid only when runMode=train, in which case maxFactors must be specified.
printModel	Boolean (true or false) indicating whether or not a summary of the model should be written to outputFile. The default is printModel=false.
printPred	Boolean indicating whether or not predicted activities should be written to outputFile. If runMode=train, the training set and test set predictions are written separately. The default is printPred=false. printPred=true is allowed only when maeFile has been specified.
printBits	Boolean indicating whether or not volume occupation bit strings should be written to outputFile. printModel=true produces the full list of volume elements and atom classes that define the bit set. If runMode=train, bit strings for the training set and test set molecules are written separately. printBits=true is allowed only when maeFile has been specified.
csvPred	Name of CSV file containing the set membership, observed activity, and predicted activities for each ligand in the QSAR model.
csvBits	Name of CSV file containing the observed activity and volume occupation bits for each ligand in the QSAR model. This file can be supplied to the utility canvasPLS with the option -autoScaleOff to get the exact same model and predictions produced by phase_qsar.

```
runMode=train
maeFile=steroids.mae          ! 31 molecules
actFile=steroids_act.txt      ! 31 activity values
modelFile=steroids_model.dat
numTrain=21
```

```
duplex=1234567                ! Random split: 21 train / 10 test
gridSpacing=1.0
maxFactors=4
printModel=true
printPred=true
printBits=true
```

B.8 Feature Frequencies File

This file is used to set the minimum and maximum allowed feature frequencies for common pharmacophore perception. It is named `FeatureFreq.tab`. Each line contains the letter code for the feature type, followed by the minimum and the maximum number of occurrences of that feature in any hypothesis. The example below shows the default frequencies.

```
#####
#                                                                 #
# Feature frequency file. Used to set minimum and maximum allowed feature #
# frequencies for common pharmacophore perception. You may change these #
# limits, but do not make any other modifications to this file.         #
#                                                                 #
#####
A 0 4
D 0 4
H 0 4
N 0 4
P 0 4
Q 0 0
R 0 4
X 0 4
Y 0 4
Z 0 4
END_OF_FEATURE_DATA
```

For example, the text `A 0 4` indicates that each common pharmacophore will be restricted to contain between zero and four acceptors (inclusive). If you had some prior knowledge of the problem at hand, you could adjust these frequencies to narrow the focus in accordance with that knowledge. For example, suppose it has been established that all actives bind to a specific site on the receptor through a hydrogen bond, where the ligand acts as an acceptor. In that case, you have justification to require that each common pharmacophore contain at least one acceptor.

B.9 Feature-Matching Tolerances File

This file is used to set feature-matching tolerances. When searching for matches, this file should be named *hypoID.tol*, where *hypoID* is the hypothesis identifier used to define other hypothesis-related files. Although this file contains no hypothesis-specific information, the naming convention is required for the file to be used when searching for matches for a specific hypothesis. You must make a copy of it with the appropriate name for each hypothesis for which you want to use feature-matching tolerances.

The file contains one line for each feature type for which tolerances are to be used. Each line consists of a single character feature type and a tolerance value in angstroms, separated by a space. If a feature type is omitted, a default tolerance of 1.0 is used for that feature type. The feature type ? can be used to define a default cutoff for any feature type not listed in the file. The following is a sample feature-matching tolerances file:

```
#####
#
# Feature matching tolerances applied when hypotheses are scored with respect #
# to actives. You may change the tolerances, but do not make any other #
# modifications to this file. To completely disable the use of tolerances, #
# remove the FEATURE_ALIGN_CUTOFF_FILE option from score_actives_phase.inp. #
#
#####
A 1
D 1
H 1.5
N 0.75
P 0.75
R 1.5
X 1
Y 1
Z 1
END_OF_FEATURE_DATA
```

B.10 Hypothesis-Specific Tolerances File

This file is used to set tolerances for the specific features of a hypothesis in a search for matches. The file must be named *hypoID.dxyz*, where *hypoID* is the hypothesis identifier used to name hypothesis-related files. It contains multiple lines, each consisting of a single character feature type and a tolerance value, separated by a space. The file must contain one line for each site in the hypothesis, and there is a one-to-one mapping of the tolerances to the sites in the hypothesis. To see how the hypothesis maps to the reference ligand, you can use the Edit Hypothesis panel in Maestro.

The following is a sample hypothesis-specific tolerances file:

```
D 1.50
H 2.00
H 1.50
R 2.00
R 1.50
```

B.11 Site Mask File

This file is used to determine whether specific sites are matched or not in a partial match. The file must be named *hypoID.mask*, where *hypoID* is the hypothesis identifier used to name hypothesis-related files. The file must contain one line for each site in the hypothesis, with a 1, a 0 or a -1 on each line. These numbers determine how the site is matched:

- 1: Site must be matched.
- 0: Site can be matched but need not be matched
- 1: Site must not be matched.

For example, consider a hypothesis that contains the site types D, H, H, R, and R. To require that every partial match contains the donor site but not the second hydrophobic site, the corresponding site mask file should contain the following 5 lines:

```
4 D 1
5 H 0
6 H -1
9 R 0
11 R 0
```

The first two columns are the same as in the *hypoID.xyz* file. The third column contains the numbers that indicate how the site is matched.

Note: The Phase 2.0 format, in which only the column of numbers is given (the third column in the above example) is still supported, but the new format is recommended.

An alternative for excluding sites from matching is to use a hypothesis rules file—see [Section B.12 on page 230](#).

B.12 Hypothesis Rules File

Feature rules allow generalized matching according to permitted features and prohibited features for each site in the hypothesis. The file must be named *hypoID.rules*, where *hypoID* is the hypothesis identifier used to define other hypothesis-related files. This file will be used when you search for matches if it is present with the other input files, and `useFeatureRules` is not set to `false` in the input file.

The feature rules file must contain one line for each site in the hypothesis. Each line must contain the feature number followed by a string of features that are permitted at this site; these can optionally be followed by a string of features that are prohibited at this site. At a minimum, the feature rules file must contain the first two entries from each line in the corresponding *hypoID.xyz* file. The prohibited feature string, if included, is only used when doing partial matching.

Suppose you have the following *hypoID.xyz* file:

```
4 A 5.8663 -0.0455 1.5777
5 D 8.5193 0.6900 2.3939
6 H 9.5900 3.6582 1.8851
9 R 7.2866 2.6408 1.0380
11 R 0.8596 0.3614 1.2689
```

When finding matches to this hypothesis you want the following rules to apply:

- The hydrophobic site should match both hydrophobic and aromatic features.
- The aromatic sites should match both aromatic and hydrophobic features.
- The hydrophobic feature should never be overlaid onto an ionic site.

To achieve this behavior, you would construct the following feature rules file:

```
4 A
5 D
6 HR NP
9 RH
11 RH
```

The rule for site 6, which is a hydrophobic site, means that it can be matched to a hydrophobic or aromatic feature in the database. If partial matching is in use, each partial match that fails to include site 6 (i.e., it's not matched to an H or R in the molecule), is checked to make sure that it does not inadvertently match a negative or positive feature in the database molecule.

