

MacroModel 9.7

XCluster Manual

MacroModel XCluster Manual Copyright © 2009 Schrödinger, LLC. All rights reserved.

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

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

Schrödinger software includes software and libraries provided by third parties. For details of the copyrights, and terms and conditions associated with such included third party software, see the Legal Notices for Third-Party Software in your product installation at `$(SCHRODINGER)/docs/html/third_party_legal.html` (Linux OS) or `%SCHRODINGER%\docs\html\third_party_legal.html` (Windows OS).

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

June 2009

Contents

Document Conventions	vii
Chapter 1: What It's For	1
1.1 Filtering and Clustering	1
1.2 Distance Criteria	2
1.3 Batch or Interactive Use	3
1.4 Running Schrödinger Software	3
1.5 Citing XCluster in Publications	3
Chapter 2: What It Does	5
2.1 Clustering	5
2.2 Threshold Distance	6
2.3 Critical Threshold Distance	6
2.4 Clustering Level	7
2.5 Generic Order	7
2.6 Choosing a Clustering	7
2.7 Visualizing Clustered Conformations	8
2.8 Visualizing Representative Structures	8
Chapter 3: Using XCluster	9
3.1 Cluster and XCluster	9
3.2 Command-line Options	9
3.3 The Main Window	10
3.4 The File Menu	10
3.5 The Commands Panel	12
3.5.1 Control Region	13
3.5.2 File Specification Region	13
3.5.3 Command Specification Region	14

3.6 The Visualization Menu	17
3.6.1 Clustering Statistics	17
3.6.2 Distance Map	18
3.6.3 Clustering Mosaic	20
3.6.4 Cluster Membership	22
3.7 The Help Menu	22
Chapter 4: Using Cluster	23
4.1 Batch Interface	23
4.1.1 Command-line Options	23
4.1.2 File Conventions	23
4.1.3 Command File	24
4.2 Comment Insertion Command	24
4.3 Input File Specification Commands	24
4.4 Symmetry Specification Commands	25
4.5 Distance Matrix Generation Commands	27
4.6 Clustering Commands	29
4.7 File Output Commands	29
Chapter 5: Figures of Merit	35
5.1 Separation Ratio	35
5.2 Effective Number of Clusters	36
5.3 Reordering Entropy	37
Chapter 6: How It Works	39
6.1 Generic Conformation Order	39
6.2 Three-dimensional Superposition	40
Chapter 7: Examples	43
7.1 Two-dimensional Examples	43

7.1.1 Example 2D-1	44
7.1.2 Example 2D-2	49
7.1.3 Example 2D-3	53
7.1.4 Example 2D-4	56
7.2 Conformational Search Examples	59
7.2.1 Roseotoxin-B, a Cyclic Peptide	59
7.2.2 Cycloheptadecane	64
7.3 Molecular Dynamics of Pentane	67
Chapter 8: Command Reference	73
Chapter 9: X Resources	77
Getting Help	79
Index	83

Document Conventions

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

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

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

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

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

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

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

What It's For

Computational techniques such as molecular dynamics, Monte-Carlo sampling, and conformational search methods produce large numbers of conformations of a given chemical structure. Usually these techniques are used to find a particular quantity of interest—for example, the lowest energy conformation, or the free energy of the system. There are occasions when it is useful to know how the conformations obtained are related to one another: for example, whether they fall into structural classes, and, if so, the nature of the classes.

XCluster is a tool for observing, analyzing and visualizing the clustering of molecular conformations, given any of several commonly used criteria of molecular similarity. It may be also used to observe, analyze and visualize clustering in arbitrary user-supplied pairwise distance data.

