ORCHESTRAR™ Manual

SYBYL® 7.3
Late 2006

1699 South Hanley Rd.
St. Louis, MO
63144-2917

Phone: +1.314.647.1099
Fax: +1.314.647.9241
http://www.tripos.com

LEGAL NOTICE

Selected software programs for Comparative Molecular Field Analysis (CoMFA) and HQSAR methodologies contained or documented herein are covered by one or more of the following patents: US 6,208,942; US 5,025,388; US 5,307,287; US 5,751,605; AT E150883; BE 0592421; CH 0592421; DE 691 25 300 T2; FR 0592421; GB 0592421; IT 0592421; NL 0592421; SE 0592421. Surflex software programs for chemical analysis by morphological similarity methodologies contained or documented herein are covered by the patent US 6,470,305 B1.

SYBYL, UNITY, CoMFA, CombiFlexX, GeneFold, LeapFrog, LITHIUM, OptDesign, Alchemy, and ChemPrint are registered trademarks of Tripos, Inc. AUSPYX, Benchware, CAPRI, ChemEnlighten, CScore, DISCOtech, Distill, FlexiDock, GALAHAD, GASP, HiVol, HQSAR, Legion, MOLCAD, Molecular Spreadsheet, OptiDock, OptiSim, PD Triplets, ProTable, RECEPTOR, SARNavigator, Selector, SiteID, Tuplets, TRIAD, and Tripos Bookshelf are trademarks of Tripos, Inc.

Concord, DiverseSolutions, and StereoPlex are registered trademarks of the University of Texas System.
hint! is a registered trademark eduSoft LC.
Mardigras is a trademark of the Regents of the University of California.
MM2, MM3 and MM4 are registered trademarks of the University of Georgia.
Molconn-Z is a trademark of Hall Associates Consulting.
RACHEL is a trademark of Drug Design Methodologies.
Surflex, Surflex-Dock, and Surflex-Sim are trademarks of BioPharmics LLC.
ZAP is a trademark of OpenEye Scientific Computing.
"FairCom" and "c-tree Plus" are trademarks of FairCom Corporation and are registered in the United States and other countries.
All other trademarks are the sole property of their respective owners.

ORCHESTRAR Table of Contents

This page intentionally blank.

Chapter 1.

ORCHESTRAR Introduction

The recognition of homology between protein sequences provides invaluable information toward understanding the biological behavior and biochemical function of uncharacterized sequences. this recognition also enables prediction of three-dimensional structures through comparative modeling.

ORCHESTRAR is a suite of applications designed for comparative protein modeling. With ORCHESTRAR you can create an all-atom model based on a target protein sequence and one or more structural homologs. The homologs may be user-provided or identified using FUGUE (described in the FUGUE Manual).

The SYBYL interface to ORCHESTRAR allows you to:

- perform sequence alignment and homolog structural alignment;
- identify structurally conserved regions;
- model loops (structurally variable regions) using model-based and ab initio methods;
- model sidechains;
- analyze the final model and prepare it for use in further studies.

Acknowledgments

ORCHESTRAR was developed in the laboratory of Professor Sir Thomas Blundell at Cambridge University (UK).

License Requirements for ORCHESTRAR

The following licenses are required to run ORCHESTRAR:

- Biopolymer
- ORCHESTRAR—to access the functionality
- ORCHESTRAR_Interface—to access the ORCHESTRAR Project Manager within SYBYL

An additional MOLCAD license is optional

HARMONY requires the program Gnuplot.

This page intentionally blank.

Chapter 2.

ORCHESTRAR Tutorials

These tutorials will lead you through comparative protein modeling: from a protein sequence to the evaluation of an all-atom homology model.

- ORCHESTRAR Project Manager Tutorial on page 8
- ORCHESTRAR Unix Commands Tutorial on page 26

See also:

- ORCHESTRAR Hints and Tips on page 37

Hardware Note: Results may vary slightly on different hardware platforms because of the handling of floating point numbers and the iterative nature of some of the algorithms (especially CHORAL).

2.1 ORCHESTRAR Project Manager Tutorial

In this tutorial you will learn how to construct a comparative model for a query sequence using the SYBYL interface to ORCHESTRAR.

A number of homologs were obtained for a clinically relevant query sequence using the FUGUE program. The query sequence and homologs will serve as input into ORCHESTRAR.

The solved structure from which the query sequence was obtained is of human factor Xa (Stuart-Prower factor), a serine protease that is part of the clotting cascade. It is part of the prothrombinase complex catalyzing the conversion of prothrombin to thrombin. A deficiency in factor Xa causes abnormal blood coagulation (Stuart-Prower deficiency) and excessive bleeding. Sales of drugs used to inhibit factor Xa were greater than $1.35 billion in 2003.

More details about this structure may be found in:

> Scharer, K. et al. (2005) Quantification of Cation-Pi Interactions in Protein-Ligand Complexes: Crystal-Structure Analysis of Factor Xa Bound to a Quaternary Ammonium Ion Ligand. Angew.Chem., Int.Ed.Engl. 44:4400.

A matter of time: This tutorial requires approximately 30 minutes of personal time and 10 minutes of CPU time (tested on a Xeon 2.8 GHz Linux machine).

2.1.1 Set Up

The set of files used in this tutorial is stored in compressed format in $TA_DEMO/orchestrar. To access any of the files you must first uncompress the entire set into your current working directory.

1. Access the operating system from within SYBYL.

   ➢ At the SYBYL prompt, type: **sh**

2. Uncompress the files into your current working directory.

   ➢
   ```
   gzip -cd $TA_DEMO/orchestrar/fugue_run_query.tar.gz |
   tar xvf -
   ```

3. Exit the operating system shell and return to SYBYL.

   ➢ Type: **exit**

2.1.2 Create an ORCHESTRAR Project

1. Start ORCHESTRAR

   ➢ **Biopolymer >>> Model Proteins >>> ORCHESTRAR**

   The *ORCHESTRAR Project Manager* opens (dialog description on page 42).

2. Examine the *ORCHESTRAR Project Manager*.

   The *Project Manager* is the central hub where entry points into several modeling tasks can be performed. Buttons for these tasks are laid out on the left side of the dialog. Buttons for project management are located at the top of the dialog. Project notes are added automatically as steps in the modeling process are performed.

   This tutorial walks through each modeling tasks sequentially as they appear in the dialog.

3. Create a new project and import the homologs to a query sequence obtained using FUGUE.

   ➢ In the *ORCHESTRAR - Project Manager*, click **New Project**.

   ➢ In the *Select New Project Type* dialog, make sure that **FUGUE Run Results** is ***on*** and click **Import**.

   The *Import FUGUE Run* dialog opens (dialog description on page 45).

   ➢ Type **`tutorial`** as the Project Name.

   All the files related to this project will be stored in a new directory named tutorial.orc.

   ➢ To retrieve the FUGUE Run type **`fugue_run_query`** and press the keyboard carriage return or use the adjacent file browser to retrieve the file.

   The list in the dialog is populated with a number of HOMSTRAD families, each of which contains a set of homologs to the query sequence. Several families are listed as CERTAIN: two single member family homologs and two multi-member families.

   ➢ For this study, select the **sermam** family from which the comparative model will be constructed.

   ➢ In the Alignment File option menu, select **model_sequence_sermam_ma.ali** as the file that will provide sequence alignment information between the query sequence and the homologs.

➢ Click **Import**.

As the homologs are read into SYBYL, ORCHESTRAR structurally aligns them and displays their C-alpha trace, each in a different color (cycling through a set of 8 colors).

Once the homologs have been displayed, the Project Notes are updated in the *Project Manager*. The notes include the name and location of the new project as well as the data source and the name of the alignment file. Project notes will be added by ORCHESTRAR as each step in the modeling process is performed. You may add your own notes at any time by clicking within the Project Notes area and typing. Notes can be saved by clicking Save Notes.

2.1.3 Examine the Homolog Sequences and Structures

1. The first ORCHESTRAR task is to model the conserved regions.

➢ In the *Project Manager,* click **Model Conserved Regions**.

Two dialogs appear:

- the *Model Conserved Regions* dialog (dialog description on page 49)
- the *Sequence Viewer* (dialog description in the Biopolymer Manual). The *Sequence Viewer* is posted automatically by all operations that make use of it.

➢ You may want to reduce the size of the *Sequence Viewer* and reposition the SYBYL window and dialog for optimal viewing.

➢ In the *Model Conserved Regions* dialog, examine the list of homologs in the sermam family.

For each homolog the columns contain the homolog name, its color on the SYBYL screen, the percent identity to the query sequence, and the RMSD between its C-alpha atoms and those of the homolog with the top FUGUE score to the query sequence.

➢ In the *Sequence Viewer*, use the scroll bars to examine the query sequence and its alignment to the homolog sequences.

The sermam family has many members, and it may be beneficial to remove some homologs with lower identity to the query.

2. Edit the homolog list to remove homologs with low identity to the query.

   ➢ In the *Model Conserved Regions* dialog, click the **%ID** button above the list to sort it by decreasing % identity values.

   ➢ In the sorted homolog list, select the top 5 entries (1kigh, 1fxya, 1ppb, 1bbr, 2tbs), then **Invert** the selection.

   As each homolog is selected, its structure is highlighted on the SYBYL screen and its sequence is highlighted in the *Sequence Viewer*.

   ➢ Click **Remove Homolog**.

   ➢ Click **Yes** in the dialog prompting you to confirm the permanent removal of the selected homologs.

   The homologs are removed from the list, from the SYBYL screen and from the *Sequence Viewer*.

3. Structurally re-align the homologs to each other now that the list has changed.

   ➢ Click **Align by Structure**.

   ORCHESTRAR realigns the homologs based on their 3D structures.

   ➢ When prompted to use the sequence alignment that was created by the structural re-alignment, click **Yes**.

   This operation realigns the sequences in the *Sequence Viewer*.

2.1.4 Build and Analyze Structurally Conserved Regions (SCRs)

1. You are ready to build the structurally conserved regions.

   ➢ At the bottom of the *Model Conserved Regions* dialog, click **Build SCRs**.

   Usage Note: If ORCHESTRAR is unable to generate a good quality SCR model based on the scoring of environment-specific substitution tables, a dialog appears reporting the number of homologs with poor scores. The recommendation is to remove lower scoring homologs (flagged with a “?” in the textport and in the dialog) and to repeat the previous steps (Align by Structure, then Build SCRs). Another option is to remove the highest scoring homolog, so that the maximum global score is lowered and the distribution of values is narrower.

2. Examine the *Analyze Conserved Region* dialog.

   Upon successful building of the SCRs, ORCHESTRAR displays the *Analyze Conserved Region* dialog (dialog description on page 56).

➢ At the top of the dialog, look at **Select Model**.

The best model is selected automatically. In this example, it is **model_f4**, the model in which a small gap was filled. The status line reports the number of residues in the model and the number of residues and sidechains still missing from the model.

```
Model Status: 207 res, 34 missing res, 5 gaps, 175 missing sidechains
```

➢ Look at the other models in the **Select Model** menu.

Four models are available: model_c1, model_c2, model_c3, and model_f4. Model names are assigned automatically as model_*an*, where *a* is a single letter identifying the operation and *n* is an integer that is automatically incremented with each new model. Examples: model_c1 (conserved regions), model_f4 (after fixing small gaps).

➢ Select **model_f4**.

➢ Look at the **Homolog** list.

The homologs that were used to build the model are listed at the top of the dialog, but they not visible in the SYBYL window (as indicated by -Hidden- in the Color column).

➢ Turn your attention to the **SCR List** in the dialog.

- **Start** and **End** identify the residues at the beginning and end of an SCR.
- **Length** indicates the number of residues comprising the SCR.
- **Color** indicates the coloring of the SCR: ByAtomType or a named color.
- **Gap Length** shows the number of missing residues between an SCR and the one preceding it.

All SCRs are displayed because they are all selected, and the **Automatic** check box is ***on***.

3. Enhance the display of the SCRs.

➢ Color the SCRs in different colors by setting **Conserved Region** (below the list) to **Spectrum**.

➢ Set **Model Structure** to **Backbone Only** and **Label Termini**.

The SCRs of the newly constructed homology model are listed and selected in the dialog, and they are displayed in the SYBYL window (in M1).

4. Examine the *Sequence Viewer* (dialog description in the Biopolymer Manual).

   The selected model (model_f4) is placed just below the query sequence. By default, the text of the model and homolog sequences is colored by secondary structure: helices in red and sheets in blue. The corresponding menu item is **SeqViewer: View >>> Text Style >>> Color by Secondary Structure**.

5. Check the model for abnormal peptide bond lengths and Cα-Cα distances.

   ➢ Near the bottom of the *Analyze Conserved Region* dialog, click **Check Distances**.

   In the SYBYL window, the model is colored and labeled as follows:

   - Yellow: residues on either side of a gap. These residues are labeled.
   - Red: residues with peptide bonds >20% from the ideal length of 1.33 Å. These residues are labeled.
   - Blue: residues with peptide bonds within the acceptable distance range.

   In the textport, the following information is reported:

   - Residues with peptide bonds >20% from the ideal length of 1.33 Å. This list matches the residues colored red.
   - Residues with Cα-Cα >20% from the ideal distance of 3.8 Å.
   - Disconnected pairs, which consists of two sequential residues for which the amide bond is missing because the two atoms are >1 Å from the ideal peptide bond length of 1.33 Å.
   - Non-sequential residues connected to each other.

6. Check for non-ideal phi and psi angles.

   ➢ Click **Check Torsions**.

   In the SYBYL window, the model is colored and labeled as follows:

   - Red: Residues that lie in the OUTSIDE region of the Ramachandran plot. These residues are labeled.
   - Yellow: Residues that lie in the GENEROUS regions of the Ramachandran plot.
   - Blue: Residues that lie within CORE or ALLOWED regions of the Ramachandran plot.

   This information is also output to the SYBYL textport.

7. Check for steric clashes

   ➢ Click **Check for Clashes**.

   - Yellow: Residues involved in at least one steric clash. These residues are labeled.
   - Red: Atoms involved in steric clashes.
   - Blue: Residues not involved in steric violations.

2.1.5 Search Loops

1. The next ORCHESTRAR task is to search for loops to fill the remaining gaps.

   ➢ In the *Analyze conserved Regions* dialog, click **Search Loops**.

   The *Search Loops* dialog appears (dialog description on page 62).

   ➢ Make sure that the model selected at the top of the dialog is **model_f4**.

2. In the View Gaps section, examine the gap information in the **Gap** option menu.

   This menu contains a list of gaps defined by the residues flanking a span of missing residues followed by the number of missing residues. Selecting a gap from this menu will center the view on this gap and label the residues that flank the gap.

   ➢ In the **Gap** option menu, select the first gap, **GLU21:PHE26(4)**.

   SYBYL centers the view on this gap and labels flanking residues GLU21 and PHE26.

3. Examine the list of loop searches that will be performed.

   ➢ Turn your attention to the **Loop Searches** list.

   The gap selected at the top of the dialog is highlighted in this list.

   - **Q** or **H** indicates whether this search is queued for processing or has been placed on hold.
   - **Gap/Loop** column contain residues that flank a gap
   - **Length** the number of missing residues.
   - **Method** column contains loop search method according to the following:

    - **D** = fragment ***d***atabase (FREAD) created from HOMSTRAD; appropriate for 3-12 residue loops
    - **A** = ***a***b initio database (PETRA) of 10^8 secondary structure fragments; appropriate for 3-8 residue loops
    - **H** = ***h***omologs
  - **N-ter**—The first residue in the loop to be found. The number in parentheses (-0 by default) indicates the position of this residue with respect to the known residue that flanks this gap.
  - **C-ter**—The last residue in the loop to be found. The number in parentheses (+0 by default) indicates the position of this residue with respect to the known residue that flanks this gap.

4. Run loop searches

   ➢ Click **Search Loops** in the *Search Loops* dialog to begin loop searches.

   The search will take several minutes (5 minutes on a Xeon 2.8 GHz Linux machine).

2.1.6 Examine and Add Loop Solutions

1. Examine the loops solutions for one of the gaps.

   Upon successful completion of the loop search, ORCHESTRAR displays the *Add/Analyze Loops* dialog (dialog description on page 69).

   ➢ Make sure that **model_f4** is selected in the **Select Model** option menu.

   The status line reports the number of residues in the model and the number of residues and sidechains still missing from the model.

```
Model Status: 207 res, 34 missing res, 5 gaps, 175 missing sidechains
```

   ➢ Select the **GLU21:PHE26(4)** gap in the **Gap** option menu.

   This will center the view on this gap and list the loop solutions obtained for this gap.

   The top solution for the GLU21:PHE26(4) gap is 1kigh22, which spans residues ASN20:CYS27 (a total length of 8 residues), and was retrieved from one of the homologs. It has an RMSD to the gap flanking residues of 0.23 Å and a favorable Ek value (-0.05). It is colored green on the SYBYL screen.

   - **Loop Name** identifies the file containing this loop (`.atm` file in the LOOPS subdirectory).

- **Endpoints** name the first and last residues in the loop
- **Len** is the number of residues in the loop
- **Src** identifies the method that produced this loop: A (ab initio), D (database search) or H (homolog).
- **RMSD** is the root mean square deviation between the residues flanking the gap and the residues flanking the loop.
- **Ek** is the energy of the predicted loop in the target structure.
- **ESST** is the environmentally constrained substitution score of a loop (for database solutions only)
- **Run** indicates the run number of the loop search

In the list of loop solutions for this gap the top scoring loop is automatically marked with a '+' sign. This identifies it as the loop that will be added to the model in a subsequent step. Only one loop per gap may be marked for addition.

➢ Examine the many solutions for this particular gap listed in the **Loop Search Results** section of the dialog.

➢ Use the arrow buttons below the list to scroll through all loop solutions for this gap.

Only one loop may be examined at a time. It is flagged with an asterisk before its name in the list. It is colored red on the SYBYL screen (unless it also happens to be the loop marked with a + sign and is colored green).

➢ When you are done examining the loop solutions for this gap, click the C button between the two arrows.

➢ Scroll all the way back to the 1kigh22 loop and make sure that it is marked with a + sign.

2. Examine the loop solutions for the other gaps.

➢ Select each of the other gaps in the **Gap** option menu and look at the list of loop solutions.

Many loop solutions were found for all gaps except for ASP174:ASP179 for which only a single loop solution was found.

Occasionally, longer loop solutions or a larger number of solutions are desired for a gap, especially if the gap is not very large or if no loop solutions could be found for that gap. In this example, only one solution was found for the ASP174:ASP179 gap. You will change the loop search criteria for this gap and perform another loop search.

3. Change the search criteria for the loop that has only one solution.

   ➢ At the bottom of the *Add/Analyze Loops* dialog, click **Remodel Loops** to return to the *Search Loops* dialog.

   The list of loop searches is empty because all the searches have already been run.

   ➢ In the *Search Loops* dialog, click **Previous Loop Searches**.

   ➢ In the *Previous Loop Searches* dialog, activate **List Loop Searches for All Gaps**.

   The list in the dialog is populated with the conditions used for the first run.

   ➢ Highlight the **ASP174:ASP179** loop search and click **Copy Search** below the list.

   The *Copy Loop Search* dialog appears (dialog description on page 65).

   The dialog displays information for the ASP174:ASP179 gap: the flanking residues, the gap length, and the Cα/Cα distance between the flanking residues.

   ➢ In the Loop Search Parameters section:

   - Change the N-ter Flanking Residue to **TYR173 (-1)** using the left arrow. TYR173 is highlighted on the screen.
   - Change the C-ter Flanking Residue to **ALA180 (+1)** using the right arrow. ALA180 is highlighted on the screen.
   - Click **Add** to add this loop search to the Loop Search list and close the dialog.

   In the *Search Loop*s dialog, the list of Loop Searches has been updated and reflects the loop search addition.

   ➢ Click **Search Loops**.

   When the search finishes, the *Add/Analyze Loops* dialog will appear again.

   ➢ Select the **ASP174:ASP179** gap.

   This time, several loop solutions were found for this gap, and they appear in the list along with the loop solution from the previous run.

4. Explore the loop solutions for the ASP174:ASP179 gap.

   First, sort the list based on the run number.

- ➢ Click the **Run** button above the list to sort it by increasing run number.

The solution from the first run appears at the top of the list. It is still marked with a + sign and displayed in green on the screen.

Next, sort the list by increasing Ek values because the loop solution with the lowest Ek value is often a good one.

- ➢ Click the **Ek** button above the list to sort it by increasing Ek.
- ➢ Highlight the loop with the lowest Ek in the list (67005237).
- ➢ In the **Examine** section below the list of loop solutions, click the right arrow.

The loop is flagged with an asterisk in the list and colored red on the screen.

- ➢ Compare the two loop solutions visually.
  - 1kigh175, still marked with a + sign, is colored green.
  - 67005237 is colored red.

These two loops have geometries that are noticeably different and probably reflect two different conformations for the protein in this region. In fact, the ASP174:ASP179 loop comprises a portion of a Na binding site. The conformation of the 67005237 loop solution is likely to be the conformation of the loop when Na is not bound, while that of 1kigh175 is likely to be the conformation of the loop when the loop residues are coordinated by a Na atom.

Usage Note: When encountering this situation in your own work, you may want to build two different models based on the different loop solutions. Recommendations on how to proceed are provided at the end of this tutorial (see Recommended Work Flow to Build Another Model on page 25). For now, you will build the model using the loop solution with the lowest Ek value.

- ➢ With the 67005237 loop solution highlighted in the dialog, click **Mark** below the list.

A + sign appears in front of the loop solution in the list, and the loop is colored green on the screen. Because only one loop solution may be marked for each gap, the other solution is no longer marked nor is it displayed.

5. Look at the proposed model.

- ➢ In the SYBYL window, use the **Reset** icon and click **Everything**.

- ➢ Scale down the image to look at the entire model and the five green loops.
- ➢ Make sure that **Show All Marked Loops** is ***on*** (above the list).

The top scoring loop solution for each gap is colored green and lined up with the existing model (model_f4).

6. Meld the top loop solutions into the model.

- ➢ Press **Join All Marked Loops (+)** to meld the loop solutions into the gaps.

After each loop is melded into the appropriate gap it undergoes a quick minimization: 10 iterations on backbone atoms using the Powell method and the Tripos force field.

When the loops have been melded, the *Add/Analyze Loops* dialog will return and display the following status line:

```
231 res, 10 missing res, 0 gaps, 196 missing sidechains
```

The **Gap** option menu is empty.

2.1.7 Model and Analyze the Sidechains

1. Prepare the sidechain addition.

- ➢ At the bottom of the *Add/Analyze Loops* dialog, click **Model Sidechains**.

The *Model Sidechains* dialog appears (dialog description on page 77).

By default, the **Borrow Sidechain Conformations From Homologs** option is on. This enables ORCHESTRAR to borrow sidechain conformations from the homologs used in modeling.

- ➢ Make sure that Borrow Options is set to **Full Borrowing/Full Restrictions**.

This is the most conservative option in this menu instructing ORCHESTRAR to first borrow as many sidechain conformations as possible (Full Borrowing) from the homologs before using a rotamer library to add the remaining sidechains. This option uses only the chi values for the top scoring rotamers in the library (Full Restrictions).

Note: The homologs used for modeling are available for viewing in the **Select Homologs** list. They are hidden by default, but may be selected and viewed.

2. Add the sidechains to the model.

   ➢ Click **Add Sidechains**.

3. Analyze steric clashes between atoms in the sidechains.

   When the sidechain search is finished, the *Analyze Sidechains* dialog appears (dialog description on page 82)

   ➢ Toggle ***on*** **Visualize Sidechain Clashes**.

   Residues that contain atoms involved sidechain clashes are colored yellow, while all other residues are colored blue. Examine the yellow residues. Only two atoms are involved in van der Waals overlap. These are colored red.

2.1.8 Analyze the Model

1. The final ORCHESTRAR task is to analyze the model.

   ➢ At the bottom of the *Analyze Sidechains* dialog, click **Analyze Model**.