Vector scoring is turned off completely if any permitted feature string contains more than just the original feature type (as in the above example). This is done to avoid possible errors and inconsistencies when vector and non-vector features are aligned.

B.13 Database Search Input File

This file is named *jobname_dbsearch.inp* and is automatically generated by the utility *phasedb_findmatches*. The file contains keyword=value pairs, as for the other input files. You should not normally need to modify this file, but in case you do, the full description of the keywords and their dependencies are given in Table B.11. Keywords that you should not modify are marked in the table. In this file, you must supply only the keywords that are relevant to the selected run mode. If you leave other keywords in the file, they will be flagged as errors, and the job will not run. You can comment out optional keywords or keywords that are illegal for the run mode using an exclamation point (!). In this table, the value of a keyword is represented in italics>.

Table B.11. Description of keywords in the database search input file.

Keyword	Description
<i>alignCutoff</i>	Alignment score cutoff in fitness function. Not used as a hit filter unless <i>useHardAlignCutoff</i> is set to <i>true</i> . Default: 1.2 Å.
<i>alignPenalty</i>	Partial match alignment penalty (see text on page 101). Default: 1.2.
<i>alignWeight</i>	Weight of alignment score in fitness function. Must be non-negative. Default: 1.0.
<i>dbPathName</i>	Name of the database, including the absolute path. Required. Do not modify.
<i>deltaDist</i>	Tolerance (Å) used to match intersite distances in the find step. While <i>deltaDist</i> may be changed, the recommended value is twice the final box size that was used in the Find Common Pharmacophores step when the hypothesis was developed. The default final box size is 1.0 Å. Illegal in <i>fetch</i> mode. Default: 2.0 Å.
<i>flexAmideOption</i>	Amide torsional angle sampling option for flexible searches. Legal values are <i>vary</i> (vary angles), <i>orig</i> (keep original angles), and <i>trans</i> (make angles trans). The default is <i>vary</i> . Legal only in modes that include <i>flex</i> .
<i>flexMaxConfs</i>	Maximum number of conformers per molecule in flexible searches. Default: 100. Legal only in <i>flex</i> modes.
<i>flexConfsPerBond</i>	Maximum number of conformations per rotatable bond to generate for each molecule. Default: 10. Legal only in <i>flex</i> modes.
<i>flexMaxRelEnergy</i>	Conformational energy window in kJ/mol for flexible searches. Default: 41.84, i.e., 10.0 kcal/mol. Legal only in <i>flex</i> modes.
<i>flexSearchMethod</i>	Conformational search method for flexible searches. Allowed values are <i>rapid</i> and <i>thorough</i> . The default is <i>rapid</i> . Legal only in <i>flex</i> modes.

Table B.11. Description of keywords in the database search input file. (Continued)

Keyword	Description
hitFile	Hit file, in Maestro format. Required; default is <i>jobname-hits.mae</i> . Holds the conformations that produced matches, sorted by decreasing fitness and aligned to the hypothesis. Do not modify.
hitListFile	Hit list file. This file exists only while <code>phase_dbsearch</code> is running and it contains all the information necessary to reconstitute the hit list that <code>phase_dbsearch</code> maintains in memory. Do not modify.
hypoID	Prefix for all hypothesis files. At minimum, the files <i>hypoID.def</i> and <i>hypoID.xyz</i> must exist. If using a reference ligand, <i>hypoID.mae</i> and <i>hypoID.tab</i> must also exist. Required.
matchFile	Match file. Holds a lookup table that is used to rapidly retrieve hits in the fetch step. Required in <code>fetch</code> mode, illegal in <code>flex</code> mode. Do not modify.
maxHits	The maximum number of hits that will be written to <code>hitFile</code> , selected in order of fitness. If you want to view the hits in Maestro, this number should not be very large. Default:1000.
maxHitsPerMol	Maximum number of hits per molecule that will be written to <code>hitFile</code> . Default is 1.
minSites	Minimum number of sites in the hypothesis that must be matched. Optional. If the hypothesis contains 3 or more sites, <code>minSites</code> must be at least 3. If the hypothesis contains 1 or 2 sites, this option should not be used. The default is to match all sites in the hypothesis. Not valid in <code>fetch</code> mode.
preferBigMatches	Preference for partial matches containing a greater number of sites should be favored. Legal values are <code>true</code> and <code>false</code> . If set to <code>true</code> , matches involving fewer than <i>n</i> sites will not be sought if there are any matches with <i>n</i> sites. The default is <code>true</code> . Not valid in <code>fetch</code> mode.
refConfIndex	Identifies the reference conformation. This is a zero-based index, so a value of 16 indicates that the reference conformation is the 17th structure. Required if vector or volume scores are to be computed. Not valid in <code>fetch</code> mode. Do not modify.
runMode	Run mode. Allowed values are <code>find+fetch</code> , <code>find+fetch+flex</code> , <code>fetch</code> , <code>fetch+flex</code> , <code>flex</code> . Required.

Table B.11. Description of keywords in the database search input file. (Continued)

Keyword	Description
timeLimit	CPU time limit in seconds for finding matches for each molecule. CPU usage is checked before each conformer from a given molecule is searched, and matching is terminated if the time limit is exceeded at that point. The time limit does not apply to the process that generates conformers during the search, so if any of the <code>flex</code> options are specified and a molecule has a large number of rotatable bonds, the overall CPU time may significantly exceed the imposed limit. The default is unlimited CPU time. Irrelevant in <code>fetch</code> mode. A value of <code>-1</code> indicates no time limit.
useDbKeys	Use 3D database keys to filter out molecules that cannot match. Allowed values are <code>true</code> and <code>false</code> . Default: <code>true</code> . Valid only when <code>runMode</code> includes <code>find</code> .
useDeltaHypo	Apply hypothesis-specific matching tolerances. Allowed values are <code>true</code> and <code>false</code> . If set to <code>true</code> , requires the file <code>hypoID.dxyz</code> . Not valid when <code>runMode</code> includes <code>fetch</code> .
useExclVol	Apply excluded volumes to hits. Allowed values are <code>true</code> and <code>false</code> . If set to <code>true</code> , requires the file <code>hypoID.xvol</code> or <code>hypoID.ev</code> . If both files exist, <code>hypoID.ev</code> is used.
useExistingSites	Use existing database sites when searching. Should be <code>false</code> if the database and hypothesis feature definitions differ, in which case sites will be generated using the hypothesis feature definitions. Valid only when <code>runMode</code> includes <code>find</code> . Default: <code>true</code> .
useFeatureCutoffs	Apply feature-specific matching tolerances. Allowed values are <code>true</code> and <code>false</code> . If set to <code>true</code> , requires the file <code>hypoID.tol</code> . Not valid when <code>runMode</code> includes <code>fetch</code> .
useFeatureRules	Apply feature-matching rules, which associate permitted and prohibited features with each site in the hypothesis. Allowed values are <code>true</code> and <code>false</code> . If set to <code>true</code> , requires the file <code>hypoID.rules</code> . If the feature rules permit any site to be matched to more than one type of feature, vector scoring is turned off. Irrelevant when <code>runMode</code> includes <code>fetch</code> .
useHardAlignCutoff	Apply <code>alignCutoff</code> as a hit filter. Allowed values are <code>true</code> and <code>false</code> . By default, <code>alignCutoff</code> is used only to compute fitness, not to eliminate hits. If set to <code>true</code> , hits with <code>alignScore > alignCutoff</code> are rejected.
useQSARModel	Apply QSAR model to hits. Allowed values are <code>true</code> and <code>false</code> . If set to <code>true</code> , requires the file <code>hypoID.qsar</code> .
useRefLigand	Use a reference ligand if set to <code>true</code> . If set to <code>true</code> , requires the files <code>hypoID.mae</code> and <code>hypoID.tab</code> . If <code>false</code> , vector and volume scoring will not be done. Not valid when <code>runMode</code> includes <code>fetch</code> . Allowed values are <code>true</code> and <code>false</code> .