1.1 Filtering and Clustering

One method that can be used for this purpose is filtering. Filtering screens the conformations against one or more external criteria. For each criterion, the user provides a variable, a target value, and a range around this target, and the filter returns those structures for which the variables lie within range of the target. In order to use such a method, a user has to have a given variable, target and range in mind. For example, suppose the user has generated many conformations of molecule *A*, and he then wants to know which of the conformations exhibit geometries similar to that of some other molecule, *B*. He might specify a list of atoms in *A* and a corresponding list in *B*, and direct the filtering program to return those conformations of *A* which are within, say, 0.25 Å R.M.S. atomic displacement of *B*, following best rigid-body superposition of the comparison atoms of the two molecules. Here the variable is R.M.S. atomic displacement following rigid-body superposition, the target is 0.0 Å with respect to *B*, and the range is 0.25 Å. In order to use such a method, the user has to have a reason for wanting *A* to look like *B*—that is, he has to supply not only a variable, but also a target and a range.

In contrast, a user might want to ask whether the conformations of *A* form natural groupings amongst themselves; whether the conformations occur in structurally related clusters, or whether, on the other hand, they are distributed in a continuous manner in some sense. The purpose of XCluster is to investigate questions of this nature. Given as input a data file containing a list of conformations of a molecule in MacroModel or Maestro format, the program searches for clusters of structures based on one of several distance criteria which the

user may specify, as described in the next section. Any structural output created by the program is in the same format as the input structures.

1.2 Distance Criteria

- Atomic R.M.S. displacement between pairs of structures following rigid-body superposition.
- Atomic R.M.S. displacement between pairs of structures without rigid-body superposition.
- R.M.S. difference between corresponding torsion angles in pairs of structures.

What these criteria have in common is that they all can be used to specify how different two structures are. This difference can be thought of as a conformational distance d_{ij} between a pair of structures i and j .¹ The method used by XCluster starts by constructing the distance matrix, **D**, whose elements are d_{ij} . There is provision in XCluster for specifying which atoms or torsions are to be included in the distance calculation, and for specifying any symmetry operations that should be performed in the course of comparison.

XCluster can also be used in *dfile mode* to perform cluster analysis of a distance matrix supplied by the user. Thus, it can be used to examine clustering based on criteria other than the ones listed above. The user-supplied distances need not be derived from an ensemble of molecular conformations: XCluster could be used to analyze the clustering of dandelions in a field, mushrooms in a wood, craters on the moon, or cells on a substrate. Although all the measures XCluster calculates satisfy the triangle inequality,² *dfile mode* works for “distance” measures that do not.

Once XCluster has created or read in the distance matrix, it searches **D** for clusters. In general, if there are N items of data (e.g., molecular conformations), XCluster finds N *clusterings*, or ways of placing the items into clusters. These are indexed by *clustering level*, an integer which runs from 1 through N . The user can then examine various figures of merit or any of several visual representations of the distance information to determine which of the clusterings, if any, seem especially interesting. XCluster also provides the capability of writing out structural data for clusters at any clustering level, appropriately superimposed and colored by cluster.

1. When torsional angles are being compared our use of the term *conformational distance* corresponds to that of Saunders. (Saunders, M. J. *Comput. Chem.* **1991**, *12*, 645.)

2. The triangle inequality states that for any three points, i , j and k , $d_{ik} \leq d_{ij} + d_{jk}$; strictly speaking, a measure that does not satisfy this relationship cannot be called a distance.

1.3 Batch or Interactive Use

XCluster can be run as a batch program (`cluster`) or interactively from an X Windows interface (`xcluster`). XCluster calculations can also be set up and run from the XCluster panel in Maestro, using the picking tools to set up atoms or torsions for comparison.

1.4 Running Schrödinger Software

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

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

Once you have set the `SCHRODINGER` environment variable, you can start Maestro with the following command:

```
$$SCHRODINGER/maestro &
```

It is usually a good idea to change to the desired working directory before starting Maestro. This directory then becomes Maestro's working directory. For more information on starting Maestro, including starting Maestro on a Windows platform, see [Section 2.1](#) of the *Maestro User Manual*.

1.5 Citing XCluster in Publications

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

MacroModel XCluster, version 9.7, Schrödinger, LLC, New York, NY, 2009.