   The *Analyze Model* dialog appears (dialog description on page 86).

   The status line indicates the following:

   ```
   231 res, 10 missing res, 0 gaps, 0 missing sidechains
   ```

   Note: 10 residues at the C terminus could not be modeled because ORCHESTRAR requires that a gap have two flanking regions in order for a loop to be added.

2. Perform a few simple checks on the model.

   ➢ Click **Check Backbone Distances** to color in red the residues that have non-ideal peptide bonds and Cα/Cα distances.

   Values and the color legend are output to the textport. A relatively small number of residues have non-ideal peptide bonds or longer than ideal Cα/Cα distances. The values are within ranges that can be quickly corrected by a short minimization.

   ➢ Click **Check Backbone Torsions** to color the residues that have non-ideal phi and psi angles.

   Values and the color legend are output to the textport. A small number of residues are colored red indicating that a phi or psi angle is outside the expected range. You will perform a quick minimization to correct these.

   ➢ Click **Show Conserved Regions**.

Residues in structurally variable regions are colored red.

3. Use ProTable to display the phi/psi dihedral values in a Ramachandran plot.

➢ Click **Analyze Model (ProTable)**.

A ProTable table is opened where each row is a residue in the model.

➢ **MSS: Protein >>> Conformational Analysis**

➢ Click **OK** to accept the defaults:

- The protein model is in **M1**.
- The model's phi-psi plot will be displayed in **D2**.
- The statistical Ramachandran plot will be displayed in **D3**.

➢ Press **OK** when prompted with the information on how to delete the statistical Ramachandran plot (in D3).

Toggle off the display of the model for easier viewing of the two plots.

➢ Click the ☑ icon to open the *Molecule Display Options* tool.

➢ Hide the model temporarily by toggling ***off*** **Mol Vis** for **M1**.

➢ Use the **Reset** icon and click **Everything**.

➢ Scale down the image for comfortable viewing.

For more information about the contents of the table see Protein Backbone Analysis (in the ProTable Manual).

When you are finished viewing the table and the plot, close the table.

➢ **MSS: File >>> Close**

➢ Select **No** when prompted to save the table.

Closing the ProTable also closes the phi-psi plot associated with the model. However, the statistical Ramachandran plot is still displayed. You can hide it and redisplay the model.

➢ Use the ☑ icon to toggle ***on*** the display of the model in **M1**.

➢ Toggle ***off*** the display of the **Independent Background** in **D3**.

➢ Press **Close** to close the tool.

4. Prepare the model for minimization

   ➢ In the *Analyze Model* dialog, make sure that the **Model Style** is set to display the **Whole Protein**.

   Before it can be minimized, the model needs to have hydrogens, force field atoms types, and atomic charges added.

   ➢ Click **Structure Preparation Tool**.

   The *Prepare Protein Structure* dialog appears (dialog description in the Biopolymer Manual)

   ➢ Click **Analyze Selected Structure**.

   The analysis proceeds rapidly and updates the fields in the dialog.

   ➢ Click **Fix** next to **Termini Treatment**.

   ➢ In the *Edit Termini* dialog, change **None** to **Charged** in the New Block option menu, then click **Apply to Selected Protein**.

   This adds termini that are charged to the model.

   ➢ Click **Close**.

   ➢ Click **Add** next to **Add Hydrogens**.

   ➢ In the *Add Hydrogens* dialog, click **OK** to add hydrogens to all applicable atoms.

   ➢ Click **Add** next to **Add Charges**.

   ➢ In the *Load Charges* dialog, press **OK** to add **AMBER7 FF99** charges to the model.

   ➢ Press **Yes** to replace b-factor values with charges. (ORCHESTRAR makes use of the b-factor field in ORCHESTRAR-created PDB files for bookkeeping purposes).

   ➢ Press **Fix** next to **Fix Sidechain Amides**.

   ➢ Press **Close** in the *Prepare Protein Structure* dialog.

5. Minimize the model in stages.

   ➢ In the *Analyze Model* dialog, click **Staged Minimization**.

   ➢ In the *Minimize* dialog that appears:

- Set the **Max Iteration** to `100`. (In your own work, you would increase this to a much larger number.)
- Press the **Energy Setup Modify** button.
- Set the **Force Field** to **AMBER7 FF99**.
- Select the **Charges** to **Use Current**.
- Chang the **Dielectric Constant** to `4`.
- Press **OK**.

➢ Press **OK** in the *Minimize* dialog.

Several minimizations are performed on the model as follows:

1. Hydrogens only
2. Hydrogens + sidechains
3. Hydrogens + sidechains + backbone atoms except C-alpha
4. All atoms

When the minimization completes the *Analyze Model* dialog is posted again.

2.1.9 Save the Model

When the staged minimization is finished, save the model.

➢ At the bottom of the *Analyze Mode*l dialog, click **Save Model**.

➢ In the *Save Model* dialog, enter **`min_model`** and press **Save**.

This saves the model in .mol2 format.

➢ In the *Analyze Model* dialog click **Exit** to close ORCHESTRAR and the *Sequence Viewer*.

A dialog presents the opportunity to retain the model on the SYBYL screen after closing ORCHESTRAR.

➢ Click **Yes** to retain the model in M1.

2.1.10 Compare the Model to the Known Structure

The solved structure from which the query sequence was obtained is of human factor Xa (Stuart-Prower factor), a serine protease that is part of the clotting cascade. It is part of the prothrombinase complex catalyzing the conversion of prothrombin to thrombin.

1. The minimized model is already displayed in M1.
2. Read in the solved structure from which the query sequence was obtained.
   - ➢ **File >>> Retrieve PDB >>> from RCSB**
   - ➢ Type `2BOK` and click **OK**.
   - ➢ In the textport press the keyboard return key at the request for a password.
   - ➢ Enter `Yes` when prompted to read in pdb2bok.ent.
   - ➢ At the request to center the molecule click **NO** then **OK**.
3. Align the structures using homology.
   - ➢ **Biopolymer >>> Compare Structures >>> Align Structure by Homology**
   - ➢ In the *Align Structures by Homology* dialog, specify **m1** as the fixed protein and **m2** as the movable protein.
   - ➢ Change **Calpha** to **Backbone** to use backbone atoms in the fit.
   - ➢ Click **Align**.

   The Weighted Root Mean Square Distance reported in the SYBYL textport is 1.58 Å.

   - ➢ Click **Close**.
4. Compare the structures visually by rendering the model and the crystal structure with ribbons in any two contrasting colors.
   - ➢ **View >>> Biopolymer Display >>> Ribbon/Tube**
   - ➢ Selected **M1**, click **All** then **OK**.
   - ➢ Selected **ORANGE** (or your own color preference) and click **OK**.
   - ➢ **View >>> Biopolymer Display >>> Ribbon/Tube**
   - ➢ Selected **M2**, click **All** then **OK**.
   - ➢ Selected **GREEN** (or your own color preference) and click **OK**.

   Render the Na atom as a sphere.

   - ➢ **View >>> Mixed Rendering**

- Select **M2** and click **Substructures**.
- Click **A/NA1245**, then **OK**, then **OK** again.
- Select **Spacefill** then click **OK**.

Hide the molecules.

- Use the ☑ icon to toggle off the molecules in **M1** and **M2**.

You can easily see the difference in loop conformations around the Na binding site.

When comparing the ORCHESTAR model to the known 3D structure, you will notice that some residues were found in the model that are not in the crystal structure, namely residues ASP70 through GLU80. These residues could not be determined experimentally. ORCHESTRAR modeled them because they are present in the starting PIR file.

Conversely, MET242 and LYS243 at the C-terminus of the model were not modeled by ORCHESTRAR.

Recommended Work Flow to Build Another Model

After constructing the final model with one of the loop solutions for the ASP174:ASP179 gap, you may use the ORCHESTRAR Project Manager to build another model based on another loop solution.

1. In the **Project Manager**, load the project created in this tutorial.
2. Return to **Add/Analyze Loops** and select **model_f4** at the top of the dialog. This model still has 5 gaps. However, all the loop solutions are listed for each gap.
3. Select the **ASP174:ASP179** gap, sort the list by run number, and mark the single solution (1kigh175) produced by the first run.
4. Join All Marked Loops (+). This will create a new model (numbered sequentially) and add it to the list of models available at the top of all the dialogs. Proceed to build the full model. This one does include the loop in a geometry favorable to Na coordination. The RMSD produced by alignment with the crystal structure is 1.53 Å.
5. Use the standard SYBYL tools to compare the two models to each other and to the known structure.

2.2 ORCHESTRAR Unix Commands Tutorial

In this tutorial you will learn how to construct a comparative model for a query sequence using the ORCHESTRAR Unix commands. You will use SYBYL to view the results along the way.

The sequence used in this tutorial is from the solved structure `1HCS.pdb`, chain A which is of the human pp60c-src SH2 domain. The Src homology 2 domain (or SH2 domain) is found in proteins in signal transduction which in turn regulates cell growth and proliferation.

Hardware Note: Results may vary slightly on different hardware platforms because of the handling of floating point numbers and the iterative nature of some of the algorithms (especially CHORAL).

A matter of time: This tutorial requires approximately 50 minutes personal time and 10 minutes CPU time (tested on a Xeon 2.8 GHz Linux machine).

Note: Some of the commands in this tutorial are very long and do not fit on one line. A backslash (\) indicates the line break.

2.2.1 Set Up

1. Create a new directory to hold the files used and generated in this tutorial.

   ➢ Within your working directory, type: **`mkdir tut_com`**

2. You will be running SYBYL and ORCHESTRAR side-by-side in the same directory, but in separate shells.

   ➢ Open a new, separate shell.

3. Change the directory in both shells.

   ➢ In both shells, `cd` to your newly created directory.

   All instructions in this tutorial are given in relationship to this directory.

4. In the shell where you will run SYBYL, start SYBYL then load the staccato.spl script.

   ➢ Start SYBYL.

   ➢ Type: **`take $TA_DEMO/orchestrar/staccato.spl`**

   The script is available as the command `staccato` as long as this SYBYL session is active.

5. In the shell where you will run the ORCHESTRAR commands, set up your SYBYL environment.

   ➢ In the ORCHESTRAR shell, type: **`trigo -shell sybyl7.3`**

   ➢ Type: **`cp $TA_DEMO/orchestrar/unknown.pir .`** (include the space and period at the end)

2.2.2 Find Homologs with FUGUE

1. Run FUGUE to find homologs for the input sequence. For this tutorial, Tripos provides the sequence in the file unknown.pir.

   ➢ **Biopolymer >>> Model Proteins >>> FUGUE >>> Run FUGUE**

   SYBYL opens the *FUGUE* dialog (dialog description in the FUGUE Manual).

   ➢ Set **Input Sequence from** to **PIR File**.

   ➢ Type **`unknown.pir`** in the associated field or use the file browser to retrieve the file.

   ➢ Set **Profiles to Search** to **All Profiles**.

   ➢ Toggle **Run PSI-Blast first** ***on***.

   ➢ In **Jobname**, enter **`fugue_run`**.

   ➢ Activate **Run in Batch** and select the **Machine** on which to run the job.

   ➢ Click **OK**.

Note: Actual performance is based on your particular hardware. On a Xeon 2.8 GHz Linux machine this job takes about 7 minutes.

2. View the alignments.

➢ **Biopolymer >>> Model Proteins >>> FUGUE >>> View FUGUE Results**

SYBYL displays the *FUGUE Results* dialog.

➢ Next to **Run Directory** click **[...]** to access a file browser.

➢ Select **fugue_run** and click **OK**.

The sh2 (Src homology 2 domain) family of homologs is the top result and will be used in this tutorial.

➢ Highlight the **sh2** family and click **View Alignment**.

In a browser window, SYBYL displays the alignment for this family and for the `unknown.pir`. The known sequences are colored with JOY sequence annotation. See JOY Alignment Key for HTML Format (in the Biopolymer Manual).

➢ Click **OK** to close the *FUGUE Results* dialog.

2.2.3 Produce Multiple Sequence Alignment with BATON

1. When the FUGUE run has finished, return to the shell where the SYBYL environment was set up. Copy the structure files from the sh2 family to your working directory

➢ In the external shell, type:

➢ **`cp $HOMSTRAD/sh2/*.atm .`** (include the space and period at the end)

The copied files contain atom type and coordinate information in a format similar to the PDB format:

1ab2.atm, 1aouf.atm, 1ayd.atm, 1bfj.atm, 1csza.atm, 1lkka.atm, 1shba.atm, 1zfpe.atm, 2plda.atm, 2pnb.atm, 3hck.atm

2. Use BATON to structurally align the homologs.

➢ In the external shell, type[1]:

➢
```
$TA_ORCHESTRAR_BIN/Baton -pdbout 1ab2.atm \
1aouf.atm 1ayd.atm 1bfj.atm 1csza.atm 1lkka.atm \
1shba.atm 1zfpe.atm 2plda.atm 2pnb.atm 3hck.atm
```

For more information about the command syntax, see BATON on page 96.

3. Use FUGUE to recreate the sequence alignment of the family without 3icd. A sequence re-alignment is recommended whenever a homolog has been added or removed.

➢
```
$TA_ORCHESTRAR_BIN/joy baton.ali
```

➢
```
$TA_ORCHESTRAR_BIN/melody -t baton.tem
```

➢
```
$TA_ORCHESTRAR_BIN/fugueseq -seq unknown.pir \
-prf baton.fug -joy -toprank 1 -top 1 -A 9 -detail
```

2.2.4 Determine Structurally Conserved Regions with CHORAL

1. Use CHORAL to determine the SCRs. CHORAL uses the .brk files from the BATON run and the alnpap.ali file from the FUGUE run to generate the SCRs.

➢
```
$TA_ORCHESTRAR_BIN/choral -a unknown_baton_ma.ali -b
```

For more information about the command syntax, see CHORAL on page 98.

By default, CHORAL produces 5 solutions stored in the files CORE0*x*.pdb, where *x* ranges from 1 to 5.

2. View the model for CHORAL's top solution in SYBYL.

➢ **File >>> Read**

➢ Select **CORE01.pdb** and click **OK**.

➢ Select **CENTER_VIEW** and click **OK**.

3. Visually inspect the SCR model.

➢ In the SYBYL textport, type: `staccato m1`

The `staccato` script:

- Colors the protein blue.

[1] Some of the commands are very long and do not fit on a single line. The backslash (\) character indicates a line break in the instruction and can be used as a continutation character when typing the command at the keyboard.

- Defines gaps by determining which residues are missing. Residues on either side of a gap are labeled and colored red.
- Looks for non-ideal peptide bond lengths. Those which are more than 20% away from the ideal length of 1.33 Å are reported in the textport.
- Looks for Cα-Cα distances longer than 4.3 Å. Residues are colored yellow and labeled if they have Cα-Cα distances greater than this value. This information is also reported in the textport.

➢ Note the residues colored red on the SYBYL screen.

Results may vary slightly on different hardware platforms because of the handling of floating point numbers and the iterative nature of some of the algorithms.

On ***Linux***, the following output is seen in the SYBYL textport:

```
6  RESIDUE GAP ---------------------------> TRP9 and ARG16
4  RESIDUE GAP ---------------------------> LEU24 and PRO29
4  RESIDUE GAP ---------------------------> THR40 and TYR45
3  RESIDUE GAP ---------------------------> LEU98 and LEU102
```

On ***SGI***, the following output is seen in the SYBYL textport:

```
6  RESIDUE GAP ---------------------------> TRP9 and ARG16
4  RESIDUE GAP ---------------------------> LEU24 and PRO29
7  RESIDUE GAP ---------------------------> THR40 and SER48
3  RESIDUE GAP ---------------------------> LEU98 and LEU102
```

The `staccato` script, which requires the presence of both residues flanking a gap, detected four gaps.

2.2.5 Fill Small Gaps with BRIDGE

You can use BRIDGE to fill small (1-3 residue) gaps. In this tutorial, staccato revealed four gaps in CHORAL’s top solution (`CORE01.pdb`).

1. It is good practice in your own work to run BRIDGE on the model produced by CHORAL.

   ➢
   ```
   $TA_ORCHESTRAR_BIN/bridge -pdb CORE01.pdb \
   -seq choral-bridge.pir
   ```

BRIDGE was able to find and fill the gap between LEU98 and LEU102.

For more information about the command syntax, see BRIDGE on page 99.

Note: Because BRIDGE works only on true gaps (missing residues) in a sequence, it ignores residue pairs that are adjacent, but separated by a larger than ideal distance. You can create a gap by deleting two such residues, so that BRIDGE can attempt to fill the gap.

2. Examine the BRIDGE results in SYBYL.

   ➢ **File >>> Read**

   ➢ Select `bridge.pdb` and click **OK**.

   ➢ Select **CENTER_VIEW** and click **OK**.

   By overlaying the model generated by CHORAL (`CORE01.pdb`) and the model generated by BRIDGE (`bridge.pdb`), you can see the loop that fills the LEU98:LEU102 gap.

3. Analyze the gaps in `bridge.pdb`, the new model.

   ➢ In the SYBYL textport, type: **`staccato m2`**

   On ***Linux***, the following output is seen in the SYBYL textport:

```
6  RESIDUE GAP ---------------------------> TRP9 and ARG16
4  RESIDUE GAP ---------------------------> LEU24 and PRO29
4  RESIDUE GAP ---------------------------> THR40 and TYR45
```

   On ***SGI***, the following output is seen in the SYBYL textport:

```
6  RESIDUE GAP ---------------------------> TRP9 and ARG16
4  RESIDUE GAP ---------------------------> LEU24 and PRO29
7  RESIDUE GAP ---------------------------> THR40 and TYR48
```

   For subsequent tasks, you will build on the model contained in `bridge.pdb`.

2.2.6 Find Loop Solutions with FREAD, PETRA, and CODA

CODA creates consensus loop solutions that are derived from the output of two programs: FREAD and PETRA. FREAD is a gap filling algorithm that uses a knowledge-based approach to find solutions in a database of loop structures. PETRA constructs the region from a database of ab initio fragments. The executable name for PETRA is MINUS.

1. Search for loops to fill the TRP9-ARG16 gap.

   Run FREAD and examine the solutions.

   ➢
```
$TA_ORCHESTRAR_BIN/FREAD3 -pdb bridge.pdb -seq \
unknown.pir -len 6 -startres 10 -o FREAD9.HITS
```

   ➢
```
more FREAD9.HITS
```

   Loop `hs1gxja522.pdb` has the lowest Ek value (energy of the predicted loop in the target structure).

   Run MINUS and examine the solutions.

```
➢ $TA_ORCHESTRAR_BIN/MINUS2 -pdb bridge.pdb \
  -seq unknown.pir -len 6 -startres 10 -read7 \
  -o MINUS9.HITS
```

```
➢ more MINUS9.HITS
```

The top solution is 03605407.pdb.

Run CODA and examine the solutions.

```
➢ $TA_ORCHESTRAR_BIN/CODA3 -ff FREAD9.HITS -mf
  MINUS9.HITS -len 6 -o CODA9.HITS
```

```
➢ more CODA9.HITS
```

The solution with the lowest Ek value is 42236710.pdb. This loop will be used in subsequent modeling.

2. Search for loops for the LEU24:PRO29 gap.

   Run FREAD and examine the solutions.

```
➢ $TA_ORCHESTRAR_BIN/FREAD3 -pdb bridge.pdb \
  -seq unknown.pir -len 4 -startres 25 -o FREAD24.HITS
```

```
➢ more FREAD24.HITS
```

The solution with the lowest Ek value is 1qba677.pdb.

Run MINUS and examine the solutions.

```
➢ $TA_ORCHESTRAR_BIN/MINUS2 -pdb bridge.pdb \
  -seq unknown.pir -len 4 -startres 25 -read7 \
  -o MINUS24.HITS
```

```
➢ more MINUS24.HITS
```

The top solution is 245742.pdb.

Run CODA and examine the solutions.

```
➢ $TA_ORCHESTRAR_BIN/CODA3 -ff FREAD24.HITS -mf
  MINUS24.HITS -len 4 -o CODA24.HITS
```

```
➢ more CODA24.HITS
```

The solution with the lowest Ek value is 1qba677.pdb. This loop will be used in subsequent modeling.

3. Search for loops to fill the third gap. Because this gap was found to have a different length on Linux and on SGI, the instructions below are platform-dependent.

On Linux:

Find a loop solution for the THR40:TYR45 gap.

Run FREAD and examine the solutions.

```
$TA_ORCHESTRAR_BIN/FREAD3 -pdb bridge.pdb -seq
unknown.pir -len 4 -startres 41 -o FREAD40.HITS
```

```
more FREAD40.HITS
```

The solution with the lowest Ek value is 1shba37.pdb.

Run MINUS and examine the solutions.

```
$TA_ORCHESTRAR_BIN/MINUS2 -pdb bridge.pdb -seq
unknown.pir -len 4 -startres 41 -read7 -o MINUS40.HITS
```

```
more MINUS40.HITS
```

The top solution is 700573.pdb.

Run CODA and examine the solutions.

```
$TA_ORCHESTRAR_BIN/CODA3 -ff FREAD40.HITS -mf
MINUS40.HITS -len 4 -o CODA40.HITS
```

```
more CODA40.HITS
```

The solution with the lowest Ek value is 1shba37.pdb. This loop will be used in subsequent modeling.

On SGI:

Fin a loop solution for the THR40:TYR48 gap.

Run FREAD and examine the solutions.

```
$TA_ORCHESTRAR_BIN/FREAD3 -pdb bridge.pdb -seq
unknown.pir -len 7 -startres 41 -o FREAD40.HITS
```

```
more FREAD40.HITS
```

The solution with the lowest Ek value is hs1zz1a128.pdb.

MINUS can quickly find solutions for gaps up to 6 residues. However, it takes a very long time on longer gaps. For that reason, the loop found by FREAD will be used in this tutorial.

2.2.7 Meld Loops to the Model with TUNER

Use TUNER to meld loop solutions, one at a time, to the model. Before running TUNER, you must concatenate each loop `.pdb` file with the model `.pdb` file.

1. Meld the `42236710.pdb` loop into the TRP9-ARG16 gap.

➢
```
cat 42236710.pdb bridge.pdb | \
$TA_ORCHESTRAR_BIN/pretuner > combined1.pdb
```

➢
```
$TA_ORCHESTRAR_BIN/tuner -pdb combined1.pdb \
-out tunerout1.pdb -meld -dino
```

For more information about the command syntax, see TUNER

2. Meld the `1qba677.pdb` loop into the LEU24:PRO29 gap.

➢
```
cat 1qba677.pdb tunerout1.pdb | \
$TA_ORCHESTRAR_BIN/pretuner > combined2.pdb
```

➢
```
$TA_ORCHESTRAR_BIN/tuner -pdb combined2.pdb \
-out tunerout2.pdb -meld -dino
```

3. Meld the last loop into the third gap. Because this gap was found to have a different length on Linux and on SGI, the instructions below are platform-dependent.

On Linux:

Meld the `1shba37.pdb` loop into the THR40:TYR45 gap.

➢
```
cat 1shba37.pdb tunerout2.pdb | \
$TA_ORCHESTRAR_BIN/pretuner > combined3.pdb
```

➢
```
$TA_ORCHESTRAR_BIN/tuner -pdb combined3.pdb \
-out tunerout3.pdb -meld -dino
```

On SGI:

Meld the `hs1zz1a128.pdb` loop into the THR40:TYR48 gap.

➢
```
cat hs1zz1a128.pdb tunerout2.pdb | \
$TA_ORCHESTRAR_BIN/pretuner > combined3.pdb
```

➢
```
$TA_ORCHESTRAR_BIN/tuner -pdb combined3.pdb \
-out tunerout3.pdb -meld -dino
```

4. After all of the loops have been added, read in tunerout3.pdb to see how the three loops were added.

   ➢ **Build/Edit >> Zap >> All**

   ➢ **File >>> Read**

   ➢ Select **tunerout3.pdb** and click **OK**.

   ➢ Select **CENTER_VIEW** and click **OK**.