Table B.11. Description of keywords in the database search input file. (Continued)

Keyword	Description
useSiteMask	Apply a site mask to partial matches. Allowed values are <code>true</code> and <code>false</code> . If set to <code>true</code> , requires the file <code>hypoID.mask</code> . Irrelevant when <code>runMode</code> includes <code>fetch</code> .
useVolumeGroups	Compute volume scores using overlap only between atoms of the same MacroModel atom type. Allowed values are <code>true</code> and <code>false</code> . If set to <code>true</code> , alignments that superimpose chemically similar atoms are scored higher.
vectorCutoff	Vector score cutoff for filtering matches. Must be in the range $[-1, 1]$. Hits whose vector score is lower than this value are discarded. Requires a reference ligand, specified by <code>refCtFile</code> . Default: <code>-1.0</code> .
vectorWeight	Weight of vector alignment score in fitness function. Requires a reference ligand, specified by <code>refCtFile</code> . Default: <code>1.0</code> .
volumeCutoff	Volume score cutoff for filtering matches. Must be in the range $[0, 1]$. Hits whose volume score is lower than this value are discarded. Requires a reference ligand, specified by <code>refCtFile</code> . Default: <code>0.0</code> .
volumeWeight	Weight of volume score in fitness function. Requires a reference ligand, specified by <code>refCtFile</code> . Default: <code>1.0</code> .

B.14 Maestro File Search Input File

The input file for searching a Maestro file for matches to a hypothesis is named `jobname_fileSearch.inp` and contains `keyword=value` pairs, as do the other input files. The full description of the keywords and their dependencies are given in Table B.12. You can comment out optional keywords or keywords that depend on other keyword settings using an exclamation point (!).

Table B.12. Description of keywords in the Maestro file search input file.

Keyword	Description
<code>maeFile</code> <code>sdFile</code>	Maestro file or SD file to be searched. May be compressed (<code>.maegz</code> or <code>.mae.gz</code> ; <code>.sd.gz</code> or <code>.sdf.gz</code>). May contain multiple conformations per molecule. Successive structures are treated as conformations of a single molecule only if the titles match and the connectivity specifications are identical. If successive structures differ only in their stereochemistry, they should have different titles if they are to be treated as separate molecules.

Table B.12. Description of keywords in the Maestro file search input file. (Continued)

Keyword	Description
<code>flex</code>	Boolean (<code>true</code> or <code>false</code>) indicating whether or not flexible searching should be done. If <code>flex=true</code> , conformations are generated as needed for each molecule in <code>maeFile</code> . Use the other <code>flex</code> options to control how conformations are generated. The default is <code>flex=false</code> .
<code>flexSearchMethod</code>	Conformational search method. Legal values are <code>rapid</code> and <code>thorough</code> . The default is <code>rapid</code> . Valid only when <code>flex=true</code>
<code>flexMaxConfs</code>	Maximum number of conformations per molecule to generate. The default is 100. Valid only when <code>flex=true</code> .
<code>flexConfsPerBond</code>	Maximum number of conformations per rotatable bond to generate for each molecule. Default: 10. Valid only in modes that include <code>flex</code> .
<code>flexMaxRelEnergy</code>	Conformational energy window, in kJ/mol. The default is 41.84, i.e., 10.0 kcal/mol. Valid only when <code>flex=true</code> .
<code>flexAmideOption</code>	Amide torsional angle sampling option. Legal values are <code>vary</code> (<code>vary</code> angles), <code>orig</code> (<code>keep</code> original angles), and <code>trans</code> (<code>make</code> angles <code>trans</code>). The default is <code>vary</code> . Valid only when <code>flex=true</code> .
<code>preferBigMatches</code>	Preference for partial matches containing a greater number of sites should be favored. Legal values are <code>true</code> and <code>false</code> . If set to <code>true</code> , matches involving fewer than n sites will not be sought if there are any matches with n sites. The default is <code>true</code> .
<code>scoreInPlace</code>	Option for scoring matches in place. Legal values are <code>true</code> and <code>false</code> . If set to <code>true</code> , no conformations are generated, and fitness is computed directly from the supplied poses, without aligning to the hypothesis. The default is <code>false</code> .
<code>timeLimit</code>	CPU time limit in seconds for finding matches for each molecule. CPU usage is checked before each conformer from a given molecule is searched, and matching is terminated if the time limit is exceeded at that point. The time limit does not apply to the process that generates conformers during the search, so if any of the <code>flex</code> options are specified and a molecule has a large number of rotatable bonds, the overall CPU time may significantly exceed the imposed limit. The default is unlimited CPU time.
<code>atomTypeVol</code>	Compute volume scores using overlap only between atoms of the same MacroModel atom type. This favors alignments that superimpose chemically similar atoms. Legal values are <code>true</code> and <code>false</code> . The default is <code>false</code> .
<code>hypoID</code>	Hypothesis ID. Required. This is the stem of the hypothesis file names (e.g. <code>hypoID.tab</code> , <code>hypoID.xyz</code>) and must include the path if these files are not located in the current directory.