What It Does

This chapter describes the basic ideas and definitions necessary to understand what XCluster does, and to place our notion of clustering in the context of others' uses of the term. We describe several subtopics in greater detail in later sections.

2.1 Clustering

There is no universally agreed upon definition of *cluster* or *clustering*³, even among statisticians. The way we use the term, a clustering is first of all a partitioning of a set of points (molecular conformations) into subsets called clusters. The term “partitioning” implies that the subsets are mutually exclusive and that they cover the set, so that in any clustering, every point (conformation) is a member of exactly one cluster.

Our scheme of clusterings is *hierarchical*, meaning that clusters at a high clustering level are unions of clusters which form at lower levels, as described below. Clusters never break apart during the formation process, which can thus be described as one of “successive agglomeration.”

The clustering algorithms used in XCluster are “exact,” which means that they are based on the full distance matrix. The run time thus tends to go as N^2 , where N is the number of points (conformations). Approximate methods become necessary for data sets containing greater than about 10^4 points, and such methods are usually employed in the cluster analyses performed by statistical software packages such as SPSS and SAS.

For each of our distance measures, all the coordinate axes in the corresponding space are commensurate; that is, in each of our conformational spaces, all the axes have the same units. This obviates the need for the rescaling of variables that generally precedes cluster analysis in a space of statistical parameters. On the other hand, XCluster incorporates algorithms that properly treat several aspects of the molecular clustering problem that are not usually encountered in clustering problems of a more general nature; namely, the periodic nature of torsional space, the possible existence of molecular symmetry, and the need to perform three-dimensional superposition in order to display clustered conformations in their optimal mutual orientations. XCluster also computes several statistics which we believe to be novel.

3. Two references that describe cluster analysis from a more general perspective than ours are: Sokol, R. R. Clustering and Classification: Background and Current Directions. In *Classification and Clustering*; Van Ryzin, J., Ed.; Academic Press: New York, 1977. Zupan, J. Clustering of Data. *Algorithms for Chemists*, John Wiley and Sons: New York, 1989; Chapter 7. The first of these is an especially readable introduction.

2.2 Threshold Distance

Let us assume that we have a list of N items, and have calculated the distances between every pair chosen from the list. These distances, d_{ij} , form a symmetric matrix, since $d_{ij} = d_{ji}$, and, since $d_{ii} = 0$, the main diagonal of the matrix is zero. If the data supplied to XCluster are in an input file of molecules in MacroModel or Maestro format, the program calculates the distance matrix, using a user-supplied choice from among several definitions of “distance;” alternatively, the user may precalculate a distance matrix and supply it directly to the program as a list of the $N(N-1)/2$ non-redundant off-diagonal elements.

Let T be some distance, which we will call a “threshold distance.” Given a value of T , we define two items, i and j , to be in the same cluster if $d_{ij} \leq T$. Thus if items a and b are closer together than T and items b and c are closer together than T , then a , b and c will be in the same cluster even if $d_{ac} > T$.

Clearly, if $T \leq \text{Min}(d_{ij})$, then there are N clusters, each consisting of a single item. If $T > \text{Max}(d_{ij})$, then there is a single cluster that contains all N items. In practice, a single cluster is formed long before T approaches $\text{Max}(d_{ij})$, because of the “chaining together” of items into the same cluster described in the last paragraph. As T is increased through the range of the d_{ij} , clusters agglomerate. Suppose, for example, that clusters E and F (and possibly others) are present at a particular value of T , and that e and f are specific items belonging to E and F . Since E and F are distinct clusters, we know that $d_{ef} > T$. Suppose that upon increasing T by a small amount we find that now $d_{ef} \leq T$. Under the new value of T , E and F are no longer distinct clusters; all their members become part of a single cluster.

2.3 Critical Threshold Distance

Imagine now that we have a sorted list of the d_{ij} . As we set T successively to each value on this list, at certain values an agglomeration will occur. If, however, T is increased to a new value of d_{ij} such that i and j are already in the same cluster, no agglomeration will, in general, occur. An example is provided by the three items a , b and c of several paragraphs ago. a , b and c were in the same cluster even though d_{ac} exceeded T . If T is now increased to d_{ac} , this cluster is unaffected.