2.2.8 Add Sidechains with ANDANTE

Use ANDANTE in the borrowing mode to add sidechains.

When adding sidechains to the target sequence, ANDANTE first borrows sidechain conformations, then uses a rotamer library to add the remaining sidechains.

1. Run JOY on the sequence alignment file to create a file that is used by ANDANTE.

   ➢
   ```
   $TA_ORCHESTRAR_BIN/joy unknown_baton_ma.ali
   ```

2. Run ANDANTE.

   ➢
   ```
   $TA_ORCHESTRAR_BIN/andante -i unknown_baton_ma \
   -cm tunerout3.pdb -chi1 -chi12 -minpid 40 -minb 40 \
   -ccc 7.0 -cbb 10.0 -o final_out
   ```

   ANDANTE exits successfully and produces an all-atom model.

3. Copy the file to give it a .pdb extension for easy handling in SYBYL.

   ➢
   ```
   cp final_out.atm final_out.pdb
   ```

4. View the all-atom model in SYBYL.

   ➢ **File >>> Read**

   ➢ Select **final_out.pdb**, **m1** and click **OK**.

   ➢ Select **CENTER_VIEW** and click **OK**.

2.2.9 Align the Model to the Solved Structure

The sequence used in this tutorial is from the solved structure `1HCS.pdb`, chain A which is of the human pp60c-src SH2 domain. The Src homology 2 domain (or SH2 domain) is found in proteins in signal transduction which in turn regulates cell growth and proliferation.

1. Retrieve this structure from the RCSB depository.

   ➢ **File >>> Retrieve PDB >>> from RCSB**

   ➢ In PDB code enter **1HCS** and click **OK**.

   ➢ In the textport press the keyboard return key at the request for a password.

   ➢ Click **Yes**, to read in pdb1hcs.ent.

   ➢ At the request to center the molecule click **NO** then **OK**.

   The solved structure is displayed in M2.

2. Align the solved structure to the model.

   ➢ **Biopolymer >>> Compare Structures >>> Align Structures by Homology**

   ➢ In the *Align Structures by Homology* dialog, specify **m1** as the fixed protein and **m2** as the movable protein.

   ➢ Change **Calpha** to **Backbone** to use backbone atoms in the fit.

   ➢ Click **Align**.

   Note in the textport that this alignment yields a weighted RMSD of 1.03 Å on Linux and 1.19 on SGI.

   ➢ Click **Close**.

3. You may now use SYBYL’s functionality to color and compare the model and the solved structure.

   This concludes this tutorial.

2.3 ORCHESTRAR Hints and Tips

ORCHESTRAR Steps

Import from MSA

- In order to have homolog structure files be automatically associated with their sequences in a sequence alignment file, use the same name for the homolog structures and sequences.
- See the *Import Alignment File* dialog description on page 46.

Model SCRs

- Spend some time ensuring that the sequences are aligned well. Drag the right mouse button in the sequence viewer ruler bar to see how the ‘equivalent’ residues align structurally. Use the sequence color schemes (such as RMSD) to identify poorly aligned regions.
- Don’t feel that you must use all available homologs. If the homologs appear to cluster into different classes, try building from those in each class rather than from all at once.
- Even after deleting undesirable homologs, the alignment may show gaps from some of the deleted homologs. You can remove these unnecessary gaps by using the **Align by Structure** button. This will perform a structural alignment of the remaining homologs, then give you the opportunity to update the sequence alignment based on the structural alignment.
- If you have imported your perfect alignment from other software, use the **Align by Sequence** button to align the structures using this sequence.
- Use **Advanced Details** and tweak the **Minimum Score** to a lower value (-4 or even -10) to try and build more of the protein as SCR and shorten the gap lengths. The quality of the SCR model will decrease accordingly though.
- See the *Model Conserved Regions* dialog description on page 49.

Analyze SCRs

- Use **Check distances** and watch out for orange, which highlights disconnected residues or connected non-sequential residues. Trim the region with **Remove Residues** and model the gap as a loop.
- See the *Analyze Conserved Regions* dialog description on page 56.

Search Loops

- Loops longer than 12 residues are difficult to model. You may get solutions, but there will be more uncertainty with these. To reduce gap length try to increase the model's SCR.
- Ab initio searching can be exponentially proportionate to the length of the loop to be searched. You may notice it with loops >6 in length.
- For difficult loops, set up multiple searches and change the anchor residue positions to look for different potential solutions.
- Install the FRAGDB and PETRA databases on a local drive to speed up the database loop search. Adapt the following instructions to suit your situation:
    - `mkdir local_Coda`
    - `cd local_Coda`
    - Set your SYBYL environment: `trigo -shell sybyl`*x.x*
    - `cp -R $CODA .` (include the period)
    - Before starting SYBYL, set the environment variable `CODA` to point to your local_Coda directory. You may want to include a line to this effect in your .sybylrc file so that it is done automatically every time you start SYBYL.
- See the *Search Loops* dialog description on page 62.

Add/Analyze loops

- Loop solutions with high ESST scores, low Ek values and low anchor region RMSD values are best.
- Tweak the **Advanced Details** parameters (by increasing the number of anchor residues) to vary how much the SCR region is allowed to change to accommodate the loop. This may affect the loop geometry.
- By default, the loop with the best Ek value is chosen for each gap. This makes it easy to **Join All Marked Loops** and proceed to the next step.
- To the *Analyze Conserved Regions* dialog description.

Model Sidechains

- To save time, try to use the borrowing mode. ARG and LYS increase search time unless their rotamers are restricted.
- See the *Model Sidechains* dialog description on page 69.

Analyze Model

- Use the various **Check** buttons to highlight problems. If necessary, go back to **Analyze SCRs** (via the Project Manager), remove the problem residues, and model the new gaps as loops.
- Use the **Structure Preparation Tool** to add hydrogens and charges. You can also check for sidechain clashes here and focus on individual rotamers.
- Use **Staged Minimization** for a quick refinement of the model.
- Use the **Analyze Model (ProTable)** button to launch a spreadsheet for detailed analysis. The Protein menu provides access to many analysis functions for highlighting incorrect chiralities, bad bond lengths and angles, etc.
- See the *Analyze Model* dialog description on page 86.

ORCHESTRAR Handling of Special Features

Ligands and Cofactors

HETATM records of PDB files are currently not supported through the SYBYL interface to ORCHESTRAR. Here is a work-around:

1. Save the model and exit the ORCHESTRAR interface.
2. Read your aligned homolog with cofactor/ligand into SYBYL. Extract the cofactor/ligand. Verify that the atom types are correct. Merge the cofactor/ligand into the model. Save the combine structure in the Mol2 or PDB format.
3. Reopen the ORCHESTRAR interface and load your project. At various stages you can ‘load custom model’, so do this and pick your modified file. Now you can model loops or sidechains including the cofactor or ligand. ***Note:*** Sidechain modeling deals explicitly with the cofactor or ligand. Loop modeling will allow you to assess to the quality of each loop and select the best solution.

Multiple Chain Proteins

Currently these are not supported in HOMSTRAD and, therefore, in FUGUE results. If you use the ‘Import MSA’ workflow, you can use multiple chain proteins as long as the sequence file matches every residue in the structure file.

This page intentionally blank.

Chapter 3.

Run ORCHESTRAR via the Menubar

Biopolymer >>> Model Proteins >>> ORCHESTRAR

Central ORCHESTRAR hub:

- ORCHESTRAR Project Manager on page 42
    - Create an ORCHESTRAR Project

ORCHESTRAR Tasks

- Model Conserved Regions on page 49
- Analyze Conserved Regions on page 56
- Search Loops on page 62
- Add/Analyze Loops on page 69
- Model Sidechains on page 77
- Analyze Sidechains on page 82
- Analyze Model on page 86
- Save the Model on page 90

Frequently Used Elements in ORCHESTRAR Dialogs on page 91

- Selection in ORCHESTRAR Lists on page 91
- ORCHESTRAR Color and Display Styles on page 92

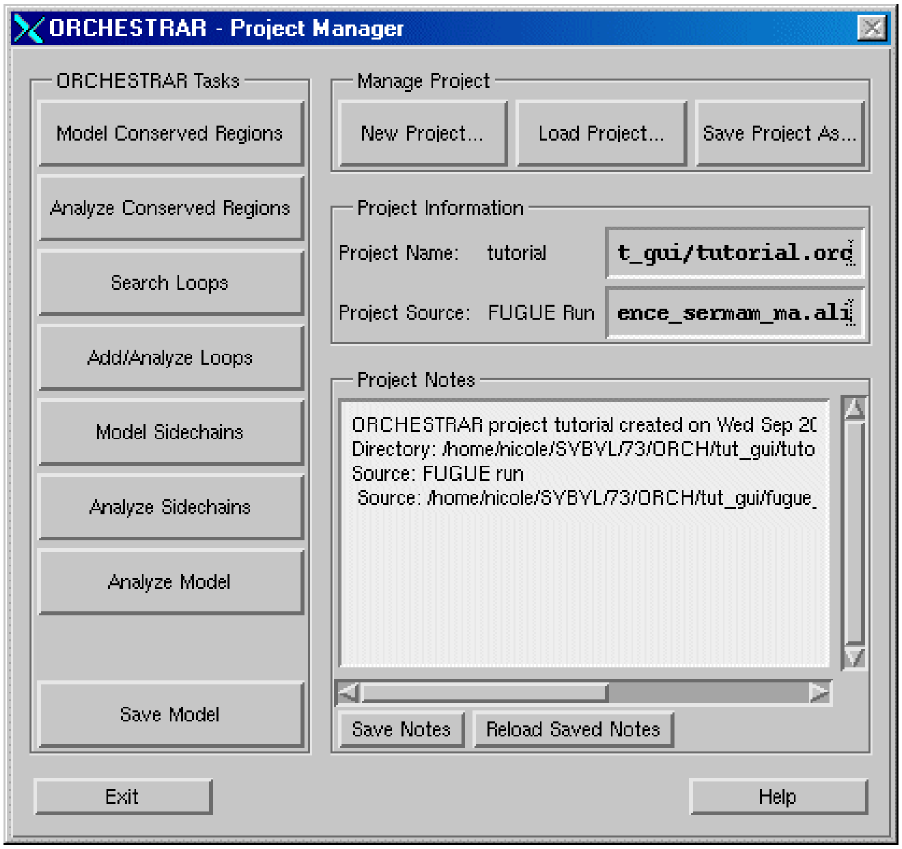
See also in the Biopolymer Manual:

- Sequence Viewer

3.1 ORCHESTRAR Project Manager

This central dialog provides access to all aspects of ORCHESTRAR project management and functionality.

Access: **Biopolymer >>> Model Proteins >>> ORCHESTRAR**



Project Name: tutorial t_gui/tutorial.orc

Project Source: FUGUE Run ence_sermam_ma.ali

ORCHESTRAR project tutorial created on Wed Sep 20
Directory: /home/nicole/SYBYL/73/ORCH/tut_gui/tuto
Source: FUGUE run
 Source: /home/nicole/SYBYL/73/ORCH/tut_gui/fugue_

Manage Project

An ORCHESTRAR project consists of a directory with the `.orc` extension. All the files associated with the project or generated while the project is loaded will be stored in that directory.

New Project	Access the *Select New Project Type* dialog and specify whether to use the results of a FUGUE run or a multiple sequence alignment (MSA) file.
Load Project	Access a file browser to select an existing ORCHESTRAR project directory (`.orc`).

Save Project As	Save the project in a different ORCHESTRAR project directory. You will be prompted for a new project name that is unique within the parent directory. The extension `.orc` will be added automatically if it is not included in the project name. This button is enabled if a project has been created or loaded.
Project Information	Information about the current project consists of: • The full path to the project's name. • The type of source or starting point for the project and the full path to the source file.
Project Notes	As ORCHESTRAR operations are performed, notes added automatically to this text area. You may also add your own notes by clicking within the Project Notes area and typing.
Save Notes	Save the ORCHESTRAR project notes to a text file (`notes.txt`) within the project directory. This button becomes active once you have added your own notes in the text window.
Reload Saved Notes	Returns the notes to the state they were in before the last time the Save Notes button was pressed.

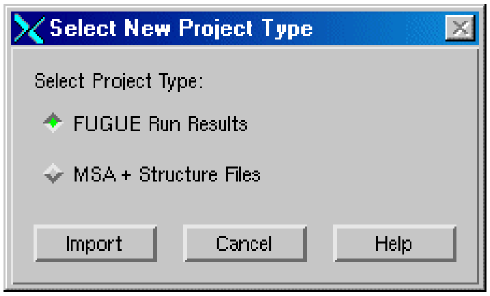
ORCHESTRAR Tasks

Model Conserved Regions	Access the *Model Conserved Regions* dialog (page 49). This button is enabled if a multiple structure alignment or the results from a FUGUE run have been loaded.
Analyze Conserved Regions	Access the *Analyze Conserved Regions* dialog (page 56). This button is enabled if an SCR model exists.
Search Loops	Access the *Search Loops* dialog (page 62). This button is enabled if an SCR model or a protein template exists.
Add/Analyze Loops	Access the *Add/Analyze Loops* dialog (page 69). This button is enabled if a loop model exists.
Model Sidechains	Access the *Model Sidechains* dialog (page 77). This button is enabled if an SCR model or a protein template exists.
Analyze Sidechains	Access the *Analyze Sidechains* dialog (page 82). This button is enabled if a sidechain model exists.

Analyze Model	Access the *Analyze Model* dialog (page 86). This button is enabled if an SCR model or a protein template exists.
Save Model	Access the *Save/Export Model Structure* dialog where you can select among the project's models which one to save using SYBYL's *Save Molecule* dialog.

3.2 Create an ORCHESTRAR Project

In the *ORCHESTRAR Project Manager*, press **New Project**.



Select Project Type:

● FUGUE Run Results

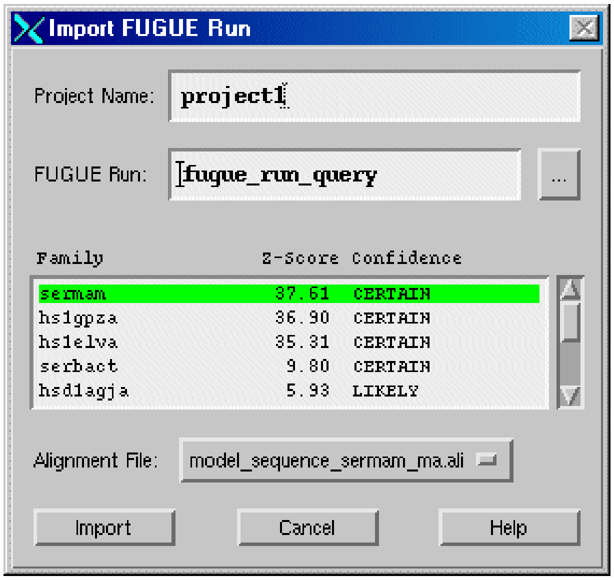
○ MSA + Structure Files

An ORCHESTRAR project may be started from:

FUGUE Run Results	Access the *Import FUGUE Run* dialog and specify the directory containing the results of a FUGUE run.
MSA + Structure Files	Access the *Import Alignment File* dialog and read in a custom multiple-sequence alignment file along with the homologs corresponding to the sequences in the file.

3.2.1 Create a Project from a FUGUE Run

In the *ORCHESTRAR Project Manager*, press **New Project**.



Project Name: project1

FUGUE Run: fugue_run_query

```
Family              Z-Score  Confidence
sermam                37.61  CERTAIN
hs1gpza               36.90  CERTAIN
hs1elva               35.31  CERTAIN
serbact                9.80  CERTAIN
hsd1agja               5.93  LIKELY
```

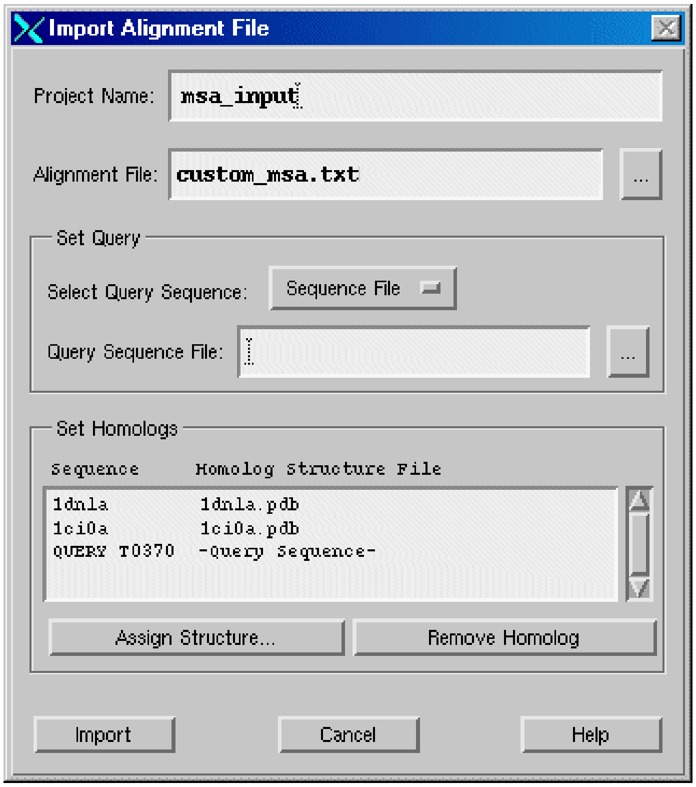
Alignment File: model_sequence_sermam_ma.ali

Project Name	Name of the new ORCHESTRAR project. All the files related to this project will be stored in a directory with the `.orc` extension. Project names must be unique within the parent directory.
FUGUE Run	Enter the name of a FUGUE run directory or use the file browser to retrieve it.
List	List of HOMSTRAD families from the FUGUE run, sorted by decreasing Z-Score values. • **Family**—Names of the HOMSTRAD families • **Z-Score**—An indication of homology recognition. Sequence divergence for an alignment is calculated as the average distance between all pairs of sequences. • **Confidence**—An indication of confidence in the alignment quality. CERTAIN = 99% for Z-Score ≥ 6.0 LIKELY = 95% for Z-Score ≥ 4.0 MARGINAL = 90% for Z-Score ≥ 3.5 GUESS = 50% for Z-Score ≥ 2.0 UNCERTAIN for Z-Score < 2.0
Alignment File	Select the MA (master sequence against all structures) or the AA (all sequences against all structures) alignment file.
Import	Select a single family in the list before pressing this button. All the homologs in that HOMSTRAD family will be imported.

3.2.2 Create a Project from an MSA File

Read in a custom multiple-sequence alignment file along with the homologs corresponding to the sequences in the file.

In the *ORCHESTRAR Project Manager*, press **New Project**.



Import Alignment File

Project Name: msa_input

Alignment File: custom_msa.txt

Set Query

Select Query Sequence: Sequence File

Query Sequence File:

Set Homologs

Sequence	Homolog Structure File
1dnla	1dnla.pdb
1ci0a	1ci0a.pdb
QUERY T0370	-Query Sequence-

Project Name	Name of the new ORCHESTRAR project. All the files related to this project will be stored in a directory with the `.orc` extension. Project names must be unique within the parent directory.
Alignment File	A multiple sequence alignment (MSA) file in any of the following formats: • PIR format of the National Biomedical Research Foundation (http://www.psc.edu/general/software/packages/seq-intro/seq-file.html) • FASTA (http://www.compbio.ox.ac.uk/faq/format_examples.shtml#fasta) • MSF = Multiple Sequence Format (http://www.compbio.ox.ac.uk/faq/format_examples.shtml#msf) • CLUSTALW (http://www.ebi.ac.uk/clustalw/)

Set Query

Select Query Sequence	Identify the query sequence: • **Sequence File**—Activate the text box and file browser below to retrieve the query sequence if it is not in the alignment file. • Sequence name—Select among the sequences in the alignment the one to use as the query.
Query Sequence File	Enter the name of the file containing the query sequence or use the adjacent file browser to retrieve it. All instances of the character “X” in the query sequence will be replaced with the character “-” in the ORCHESTAR graphical interface.

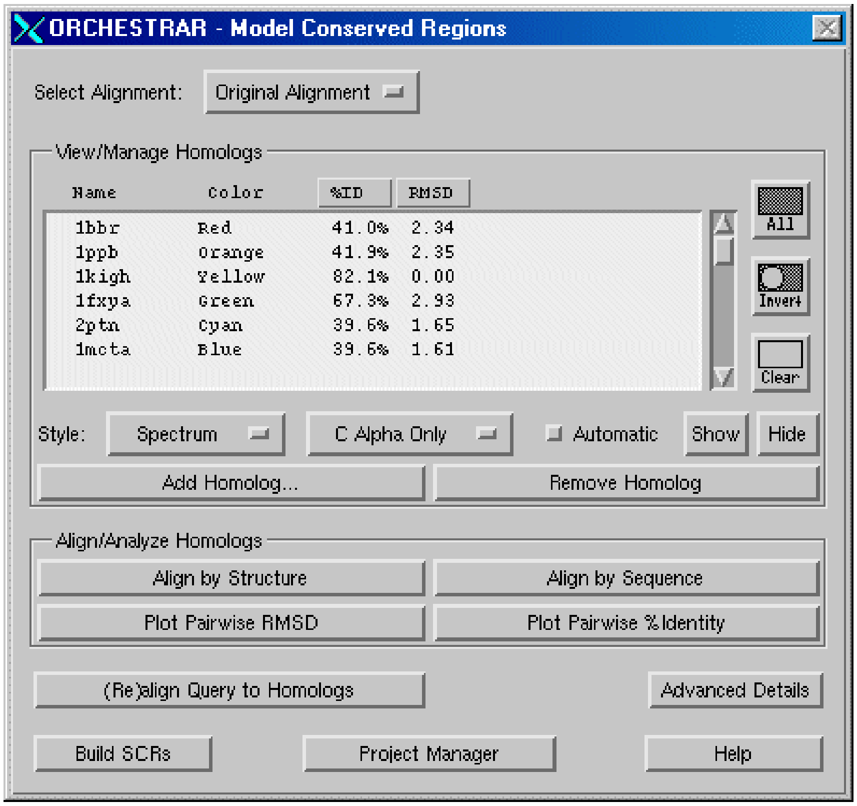
Set Homologs

Homolog List	Information about the homologs in the alignment: • **Sequence**—Code name of each sequence in the alignment file. • **Homolog Structure File**—Name and location of the PDB file containing the 3D structure of each homolog. Homologs are looked for in the current working directory then in the directory containing the MSA file. If the structural files cannot be found automatically, a file browser is posted to allow user intervention. The code -Missing- next to a sequence name indicates that no structure file could be found for that sequence. The code -Query Sequence- identifies the sequence to be modeled.
Assign Structure	Access a file browser to retrieve a PDB or Mol2 file containing the structure of the selected homolog. A warning will be issued if the structure is not compatible with the homolog’s sequence.
Remove Homolog	Remove the selected homolog(s) from the list.

3.3 Model Conserved Regions

To explore and improve the structural alignment of the homologs that will be used to build the structurally conserved regions.

In the *ORCHESTRAR Project Manager*, click **Model Conserved Regions**.



ORCHESTRAR - Model Conserved Regions

Select Alignment: Original Alignment

View/Manage Homologs

Name	Color	%ID	RMSD
1bbr	Red	41.0%	2.34
1ppb	Orange	41.9%	2.35
1kigh	Yellow	82.1%	0.00
1fxya	Green	67.3%	2.93
2ptn	Cyan	39.6%	1.65
1mcta	Blue	39.6%	1.61

Style: Spectrum   C Alpha Only   □ Automatic

The *Sequence Viewer* is also posted (dialog description in the Biopolymer Manual).

Select Alignment	The alignment associated with each SCR model is retained and inherited by subsequent model in the ORCHESTAR work-flow. An asterisk after the name of any alignment indicates that it was changed after it was last saved. • **Original Alignment**—Alignment as read in at the start of the project. You may recall this alignment later in the project to regain homologs that were previously deleted. • **Alignment N**—New alignments are created during the Build SCRs operation and may also be generated manually within the *Sequence Viewer*.