Table B.12. Description of keywords in the Maestro file search input file. (Continued)

Keyword	Description
<code>deltaDist</code>	Tolerance (Å) used to match intersite distances. The recommended value is twice the final box size that was used in the Find Common Pharmacophores step when the hypothesis was developed. The default final box size is 1.0 Å; the default value of <code>deltaDist</code> is 2.0 Å
<code>minSites</code>	Minimum number of sites in the hypothesis that must be matched. Optional. The default is to match all sites in the hypothesis. Must be greater than or equal to 3. Do not use this option if the hypothesis contains only 1 or 2 sites.
<code>useSiteMask</code>	Boolean for applying a site mask to partial matches. The default is <code>true</code> if the file <code>hypoID.mask</code> is present. Set to <code>false</code> to disable. See Section B.11 on page 230 for information on site masks.
<code>useFeatureCutoffs</code>	Boolean for applying feature-based tolerances to matches. The default is <code>true</code> if the file <code>hypoID.tol</code> is present. Set to <code>false</code> to disable.
<code>useQSARModel</code>	Boolean for applying a QSAR model for the hypothesis to the hits. The default is <code>true</code> if the file <code>hypoID.qsar</code> is present. Set to <code>false</code> to disable.
<code>useExclVol</code>	Boolean for applying excluded volume filtering to hits. The default is <code>true</code> if the file <code>hypoID.xvol</code> is present. Set to <code>false</code> to disable.
<code>alignWeight</code>	Weight of alignment score in fitness function. Default is 1.0.
<code>alignCutoff</code>	Alignment cutoff in fitness function. Default is 1.2 Å.
<code>alignPenalty</code>	Alignment penalty in fitness function. Default is 1.2 Å.
<code>vectorWeight</code>	Weight of vector alignment score in fitness function. Default is 1.0.
<code>volumeWeight</code>	Weight of volume score in fitness function. Default is 1.0.
<code>hitFile</code>	Hit file, in Maestro format. Required. Holds the conformers that produced matches, aligned to the hypothesis. For a given molecule, matches are sorted by decreasing fitness. Scores and predicted activities (if any) are written as properties.
<code>maxHitsPerMol</code>	Maximum number of hits per molecule that will be written to <code>hitFile</code> . Default is 1.

Phase Utilities

In addition to the utilities described in [Chapter 12](#) and [Chapter 13](#), there are several utilities provided with Phase that may be useful in some circumstances. These utilities are described in this appendix.

C.1 combine_hits

The utility `combine_hits` combines the structures in one or more hit files to generate a single file. Redundant structures are not eliminated from the hit file. The syntax of the command is as follows:

```
$SCHRODINGER/utilities/combine_hits hitFileList hitFileOut maxHits
    [hitsAreGrouped]
```

The command-line arguments are given in [Table C.1](#).

Table C.1. Arguments for combine_hits command.

Argument	Description
<i>hitFileList</i>	Text file containing the list of <code>phase_dbsearch</code> hit files to combine, one file name per line.
<i>hitFileOut</i>	Name of the combined hit file to be created.
<i>maxHits</i>	Maximum number of hits to write to the output file.
<i>hitsAreGrouped</i>	Indicates whether the original hit files are grouped by molecule. Allowed values are <code>true</code> and <code>false</code> . The default is <code>true</code> .

C.2 combine_matches

The utility `combine_matches` is intended to be used when a distributed search job fails, and the matches from the individual subjobs need to be combined into a single file. The syntax of the command is as follows:

```
$SCHRODINGER/utilities/combine_matches matchFileList matchFileOut
```

where *matchFileList* is a text file containing the list of `phase_dbsearch` match files to combine, one file name per line, and *matchFileOut* is the name of the combined match file to be created.

C.3 convert_hypoDistToXYZ

The utility `convert_hypoDistToXYZ` creates a hypothesis `.xyz` file from a file containing intersite distances. The syntax of the command is as follows:

```
convert_hypoDistToXYZ hypoID
```

where *hypoID* is the prefix used to identify input and output files. The input file should be named *hypoID*.dist and it should contain the alphabetized variant, followed by the strict lower triangle of intersite distances, as indicated by the following example:

```
ADHP
d(D,A)
d(H,A) d(H,D)
d(P,A) d(P,D) d(P,H)
```

Here the $d(x,y)$ are the distances.

Since intersite distances are unaffected by reflection operations, this program creates two mirror image hypothesis files: *hypoID*_1.xyz and *hypoID*_2.xyz. A least-squares technique is applied to align the two sets of sites, and the RMSD is reported, so it will be obvious whether the mirror images are equivalent.

There are no reference ligands for these two hypotheses, so two SD files, *hypoID*_1.sdf and *hypoID*_2.sdf are created to help visualize the points. To achieve colors similar to those used in the Phase GUI, the pharmacophore features in the SD files are represented by different elements—see [Section C.5](#) for a description.

C.4 convert_hypoFeatures

The utility `convert_hypoFeatures` attempts to convert a hypothesis to use a new set of feature definitions. The syntax of the command is as follows:

```
convert_hypoFeatures hypoID featureFile newHypoID
```

where *hypoID* is the prefix used to name the files in the existing hypothesis, e.g., *hypoID*.def, *hypoID*.mae, *hypoID*.tab, *hypoID*.xyz. If these files are not in the current directory, you must include the path in *hypoID*. *featureFile* is the file containing the new feature definitions. *newHypoID* is the prefix for the converted hypothesis files, and must be different from *hypoID*.

C.5 create_hypoSDFFile

The utility program `create_hypoSDFFile` creates an SD file to help visualize hypotheses that have no reference ligand. To visualize the hypothesis, simply import the SD file into Maestro. The syntax of the command is as follows:

```
create_hypoSDFFile hypoID
```

The input hypothesis file should be named *hypoID*.xyz. The pharmacophore sites in that file will be used to create the structure file *hypoID*.sdf. Pharmacophore features in the SD file are represented by atoms whose colors are similar to those used to render features in the Phase GUI, as follows:

A	oxygen (red)
D	nitrogen (blue)
H	boron (green)
N	bromine (dark red)
P	sodium (dark blue)
R	silicon (orange)

C.6 create_hypoFiles

This utility creates the .def, .mae, .xyz and .tab hypothesis files from a single reference ligand structure and a feature definition file. The syntax of the command is as follows:

```
create_hypoFiles maeFile hypoID [fdFile]
```

where *hypoID* is the prefix used to name the hypothesis files, *maeFile* is the Maestro file that contains the reference ligand structure, and *fdFile* is an optional feature definition file. If this last file is omitted, the default feature definitions in the installation are used. Note that all surface-accessible sites in the reference ligand are included in the .xyz file. You can edit the hypothesis in the Edit Hypothesis panel.

C.7 phase_volCalc

This utility calculates a matrix of overlapping volume values between structures in one or two Maestro files. Volumes are computed by treating each molecule as a set of atomic spheres. The syntax of the command is as follows:

```
phase_volCalc -mae1 maeFile1 [-range1 beg1:[end1]] [-titles1]  
-mae2 maeFile2 [-range2 beg2:[end2]] [-titles2] [options]
```

The options are given in [Table C.2](#).

Table C.2. Options for `phase_volCalc`.

Option	Description
<code>-mae1 maeFile1</code>	File whose structures will span the rows of the volume matrix. Only one structure is held in memory at any given time.
<code>-range1 beg1:[end1]</code>	Range of structures to consider in <code>maeFile1</code> . Examples: 5:10 structures 5 through 10 2: structures 2 through end of file. The default is to consider all structures.
<code>-titles1</code>	Add the structure title to each row.
<code>-mae2 maeFile2</code>	File whose structures will span the columns of the volume matrix. <code>maeFile2</code> and <code>maeFile1</code> may be the same file. All column structures are held in memory at the same time, so this should be the smaller of the two structure sets.
<code>-range2 beg2:[end2]</code>	Range of structures to consider in <code>maeFile2</code> . The default is to consider all structures.
<code>-titles2</code>	Add the structure title to each column.
<code>-out csvFile</code>	File to which the comma-separated volume matrix should be written. If omitted, the matrix will be written to standard output.
<code>-radius r</code>	Use a fixed radius for each atomic sphere. The default is to use the van der Waals radii. Note that Phase volume scoring uses a fixed radius of 1.7.
<code>-scale propName</code>	Scale each radius by an atom-level property in the associated Maestro file. <code>propName</code> must correspond to a real-valued property, and must therefore begin with <code>r_</code> .
<code>-grid spacing</code>	Grid spacing for volume calculation. The default is 1.0, which is also the grid spacing used in Phase volume scoring.
<code>-hydrogens</code>	Consider hydrogens when computing volumes. Phase volume scoring ignores hydrogens.
<code>-atomTypeVol</code>	Compute overlapping volumes only between atoms of the same Macro-Model type. Phase volume scoring does not consider atom type by default, but <code>phase_fileSearch</code> and <code>phase_multiQsar</code> have options to do so. Note that if <code>-hydrogens</code> is omitted, only hydrogens attached to carbons are ignored.
<code>-scores</code>	Compute volume scores, $V_{\text{common}}/V_{\text{total}}$, rather than just the overlapping volumes V_{common} .

C.8 rmsdcalc

This utility computes the RMSD between each structure in a given Maestro file and a corresponding reference structure from a second Maestro file. The correspondence between structures is done by title, so reference titles must be unique, and each structure for which the RMSD is sought must have a title that matches one of the reference structure titles. Note that RMSD cannot be computed between two structures unless they have the same connectivity. The syntax of the command is as follows:

```
rmsdcalc -screen screenFile -ref refFile -out {csvFile|maeFile} [options]
```

The file specifications are given in [Table C.3](#), and the options are described in [Table C.4](#).

Table C.3. File specifications for the *rmsdcalc* command.

File Specification	Description
-screen <i>screenFile</i>	Maestro file (.mae, .maegz, or .mae.gz) containing the structures for which RMSDs are sought.
-ref <i>refFile</i>	Maestro file (.mae, .maegz, or .mae.gz) containing the reference structures.
-out { <i>csvFile</i> <i>maeFile</i> }	Output file. If the extension is .csv, a comma-separated file is created with the title and RMSD for each structure in <i>screenFile</i> . If the extension is .mae, .maegz, or .mae.gz, a Maestro file is created with each structure from <i>screenFile</i> and the property <code>r_rmsdcalc_RMSD</code> .

Table C.4. Options for the *rmsdcalc* command.

Option	Description
-align	Perform least-squares alignment of each structure onto its reference before computing the RMSD. If the output file is a Maestro file, it will contain aligned structures. The default is to compute the RMSD using the supplied coordinates.
-hydrogens {ignore all hetero}	Hydrogen treatment: ignore Ignore all hydrogens (the default) all Treat all hydrogens hetero Treat only hydrogens on heteroatoms
-renumber	Renumber structures to identify symmetry-related structures that yield the lowest RMSD values. If used with hydrogens included, a large number of symmetry-related structures can be generated, resulting in large memory and CPU usage.
-title <i>propName</i>	Use property <i>propName</i> as the source of titles. Must begin with <code>s_</code> or <code>i_</code> .

C.9 flex_align

This utility uses the `phase_shape` technology to flexibly align a set of structures to a flexible template, then reports the alignments associated with the template conformer that yielded the highest average similarity to all structures. The conformers can be generated during the job or they can be pregenerated. The syntax of the command is as follows:

```
flex_align -screen screenFile -shape shapeFile -JOB jobName [options]
           [job-options]
```

The file containing the structures to align is specified with `-screen`, and the file containing the template structure is specified with `-shape`. Both files can be in either Maestro or SD format. If *shapeFile* and *screenFile* are the same, all pairs of structures in the file are rigidly aligned to see which one is most suitable as a template. *shapeFile* cannot be the same as *screenFile* when using pregenerated conformers.

The job name is specified with `-JOB`, and the aligned structures are written to a compressed Maestro file named *jobName_flex_align.maegz*.

The options are listed in [Table C.5](#). The standard Job Control options (see [Section 2.3](#) of the *Job Control Guide*) are supported, with the restriction that you can only specify a single host with `-HOST`. The common job options `-INTERVAL`, `-LOCAL`, and `-WAIT` are also supported.

Table C.5. Options for the `flex_align` command.

Option	Description
<code>-rigid {screen shape both}</code>	<i>screenFile</i> or <i>shapeFile</i> (or both) contains pre-computed conformers. Consecutive structures with identical titles and connectivities are treated as conformers of the same molecule.
<code>-title <i>propName</i></code>	Use property <i>propName</i> as the source of titles. Valid only with <code>-rigid</code> .
<code>-norm 1 2 3 4</code>	Similarity normalization scheme. The similarity between the template A and a screening molecule B is a function of the overlap $O(A,B)$ between the two, and the self-overlaps $O(A,A)$ and $O(B,B)$: $\text{Sim}(A,B) = O(A,B)/f(O(A,A), O(B,B))$. The normalization scheme determines the form of the function f : <ol style="list-style-type: none"> 1 $f = \max\{O(A,A), O(B,B)\}$ (default) 2 $f = \min\{O(A,A), O(B,B)\}$ 3 $f = O(A,A)$ 4 $f = O(B,B)$
<code>-best</code>	Report the best alignment found for each structure in <i>screenFile</i> . This may not yield a good consensus alignment because the best alignment doesn't always come from the same template conformer.

Table C.5. Options for the *flex_align* command.

Option	Description
<code>-atomTypes</code> <i>atomTypes</i>	Consider atom types when computing similarities. If this option is used, overlapping volumes are computed only between atoms of the same type, so that alignments favor superposition of chemically similar atoms. The supported atom typing schemes are: <code>mmod</code> MacroModel atom types. <code>element</code> Elemental types. <code>pharm</code> Generalized pharmacophoric types as defined for Phase QSAR models: D—H-bond donor hydrogen H—hydrophobic/non-polar N—negative ionic P—positive ionic W—electron-withdrawing X—other
<code>-atomWeights</code> <i>propName</i>	Use the real atom-level property <i>propName</i> in <i>shapeFile</i> to weight the overlap with the template atoms. This is achieved by scaling the radius of each atom by the cube root of its weight. Values in the range 0 to 1 are recommended. Valid only when <i>shapeFile</i> is a Maestro file.
<code>-flexAmideOption</code> <i>option</i>	Amide torsion sampling option. Allowed values are <code>vary</code> , <code>orig</code> , and <code>trans</code> . Not valid with <code>-rigid</code> . Default: <code>vary</code> .
<code>-flexMaxConfs</code> <i>maxConfs</i>	Maximum number of conformations per molecule to generate. Not valid with <code>-rigid</code> . Default: 100.
<code>-flexConfsPerBond</code> <i>numPerBond</i>	Maximum number of conformations per rotatable bond to generate for each molecule. The total number of conformers is bounded by the product of <i>numPerBond</i> and the number of rotatable bonds. If <i>maxConfs</i> is increased, <i>numPerBond</i> may have to be increased as well in order to retain additional conformers for more rigid structures. Not valid with <code>-rigid</code> . Default: 10.
<code>-flexMaxRelEnergy</code> <i>energy</i>	Conformational energy window in kJ/mol. Not valid with <code>-rigid</code> . Default: 104.6 kJ/mol (25 Kcal/mol).
<code>-flexSearchMethod</code> <i>method</i>	Conformational sampling method. Allowed values are <code>rapid</code> and <code>thorough</code> . Not valid with <code>-rigid</code> . Default: <code>rapid</code> .
<code>-hydrogens</code>	Consider hydrogens attached to non-carbon atoms when computing shape similarity. Hydrogens attached to carbon atoms are always ignored. Default: ignore all hydrogens.
<code>-refine</code>	Generate additional conformers for each consensus alignment to see if a higher similarity to the template conformer can be achieved. These options are not valid with <code>-best</code> .