View/Manage Homologs

The initial list of homologs consists of:

- If the project is based on a FUGUE run: all the homologs in the HOMSTRAD family. The query sequence is in the FUGUE run's `sequences` subdirectory.
- If the project is based on a user-selected MSA: all the homologs referenced in the MSA file.

Note: Before being placed in the list, homologs are checked for structure-sequence compatibility. Homologs failing this test are not added to the list nor to the *Sequence Viewer* and are reported in the textport.

Homolog List	The list (which is synchronized with the *Sequence Viewer*) provides the following information for each homolog: • **Name**—Name of the file containing it (without file extension). Only the first 10 characters of homolog names are shown in the list. • **Color**—Name of a solid color, or ByAtomType, or -Hidden- (if the homolog is not currently visible). • **%ID**—%identity of the homolog to the query (target) sequence. Click **%ID** above the list to sort it by increasing or decreasing values in this column. • **RMSD**—Root mean squared distance (based on C-alpha) to the homolog with the largest % identity to the query sequence. Click **RMSD** above the list to sort it by increasing or decreasing values in this column. To increase the size of the list, stretch the dialog vertically.
Homolog Selection	See Selection in ORCHESTRAR Lists on page 91 for a full description. Homologs selected in this list are also bold in the left panel of the *Sequence Viewer*.
Display Style	See ORCHESTRAR Color and Display Styles on page 92 for a full description.

Add Homolog	Access a file browser to retrieve an additional homolog in PDB format and add it to the list and to the *Sequence Viewer*. The project notes list each added homolog. Adding a homolog does not adjust the sequence and structural alignments.
Remove Homolog	Select one or more homologs in the list then press this button to remove them from the list and from the *Sequence Viewer*. A dialog will prompt you to confirm the permanent removal of the selected homologs. The project notes list each deleted homolog. Deleting a homolog does not adjust the sequence and structural alignments.

Align/Analyze Homologs

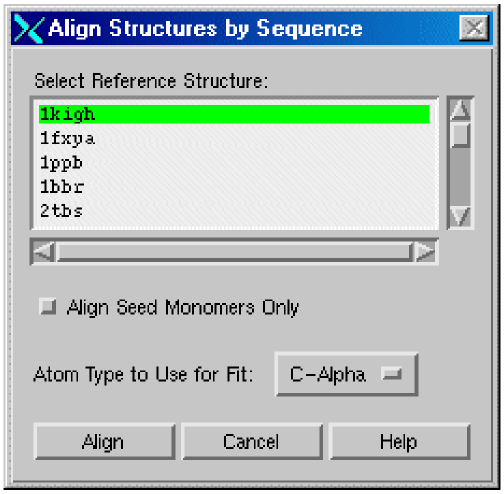
Align by Structure	The selected homologs (or all if none are selected) are structurally aligned using BATON. The project notes list the align homologs and their RMSD value. Upon completion of the structural alignment you will be prompted in a dialog whether to use the new sequence alignment.
Align by Sequence	Access the *Align Structures by Sequence* dialog to align the structures of selected homologs by sequence homology.
Plot Pairwise RMSD	Display, in a separate dialog, the pairwise Root Mean Square Deviation (in Å) for the C-alphas of the selected homologs. If nothing is selected in the list, all homologs are used. The numerical representation of the same grid is printed in the textport.
Plot Pairwise %Identity	Display, in a separate dialog, in a colorized grid the pairwise % identity, based on the C-alphas, of the selected homologs and the query. If nothing is selected in the list, all homologs are used. The numerical representation of the same grid is printed in the textport.
(Re)align Query to Homologs	Create a sequence alignment from the selected homologs. If none are selected, all are used. This operation will overwrite the MSA.

Action Buttons

(Re)align Query to Homologs	Create a sequence alignment from the structural alignment of the selected homologs. If none are selected, all are used.
Advanced Details	Access the *Model SCR Details* dialog where you can adjust the parameters used to build the SCRs.
Build SCRs	Determine the structurally conserved regions using `CHORAL` with the current sequence alignment. If no homologs are selected in the list, they are all used. If only some of the homologs are selected you will be prompted to confirm the partial selection. If `CHORAL` is unable to generate a good quality SCR model based on the scoring of environment-specific substitution tables, a dialog appears reporting the number of homologs with poor scores. The recommendation is to remove lower scoring homologs (flagged with a “?” in the textport and in the dialog). It is possible to force the building of the SCRs with the existing homologs within the warning dialog or via `TAILOR!MODEL_SCR!FORCE_BUILD YES`. Upon successful completion, the *Analyze Conserved Regions* dialog will be posted.
Project Manager	Close this dialog and return to the *ORCHESTRAR Project Manager*.

3.3.1 Align by Sequence

Align the selected homologs on the basis of their sequence similarity to one of them used as reference.



Select Reference Structure:

1kigh
1fxpa
1ppb
1bbr
2tbs

□ Align Seed Monomers Only

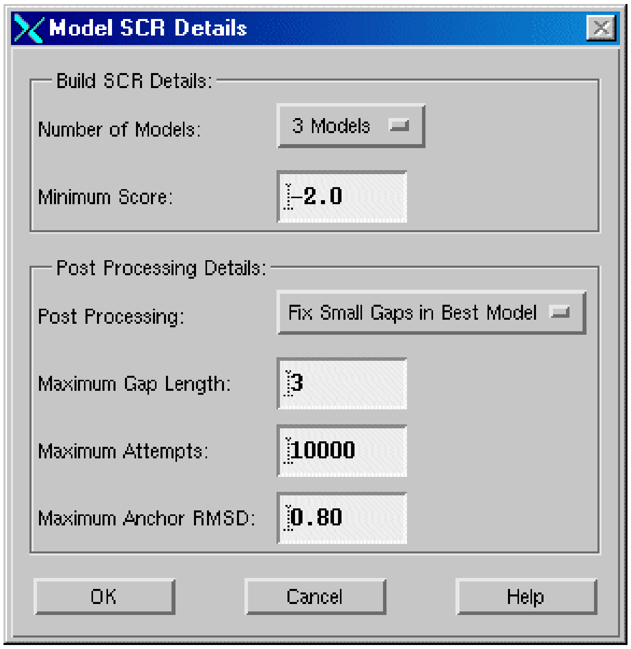
Atom Type to Use for Fit: C-Alpha

Select Reference Structure	Designate the homolog to which the others will be aligned.
Align Seed Monomers Only	If ***on***, perform a structural fit using only the atoms in monomers which are identical across the sequence alignment. If ***off***, the atoms of all aligned monomers are used.
Atoms to Use for Fit	Specify the atoms to be used for the least-squares fit: • **C-Alpha** • **Backbone** • **Sidechain** • **All**

3.3.2 Model SCR Details

Adjust the parameters used to build the structurally conserved regions.

In the *Model Conserved Regions* dialog, click **Advanced Details**.



Model SCR Details

Build SCR Details:

Number of Models: 3 Models

Minimum Score: -2.0

Post Processing Details:

Post Processing: Fix Small Gaps in Best Model

Maximum Gap Length: 3

Maximum Attempts: 10000

Maximum Anchor RMSD: 0.80

Build SCR Details

Number of Models	The number of model to be generated from the selected homologs: 1, 2, 3, 5, or 10. The default is 3 models. Model name: model_c*n*, where *n* is an integer that is automatically incremented with each new model. `TAILOR SET ORCHESTRAR MODEL_SCR NUMBER_OF_MODELS`
Minimum Score	Minimum ESST score (environment-specific substitution tables) for the SCR. For each residue in a given SCR, the propensity for a residue to be replaced by another in a specific conformation in a particular protein environment is determined from environment substitution tables [Ref. 2]. The ESST score for an SCR is representative of all its residues and is an indication of the SCR's propensity to represent the structure of the target sequence for a given three-dimensional region. Default is -2.0 in a useful range of -4 to 0. More negative values may increase the number of residues built, but may decrease the overall quality of the SCR model. Positive values lead to more restrictive SCRs. `TAILOR SET ORCHESTRAR MODEL_SCR MIN_SCORE`

Post Processing Details

Small gaps may be filled in an automatic post-processing step after building the SCRs. This operation is performed by `BRIDGE`.

Post Processing	You may choose to: • **Fix Small Gaps in Best Model** (default) • **Fix Small Gaps in All Model** • **Do not Fix Small Gaps** `TAILOR SET ORCHESTRAR MODEL_SCR FIX_SMALL_GAPS` Any model created by fixing small gaps is given the name model_f*n*, where *n* is an integer that is automatically incremented with each new model.
Maximum Gap Length	The maximum size of a gap that can be fixed by this process. The default is 3 residues. `TAILOR SET ORCHESTRAR FIX_SMALL_GAPS MAX_GAP_LENGTH`

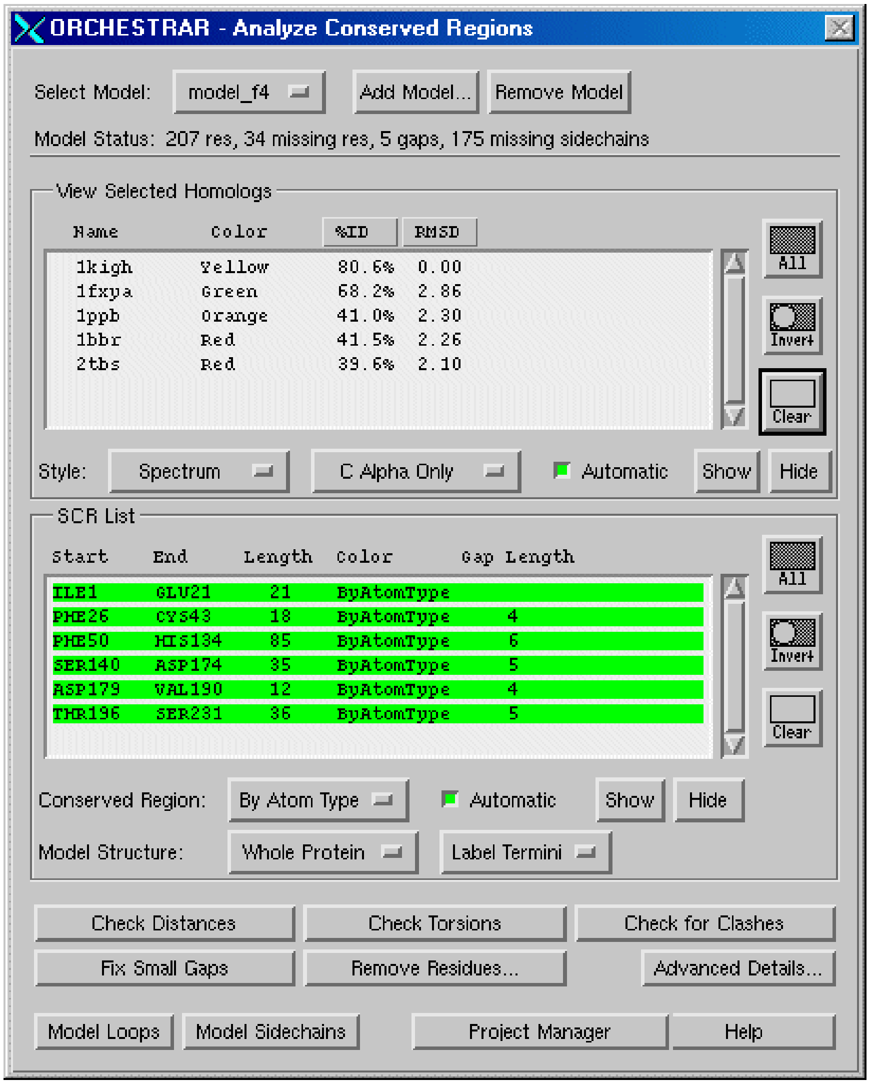
Maximum Attempts	The maximum number of attempts in a build cycle. The default is 10000. `TAILOR SET ORCHESTRAR FIX_SMALL_GAPS MAX_ATTEMPTS`
Maximum Anchor RMSD	Maximum all-atom RMSD at the C-terminal anchor residue. The default is 0.80 Å. `TAILOR SET ORCHESTRAR FIX_SMALL_GAPS MAX_ANCHOR_RMSD`

3.4 Analyze Conserved Regions

To analyze the quality of the structurally conserved regions, perform minor alterations to the SCRs, and fill small gaps.

Access:

- In the *ORCHESTRAR Project Manager*, click **Analyze Conserved Regions**.
- In the *Model Conserved Regions* dialog, click **Build SCRs**.



ORCHESTRAR - Analyze Conserved Regions

Select Model: model_f4

Model Status: 207 res, 34 missing res, 5 gaps, 175 missing sidechains

View Selected Homologs

Name	Color	%ID	RMSD
1kigh	Yellow	80.6%	0.00
1fxya	Green	68.2%	2.86
1ppb	Orange	41.0%	2.30
1bbr	Red	41.5%	2.26
2tbs	Red	39.6%	2.10

Style: Spectrum   C Alpha Only   ☑ Automatic

SCR List

Start	End	Length	Color	Gap Length
ILE1	GLU21	21	ByAtomType	
PHE26	CYS43	18	ByAtomType	4
PHE50	HIS134	85	ByAtomType	6
SER140	ASP174	35	ByAtomType	5
ASP179	VAL190	12	ByAtomType	4
THR196	SER231	36	ByAtomType	5

Conserved Region: By Atom Type   ☑ Automatic

Model Structure: Whole Protein   Label Termini

Usage Note: While this dialog is posted, the editing tools in the *Sequence Viewer* are disabled.

Model

Select Model	The list is populated with models from the `CHORAL` and optional `BRIDGE` run as well as any user-provided custom model. Model names are assigned automatically as model_*an*, where *a* is a single letter identifying the operation and *n* is an integer that is automatically incremented with each new model. Examples: model_c1 (conserved regions), model_f4 (after fixing small gaps).
Add Model	Access a file browser to retrieve a molecule (in PDB or Mol2 format). The molecule will be checked for sequence compatibly with the target sequence before it is added to the list of models.
Remove Model	Remove the selected model from the project directory. A dialog will prompt you to confirm the deletion.
Model Status	The status line reports the number of residues in the selected model and the number of residues and sidechains still missing from that model.

View Selected Homologs

List	The list (which is synchronized with the *Sequence Viewer*) provides the following information for each homolog: • **Name**—Name of the file containing it (without file extension). Only the first 10 characters of homolog names are shown in the list. • **Color**—Name of a solid color, or ByAtomType, or -Hidden- (if the homolog is not currently visible). • **%ID**—%identity of the homolog to the query (target) sequence. Click **%ID** above the list to sort it by increasing or decreasing values in this column. • **RMSD**—Root mean squared distance (based on C-alpha) to the homolog with the largest % identity to the query sequence. Click **RMSD** above the list to sort it by increasing or decreasing values in this column. To increase the size of the list, stretch the dialog vertically.

Homolog Selection	See Selection in ORCHESTRAR Lists on page 91 for a full description. Homologs selected in this list are also bold in the left panel of the *Sequence Viewer*.
Display Style	See ORCHESTRAR Color and Display Styles on page 92 for a full description.

SCR List

List	The list is populated by the selected model. It provides the following information for each SCR: • **Start**—The first residue. • **End**—The last residue. • **Length**— The number of residues. • **Color**—Name of a solid color, or ByAtomType, or -Hidden- (if the homolog is not currently visible). • **Gap Length**—The number of missing residues between an SCR and the one preceding it. To increase the size of the list, stretch the dialog vertically.
SCR Selection	See Selection in ORCHESTRAR Lists on page 91 for a full description.
Conserved Region	Color for the selected SCR(s): • **Spectrum** (default)—Automatic selection of a different color for each displayed SCR. The spectrum consists of 8 recycled colors. • Any of the 24 SYBYL colors. • **By Atom Type** **-Hidden-** identifies an SCR that is not currently displayed.
Model Structure	Display style for the selected SCR(s): • **C-alpha Only** (default) • **Backbone Only** • **Whole Protein**
Label	Label the residues (**Termini**, **All** or **None**) in the selected SCRs.

SCR Analysis and Modification

Check Distances	• Residues on either side of a gap are labeled and colored yellow. • Residues for which the peptide bond distance is >20% from 1.33 Å are colored red, labeled and listed in the textport. • The rest of the model is colored blue. • Residues for which Cα-Cα distances are >20% from 3.8 Å are listed in the textport. • Disconnected pairs are listed in the textport. These consist of two sequential residues for which the amide bond is missing because the two atoms are >1 Å from the ideal peptide bond length of 1.33 Å. • Non-sequential residues connected to each other are listed in the textport.
Check Torsions	Residues are colored according to their placement in the Ramachandran plot: • Red: Residues in the OUTSIDE region. These residues are also labeled. • Yellow: Residues in the GENEROUS region. • Blue: residues in the CORE and ALLOWED regions. This information is also output to the textport.
Check for Clashes	Perform a distance check between atoms in different residues in the selected model and color the model as follows: • Yellow: Residues involved in at least one steric clash. These residues are labeled. • Red: Atoms involved in steric clashes. • Blue: Residues not involved in steric violations.
Fix Small Gaps	Perform a `BRIDGE` run on the selected SCR model using the parameters set through **Advanced Details**. Any model created by fixing small gaps is given the name model_f*n*, where *n* is an integer that is automatically incremented with each new model. If a new model is created, it is added to the list and becomes the new default.

Remove Residues	Residues to be removed from the model may be selected on the screen or via a monomer selection dialog. Deleted residues are replaced by hyphens (-) in the *Sequence Viewer*. The new model inherits the name of the one it was derived from, followed by an underscore and an index number. It is added to the list of models and becomes the new default.
Advanced Details	Access the *Analyze SCR Details* dialog and adjust the parameters used by the **Fix Small Gaps** operation.

Navigation Buttons

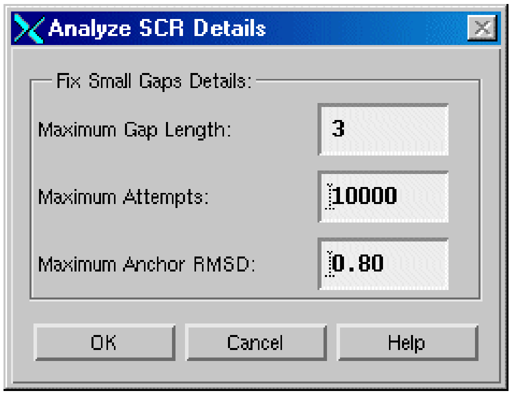
Upon leaving this dialog, the selected model is used for future operations.

Search Loops	Access the *Search Loops* dialog.
Model Sidechains	Access the *Model Sidechains* dialog.
Project Manager	Close this dialog and return to the *ORCHESTRAR Project Manager*.

3.4.1 Analyze SCR Details

Adjust the parameters used by `BRIDGE` to fill small gaps in the model.

In the *Analyze Conserved Regions* dialog, click **Advanced Details**.



Maximum Gap Length: 3

Maximum Attempts: 10000

Maximum Anchor RMSD: 0.80

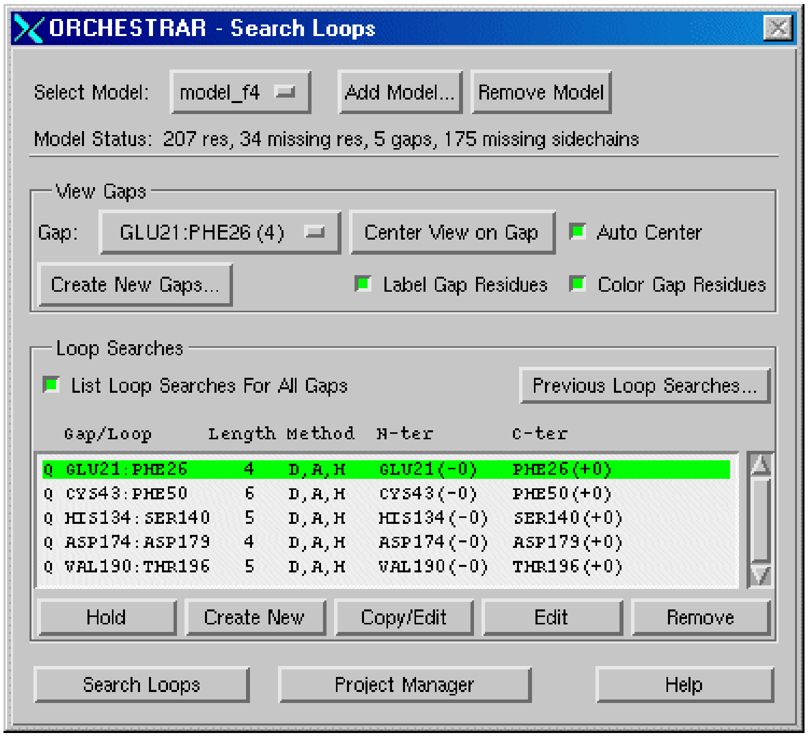
Maximum Gap Length	The maximum size of a gap that can be fixed by this process. The default is 3 residues. `TAILOR SET ORCHESTRAR FIX_SMALL_GAPS MAX_GAP_LENGTH`
Maximum Attempts	The maximum number of attempts in a build cycle. The default is 10000. `TAILOR SET ORCHESTRAR FIX_SMALL_GAPS MAX_ATTEMPTS`
Maximum Anchor RMSD	Maximum all-atom RMSD at the C-terminal anchor residue. The default is 0.80 Å. `TAILOR SET ORCHESTRAR FIX_SMALL_GAPS MAX_ANCHOR_RMSD`

3.5 Search Loops

To inspect gaps in the model's backbone and setup a loop search for each gap.

Access: Click **Search Loops** in:

- the *ORCHESTRAR Project Manager*
- the *Analyze Conserved Regions* dialog



Select Model: model_f4

Model Status: 207 res, 34 missing res, 5 gaps, 175 missing sidechains

Gap: GLU21:PHE26 (4)

☑ Auto Center

☑ Label Gap Residues

☑ Color Gap Residues

☑ List Loop Searches For All Gaps

```
  Gap/Loop       Length Method  N-ter       C-ter
Q GLU21:PHE26      4    D,A,H   GLU21(-0)   PHE26(+0)
Q CYS43:PHE50      6    D,A,H   CYS43(-0)   PHE50(+0)
Q HIS134:SER140    5    D,A,H   HIS134(-0)  SER140(+0)
Q ASP174:ASP179    4    D,A,H   ASP174(-0)  ASP179(+0)
Q VAL190:THR196    5    D,A,H   VAL190(-0)  THR196(+0)
```

Usage Note: While this dialog is posted, the editing tools in the *Sequence Viewer* are disabled.

Model

Select Model	The list is populated with all models generated so far as well as any user-provided custom model. The default model is the one that was selected at the time of exit from the previous dialog. Model names are assigned automatically as model_*an*, where *a* is a single letter identifying the operation and *n* is an integer that is automatically incremented with each new model. Examples: model_c1 (conserved regions), model_f4 (after fixing small gaps).
Add Model	Access a file browser to retrieve a molecule (in PDB or Mol2 format). The molecule will be checked for sequence compatibly with the target sequence before it is added to the list of models.
Remove Model	Remove the selected model from the project directory. A dialog will prompt you to confirm the deletion.
Model Status	The status line reports the number of residues in the selected model and the number of residues and sidechains still missing from that model.

View Gaps

Gap	Each item in the list is identified by the names of the two residues flanking the gap or the disconnected peptide bond followed by the number of missing residues (if any). Selecting a gap in this menu automatically selects it in the list below.
Center View on Gap	Center the model around the selected gap or disconnected pair for easier viewing. This button is accessible only when **Auto Center** is ***off***.
Auto Center	Whether to automatically center the model around the selected gap. This option is ***on*** by default.
Create New Gaps	Access the *Atom Selection* dialog in Monomer mode where you can select the residues to be deleted to form a new gap. Creating a gap allows you to remodel a particular region of a model. This is useful when consecutive residues are disconnected because of a very short or very peptide bond distance or when residues have undesirable geometries.