Table C.5. Options for the flex_align command.

Option	Description
-refineAmideOption <i>option</i>	Amide torsion sampling option for refinement. Allowed values are vary, orig, and trans. Only valid with -refine. Default: vary.
-refineMaxConfs <i>maxConfs</i>	Maximum number of conformations per molecule to generate for refinement. Only valid with -refine. Default: 1000.
-refineConfsPerBond <i>numPerBond</i>	Maximum number of conformations per rotatable bond to generate for each molecule during refinement. The total number of conformers is bounded by the product of <i>numPerBond</i> and the number of rotatable bonds. If <i>maxConfs</i> is increased, <i>numPerBond</i> may have to be increased as well in order to retain additional conformers for more rigid structures. Only valid with -refine. Default: 100.
-refineMaxRelEnergy <i>energy</i>	Conformational energy window in kJ/mol for refinement. Only valid with -refine. Default: 104.6 kJ/mol (25 Kcal/mol).
-refineSearchMethod <i>method</i>	Conformational sampling method for refinement. Allowed values are rapid and thorough. Only valid with -refine. Default: thorough.
-save	Save all job files.

Getting Help

Schrödinger software is distributed with documentation in PDF format. If the documentation is not installed in `$(SCHRODINGER)/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*.

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

- Check the Auto-Help text box, which is located at the foot of the main window. If help is available for the task you are performing, it is automatically displayed there. Auto-Help contains a single line of information. For more detailed information, use the online help.
- If you want information about a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Maestro menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- For information about a panel or the tab that is displayed in a panel, click the Help button in the panel, or press F1. The help topic is displayed in your browser.
- For other information in the online help, open the default help topic by choosing Online Help from the Help menu on the main menu bar or by pressing CTRL+H. This topic is displayed in your browser. You can navigate to topics in the navigation bar.

The Help menu also provides access to the manuals (including a full text search), the FAQ pages, the New Features pages, and several other topics.

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

- *Maestro User Manual*, for detailed information on using Maestro
- *Maestro Command Reference Manual*, for information on Maestro commands
- *Maestro Overview*, for an overview of the main features of Maestro
- *Maestro Tutorial*, for a tutorial introduction to basic Maestro features
- *Phase Quick Start Guide*, for a tutorial introduction to Phase
- Phase Frequently Asked Questions pages, at https://www.schrodinger.com/Phase_FAQ.html

- Known Issues pages, available on the [Support Center](#).

The manuals are also available in PDF format from the Schrödinger [Support Center](#). Local copies of the FAQs and Known Issues pages can be viewed by opening the file `Suite_2009_Index.html`, which is in the `docs` directory of the software installation, and following the links to the relevant index pages.

Information on available scripts can be found on the [Script Center](#). Information on available software updates can be obtained by choosing Check for Updates from the Maestro menu.

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail: help@schrodinger.com
USPS: Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone: (503) 299-1150
Fax: (503) 299-4532
WWW: <http://www.schrodinger.com>
FTP: <ftp://ftp.schrodinger.com>

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

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

On UNIX you can obtain the machine and system information listed above by entering the following command at a shell prompt:

```
$SCHRODINGER/utilities/postmortem
```

This command generates a file named `username-host-schrodinger.tar.gz`, which you should send to help@schrodinger.com. If you have a job that failed, enter the following command:

```
$SCHRODINGER/utilities/postmortem jobid
```

where *jobid* is the job ID of the failed job, which you can find in the Monitor panel. This command archives job information as well as the machine and system information, and includes input and output files (but not structure files). If you have sensitive data in the job

launch directory, you should move those files to another location first. The archive is named `jobid-archive.tar.gz`, and should be sent to help@schrodinger.com instead.

If Maestro fails, an error report that contains the relevant information is written to the current working directory. The report is named `maestro_error.txt`, and should be sent to help@schrodinger.com. A message giving the location of this file is written to the terminal window.

More information on the `postmortem` command can be found in [Appendix A](#) of the *Job Control Guide*.

On Windows, machine and system information is stored on your desktop in the file `schrodinger_machid.txt`. If you have installed software versions for more than one release, there will be multiple copies of this file, named `schrodinger_machid-N.txt`, where *N* is a number. In this case you should check that you send the correct version of the file (which will usually be the latest version).

If Maestro fails to start, send email to help@schrodinger.com describing the circumstances, and attach the file `maestro_error.txt`. If Maestro fails after startup, attach this file and the file `maestro.EXE.dmp`. These files can be found in the following directory:

```
%USERPROFILE%\Local Settings\Application Data\Schrodinger\apcrash
```

Glossary

Active compound—A compound that shows high affinity for the biological target. Synonymous with the term *ligand*.

Active set—The set of active compounds that is used to develop a pharmacophore model. This set does not necessarily include all active compounds.

Excluded volume—A region of space in a pharmacophore hypothesis that should not be occupied by any atom of an active compound.

Feature—see **Pharmacophore feature**

Hit—A structure in a 3D database that is found to contain an arrangement of site points that can be mapped to the pharmacophore hypothesis. A hit is not necessarily active, but it is presumed to have a greater than average probability of being active if it was retrieved using a valid hypothesis.

Hypothesis—see ***n*-Point pharmacophore hypothesis**

Inactive compound—A compound that shows little or no affinity for the biological target.

Intersite distance—The distance between any two site points in a pharmacophore.

Ligand—see **Active compound**

Negative compound—A compound that is inactive, yet highly similar in structure to one or more known actives. Some compounds are negative because they lack certain key pharmacophore features found in true actives. Other negatives may actually satisfy exactly the same pharmacophore hypotheses as the actives, but possess extraneous structural characteristics that prevent binding.

Pharmacophore feature—A characteristic of chemical structure that may facilitate a noncovalent interaction between a ligand and a biological target. Examples are hydrogen-bond acceptor ("A"), hydrogen-bond donor ("D"), hydrophobe ("H"), positive ionic center ("P"), negative ionic center ("N").

Pharmacophore site—The labeling and location of a particular pharmacophore feature within a molecule. For example, a hydrogen bond acceptor site could simply be a nitrogen atom which carries an available lone pair. A hydrophobic site might be a methyl carbon or the

centroid of a phenyl ring. The term *site point* is often used interchangeably with pharmacophore site.

***n*-Point pharmacophore**—Any 3D arrangement of *n* pharmacophore features.

***n*-Point pharmacophore hypothesis**—A specific 3D arrangement of *n* pharmacophore features, with associated uncertainties in the feature positions. High affinity ligands in their active conformations are expected to contain pharmacophore sites that can be mapped (within the limits of uncertainty) to any valid hypothesis. A given hypothesis may contain features that are associated with a single mode of binding, or it may contain features that are common to two or more modes of binding.

Reference ligand—The ligand that provides the pharmacophore that defines a hypothesis. In pharmacophore model development, this pharmacophore yields the highest multi-ligand alignment score for the active-set ligands. The reference ligand matches the hypothesis exactly, and has a perfect fitness score.

Site point—see **Pharmacophore site**

3D Database—A set of molecules, each of which is represented by one or more 3D conformational models, augmented with a pharmacophore-based representation of the molecules. A 3D database includes feature types and site point coordinates for each conformation.

Variant—The set of feature types in a pharmacophore. For example, the variant AHH indicates a 3-point pharmacophore containing one hydrogen bond acceptor and two hydrophobic sites.

Vector feature—A pharmacophore feature that contains directionality, such as a hydrogen bond acceptor, hydrogen bond donor, or aromatic ring. A vector feature does not necessarily have vector geometry.

Vector geometry—the geometric characteristics of hydrogen bond acceptors and donors. Refers to the direction of lone pairs in a hydrogen-bond acceptor or the direction of the heavy-atom–hydrogen-atom bond in hydrogen-bond donors. Features with vector geometry must be vector features.

A

- acceptors
 - display appearance..... 30
 - explicit projected points 35
 - projected point type 34
- actives
 - choosing for model development 25, 117
 - matching criterion..... 41
 - requiring matches to specific 42, 211
 - scoring 50, 124
- activities
 - calculating for hits 108, 157, 234
 - conversion to log units..... 15, 17, 118
 - cutoffs for actives and inactives..... 25, 117
 - entering by hand 15
 - for command-line QSAR model..... 226
 - positive, negative contributions to 72
 - predicted by QSAR model.... 68, 92, 108, 159
 - scoring by 52, 125
 - selecting property for..... 15, 17, 224, 226
 - units 111
- Advanced Matching Options dialog box 106
- aligning molecules to hypotheses 140
- aligning non-model ligands..... 57
- Alignment Options dialog box..... 57
- alignment score
 - adjustment for partial matches..... 50, 101
 - cutoff..... 48, 232
 - definition..... 48
 - filtering by 51
 - weight in fitness function..... 232
- angles, displaying..... 7, 77, 78
- atom types
 - in QSAR models..... 201
 - use in shape queries 185
- atom-based QSAR models..... 64

B

- binding modes, number of 194
- box size 43
- box, definition 39

C

- chirality information, use of..... 18
- Choose Reference Ligand dialog box 81
- Cluster Hypotheses dialog box 58

- clustering, of hypotheses..... 58–59, 128–130, 193
- color scheme, pharmacophore features..... 30, 241
- conformational search method
 - database search 103
 - grid file search 177, 186, 245, 246
 - pharmacophore model development..... 20
- conformations, recognition of 115
- conformers
 - adding to database 152
 - eliminating redundant..... 22
 - generating extra for matches..... 104
 - generation method 20
 - postminimization 22
 - sampling options..... 21
 - thresholds for limiting number 23
- conventions, document..... xi
- counter ions, removing..... 17
- custom pharmacophore features 28
- cutoffs
 - alignment score..... 48
 - conformational comparison 23
 - conformational energy 23, 177, 186, 245, 246
 - feature-matching 160
 - hypothesis-specific 106, 160, 179, 229
 - intersite distances..... 104, 156
 - number of hypotheses..... 51
 - pharm set activity..... 25, 117
 - site-matching 104
 - specifying for grid file search..... 178
 - t-value filter..... 67
 - vector score..... 235
 - volume score..... 235
 - see also* thresholds

D

- database search
 - adding hypotheses 103
 - distributed processing 172
 - feature cutoffs 160
 - filtering with excluded volumes 108, 157, 234
 - fitness score 100
 - Maestro properties generated 109
 - partial matches..... 104, 161, 233
 - predicting activities with QSAR model... 108, 157, 159, 234
 - pre-screening 157, 171
 - restarting job..... 159, 162

- restriction to subsets 101, 159
 - search mode 155
 - selecting hypothesis 103
 - site cutoffs 104
 - tolerances 104, 232
 - database subsets
 - counting records in 170
 - creating 162
 - in shape-based searches 187
 - searching with 101, 159
 - databases
 - access permissions 173
 - access to 97
 - adding and deleting structures 149
 - adding conformers and sites 152
 - backing up 169
 - checking integrity 168
 - compacting 170
 - converting formats 167
 - duplicate structures 148
 - merging 166
 - recovering after errors 169
 - deleting ligands from run 15
 - directory
 - database 150
 - installation 2
 - ligands 112, 116
 - Maestro working 3
 - results 112
 - disk space requirements
 - pharmacophore search 44
 - distances, displaying 7, 77, 78
 - distributed processing 172, 176
 - donors
 - display appearance 30
 - explicit projected points 35
 - projected point type 34
- E**
- Edit Hypothesis dialog box 85, 86
 - energy window 23, 177, 186, 245, 246
 - entries, Project Table
 - adding to pharmacophore project 16
 - selecting for reference ligand 81
 - environment variable
 - SCHRODINGER 2
 - SCHRODINGER_PHASE_MAX_RETRY 152
 - SCHRODINGER_PHASE_PROGRESS_DIR 174
 - SCHRODINGER_PHASE_VERBOSITY 159, 189
 - ePlayer 108
 - excluded volumes
 - adding to hypothesis 59, 80
 - applying to hits 108, 157, 160, 179, 234, 237
 - definition 47
 - displaying 6
 - ligand-shaped 136
 - receptor-based 138, 197
 - sphere radii 198
 - steric clash based 137
 - Excluded Volumes dialog box 60
- F**
- feature definitions
 - adding custom 35
 - file format 218
 - mismatch between hypothesis and
 - database 103
 - modifying 30
 - specifying default file for 12
 - specifying file for create sites 119
 - feature-matching rules 106, 230
 - feature-matching tolerances 160, 179, 229
 - features—*see* pharmacophore features 27
 - fitness score
 - definition 100
 - modifying 108, 232, 237
 - flexible searching 103
- G**
- geometry, vector 252
 - Guide 7
- H**
- hits
 - applying excluded volumes to .. 108, 157, 234
 - calculating activities for 108, 157, 234
 - combining 239
 - definition 99
 - limiting the number of 108, 158
 - ordering of 100
 - hydrogens
 - adding to ligands 17
 - inclusion in shape-based similarity .. 186, 245

inclusion in volume calculation 242
 hydrophobic group definition file..... 120
 hypotheses
 adding QSAR model to 92
 adding sites 86
 aligning 79, 142
 aligning molecules to..... 140
 changing feature types..... 86
 clustering by geometric similarity 124,
 128–130
 clustering by ligand membership..... 193
 coloring 79
 consensus 144
 converting feature definitions 240
 creating 80
 displaying 6, 56, 79
 displaying labels 6
 editing 84
 exporting..... 57, 71, 74, 79
 filtering 50
 freestyle 80
 importing for search 103
 in Project Table..... 78
 ligand-based..... 80
 repositioning sites 86
 using receptor information 197
 Hypotheses Table panel 79