Label Gap Residues	Whether to label the residues flanking the selected gap. This option is ***on*** by default.
Color Gap Residues	Whether to color (red) the residues flanking the selected gap. This option is ***on*** by default.

Loop Searches

List Loop Searches for All Gaps	Whether to show in the list the loop searches for all identified gaps (default) or only the loop searches associated with the selected gap.
List	One loop search is automatically set up for each gap. the following information is provided: • **H/Q**—Whether the loop is included in the search queue (Q) or on hold (H). The status of a loop can be changed via the **Hold/Queue** button below the list. • **Gap/Loop**—The names of the two residues flanking the gap or disconnected pair. • **Length**—The number of missing residues. • **Method**—The loop search method: D = fragment ***d***atabase (FREAD) created from HOMSTRAD; appropriate for 3-12 residue loops A = ***a***b initio database (PETRA) of 10^8 secondary structure fragments; appropriate for 3-8 residue loops H = ***h***omologs (see the *Setup Loop Search* dialog for details). • **N-ter**—The first residue in the loop to be found. The number in parentheses (-0 by default) indicates the position of this residue with respect to the known residue that flanks this gap. • **C-ter**—The last residue in the loop to be found. The number in parentheses (+0 by default) indicates the position of this residue with respect to the known residue that flanks this gap. To increase the size of the list, stretch the dialog vertically.
Previous Loop Searches	Access the *Previous Loop Searches* dialog and select from a list of loop searches that have already been run.
Hold or **Queue**	**Hold**—Place the selected loop search on hold. **Queue**—Place the selected loop search in the processing queue.

Create New	Access the *Setup Loop Search* dialog where you can define the parameters for a new loop search. ***Note:*** This button is accessible only if a single gap has been selected at the top of the dialog.
Copy/Edit	Access the *Setup Loop Search* dialog where you can define different loop search parameters for the same gap. ***Note:*** This button is accessible only if a single loop search has been selected in the adjacent list.
Edit	Selected a single loop search then press **Edit** to access the *Setup Loop Search* dialog where you can modify the parameters for the loop search run. ***Note:*** This button is accessible only if a single loop search has been selected in the adjacent list.
Remove	Remove the selected loop search from the list.

Action Buttons

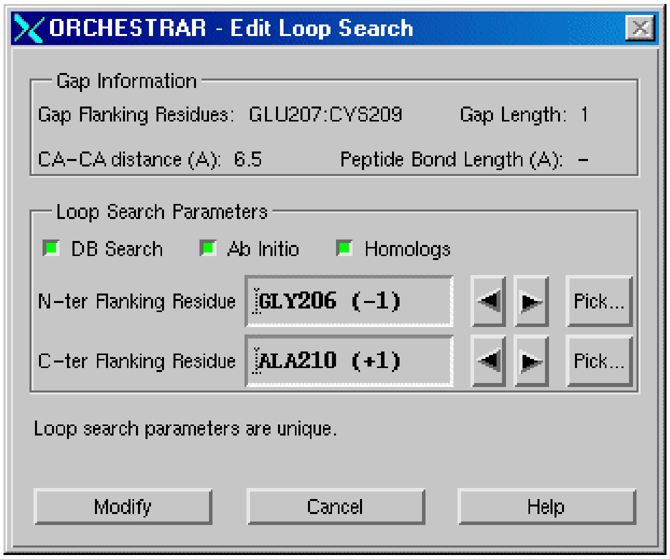
Search Loops	Perform the loop search using the criteria in the list of **Loop Searches**. This button is disabled if no loop searches are defined or they are all on hold. Upon successful completion of a loop search the *Add/ Analyze Loops* dialog will be posted. If all queued loop searches fail a dialog will be posted, recommending a course of action.
Project Manager	Access the *ORCHESTRAR Project Manager*.

3.5.1 Setup Loop Search

To adjust the conditions of a loop search. The name of the dialog reflects the name of the calling button: Create New, Copy, Edit.

Access:

- In the *ORCHESTRAR Project Manager*, click **Search Loops**.
  In the *Search Loops* dialog, click **Create New** or select a loop and click **Copy/Edit** or **Edit**.
- In the *Add/Analyze Loops* dialog, click **Remodel Loops**.
  In the *Search Loops* dialog, click **Previous Loop Searches**.
  In the *Previous Loop Searches* dialog, select a loop and click **Copy Search**.



ORCHESTRAR - Edit Loop Search

Gap Information

Gap Flanking Residues: GLU207:CYS209   Gap Length: 1

CA-CA distance (A): 6.5   Peptide Bond Length (A): –

Loop Search Parameters

☑ DB Search ☑ Ab Initio ☑ Homologs

N-ter Flanking Residue: GLY206 (-1)

C-ter Flanking Residue: ALA210 (+1)

Loop search parameters are unique.

Gap Information

Gap	The name and sequence ID number of the anchor residues that flank the gap.
Gap Length	For gaps only: the number of missing residues. ***Note:*** It is not possible to define a loop search for a gap length of 0.
CA-CA Distance	The Cα-Cα distance in Å.
Peptide Bond Length	For disconnected pairs only: the peptide bond length in Å.

Loop Search Parameters

Method	Activate one or more of the following loop searching methods (all are on by default) • **DB Search**—fragment database (FREAD) created from HOMSTRAD; appropriate for 3-12 residue loops. • **Ab Initio**—database (PETRA) of 10^8 secondary structure fragments; appropriate for 3-8 residue loops. • **Homologs**—3D structure of the homologs.

N-ter Flanking Residue	Name and number of the first residue in the loop. By default, this is the residue that flanks the gap. Use the adjacent arrow buttons to move this endpoint up or down the peptide chain. Click **Pick** to select the residue in the SYBYL window.
C-ter Flanking Residue	Name and number of the last residue in the loop. By default, this is the residue that flanks the gap. Use the adjacent arrow buttons to move this endpoint up or down the peptide chain. Click **Pick** to select the residue in the SYBYL window.

Action Button

Add or **Modify**	• **Add**— Add the newly defined loop search to the list in the *Search Loops* dialog. • **Modify**—Replace an existing loop search with the modified parameters. The new or modified loop search is compared to all others to avoid duplication.

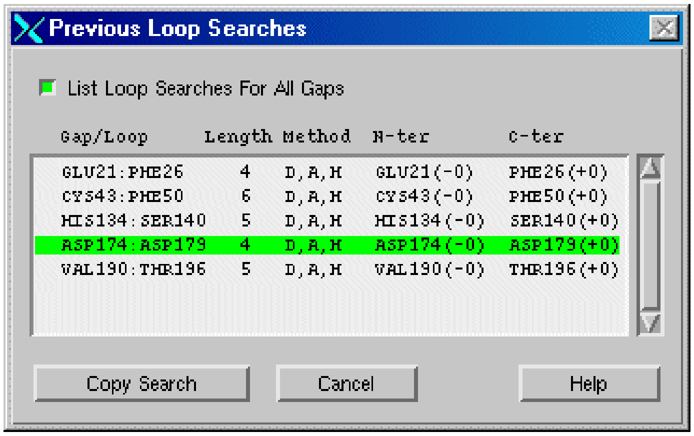
3.5.2 Previous Loop Searches

To access the list of loop searches that have already been run.

Access:

- In the *ORCHESTRAR Project Manager*, click **Search Loops**.
- Then in the *Search Loops* dialog, click **Previous Loop Searches**.

.



☑ List Loop Searches For All Gaps

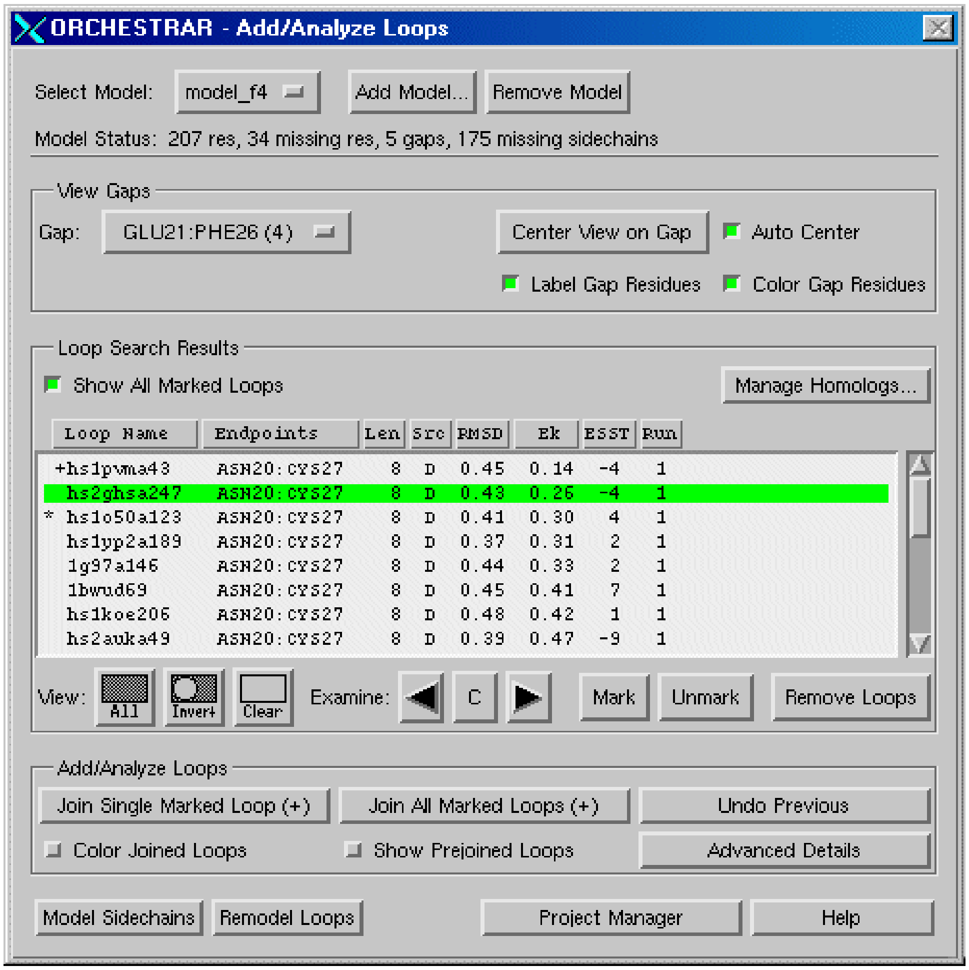
Gap/Loop	Length	Method	N-ter	C-ter
GLU21:PHE26	4	D,A,H	GLU21(-0)	PHE26(+0)
CYS43:PHE50	6	D,A,H	CYS43(-0)	PHE50(+0)
HIS134:SER140	5	D,A,H	HIS134(-0)	SER140(+0)
ASP174:ASP179	4	D,A,H	ASP174(-0)	ASP179(+0)
VAL190:THR196	5	D,A,H	VAL190(-0)	THR196(+0)

List Loop Searches for All Gaps	Whether to list all loop searches or only those for the gap that was selected before accessing this dialog.
List	For each loop search, the following information is provided: • **Gap/Loop**—The name and sequence ID number of the anchor residues that flank the gap. • **Length**—The number of missing residues. • **Method**—The loop search method: D = fragment ***d***atabase (FREAD) created from HOMSTRAD; appropriate for 3-12 residue loops A = ***a***b initio database (PETRA) of 10^8 secondary structure fragments; appropriate for 3-8 residue loops H = ***h***omologs • **N-ter**—Name and number of the first residue in the loop, followed by this residue’s relationship to the residue that flanks the gap in the model. • **C-ter**—Name and number of the last residue in the loop, followed by this residue’s relationship to the residue that flanks the gap in the model.
Copy Search	Copy the conditions of the selected loop search and access the *Copy Loop Search* dialog.

3.6 Add/Analyze Loops

To analyze the proposed loops for each gap, join them to the backbone, and verify and optimize the resulting geometry.

In the *ORCHESTRAR Project Manager*, click **Add Loops**.



Select Model: model_f4

Model Status: 207 res, 34 missing res, 5 gaps, 175 missing sidechains

Gap: GLU21:PHE26 (4)

☑ Auto Center

☑ Label Gap Residues ☑ Color Gap Residues

☑ Show All Marked Loops

Loop Name	Endpoints	Len	Src	RMSD	Ek	ESST	Run
+hs1pvma43	ASN20:CYS27	8	D	0.45	0.14	-4	1
hs2ghsa247	ASN20:CYS27	8	D	0.43	0.26	-4	1
* hs1o50a123	ASN20:CYS27	8	D	0.41	0.30	4	1
hs1yp2a189	ASN20:CYS27	8	D	0.37	0.31	2	1
1g97a146	ASN20:CYS27	8	D	0.44	0.33	2	1
1bwud69	ASN20:CYS27	8	D	0.45	0.41	7	1
hs1koe206	ASN20:CYS27	8	D	0.48	0.42	1	1
hs2auka49	ASN20:CYS27	8	D	0.39	0.47	-9	1

□ Color Joined Loops □ Show Prejoined Loops

Model

Select Model	The list is populated with all models generated so far as well as any user-provided custom model. The default model is the one that was selected at the time of exit from the previous dialog. Model names are assigned automatically as model_*an*, where *a* is a single letter identifying the operation and *n* is an integer that is automatically incremented with each new model. Examples: model_c1 (conserved regions), model_f4 (after fixing small gaps), model_l5 (after adding loops).
Add Model	Access a file browser to retrieve a molecule (in PDB or Mol2 format). The molecule will be checked for sequence compatibly with the target sequence before it is added to the list of models.
Remove Model	Remove the selected model from the project directory. A dialog will prompt you to confirm the deletion.
Model Status	The status line reports the number of residues in the selected model and the number of residues and sidechains still missing from that model.

View Gaps

Gap	Each item in the list is identified by the names of the two residues flanking the gap or the disconnected peptide bond followed by the number of missing residues (if any). Selecting a gap in this menu automatically updates the list of **Loop Search Results** below. **None** indicates that there are no gaps left in the model.
Center View on Gap	Center the model around the selected gap or disconnected pair for easier viewing. This button is accessible only when **Auto Center** is ***off***.
Auto Center	Whether to automatically center the model around the selected gap. This option is ***on*** by default.
Label Gap Residues	Whether to label the residues flanking the selected gap. This option is ***on*** by default.
Color Gap Residues	Whether to color (red) the residues flanking the selected gap. This option is ***on*** by default.

Loop Search Results

Show All Marked Loops	Whether to display the marked loops. By default, the top scoring loop solution for each gap is marked (+) in the list and displayed in green.
Manage Homologs	Access the *Manage Homologs* dialog where you can selectively display the homologs associated with the project.
List	For each loop search result in the list the following information is provided: • **Loop Name**—File containing this loop (.atm file in the LOOPS subdirectory). • **Endpoints**—First and last residues in the loop • **Len**—Number of residues in the loop • **Src**—Method that produced this loop: A (ab initio), D (database search) or H (homolog). By default, the loop solutions are sorted as follows: H > D > A. Within each group, the sort order is by increasing Ek value. • **RMSD**—Root mean square deviation between the residues flanking the gap and the residues flanking the loop. • **Ek**—Energy of the predicted loop in the target structure. The loop solution with the overall lowest Ek value is automatically marked with a + sign. • **ESST**—Environmentally constrained substitution score of a loop (for database solutions only) • **Run**—Run number of the loop search Click any of the column headings above the list to sort it by increasing or decreasing values in this column. To increase the size of the list, stretch the dialog vertically.
View	Loop solutions may be selected in the list in the following manner: • Click on any line in the list to select it. Click again to unselect it. • Ctrl-click adds to or removes from the selection. • Shift-click selects all items in a range. • Click **All** to select all items in the list. • Click **Invert** to invert the current selection. • Click **Clear** to clear the selection. See Loop Color Scheme below.

Examine	Only one loop may be examined at a time. It is flagged with an asterisk before its name in the list. • **<** and **>** Move to the previous or to the next loop in the list of loop search results for the selected gap. An asterisk flags this loop in the list. • **C**—Clear the screen of the loop that was being examined and remove the associated asterisk from the list. See Loop Color Scheme below.
Mark	The purpose of marking a loop solution is to designate the one that will be joined to the model to fill a gap. Only one loop solution may be marked (with a + sign) for each gap. The best loop solution for each gap is automatically marked. This solution is identified by the smallest **Ek** value. To mark another loop solution, simply highlight it in the list and click this button. See Loop Color Scheme below.
Unmark	Remove the + sign associated with the marked loop for the selected gap.
Remove Loops	Remove the selected loop(s) from the list.

Loop Color Scheme	The color of a loop solution depends on the action taken to display it: • Green: The single loop marked with a (+) in the list is always visible and green. • Red: The single loop flagged with a (*) in the list is red. If that loop is also marked with a (+), it is green. • By atom types: All loops selected manually in the list unless flagged by (+) or (*).

Add/Analyze Loops

Join Single Marked Loop (+)	Meld the selected loop (using `TUNER`) onto the protein backbone and minimize the resulting model as follows: 10 iterations on backbone atoms using the Powell method and the Tripos force field. The new model is added to the list and becomes the new default. It is automatically given the name model_l*n*, where *n* is an integer that is automatically incremented with each new model.
Join All Marked Loops (+)	Meld all marked loops (using `TUNER`) onto the protein backbone and minimize the resulting model. Because the top scoring loop solution for each gap is marked automatically, this is a fast and easy way to build the model. After each loop is melded into the appropriate gap it undergoes a quick minimization: 10 iterations on backbone atoms using the Powell method and the Tripos force field. The new model is added to the list and becomes the new default. It is automatically given the name model_l*n*, where *n* is an integer that is automatically incremented with each new model.
Color Joined Loops	Color in green all fixed gaps and joined loops.
Show Prejoined Loops	Display in yellow the loop(s) involved in the last join operation.
Undo Previous	Undo the last loop addition, including subsequent minimization, and delete the associated model.

Navigation Buttons

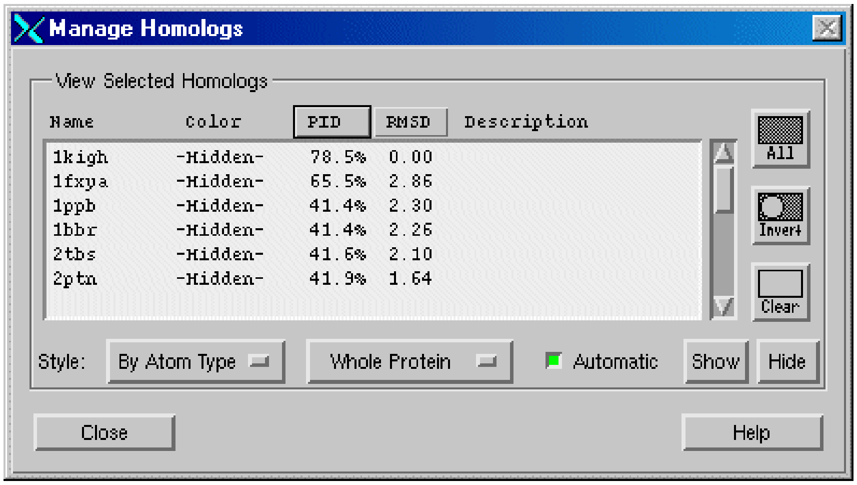
Model Sidechains	Access the *Model Sidechains* dialog. You will be prompted if there are marked loops that have not been joined.
Remodel Loops	Access the *Search Loops* dialog.
Project Manager	Access the *ORCHESTRAR Project Manager*.

3.6.1 Manage Homologs

To access the list of homologs used in the project and selectively display them.

Access:

- In the *ORCHESTRAR Project Manager*, click **Add/Analyze Loops**.
- Then in the *Add/Analyze Loops* dialog, click **Manage Homologs**.



Name	Color	PID	RMSD	Description
1kigh	-Hidden-	78.5%	0.00	
1fxpa	-Hidden-	65.5%	2.86	
1ppb	-Hidden-	41.4%	2.30	
1bbr	-Hidden-	41.4%	2.26	
2tbs	-Hidden-	41.6%	2.10	
2ptn	-Hidden-	41.9%	1.64	

Style: By Atom Type   Whole Protein   ☑ Automatic

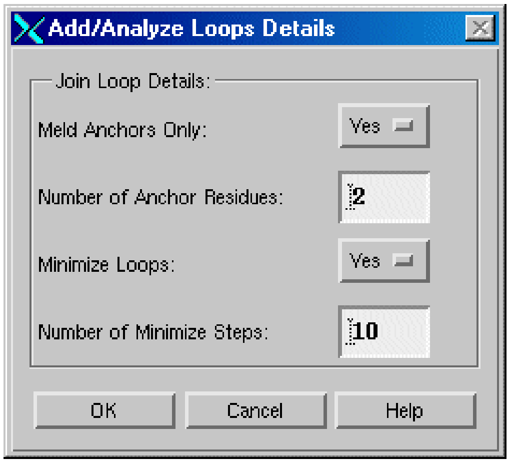
List	The list (which is synchronized with the *Sequence Viewer*) provides the following information for each homolog: • **Name**—Name of the file containing it (without file extension). Only the first 10 characters of homolog names are shown in the list. • **Color**—Name of a solid color, or ByAtomType, or -Hidden- (if the homolog is not currently visible). • **%ID**—%identity of the homolog to the query (target) sequence. Click **%ID** above the list to sort it by increasing or decreasing values in this column. • **RMSD**—Root mean squared distance (based on C-alpha) to the homolog with the largest % identity to the query sequence. Click **RMSD** above the list to sort it by increasing or decreasing values in this column. To increase the size of the list, stretch the dialog vertically.

Homolog Selection	See Selection in ORCHESTRAR Lists on page 91 for a full description. Homologs selected in this list are also bold in the left panel of the *Sequence Viewer*.
Display Style	See ORCHESTRAR Color and Display Styles on page 92 for a full description.

3.6.2 Add/Analyze Loop Details

Adjust the parameters used by TUNER to join loops to the model.

In the *Add/Analyze Loops* dialog, click **Advanced Details**.



Meld Anchors Only: Yes
Number of Anchor Residues: 2
Minimize Loops: Yes
Number of Minimize Steps: 10

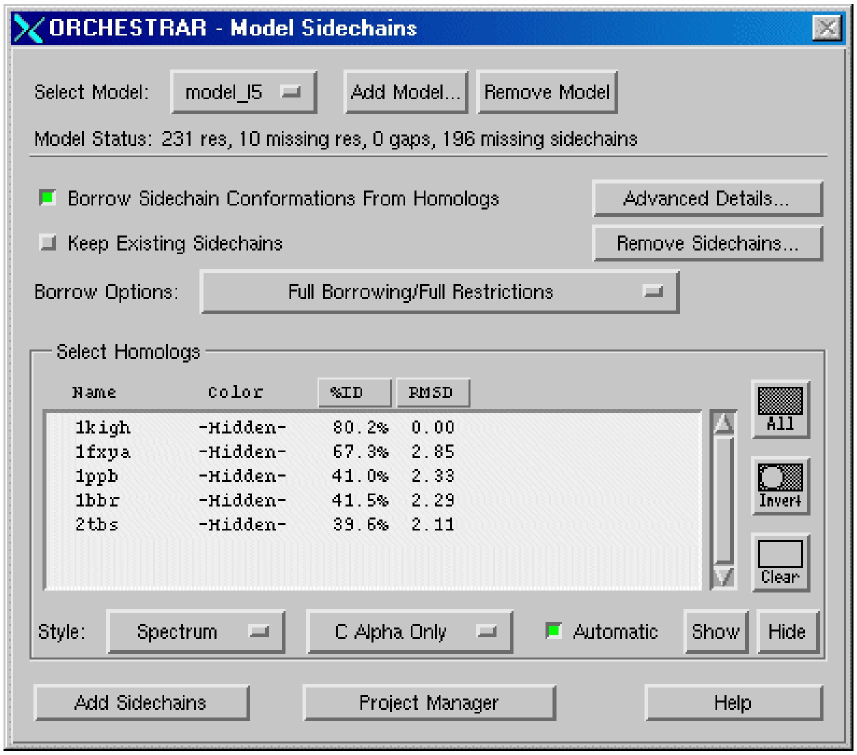
Meld Anchors Only	• **Yes**—Meld only the two anchor residues • **No**—Meld the complete loop and the selected number of anchor residues. TAILOR SET ORCHESTRAR JOIN_LOOPS MELD_ANCHOR_ONLY
Number of Anchor Residues	Number of anchor residues on each side of the gap that will be tuned and minimized when the loop is joined to the model. TAILOR SET ORCHESTRAR JOIN_LOOPS NUM_ANCHOR_RESIDUE

Minimize Loops	Whether to minimize the loop after melding it into the model. Minimization consists of 10 (default) iterations using the Powell method and the Tripos force field. `TAILOR SET ORCHESTRAR JOIN_LOOPS MINIMIZE_LOOPS`
Number of Minimize Steps	The number of iterations for the quick minimization performed after melding a loop. Default is 10 iterations. `TAILOR SET ORCHESTRAR JOIN_LOOPS MINIMIZE_STEPS`

3.7 Model Sidechains

To add sidechains to the protein model. Conformations may be borrowed from the homologs or selected in rotamer libraries.