I

inactives
 choosing for model development 25, 117
 creating excluded volumes with 137
 scoring 50, 53, 127
 included volumes 189
 intersite distances
 creating hypothesis from 240
 definition..... 39
 displaying 7, 56, 77, 78
 matching cutoff..... 104, 156
 minimum for common pharmacophore 43
 ionization state, setting in model development. 19

J

job completion, checking 174
 job progress 174

L

labels, setting color of 12
 ligands
 adding from a file..... 15
 adding from another run 16
 adding to Project Table 58, 75
 aligning to hypothesis..... 140
 creating included volumes from 190
 deleting from run 15
 directory..... 116
 exporting aligned..... 58, 75
 grouping for pharmacophore model ... 42, 211
 preparation for pharmacophore model 111
 test set for QSAR model..... 66, 90, 117
 training set for QSAR model..... 66, 90
 log files..... 173

M

Maestro properties from database search..... 109
 Maestro, starting 3
 match file..... 99
 matches
 definition..... 99
 filtering 107
 partial 104
 tolerances 104
 matching rules..... 106

N

New Hypothesis dialog box 82, 83

O

output files
 pharm_build_qsar..... 132
 pharm_cluster_hypotheses 129
 pharm_create_sites 119
 pharm_find_common 122
 pharm_project 116, 133
 pharm_score_actives 125
 pharm_score_inactives..... 128
 phase_feature 120
 phase_hypoCluster 130
 phase_inactive..... 128
 phase_partition..... 123
 phase_qsar..... 134

- phase_scoring..... 126
- P**
- partial matches
- definition..... 99
 - inactives score adjustment..... 50
 - list of sites matched 109
 - searching for 104, 161
 - survival score adjustment 101
- patterns
- adding to features 33
 - ignoring..... 35
- pharm set
- changing membership of 24
 - defining 36, 117
- pharmacophore features 27
- adding patterns..... 32
 - built-in 27
 - converting in a hypothesis 240
 - custom..... 28, 35
 - definition file..... 218
 - display appearance..... 30
 - displaying 30
 - excluding functional groups from..... 34
 - ignoring patterns 35
 - inconsistent definitions 103
 - radius for QSAR model 226
- pharmacophore sites, defining 33
- pharmacophore, reference..... 48
- pharmacophore-based QSAR models 64
- Phase QSAR - Scatter Plot dialog box 71
- PLS factors, maximum..... 67
- post-hoc score 55
- product installation..... 247
- Project Table
- adding ligands from..... 16
 - adding ligands to..... 58
 - properties of hits 108
- projected points 35, 220
- projects, Phase pharmacophore model..... 115
- properties
- choosing for activity 15, 16
 - conformational energy 118
 - hits, imported into Maestro..... 108, 159
 - scoring by user-defined..... 125
 - use for excluded volume sphere radii 198
- Q**
- QSAR models
- adding to an existing hypothesis..... 92
 - analyzing..... 71
 - applying to hits 108, 159, 179, 237
 - atom-based..... 64
 - description 63
 - displaying 7
 - exporting 71
 - feature-based..... 134
 - filtering variables 67
 - importing 92
 - options for..... 67, 226
 - pharmacophore-based..... 64
 - scatter plot 70, 91
 - statistical definitions 205
 - test and training sets 66, 90
 - visualization of 72, 135
- QSAR Visualization Settings panel 73
- R**
- receptor
- creating hypothesis from 197
 - creating included volume from..... 190
- reference ligand..... 48
- activity score..... 52, 125
 - choosing for new hypothesis 82
 - displaying 79
 - dummy 78
 - relative conformational energy score.. 52, 125
- reference pharmacophore..... 48
- relative energy
- scoring by 52, 125
 - setting property name 118
- restarting jobs
- conformation generation/site creation 153
 - database creation/addition 151
 - database search 159, 162
 - shape-based searches 188
- run
- adding ligands from..... 16
 - definition..... 6
 - saving 7
 - storing QSAR model in 75

S

Schrödinger contact information..... 248

scores

- activity 52, 125
- alignment 48
- fitness 100
- post-hoc 55
- relative energy 52, 125
- selectivity 49
- site 48, 55
- survival 49
- vector 48
- volume 48, 105

selectivity score

- definition 49
- range 52

site mask 106, 179

site measurements

- displaying 7, 77, 78

site points

- maximizing number in search . 104, 156, 178, 233, 236
- number to match 104, 122, 156, 233, 237
- required matches 106, 161
- selecting number in hypothesis 40, 121

site score

- definition 48, 55
- range 52

site-matching tolerances 104

SMARTS patterns—*see* patterns

solvent molecules, removing 17

step guide 7

steps, navigation 7

stereoisomers

- generating in model development 18
- grouping of, for pharmacophore model 14
- separating for database addition 151

structures

- adding to database 149
- cleaning up 17
- duplicates in database 148
- requirements for hypothesis creation 81
- requirements for model development 13
- scoring in place 104
- sources for searching 101

subsets, of databases 162

survival score

- adjusting 52
- adjustment for partial matches 101
- definition 49

T

test set for QSAR model 66, 90, 117

thresholds

- conformer generation 23
- database search 232, 237
- hypothesis scoring 51
- see also* cutoffs

tolerances—*see* cutoffs, thresholds

toolbar, Phase panels 6, 77

training set for QSAR model 66, 90, 117, 227

V

variants

- definition 39, 252
- excluding from search 42
- list of available 40
- selecting 44

vector feature 27, 252

vector geometry 27, 252

vector score

- definition 48
- filtering threshold 51
- range 52
- weight in fitness function 235

View Clusters dialog box 59

volume score

- definition 48
- filtering matches by 235
- range 52
- using atom types 105, 156, 178, 235, 236
- weight in fitness function 235

W

weights

- fitness score 100
- specifying for grid file search 178
- survival score 49, 52

120 West 45th Street, 29th Floor
New York, NY 10036

Zeppelinstraße 13
81669 München, Germany

101 SW Main Street, Suite 1300
Portland, OR 97204

Dynamostraße 13
68165 Mannheim, Germany

8910 University Center Lane, Suite 270
San Diego, CA 92122

Quatro House, Frimley Road
Camberley GU16 7ER, United Kingdom

SCHRÖDINGER.