In the *ORCHESTRAR Project Manager*, click **Model Sidechains**.



Select Model: model_l5

Model Status: 231 res, 10 missing res, 0 gaps, 196 missing sidechains

☑ Borrow Sidechain Conformations From Homologs

□ Keep Existing Sidechains

Borrow Options: Full Borrowing/Full Restrictions

Name	Color	%ID	RMSD
1kigh	-Hidden-	80.2%	0.00
1fxya	-Hidden-	67.3%	2.85
1ppb	-Hidden-	41.0%	2.33
1bbr	-Hidden-	41.5%	2.29
2tbs	-Hidden-	39.6%	2.11

Style: Spectrum C Alpha Only ☑ Automatic

Model

Select Model	The list is populated with all models generated so far as well as any user-provided custom model. The default model is the one that was selected at the time of exit from the previous dialog. Model names are assigned automatically as model_*an*, where *a* is a single letter identifying the operation and *n* is an integer that is automatically incremented with each new model. Examples: model_c1 (conserved regions), model_f4 (after fixing small gaps), model_l5 (after adding loops).
Add Model	Access a file browser to retrieve a molecule (in PDB or Mol2 format). The molecule will be checked for sequence compatibly with the target sequence before it is added to the list of models.

Remove Model	Remove the selected model from the project directory. A dialog will prompt you to confirm the deletion.
Model Status	The status line reports the number of residues in the selected model and the number of residues and sidechains still missing from that model.

Borrowing Conditions

Borrow Sidechain Conformation from Homologs	• ***On***—Allow the selected homologs to used to derive the model’s sidechains. If this option is on and no homologs are selected, all are used. • ***Off***—The sidechain conformations are obtained from one of the following rotamer libraries: Lovell (default) or Dunbrack (see Rotamer Libraries in the Biopolymer Manual). • If the template for the ORCHESTRAR project is a single protein, this option is disabled.
Advanced Details	Access the *Model Sidechains Details* dialog.
Keep Existing Sidechains	• ***On***—Keep all sidechains already on the model. By default, only missing sidechains will be added. You may also use the **Remove Sidechains** button and specify a few sidechains to be removed before proceeding. • ***Off***—All existing sidechains will be removed before proceeding.

Remove Sidechains	Remove some or all sidechains from the model.
Borrow Options	The parameters used by the various borrowing options are specified in the *Model Sidechains Details* dialog. • **Full Borrowing/Full Restrictions**—The fastest and most conservative option. First, borrow as many sidechain conformations as possible from the homologs, then use a rotamer library to add the remaining sidechains (Full Borrowing). For sidechain conformations, use only the chi values from the top scoring rotamers in the library (Full Restrictions). • **Full Borrowing/No Restrictions**—First, borrow as many sidechain conformations as possible from the homologs, then use a rotamer library to add the remaining sidechains (Full Borrowing). For sidechain conformations taken from the rotamer library, use any chi values from the rotamer library (No Restrictions). • **No Borrowing/Restrict Chi1 + Sidechain Clashes**—Use a rotamer library only (No borrowing). For chi1 angles, use only values from the top scoring rotamers (Restrict Chi1). Rotamer solutions must have sidechain-sidechain clash energies less than a maximum tolerance (Sidechain Clashes). • **Full Borrowing/Restrict Sidechain Clashes**—The slowest and most exhaustive search. First, borrow as many sidechain conformations as possible from the homologs, then use a rotamer library to add the remaining sidechains (Full Borrowing). Rotamer solutions must have sidechain-sidechain clash energies less than a maximum tolerance (Sidechain Clashes).

Select Homologs

List	The list (which is synchronized with the *Sequence Viewer*) provides the following information for each homolog: • **Name**—Name of the file containing it (without file extension). Only the first 10 characters of homolog names are shown in the list. • **Color**—Name of a solid color, or ByAtomType, or -Hidden- (if the homolog is not currently visible). • **%ID**—%identity of the homolog to the query (target) sequence. Click **%ID** above the list to sort it by increasing or decreasing values in this column. • **RMSD**—Root mean squared distance (based on C-alpha) to the homolog with the largest % identity to the query sequence. Click **RMSD** above the list to sort it by increasing or decreasing values in this column. To increase the size of the list, stretch the dialog vertically.
Homolog Selection	See Selection in ORCHESTRAR Lists on page 91 for a full description. Homologs selected in this list are also bold in the left panel of the *Sequence Viewer*.
Display Style	See ORCHESTRAR Color and Display Styles on page 92 for a full description.

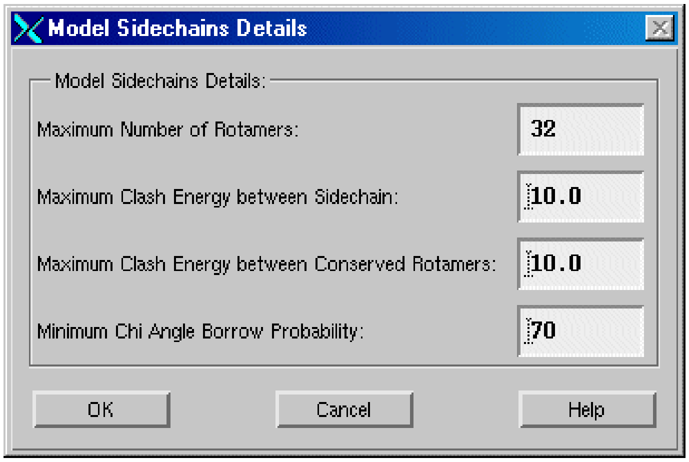
Action Button

Add Sidechains	Perform the sidechain addition.
Project Manager	Access the *ORCHESTRAR Project Manager*.

3.7.1 Model Sidechains Details

Adjust the parameters used to add sidechains to the model.

In the *Model Sidechains* dialog, click **Advanced Details**.



Model Sidechains Details

Model Sidechains Details:

Maximum Number of Rotamers: 32

Maximum Clash Energy between Sidechain: 10.0

Maximum Clash Energy between Conserved Rotamers: 10.0

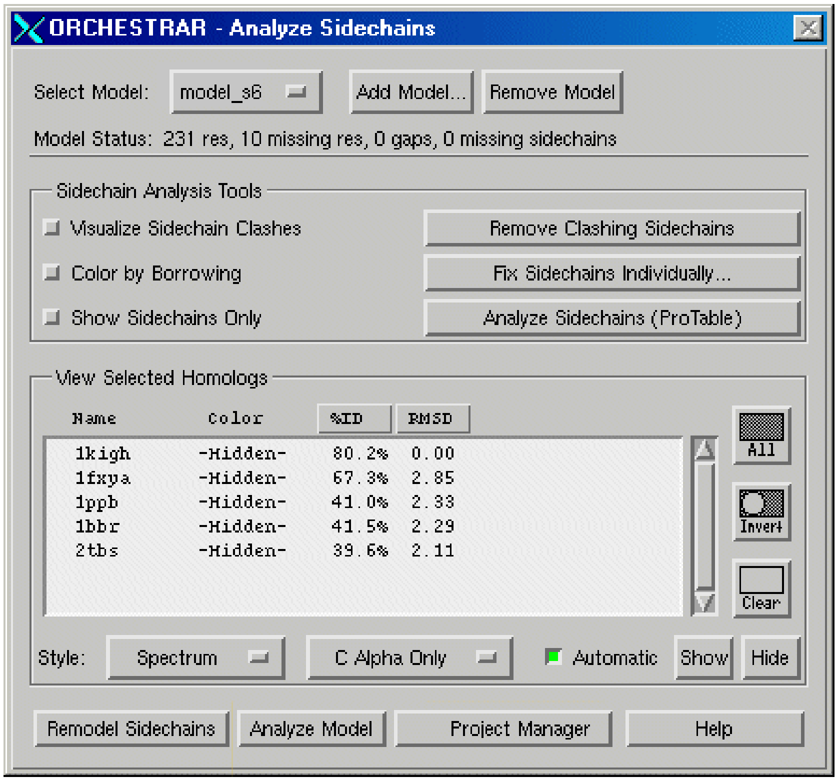
Minimum Chi Angle Borrow Probability: 70

Maximum Number of Rotamers Considered	Maximum number of rotamers considered for each sidechain. The default is 32. Lowering this number can speed up sidechain modelling, but will reduce the quality of the results. `TAILOR SET ORCHESTRAR MODEL_SIDECHAINS MAX_ROTAMERS`
Maximum Sidechain/ Sidechain Clash Energy	Maximum clash energy allowed between any two sidechain atoms belonging to different residues.The default is 10 kcal/mol. `TAILOR SET ORCHESTRAR MODEL_SIDECHAINS MAX_SIDECHAIN_CLASH`
Maximum Clash Energy between Conserved Rotamers	Maximum clash energy allowed between any two sidechain atoms belonging to different residues for which the conformations are conserved. The default is 10 kcal/mol. `TAILOR SET ORCHESTRAR MODEL_SIDECHAINS MAX_CONSERVED_CLASH`
Minimum Probability to Borrow Chi Angles	Minimum probability to borrow Chi angles from templates. For templates with low %ID values, lowering the probability may increase borrowing. `TAILOR SET ORCHESTRAR MODEL_SIDECHAINS MIN_PROBABILITY`

3.8 Analyze Sidechains

To analyze sidechain conformations and correct reported problems.

In the *ORCHESTRAR Project Manager*, click **Analyze Sidechains**.



ORCHESTRAR - Analyze Sidechains

Select Model: model_s6

Model Status: 231 res, 10 missing res, 0 gaps, 0 missing sidechains

Sidechain Analysis Tools

□ Visualize Sidechain Clashes

□ Color by Borrowing

□ Show Sidechains Only

View Selected Homologs

Name	Color	%ID	RMSD
1kigh	-Hidden-	80.2%	0.00
1fxya	-Hidden-	67.3%	2.85
1ppb	-Hidden-	41.0%	2.33
1bbr	-Hidden-	41.5%	2.29
2tbs	-Hidden-	39.6%	2.11

Style: Spectrum · C Alpha Only · ☑ Automatic

Model

Select Model	The list is populated with all models generated so far as well as any user-provided custom model. The default model is the one that was selected at the time of exit from the previous dialog. Model names are assigned automatically as model_*an*, where *a* is a single letter identifying the operation and *n* is an integer that is automatically incremented with each new model. Examples: model_c1 (conserved regions), model_f4 (after fixing small gaps), model_l5 (after adding loops), model_s6 (after adding sidechains).
Add Model	Access a file browser to retrieve a molecule (in PDB or Mol2 format). The molecule will be checked for sequence compatibly with the target sequence before it is added to the list of models.

Remove Model	Remove the selected model from the project directory. A dialog will prompt you to confirm the deletion.
Model Status	The status line reports the number of residues in the selected model and the number of residues and sidechains still missing from that model.

Sidechain Analysis Tools

Visualize Sidechain Clashes	Residues that contain atoms involved in sidechain clashes are colored yellow, while all other residues are colored blue. In the yellow residues, atoms that are in steric contacts are colored red.
Color by Borrowing	Color each sidechain in the 3D model by the color of the homolog that provided the sidechain's geometry. The color legend is printed in the textport.
Show Sidechains Only	Whether to display only the sidechain atoms.
Remove Clashing Sidechains	Remove from the model the sidechains involved in steric clashes.

Fix Sidechains Individually	Access the *Set Sidechain Conformation* dialog (described in the Biopolymer Manual). The list of residues is populated with all the residues identified by activating the adjacent **Visualize Sidechain Clashes**.
Analyze Sidechains (ProTable)	Opens ProTable and fill the spreadsheet with the following information for each residue. • **SourceFile**—C(conserved) or R(estricted). The `conserved.log` file contains a list of target residues where all chi values for a residue where taken from a homolog. The `restricted.log` file contains a list of residues where some of the chi values where taken from homologs. In either case, the residue is identical between the target and homolog. • **Parent_Homolog**—The parent homolog from which chi information was obtained. • **PID**—Percent sequence identity between the target and homolog sequence. • **Conserved_Prob**—In the `conserved.log` file, the calculated probabilty that all chi angles used to construct the sidechain from the parent are conserved. In the `restricted.log` file, the probabilty that either the first chi angle or the first two chi angles are conserved; all solutions from the rotamer library are then restricted to rotamers within 30° (at chi1 or chi1+2) of the highest scoring parent. • **Parent_Residue**—The residue type of the parent homolog at a given position. • **Conserved_wght**—An arbitrary weight used when a number of homologs have the same residue type as the target sequence. A weight is added to the parent with the highest PID, biasing chi value information toward this parent. Conserved weight = (0.6 * parent PID) + (0.4 * Conserved probability) • **Chi***n*—Chi values used in the model.

View Selected Homologs

List	The list (which is synchronized with the *Sequence Viewer*) provides the following information for each homolog: • **Name**—Name of the file containing it (without file extension). Only the first 10 characters of homolog names are shown in the list. • **Color**—Name of a solid color, or ByAtomType, or -Hidden- (if the homolog is not currently visible). • **%ID**—%identity of the homolog to the query (target) sequence. Click **%ID** above the list to sort it by increasing or decreasing values in this column. • **RMSD**—Root mean squared distance (based on C-alpha) to the homolog with the largest % identity to the query sequence. Click **RMSD** above the list to sort it by increasing or decreasing values in this column. To increase the size of the list, stretch the dialog vertically.
Homolog Selection	See Selection in ORCHESTRAR Lists on page 91 for a full description. Homologs selected in this list are also bold in the left panel of the *Sequence Viewer*.
Display Style	See ORCHESTRAR Color and Display Styles on page 92 for a full description.

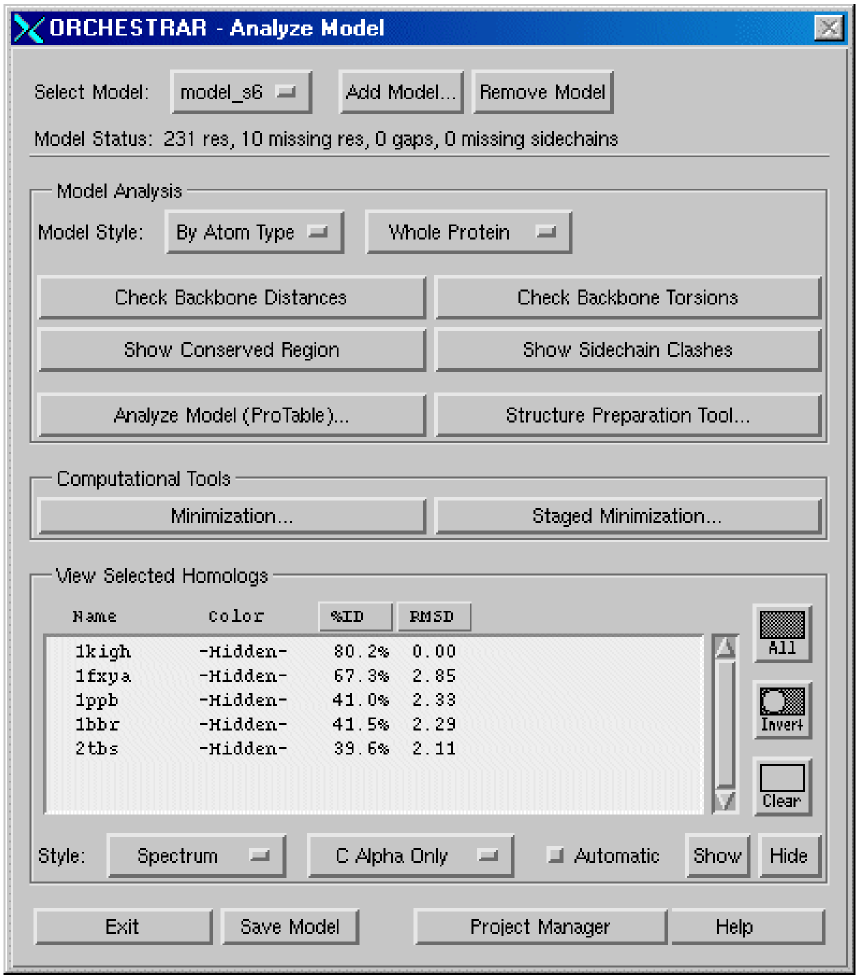
Navigation Buttons

Remodel Sidechains	Access the *Model Sidechains* dialog.
Analyze Model	Access the *Analyze Model* dialog.
Project Manager	Access the *ORCHESTRAR Project Manager*.

3.9 Analyze Model

To analyze the quality of the final model.

In the *ORCHESTRAR Project Manager*, click **Analyze Model**.



Select Model: model_s6

Model Status: 231 res, 10 missing res, 0 gaps, 0 missing sidechains

Model Style: By Atom Type; Whole Protein

Name	color	%ID	RMSD
1kigh	-Hidden-	80.2%	0.00
1fxya	-Hidden-	67.3%	2.85
1ppb	-Hidden-	41.0%	2.33
1bbr	-Hidden-	41.5%	2.29
2tbs	-Hidden-	39.6%	2.11

Style: Spectrum; C Alpha Only; □ Automatic

Model

Select Model	The list is populated with all models generated so far as well as any user-provided custom model. The default model is the one that was selected at the time of exit from the previous dialog. Model names are assigned automatically as model_*an*, where *a* is a single letter identifying the operation and *n* is an integer that is automatically incremented with each new model. Examples: model_c1 (conserved regions), model_f4 (after fixing small gaps), model_l5 (after adding loops), model_s6 (after adding sidechains).
Add Model	Access a file browser to retrieve a molecule (in PDB or Mol2 format). The molecule will be checked for sequence compatibly with the target sequence before it is added to the list of models.
Remove Model	Remove the selected model from the project directory. A dialog will prompt you to confirm the deletion.
Model Status	The status line reports the number of residues in the selected model and the number of residues and sidechains still missing from that model.

Model Analysis

Model Style	See ORCHESTRAR Color and Display Styles on page 92 for a full description.
Check Backbone Distances	• Residues on either side of a gap are labeled and colored yellow. • Residues for which the peptide bond distance is >20% from 1.33 Å are colored red, labeled and listed in the textport. • The rest of the model is colored blue. • Residues for which Cα-Cα distances are >20% from 3.8 Å are listed in the textport. • Disconnected pairs are listed in the textport. These consist of two sequential residues for which the amide bond is missing because the two atoms are >1 Å from the ideal peptide bond length of 1.33 Å. • Non-sequential residues connected to each other are listed in the textport.

Check Backbone Torsions	Residues are colored according to their placement in the Ramachandran plot: • Red: Residues in the OUTSIDE region. These residues are also labeled. • Yellow: Residues in the GENEROUS region. • Blue: residues in the CORE and ALLOWED regions. This information is also output to the textport.
Show Conserved Region	Structurally conserved regions of the model are colored red.
Show Sidechain Clashes	Residues that contain atoms involved in sidechain clashes are colored yellow, while all other residues are colored blue. In the yellow residues, atoms that are in steric contacts are colored red.
Analyze Model (ProTable)	Launch ProTable (described in the ProTable Manual).
Structure Preparation Tool	Access the *Prepare Protein Structure* dialog (described in the Biopolymer Manual).Use it to add hydrogens, type the atoms for the AMBER force fields and add atomic charges.

Computational Tools

Minimization	Access the *Minimize* dialog to minimize the strain energy of the model. See also Minimize Energies in the Force Field Manual.
Staged Minimization	Perform a series of minimizations on the model as follows: • Hydrogens only • Hydrogens + sidechains • Hydrogens + sidechains + backbone atoms except C-alpha • All atoms

View Selected Homologs

List	The list (which is synchronized with the *Sequence Viewer*) provides the following information for each homolog: • **Name**—Name of the file containing it (without file extension). Only the first 10 characters of homolog names are shown in the list. • **Color**—Name of a solid color, or ByAtomType, or -Hidden- (if the homolog is not currently visible). • **%ID**—%identity of the homolog to the query (target) sequence. Click **%ID** above the list to sort it by increasing or decreasing values in this column. • **RMSD**—Root mean squared distance (based on C-alpha) to the homolog with the largest % identity to the query sequence. Click **RMSD** above the list to sort it by increasing or decreasing values in this column. To increase the size of the list, stretch the dialog vertically.
Homolog Selection	See Selection in ORCHESTRAR Lists on page 91 for a full description. Homologs selected in this list are also bold in the left panel of the *Sequence Viewer*.
Display Style	See ORCHESTRAR Color and Display Styles on page 92 for a full description.

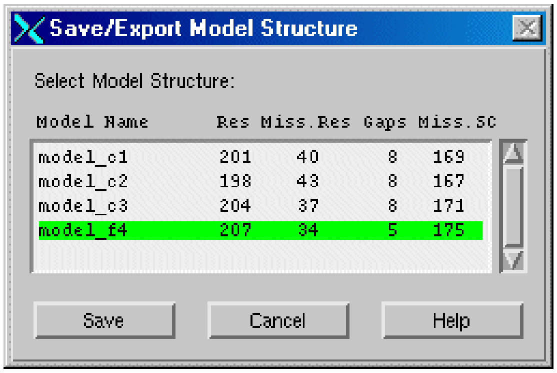
Navigation Buttons

Exit	Exit ORCHESTRAR. You will be prompted whether to retain the current model in the molecule area.
Save Model	Access SYBYL’s Save As dialog where you can save the model in a variety of formats.
Project Manager	Access the *ORCHESTRAR Project Manager*.

3.10 Save the Model

In the *ORCHESTRAR Project Manager*, click **Save Model**.

If more than one model is available, you must select the one to be saved.



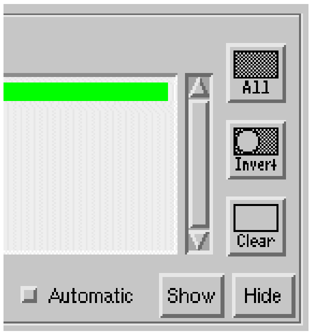
Model Name	Res	Miss.Res	Gaps	Miss.SC
model_c1	201	40	8	169
model_c2	198	43	8	167
model_c3	204	37	8	171
model_f4	207	34	5	175

Model Name	Name given by ORCHESTRAR to the model.
Res	Total number of residues in the model.
Miss.Res	Number of missing residues.
Gaps	Number of gaps.
Miss.SC	Number of missing sidechains.
Save	Access SYBYL’s *Save As* dialog where you can select the format of the output file and specify its name.

3.11 Frequently Used Elements in ORCHESTRAR Dialogs

3.11.1 Selection in ORCHESTRAR Lists

The following features appear in multiple dialogs to select and control the display of homologs and structurally conserved regions.



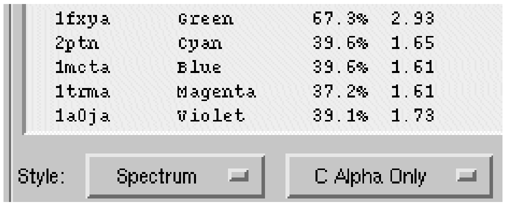
Use your browser’s Back button or link to the description of the dialogs that include this feature:

- *Model Conserved Regions*
- *Analyze Conserved Regions*
- *Manage Homologs* (in *Add/Analyze Loops*)
- *Model Sidechains*
- *Analyze Sidechains*
- *Analyze Model*

Selection	Items may be selected in the list in the following manner: • Click on any line in the list to select it. Click again to unselect it. • Ctrl-click adds to or removes from the selection. • Shift-click selects all items in a range. • Click **All** to select all items in the list. • Click **Invert** to invert the current selection. • Click **Clear** to clear the selection.
Automatic	When this check box is active: • Items selected in the list are automatically displayed in the SYBYL window. When using this method, none of the atoms in the selected items will be highlighted in the SYBYL window. • Items that are not selected are not displayed and are identified as **-Hidden-** in the Color column.
Show and **Hide**	Accessible only if **Automatic** is ***off***. • Click **Show** to display the selected item(s). When using this method, all atoms in the selection will be highlighted in the SYBYL window. To remove the highlights, click **Clear**. • Click **Hide** to undisplay the selected item(s).

3.11.2 ORCHESTRAR Color and Display Styles

The following features appear in multiple dialogs to enhance the display of homologs and structurally conserved regions.



```
1fxya     Green      67.3%  2.93
2ptn      Cyan       39.6%  1.65
1mcta     Blue       39.6%  1.61
1trma     Magenta    37.2%  1.61
1a0ja     Violet     39.1%  1.73
```

Style: Spectrum   C Alpha Only

Use your browser’s Back button or link to the description of the dialogs that include this feature:

- *Model Conserved Regions*
- *Analyze Conserved Regions*
- *Manage Homologs* (in *Add/ Analyze Loops*)
- *Model Sidechains*
- *Analyze Sidechains*
- *Analyze Model*

Color Style	Color for the selected item(s): • **Spectrum** (default)—Automatic selection of a different color for each displayed item. The spectrum consists of 8 recycled colors. • Any of the 24 SYBYL colors • **By Atom Type** • **-Hidden-** Identifies an item that is not currently displayed. ***Note:*** The model and any of the homologs may also be colored via the *Sequence Viewer*'s **Color Molecule** (as described in the Biopolymer Manual).
Display Style	Display style for the selected item(s): • **C-alpha Only** (default) • **Backbone Only** • **Whole Protein** • **Ribbon Only**—The Kabsch-Sander method is used to identify secondary structure elements. If the homolog is monochrome, the whole ribbon matches that color. Color changes and selections made in the *Sequence Viewer* have not effect on ribbons. • **Protein + Ribbon**—Add the whole protein to the ribbon display. Ribbon rendering depends on licensing and on the variable `TAILOR!ORCHESTRAR!MOLCAD_RIBBONS` (which must be set before accessing ORCHESTRAR). If the homolog is colored by atom types the color of the corresponding ribbon depends on the method. • With a MOLCAD license (default if a license is available): Helices = red ribbons Sheets = blue directional arrows Turns = magenta tubes Everything else = yellow tubes • Without a MOLCAD license (uses coloring from `TAILOR!RENDER`): Helices = magenta ribbons Sheets = yellow directional arrows Everything else = cyan tubes

This page intentionally blank.

Chapter 4.

ORCHESTRAR O/S Commands

The ORCHESTRAR suite includes the following programs:

BATON on page 96	Produces a multiple structural alignment
CHORAL on page 98	Determines the structurally conserved regions between the homologs and the target sequence.
BRIDGE on page 99	Fills small gaps in the non-contiguous structure.
CODA on page 101	Finds and ranks loop solutions.
SEARCHSLOOP on page 105	Identifies loop candidates.
TUNER on page 110	Melds loops into the model.
ANDANTE on page 112	Adds sidechains to the 3D structure.
HARMONY on page 120	Identifies mis-aligned regions in the comparative model.

See also: ORCHESTRAR Input and Output Files on page 123

When running ORCHESTRAR at the command-line, use SYBYL, in a different shell, to view your results. To learn how to use ORCHESTRAR at the O/S command line, see ORCHESTRAR Unix Commands Tutorial on page 26.

Limitation: Results may vary slightly on different hardware platforms because of the handling of floating point numbers and the iterative nature of some of the algorithms (especially CHORAL).

Conventions

Note: Some of the ORCHESTRAR command examples are very long and do not fit on one line. A backslash (\) indicates the line break.

In the tables on the following pages, these conventions are used:

`command`	Enter the command or parameter as written
filename or number	Supply the required information
[optional]	Indicates an optional parameter
{repeating}	Indicates a series of entries

4.1 BATON

Method

BATON takes a set of related protein structures (.atm files by default), performs a multiple alignment and produces a set of aligned structures with the file extension .brk. The input data may come from FUGUE or from other sources.

BATON uses a number of different protein features that include sequence, as well as structural information. Given two proteins, *I* (*i* residues long) and *J* (*j* residues long), BATON creates an *i* x *j* matrix. Each element of the matrix is then filled with a score. The score is the sum total of the difference of a number of structural descriptors. Note that there is a large number structural features.

Consider %solvent accessibility (PSA) of the sidechains at the *i*th and *j*th element in the matrix. The contribution of this feature would be given by:

W_{psa} x $(\mathrm{PSA}_i - \mathrm{PSA}_j)/100$

where W_{psa} = weight assigned to this structural feature. If residues have matching structural features then the contribution to the sum total to the total matrix element score is 0. Once each element of the matrix has been filled, a generic dynamic programming routine finds the lowest scoring path from the upper left hand section of the matrix (0,0) to the lower right hand version of the matrix (*i*,*j*). A similar to a sequence alignment only using structural features instead of simple sequence identity. The result of this is a structural alignment of *i* to *j*. This alignment is of length *n*.

Now if protein *K* (of length *k*) is added, an *n* x *k* matrix is generated, each *k*th position is checked against all members of the *n*th position of the matrix (each *n*th position consists of a residue from protein *I* and protein *J* or maybe gaps) to generate a weighted score for the matrix element. Dynamic programming is then used again to align *K* to the previously generated alignment of *i* and *j*. Note the sequential build up of the structural alignment. Different results can be obtained depending on the build order.

To learn more about BATON, see Zhu et al., 1992 [Ref. 1].

Note: By default, BATON accepts PDB files with the .atm extension. These are .pdb files that have been stripped, leaving ATOM, HEADER and REMARK records. To use PDB files, set the environment variable PDB_EXT to .pdb so that standard .pdb files may be used.

Command Format

```
Baton pdbfile1 pdbfile2 {pdbfileN} [options]
```

pdbfile	Input protein structure files with .atm extensions.
[**-pdbout**]	Produces superimposed PDB files based upon equivalence residues derived from alignment.
[**-update**]	Iteratively update the list of equivalent residues. This improves the superposition over a smaller number of residues.
[**-cutoff** number]	Maximum Ca-CA distance between equivalent residues. The default is 3.5 Å.
[**-maxrmsd** number]	Output structures that have an RMSD to pdbfile1 below this value. The default is 10 Å.
[**-minequiv** number]	Minimum number of equivalent residues for superposition. Enter the value as a percentage of protein lenght.
[**-allatom**]	Use all atoms in RMSD calculation.
[**-caatom**]	Use only CA atoms in RMSD calculation.
[**-mainchain**]	Use only the main chain atoms in RMSD caculation.
[**-list** filename]	Read the list of proteins from a file.
[**-input** filename]	Parameter file containing the default settings.
[**-output** filename]	Set the name of the output file (default baton.out).
[**-error** filename]	Set the name of the error file (default baton.err).
[**-aliout** filename]	Set the name of the alignment file (default baton.ali).
[**-outall**]	Output all sub-alignments.
[**-baton** directory]	Location of the data files (overrides $BATON).
[**-batonbin** directory]	Location of the executables (overrides $BATONBIN).

Note: The structure alignment results are dependent on the order of the PDB files in the command. This means that if you change the order of the PDB files, the alignment may look different.

Example:

```
Baton 1HCL.pdb 1P38.pdb 3ERK.pdb
```

4.2 CHORAL

Method

CHORAL takes a multiple alignment and its associated structure files, and determines structurally conserved regions (SCRs).

To learn more about this method, see Montalvão & al, 2005 [Ref. 2].

Hardware Note: Results may vary slightly on different hardware platforms because of the handling of floating point numbers and the iterative nature of some of the algorithms.

CHORAL generates the following files:

- `CORE0x.pdb`: a PDB file containing the SCRs
- `choral-andante.ali`: an alignment file used by ANDANTE when adding side chains
- `choral-bridge.pir`: a PIR file used by BRIDGE when filling small gaps

Command Format

```
choral -a alignment_file [options]
```

-a filename	The input alignment file (with a .ali extension).
[**-d** distance]	C-alpha maximum distance (default = 4.5 Å)
[**-F** fusion-score]	Minimum score for cluster fusion (default = -10)
[**-m**number]	Number of models to be created (default = 5)
[**-k**]	Output kin files
[**-p**]	Output ps files
[**-V**]	Set on verbose mode
[**-b**]	Override build warning
[**-h**]	Display help for this command

Examples:

```
choral -a target_sequence_kinase_ma.ali -m3
choral -a 1fm6_hormone_rec_mod_fugue_ma.ali
```

4.3 BRIDGE

Method

BRIDGE is typically used to close small gaps between conversed regions generated by CHORAL. It can fill a gap of up to three residues.

ORCHESTRAR provides a default file configuration file ($TA_ORCHESTRAR/Bridge/data/bridge_config). You may copy this file to your working directory and modify it to match your specific input files.

You can use the PDB template file (CORE01.pdb) alone or with the choral-bridge.pir from CHORAL.

BRIDGE uses the TUNER algorithm (see TUNER on page 110) to incorporate a loop into the model. The output file, bridge.pdb, contains the updated model.

Command Format

```
bridge -pdb template-file -seq sequence-file [options]
```

Alternative: **bridge --cfg bridge_config**

--cfg config_file	Parameter file
-pdb template_file	Template structure PDB file: the PDB file from CHORAL containing the SCRs (CORE01.pdb)
-seq sequence_file	PIR sequence file from CHORAL (choral-bridge.pir).
[**-max-attempts** number]	Maximum number of attempts in build cycle (default = 10000)
[**-out-file** filename]	File to output final model (default = bridge.pdb)
[**-notune**]	Turns off tuning.
[**-seed** number]	Randomize seed value (default = -55)
[**-rama-file** filename]	Ramachandran data file (default = all5deg.data)
[**-rama-resolution** number]	Ramachandran resolution (default = 5 degrees)
[**-pp-attempts** number]	Number of attempts to extract phi/psi state (default = 10)
[**-min-atom-dist** number]	Minimum allowed atom-atom distance (default = 2.0 Å)

[**-max-anchor-rmsd** number]	Maximum allowed all-atom RMSD at the C-terminal anchor residue (default = 0.8 Å)
[**-core-propensity** number]	Propensity for the core Ramachandran region (default = 0.70)
[**-trans-propensity** number]	Propensity for trans-peptide bond (default = 0.99)
[**-max-gap-fill** number]	Maximum gap length that can be filled (default = 3 residues)
[**-replace-chain-id** character]	Replace chain IDs of gaps, including anchors
[**start-residue-num** number]	Fill a single gap only at the specified residue number (default = 0)
[**-start-residue-chain** character]	Fill a single gap only in the specified chain (default = A)

Examples

```
bridge -pdb core01.pdb -seq choral-bridge.pir
bridge --cfg bridge_config
```

Bridge Configuration File (default)

```
#
#Template configuration file for Bridge
#
job-name bridge_core
template-file CORE01.pdb
pir-file choral-bridge.pir
out-file bridge.pdb
seed -55
rama-file all5deg.data
rama-resolution 5
max-attempts 10000
pp-attempts 10
min-atom-dist 2.0
max-anchor-rmsd 0.8
core-propensity 0.99
max-gap-fill 3
# notune
#replace-chain-id-Y
```

4.4 CODA

CODA predicts the structurally variable regions (SVRs) in proteins.

CODA combines FREAD, a knowledge-based approach, and MINUS, which constructs the region *ab initio* (a conformation searching method), then generates a consensus prediction of loop solutions from the results.

To learn more about this method, see Deane and Blundell, 2001 [Ref. 3].

4.4.1 FREAD

Method

FREAD selects loop candidates from a database of protein structure fragments with environmentally constrained substitution tables and other rule-based filters.

The FREAD database contains approximately 5000 structures, with a resolution greater than 2.5 Å, from the HOMSTRAD database. This database uses four rule-based filters:

- Dc(f) - difference of C-alpha separation of target and fragment anchor positions
- RMSD - backbone atoms in the anchor regions
- Ek - energy of fragment in the target structure
- Sc(F) - environmentally constrained substitution score of the fragment (large scores indicate homologous fragment)

FREAD can use a tuner.pdb file, a bridge.pdb file, or any user-provided PDB file with gaps as input.

FREAD generates a log file (FREAD.HITS) listing the fragments that match the criteria of anchor positions and scores based on substitution tables. FREAD also copies the top scoring solutions (from its database) in PDB format.

Command Format

FREAD3

-pdb number	Input PDB file with loops/gaps to model
-seq filename	Sequence file containing the full sequence (gaps included) in PIR format
-o filename	Output file (default = FREAD.HITS)

`-startres` number	ID number of the starting residue
`-len` number	Number of residues in the loop (maximum = 12 residues)

In order to use a proprietary fragment database, you need to use ***both*** of the following two commands.

`-dbdir` directory	Path to directory of the fragment database files
`-dbfile` filename	Name of the fragment database

Examples:

```
FREAD3 -pdb model.pdb -seq model.seq -len 4 -startres 64
FREAD3 -pdb model.pdb -seq model.seq -len 4 \
-startres 64 -o FREAD.HITS -dbdir user/FRAGDB \
-dbfile MYOWN.db
```

4.4.2 PETRA (MINUS and Read7)

Method

PETRA performs *ab initio* loop prediction. PETRA is composed of Read7 and MINUS. MINUS selects polypeptide fragments from a computer generated database and encodes all possible peptide fragments, up to eight amino acids long.

To learn more about this method, see Deane and Blundell, 2000 [Ref. 4].

You can run MINUS and Read7 at the same time, or separately. If you run them separately, run Read7 before MINUS.

The PETRA database is a computer generated conformations database. This ab initio polypeptide fragment database (APD) contains all polypeptide fragments of eight residues in length that can be made from a set of eight possible phi/psi pairs using ideal main-chain parameters. APD is a representation of the fragments as a non-degenerate list of alpha-carbon separations.

Limitation

The maximum loop length supported by MINUS and Read7 is 8 residues.

Command Format

Read7

Read7 is a preprocessor that takes a PDB file containing structurally conserved regions. It generates a log file of fragments that pass simple rule-based filters in order to reduce the number of candidates for MINUS.

-pdb filename	Input PDB file with loops to model
-seq filename	Sequence file containing the full sequence (gaps included) in PIR format
[**-o** filename]	Output log file; required for MINUS2. ***Note:*** If **-o** is not specified, the output file defaults to: TA_ORC_APD.tmp
-startres number	ID number of the starting residue
-len number	Number of residues in the loop (maximum = 8 residues)

MINUS

MINUS generates a log file that contains loop solutions. When used in conjunction with Read7, MINUS generates PDB solutions from a subset created by Read7 that have passed certain RMSD (root mean square deviation) and energy criterion.

-pdb filename	Input PDB file with loops to model
-seq	Sequence file containing the full sequence, gaps included, in PIR format
-len number	Number of residues in the loop (maximum = 8 residues)
-startres number	ID number of the starting residue
[**-o** filename]	Output file to store results. (default = MINUS2.HITS)
[**-m** filename]	Output log file for Read7. The Read7 default output file is TA_ORC_APD.tmp ***Note:*** If you change the Read7 output log file name, make the corresponding change in MINUS.

Examples:

```
Read7 -pdb my.pdb -seq mypdb.seq -o my.HITS -len 4 \
-startres 85
```

```
MINUS2 -pdb my.pdb -seq mypdb.seq -o my.HITS -len 4 \
-startres 85
```

MINUS with Read7:

```
MINUS2 -pdb my.pdb -seq mypdb.seq -o my.HITS -len 4 \
-startres 85 -read7
MINUS2 -pdb my.pdb -seq mypdb.seq -o my.HITS -len 4 \
-startres 85 -read7 -m myread7.log
```

4.4.3 CODA Program

Method

CODA extracts the results from FREAD and PETRA, clusters them, and selects a fragment from these clusters using several rule-based filters. CODA outputs FREAD and Petra loop pairs based on RMSD, Ek, and phi/psi differences. Each loop also lists an RMSD between the loop and the model anchor regions, along with an Ek.

Command Format

CODA3

-ff filename	FREAD output hits file
-mf filename	MINUS output hits file
-o filename	Output file to store results (CODA log file)
-len number	Number of residues to model, starting from the starting residue
-th	Take top homolog hit with cutoff

Examples:

In the example below, the output will default to CODA.HITS

```
CODA3 -ff FREAD.HITS -mf MINUS.HITS -len 4
```

In the example below, none of the default file names are used.

```
CODA3 -ff fread_249_6.log -mf minus_249_6.log \
-o ocoda_249_6.log -len 16
```

4.5 SEARCHSLOOP

Method

SEARCHSLOOP searches the SLoop database for potential loops for gaps in 3D structures; it uses an SSTRUC .sst file. SEARCHSLOOP generates and scores a set of potential loops.

In the SLoop database, the loops are classified according to their length, the type of bounding secondary structure, and the conformation of the main chain. Classification into structural families depends on two values: the mean distance between the first and last alpha Carbon, and the distance to the center of gravity of the cluster. Each class in the SLoop database contains information about the sequences of its member loops, the local structural environment of the loop residues, and the angle and distance between secondary structures.

To learn more about this method, see Burke et al, 2000 [Ref. 5].

Command Format

`SearchSLoop`

General Options

-pdb filename	Partial or full model/structure
[**-V**]	Verbose mode
[**-H**]	Help

Selection of Sequences

[**-seq** filename]	Input full sequence file (FASTA or PIR) (overrides PDB)
[**-protid** name]	Select protein from multiple sequence alignment file
[**-allseq**]	Use all sequences in a multiple sequence alignment

Selection of Loops

[**-writemean**]	Write individual .mean files for each loop
[**-startresindex** number]	Search for a loop to fit the gap starting at the residue identified by its index number. The residue range for the protein is from 0 to the number of residues minus 1. However, the number specified with this parameter may not be less than 1 because two anchor residues are required.

[**-endresindex** number]	Search for a loop to fit the gap ending at residue identified by its index number. The residue range for the protein is from 0 to the number of residues minus 1. However, the number specified with this parameter may not be greater than the number of residues minus 2 because two anchor residues are required.
[**-startresnum** string]	Search for a loop starting with the residue identified by its label
[**-endresnum** string]	Search for a loop ending with the residue identified by its label
[**-allseg**]	Search all loops
[**-startseg**]	Search loops starting at the specified segment
[**-endseg**]	Search loops ending at the specified segment

SLoop-related Options

[**-num** number]	Minimum number of loops in SLoop class 1
[**-runsloop**]	Run SLoop on selected loops
[**-bounding**]	Score on bounding secondary structures
[**-build**]	Build and score loops onto PDB structure
[**-writeatm**]	Write PDB of unsuperimposed loop and 3 residues of neighboring secondary structure if `-bounding` is set) into $SLOOP_ATM directory
[**-writefrag**]	Write PDB of unsuperimposed loop and complete bounding structure into $SLOOP_ATM directory (overwrites above)
[**-writestruct**]	Write HTML of sst definition
[**-writepdbfrag**]	Write PDB of fragment superimposed onto original pdbfile into current directory
[**-writepdb**]	Write PDB of fragment plus the original structure into current directory

Selection of SLOOP Fragments

[**-minlength** num]	Minimum length of secondary structure to define a loop (default = 2)
[**-combine** num]	Use *num* top SLoop sequence scores (default = 50)
[**-shift** num]	Allow loop length to change by *num* residues when searching (default = 0)
[**-std_dev** num]	Only choose classes with score std_dev >-10.00000

[**-separ** num]	Only choose classes with separation std_dev <999999.125000
[**-angle** num]	Only choose classes with angle std_dev <999999.125000
[**-vector** num]	Only choose fragments which are more than N degrees different 0.100000
[**-anchorrmsd** num]	Only choose fragments which fit better than *num* RMSD to backbone. (default = 9.00 Å)
[**-maxenergy** num]	Only choose fragments which have an energy lower than 9999.0 kcal/mole
[**-edge** num]	Decrease actual secondary structure by *num* residues in energy calculation (default = 0)
[**-allfrag**]	Output information for all selected fragments
[**-nb** num]	Minimum distance allowed for any non-bonded contact. (Expected range 1.9 Å to 2.8 Å, mean 2.5 Å)
[**-ca** num]	Minimum distance allowed for any CA-CA non-bonded contact.(Expected range 3.2 Å to 4.6 Å, mean 3.8 Å)
[**-nb** num]	Minimum distance allowed for any CB-CB non-bonded contact. (Expected range 3.1 Å to 4.0 Å, mean 3.6 Å)

PDB Build Options

[**-bestfit** number]	When calculating loop RMSD, fit predicted loop to actual structure using: • 0—Anchor residues only • 1—Loop residues only • 2—Anchor and loop residues
[**-bbatom**]	Use backbone atoms only in RMSD calculations (default)
[**-caatom**]	Use C-alpha atom only in RMSD calculations
[**-allatom**]	Use all atoms in RMSD calculations
[**-select 0**]	Representative fragment selected on lowest anchor RMSD
[**-select 1**]	Representative fragment selected on lowest energy

Output Options

[**-html** num]	Output HTML page
[**-width** n]	Set alignment width

Secondary Structure Input

[**-extended**]	Use extended K&S definition (flag before the following options)
[**-sst** file]	Input SSTRUC secondary structure file
[**-phd** file]	Input PHD predicted secondary structure file
[**-dsc** file]	Input Dsc predicted secondary structure file
[**-predator** file]	Input Predator predicted secondary structure file
[**-consensus** file]	Input Consensus predicted/altered secondary structure file.
[**-struct** file]	Input secondary structure file via HTML. This file can be produced from the program Pdb2struct using the `-html` flag and simply contains the prediction in line length of 100 characters.

The secondary structure file can be produced from the program Consensus or Pdb2struct and is of the form A21A100:

```
Ident               EEASSTRGNLDVAKLNGDWFSIVVASNKREKIEENGSMRVFMQHIDVLE...
Structure           HHHHH______HHHHHHHHHHHHHH___HHHHH_______HHHHHHHH_...
```

Installation Directories of SLOOP (override environment variables)

[**-rootdir** dir]	Installation directory - overrides everything
[**-meandir** dir]	SLoop mean directory
[**-motdir** dir]	Local mean directory
[**-scoredir** dir]	Score directory
[**-groupdir** dir]	Group directory
[**-templatedir** dir]	Template directory
[**-matdir** dir]	Mat directory
[**-classdir** dir]	Class directory
[**-fragdir** dir]	Fragment directory
[**-pdbext** extension]	PDB file extension
[**-pdburl** url]	URL for `-writepdb`
[**-cgiroot** url]	URL for root of cgi-bin
[**-binpath** path]	Path for the program `sstruc`

Examples:

```
SearchSLoop -pdb 4fgf.pdb -writestruct \
-allseg > 4fgf.struct
SearchSLoop -seq 4fgf.fasta -pdb 4fgf.pdb -allseg
SearchSLoop -pdb 4fgf.pdb -sst 4fgf.sst -allseg -runsloop \
    -bounding -num 3 -combine 5 -std_dev 1.5 -anchorrmsd 2.0
```

4.6 TUNER

Method

Melds two, three-dimensional peptide chains together to make one continuous chain. TUNER incorporates a loop structure into a gap in a protein. Several options exist for incorporating a loop, including those that control restraints on dihedral terms.

Command Format

Each time before you meld with TUNER, you need to:

- Combine the loop PDB file with the model PDB file (`cat`)
- Format the file for tuner (`pretuner`)

```
cat chosenloop.pdb model.pdb |pretuner > combined.pdb
```

Then use TUNER:

```
tuner [options] -pdb combined.pdb -out filename
```

-pdb filename	The concatenated input pdb file
-out filename	Output file
-meld	Adds loop from the CODA output to the model
[**-start** number]	Starting loop residue number
[**-stop** number]	Ending residue number
[**-h**]	Display help
[**-noadjust**]	Only read and write the pdb file
[**-verbose**]	Verbose mode
[**-auto**]	Automatic restraint determination (default). The automatic mode uses the following sequence: -distrong, -diweak, -dino.
[**-distrong**]	Strong restraint on dihedral term; lax distance restraint
[**-diweak**]	Weak restraint on dihedral term, but strong restraint on bond lengths and angles
[**-dino**]	No restraint on dihedral term, but very strong restraint on bond lengths and angles
[**-anchor** number]	Restrict the meld to the specified number of anchor residues (default =3).

TUNER Example

```
cat hs1f0ja342.pdb tuner_out_2.pdb | pretuner \
> tuner_249_6.pdb
tuner -pdb tuner_249_6.pdb -out tuner.pdb -meld
```

4.7 ANDANTE

Method

Adds sidechains to the 3D structure. ANDANTE models sidechain conformations by:

- Borrowing side chain conformations, where appropriate, from parent structures;
- Building remaining sidechains from a high-quality rotamer library, optimizing the packing.

Rules determining when to borrow conformations from parent structures are based on an analysis of substitutions in defined structural environments.

ANDANTE has two distinct modes called borrowing and non-borrowing. The borrowing mode uses information from homologs as a preliminary filter, in order to borrow sidechain conformations that have a high probability of being conserved. Both modes check rotamers for clashes against the backbone, in cycles of larger and larger interacting groups.

Borrowing Mode

ANDANTE searches within the structural alignment for positions that can be described by two chi angles. If a high probability substitution is found between the type and location of a residue position in the model (such as a TRP that is buried and is part of a beta strand) and the type and location of the residue position in the homologs, then the residue is added the model using the exact chi angles that are present in the homologs.

ANDANTE next checks the sidechain conformation for allowable clashes with another borrowed position. The conformation with the higher borrowing score is retained. When ANDANTE has analyzed all positions, the sidechain conformations below the clash thresholds are locked to the backbone.

ANDANTE next searches for positions that are described by one chi angle only. ANDANTE applies the same checks to the backbone and other borrowed positions. If a clash occurs, ANDANTE scores the borrowed chi1+2 positions using a different scoring scheme; it retains the chi1+2 positions over the chi1 conformations.

Non-Borrowing Mode

After borrowing, ANDANTE adds the remaining sidechains to the partially constructed target by a dead-end elimination. It checks for clashes against the backbone, with the borrowed positions, for rotamers for the positions not yet constructed. ANDANTE adds the lowest energy rotamers, for all positions not yet constructed, to the target structure. ANDANTE uses clashes between these starting confirmations to identify interacting groups. These groups are expanded to include all possible clashes with other usable rotamers for positions not in the original interacting group. ANDANTE searches the expanded interacting groups to find the lowest energy conformation possible, with the usable rotamers for each position.

ANDANTE adds sidechains to the partially constructed target by a dead-end elimination. ANDANTE adds the lowest energy rotamers, for all positions not yet constructed, to the target structure. ANDANTE uses clashes between these starting confirmations to identify interacting groups. These groups are expanded to include all possible clashes with other usable rotamers for positions not in the original interacting group. ANDANTE searches the expanded interacting groups to find the lowest energy conformation possible, with the usable rotamers for each position.

Note: If the gap file option of ANDANTE is used, the file needs to contain sequence position numbers that match the `.ali` file, ***not*** the residue numbers from the structure.

Note: ANDANTE defaults to full borrowing and restricting mode if no borrowing/restricting options are entered.

Command Format

`andante`

Input Files

`-i` filename	Required input for borrowing or restricting. The basename of the HOMSTRAD family `.ali` and JOY `.tem` files.
`-pdb` filename	Required input for non-borrowing. The full name of the structure file.
[**`-cm`** filename]	Specify the structure file, overriding the one specified in the `.ali` file.

Borrowing/Non-Borrowing and Restricting Options

Borrowing and restricting may be applied separately or in a number of combinations.

[**`-chi1`**]	Enables borrowing for residues defined by one chi angle: CSTV.
[**`-chi12`**]	Enables borrowing residues defined by two chi angles DFHILNYW.

[**-chi123**]	Enables borrowing residues defined by three chi angles EKMQR.
[**-rchi1**]	Enables the restricting of possible rotamer solutions to specific chi1 bin (all residues except A and G). This is applied to non-borrowed and non-chi1+2 restricted residues.
[**-rchi12**]	Enables the restricting of possible rotamer solutions to specific chi 1+2 bins (DEFHIKLMNQRWY). This is applied to non-borrowed residue positions.
-full	Implement all borrowing and restricting options. Equivalent to `-chi1 -chil12 -chi123 -rchi1 -rchi12`. ***Note***: If you do not enter any borrowing or restriction option, the default behavior assumes `-full`.
[**-fullb**]	Implement all borrowing options, but no restricting options. Equivalent to `-chi1 -chi12 -chi123`.
[**-fullr**]	Implement all restricting options, but no borrowing options. Equivalent to `-rchi1 -rchi12`.

Borrowing and Restricting Thresholds

These options adjust the probabilities used in the borrowing and restricting modes. The threshold value set for a group of residues applies to both borrowing and restricting.

Increase the borrowing threshold when the parent % identity is low. Decrease it when extremely large clusters are generated or when encountering long run time.

[**-min1** number]	Minimum probability for borrowing. This is the probability that chi1 is conserved for a substitution in a specific environment. (Default = 70.0)
[**-min12** number]	Minimum probability for borrowing. This is the probability that sidechains defined by 2 chi angles (DFHILNYW) are conserved for a substitution is a specific environment. (Default = 70.0)
[**-min123** number]	Minimum probability for borrowing. This is the probably that sidechains defined by 3 chi angles are conserved for a substitution is a specific environment (applies to EQM only). (Default = 70.0)

Borrowing Filters

[**-minb** number]	Minimum B-factor of parent sidechain allowed for borrowing. (Default = 50.0)

[**-minres** number]	Minimum resolution (Å) of the parent structure allowed for borrowing/restricting (default = 3.0 Å). This option is off by default, which means that borrowing and restricting will be carried out from all parent listed in the .ali file. This is valid only if the parent structures are all of a reasonable quality. You may not be able to borrow from all parents if they are of low resolution or from NMR structures. If the .ali file does not contain structure information for a parent, this parent is assigned a resolution of -1 and will not borrow from if -minres is enabled.
[**-minpid** number]	Minimum %identity to a parent for borrowing/restricting. This allows to borrow/restrict from higher PID parents without regenerating .ali and .tem files.
[**-maxpid** number]	Maximum %identity to a parent for borrowing/restricting. Use it in combination with -minpid to limit borrowing/restricting to a PID range. (Default = 100.0)

Clash Thresholds

[**-cbb** number]	Maximum backbone clash allowed to use rotamer (default = 12.0 kcal/mole). This reduces the clash threshold tolerated between sidechains and the backbone of the model. Sidechains that exceed this value will not be borrowed from a parent or will be eliminated from the rotamer library solutions for being incompatible with the backbone.
[**-ccc** number]	Maximum clash allowed between sidechains (default = 10.0 kcal/mole). This can be used to perform a more comprehensive search for the global mean energy conformation of the sidechains. For example, a lower value will lower the tolerance for van der Waals overlap compared to that of a higher value, preventing the expansion of clusters to larger sizes. If the run of the program for a model is fast, this value may be lowered to search for sidechain conformation solutions with a lower number of steric clashes.
[**-con** number]	Maximum clash threshold between two borrowed sidechains (default = 10.0 kcal/mole). Some clashes may be slightly over the default threshold, which prevents some positions from being borrowed. Altering this sensibly can result in more borrowed positions being retained.

Clashes against Ligands and Cofactors

[**-hets** filename]	Use this option if clash checks must be performed against atoms that are not part of the target structure. These are typically ligands, and cofactors (HETATM records). These must be stored in a separate file in PDB format and share the same coordinate space as the protein.

Scaling Atom Radii

If clusters grow to impossible sizes, instead of altering clash threshold it may be easier to scale down the atom radii to make the clashes softer.

[**-brad** number]	Scale the atom radii of the backbone atoms (C, N, O) by the specified value (between 0.0 and 1.0).
[**-rrad** number]	Scale the atom radii of the sidechain (rotamer) atoms (C, N, O, S) by the specified value (between 0.0 and 1.0). The default = 0.9, which helps compensate for modeling deviations in the core using the restriction of fixed rotamer conformations.
[**-hrad** number]	Scale the atom radii of the HETATM atoms (identified by the `-hets` option) by the specified value (between 0.0 and 1.0).

Limiting the Number of Possible Rotamer Solutions

The size of clusters may become prohibitively large to allow sufficient sampling. This can be caused by the presence of a large number of K, R or M residues. Such residues, when found in an exposed position, can have a >25 possible library solutions within the clash thresholds. It may be desirable to limit the number of library solutions to enable better sampling during simulated annealing.

[**-maxr** number]	Limit the number of rotamers that will be used as possible solutions for a position within clash thresholds. This option works best for K, M, R, E, Q residues, which have more than 10 rotamers (in the Penultimate Rotamer Library [Ref. 16]). With this option, rotamers are loaded, checked for compatibility with the backbone and then limited to a maximum number.

[**-rotp** probability]	Restrict library solutions to high probability solutions. The default is 1% (0.01 in an allowable range of 0.00 to 1.00). The probability is applied to all residues. This option can be useful to reduce cluster sizes and increase speed, but requires the user to be familiar with the contents of the rotamer library. A better alternative is to use the `-maxr` option.

Rebuilding or Retaining Sidechain Conformations

[**-keep**]	Keep intact all sidechain conformations on the input structure. Notes: • All sidechain conformations on residues with more than 5 sidechain atoms are retained. • Andante automatically rebuild ALA C-beta positions, therefore some slight changes may occur between the input and output structure at these positions.
[**-ucon** filename]	Input a list of conserved positions and chi angles. These positions will then be constructed with the provided chi angles and retained during any searches. In borrowing mode Andante produces two files that can be used to rebuild/retain sidechain conformations in this manner: • andante_all_keep.lst • andante_rebuild_borrowed.lst

The format of the input file for `-ucon` is:

Columns	Content
1-4	residue index (range 0 - max residues)
5-6	spaces
7-10	residue number (e.g. 64)
11	residue insertion code
12-14	spaces (blank)
15	one letter code of conserved residue (e.g. F)

Columns	Content
16-17	spaces (blank)
18+	The chi angles from which the sidechain conformations will be built and retained. The spacing for these values is flexible, but they must be space delimited. There must be 4 values input for each residue. If the residue has only one chi angle, the additional 3 dummy values of 999.00 must be included. Andante looks for 4 values.

Example:

```
16    17V   Y     -60.00    -75.00     999.0     999.0
31    32N   F       75.00     63.00     999.0     999.0
```

Disulfide Bond Building

There are two types of disulfide (C-C) bonds. Those with standard geometry are easy to identify. Others have geometry outside the range of the chi angles used in Andante's default search criteria.

[**-nocc**]	Turn off the function that looks for disulfide bonds and locks them to the backbone. Use this option if Andante creates a disulfide bond, even though there are no disulfide bonds in your model.
[**-ncc1**]	By default Andante will allow clashes between Cys residues that are not already in a disulfide bond pair, but have chi1 angle likely to be conserved. This option turns off this search and does not allow additional disulfide bonds to be built from borrowed positions. At the moment no geometry check is performed on these types of disulfide bonds, and all clashes are tolerated. These types of disulfide bonds are rarely encountered.

Mutation Options

[**-mutate** char number]	Mutate the residue specified by the single character amino acid type and the residue's number. ***Example***: `amdamte96 -pdb 3app.atm -mutate W 165` Mutates residue number 165 to W. The output consists of individual structure files for all library rotamers for the residue mutated.

[**-mutate** char number **-local**]	Mutate the specified residue and attempt to minimize all local clashes. ***Example***: `andante96 -pdb 3app.atm -mutate W 165 - local` Outputs a structure file, andante.atm, of the mutant with the sidechains in the local area around the mutation minimized to reduce clashing.

Simulated Annealing

[**-startT** number]	Starting temperature for the simulating annealing search (default = 1000.0°K)
[**-step** number]	Scaling factor that reduces the temperature variable during the search (default = 0.80 within a range from 0.0 to 1.0). Smaller steps increase the search time.
[**-cycles** number]	Number of random moves to make at each temperature during the search (default = 500000). The program scales down the cycles for smaller clusters (<500000 possible rotamer combinations)

Gap Options

[**-noautogaps**]	Disable automatic gap determination from sequence and model file.
[**-gaps** filename]	If structure has gaps and in borrowing mode, supply list of ali positions (default = out.seq)
[**-gapf** filename]	If structure has gaps and in borrowing mode, user can supply file of gaps list.

Verbose Mode

[**-v**]	Produces a large amount of information about borrowing, rotamer energies, etc.
[**-V**]	Produces a very large amount of information. Useful for debugging purpose only.

Examples of general usage:

Borrowing mode:

```
andante -i serman -chi12 -chi1 -o andante_out -minpid 25 \
-minres 2.7 -minb 40 -ccb 7.0
```

Non-borrowing mode:

```
andante -pdb mystructure.pdb
```

4.8 HARMONY

Method

Use HARMONY to identify the mis-aligned regions in the comparative model.

Note: HARMONY uses **Gnuplot** which must be installed on your system. Linux systems have Gnuplot installed by default, whereas IRIX systems do not. Gnuplot for IRIX is available at http://freeware.sgi.com/

Command Format

HARMONY3

-a filename	Sequence structure alignment file in PIR format
-s filename	Alignment (file) between target sequence and its sequence homologs from BLAST run.
-**m** filename	Text file with the list of models generated from the sequence structure alignment.
[**-c** number]	Cutoff for final score to assign problematic region (default = 0)
[**-n** number]	Number of models to generate (if necessary)
[**-v**]	Verbose mode

Example

```
harmony3 -a choral-andante.ali -m model.lst -s query.blast
```

Limitations

HARMONY uses a position-specific amino acid distribution derived from the sequence homologues of the target. It assess the extra predictive power provided by the structural constraints on amino acid substitutions, compared with the predictive power provided by sequence constraints only.

The observed amino acid distribution at each position of the target is used as an estimate of the amino acid replacements which occurred during the evolution of the family/superfamily. As the sequence homologues are collected in a sequence database search using PSI-BLAST, the accuracy of such an estimate is dependent on the number of sequences collected, the divergence of these sequences, and the accuracy of the alignment between these sequences and the target. This is a major limitation factor for the performance of HARMONY. If too few sequence homologues are collected, or the collected homologues are too close to each other in the sequence space, or the PSI-BLAST alignment is of extremely poor quality, the substitution score becomes less reliable, which in turn influences the final evaluation score. However, with the exponentially growing amount of sequence data arising from various genome-sequencing project, such a limitation should become less significant over time.

HARMONY Output

HARMONY outputs a number of graphs calculated using a variety of scoring methods:

- Propensity score (_comp_sin.png plot)
- Substitution score using “sum of vectors” method (harmony3_comp_sum.png plot)
- Substitution score using “city block” method (harmony3_comp_city.png plot)
- Substitution score using “chi square” method (harmony3_com_chi2.png plot)

Interpreting the HARMONY plot

The reliability of the HARMONY run depends on the sequence divergence for the alignment between the target sequence and its homologs. This value and its rating is reported on the upper-right corner of the graph (SeqDiv). The values range from 0 to 1. A higher value indicates that more sequence information is available to HARMONY, and thus more reliable the result is. The ratings can be Poor, Medium, or Good.

The X-axis of a HARMONY plot denotes the residue number, while the Y-axis denotes the HARMONY score. Three scores are color coded and plotted for each residue:

- environment independent (uni)
- environment dependent (env)
- summary

Ideally, the summary result (line on the top plot) should be above the zero HARMONY score line. Anything below this line should be examined closely.

The environment independent (uni) values are the substitution scores taken from environment independent substitution tables (EIST). These tables are constructed from the merging of all the environment specific substitution tables (ESST) that FUGUE uses (analogous to a BLOSUM substitution table). The final score for a residue is averaged over a sliding window of 7 residues.

The environment dependent (env) values are the substitution scores taken from environment dependent substitution tables (EDST). The values of the environment substitution scores are given in the file `prop.h`. The final score for a residue is averaged over a sliding window of 7 residues.

The “uni” and “env” lines are the scores without local alignment flexibility adjustment. They represent the contributions from propensity-based scores and substitution-based scores. The summary line is the difference between “uni” and “env” plus the contribution from local flexible alignment.

If the “env” line remains above the “uni”, this indicates that the residue is found in a more preferential environment and that the modeling in that part of the alignment is likely to be correct.

4.9 ORCHESTRAR Input and Output Files

Most ORCHESTRAR steps have multiple input and output files. All these files are in ASCII format.

Program	**Input Files**	**Output Files**
BATON	set of protein structure files .atm (default) or .pdb	.brk (set of aligned protein structure files)
CHORAL	.ali file and .brk files	• CORE0x.pdb • choral-andante.ali • choral-bridge.pir • out01.seq
BRIDGE	bridge_config core.pdb choral-bridge.pir	bridge.pdb
FREAD	tuner.pdb or bridge.pdb or other-pdb-with-gaps.pdb	• xxxx.pdb (multiple files) • .log from FREAD
Read7	tuner.pdb or bridge.pdb or other-pdb-with-gaps.pdb	.log from Read7
MINUS	.log from Read	• .log from MINUS • xxxx.pdb (multiple files)
CODA	.log from FREAD and .log from MINUS	.log from CODA
TUNER	tuner.pdb (concatenated loop and backbone file)	melded.pdb (structure with added loop)

Program	**Input Files**	**Output Files**
ANDANTE	borrowing: • alignment.ali and associated files • alignment.tem • model.atm • homologs.atm (multiples) non-borrowing: structure.pdb	• .pdb(backbone with added sidechains • conserved.log (List of the positions that have been borrowed from the parents. Includes parent borrowed from, borrowing score for that position, and chi angles used.) • restricted.log (List of the positions that have had rotamer solutions “restricted” to specific chi bins. Includes parent restricted from, borrowing score for that position, and chi angles used.) • andante_all_keep.lst (List of the chi angels of the final model generated by ANDANTE. ***Tip***: This file can be modified and used as input to build customized rotamer orientations. All orientations contained in the file will be built and retained during side chain placement. • andante_rebuild_borrowed.lst(List of the chi angles of the borrowed positions only. ***Tip***: This file can be modified and used as input to build customized rotamer orientations. All orientations contained in the file will be built and retained during side chain placement.

Program	**Input Files**	**Output Files**
HARMONY	• harmony3.blast • alignment.ali and associated files • model.atm • model.lst	• harmony3_comp_sin.png (propensity score plot) • harmony3_comp_sum.png (substitution score using “sum of vectors” method plot) • harmony3_comp_city.png (substitution score using “city block” method plot) • harmony3_com_chi2.png (substitution score using “chi square” method plot)

This page intentionally blank.

Chapter 5.

ORCHESTRAR Literature References

[1] Zhu Z.Y., Sali, A. and Blundell, T.L. “A variable gap penalty function and feature weights for protein 3-d structure comparisons.” *Protein Eng.*, **1992**, *5*, 43-51.

[2] Montalvao, R.W.; Smith R.E.; Lovell, S.C.; and Blundell, T.L. “CHORAL: A differential geometry approach to the prediction of the cores of protein structures.” *Bioinformatics*, **2005**, *21*, 3719-3725.

[3] Deane, C.M and Blundell, T.L. “CODA: A combined algorithm for predicting the structurally variable regions of protein models.” *Protein Sci.*, **2001**, *10*, 599-612.

[4] Deane, C.M. and Blundell, T.L. “A Novel Exhaustive Search Algorithm for Predicting the Conformation of Polypeptide Segments in Proteins.” *Proteins: Structure, Function and Genetics*, **2000**, *40*,135-144.

[5] Burke, D.F., Deane, C. M. and Blundell, T. “Browsing the SLoop database of structurally classified loops connecting elements of protein secondary structure.” *Bioinformatics*, **2000**, *16*, 513-519.

[6] Shi, J; Blundell, T. L.; Mizuguchi, K. “FUGUE: Sequence-structure Homology Recognition Using Environment-specific Substitution Tables and Structure-dependent Gap Penalties.” *J. Mol. Biol.*, **2001**, *310*, 243-257.

[7] Mizuguchi, K.; Deane, C.M.; Blundell, T.L.; Overington, J.P., “HOMSTRAD: a database of protein structure alignments for homologous families.” *Protein Sci.*, **1998**, *7*, 2469-2471.

[8] de Bakker, P.I.W.; Bateman, A.; Burke, D. F.; Miguel, R.N.; Mizuguchi, K; Shi, J.; Shirai, H.; and Blundell, T.L., “HOMSTRAD: adding sequence information to structure-based alignments of homologous protein families.” *Bioinformatics*, **2001**, *7*, 748-749.

[9] Stebbings, L.A.; Mizuguchi, K., “HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database” *Nucleic Acids Research*, **2004**, *32*, D203-D207.

[10] Deane, C.M. Kass, Quentin, and Blundell, T.L. “SCORE: predicting the core of protein models.” *Bioinformatics*, **2001**, *17*, 541-550.

[11] Mizuguchi, K.; Deane, C.M.; Blundell, T.L.; Johnson, M.S.; Overington, J.P. “JOY: protein sequence-structure representation and analysis.” *Bioinformatics*, **1998**, *14*, 617-623.

[12] http://www-cryst.bioc.cam.ac.uk/~joy/

[13] Williams, M.G., Shirai, H., Shi J., Nagendra, H.G., Mueller, J., Mizuguchi, K., Miguel, R.N., Lovell, S.C., Innis, C.A., Deane, C.M., Chen, L., Campillo, N., Burke, D.F., Blundell T.L., and de Bakker P.I. “Sequence-structure homology recognition by iterative alignment refinement and comparative modeling.” *Proteins. Suppl.*, **2001**, *5*, 92-97.

[14] Sali, A. and Blundell, T.L. “Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming.” *J. Mol. Biol.*, **1990**, *212*, 403-28.

[15] Bower, M. J.; Cohen, F.E.; Dunbrack, R.L. “Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool.” *J. Mol. Biol.*, **1997**, 267, 1268-82.

[16] Lovell S.C.; Word, J.M. et al. “The Penultimate Rotamer Library” *Proteins: Structure, Function and Genetics*, **2000**, *40*, 389-408.

[17] Dunbrack, R. L., Jr. and Karplus, M. “Backbone-dependent rotamer library for proteins. Application to side-chain prediction.” *J. Mol. Biol.*, **1993**, 230, 543-74.

[18] Ogata, K. and Umeyama, H. “The role played by environmental residues on sidechain torsional angles within homologous families of proteins: A new method of sidechain modeling.” *Proteins-Structure Function and Bioinformatics*, **1998**, 31, 355-369.

[19] Sowdhamini, R., Srinivasan, N. et al. “Stereochemical modeling of disulfide bridges. Criteria for introduction into proteins by site-directed mutagenesis.” *Protein Eng.*, **1989**, 3, 95-103.

[20] Summers, N. L. and Karplus, M. “Construction of side-chains in homology modelling. Application to the C-terminal lobe of rhizopuspepsin.” *J. Mol. Biol.*, **1989**, 210, 785-811.

[21] Summers, N. L. and Karplus, M. “Modeling of side chains, loops, and insertions in proteins.” *Methods Enzymol.*, **1991**, 202, 156-204.

ORCHESTRAR Index