It is typical for the $N(N-1)/2$ values of d_{ij} to be unique, and for none to be identically zero (so that no item is a precise duplicate of another); in this situation, each time T is advanced to the next value of d_{ij} on the sorted list, either the number of clusters decreases by one (two clusters agglomerate), or nothing happens. There are then $(N-1)$ “critical threshold distances”; these are the values of T where agglomeration takes place. The reason there are $(N-1)$ such values is that we start with N isolated data values. After $(N-1)$ pairwise agglomerations they are all contained in a single cluster, so that no further agglomeration can occur.

2.4 Clustering Level

Based on our definition of a cluster, the smallest critical threshold distance is $\text{Min}(d_{ij})$, where the first pair of items comes together to form a cluster. Each critical threshold distance is associated with a “clustering,” which is a list of clusters, together with the lists of the items in each cluster. Let us now extend the definition of critical threshold distance to allow the value 0.0; then there are N clusterings, including the one in which each item exists as a separate cluster. In the XCluster program, these N values are indexed from 1, where each item is a separate cluster, to N , where a single cluster, composed of all the items, first appears. At clustering level i , there are $(N-i+1)$ clusters.

2.5 Generic Order

It turns out that using the definitions of a clustering and of a cluster given above, it is possible, given the pairwise distances, to reorder the list of data points (conformations) into a new sequence such that at any clustering level, all points belonging to the same cluster will lie in a contiguous block. We call such a reordering (which, incidentally, is not unique), a “generic” ordering of the points. The method XCluster uses to compute such a reordering is described in [Chapter 6](#). The utility of the generic ordering is best appreciated by consulting [Chapter 7](#).

For now, note two things about a list of conformations in generic order. First, at clustering level 1, where each item is in a cluster by itself, the list may be thought of as a series of N bins separated by dividers, or partitions. When the clustering level goes to 2, a single pair of points joins into a cluster, and this corresponds to removing a single divider. At each increase of clustering level, another divider is removed. When all $N-1$ dividers have been removed, we are at clustering level N , and all points are in a single cluster.

Second, the reordered list has the property that items close to each other in sequence tend to have short distances between them. What we mean by “tend to” is that the above statement is far truer for a generically reordered list than for a randomly ordered list, provided that the data in fact exhibit a significant degree of clustering.

2.6 Choosing a Clustering

Since there are many clusterings to choose from, the question arises, “Which of them—if any—are interesting or significant?” There are two ways to approach this question. The first is to use past experience. If you have found that conformations within, say, some particular R.M.S. atomic displacement value of each other tend to have properties that are similar in some way important to you, then you have every reason to examine the clusters that appear at this (or the next lower) value of the critical threshold distance.

On the other hand, it is natural to ask whether the data naturally “clump” into especially “good” clusters at some point. We have devised several statistics that are functions of the clustering and which serve as figures of merit for extent of clustering. It appears that at least one of them, the “separation ratio,” correctly identifies situations in which all clusters are clearly defined and well separated. Choosing Cluster Statistics from the Visualize menu of XCluster displays these measures as functions of either clustering level or critical threshold distance. Numerical values of these statistics also appear in tabular form in the XCluster log file.

Choosing Distance Map from the Visualize menu displays a visual representation of the full distance matrix in either the input order or the generic order. This is an aid to selecting a clustering, and also a means of visualizing the data all at once. In certain situations, comparing the appearance of the distance map in the two orderings clarifies what the clusters mean. Choosing Clustering Mosaic from the Visualize menu displays a diagram that shows how the clusters evolve with increasing clustering level and threshold distance. This display is similar to the “dendritogram” displays sometimes used in cluster analysis. Clicking on either the map or the mosaic identifies particular conformations and their properties.

2.7 Visualizing Clustered Conformations

Choosing Cluster Membership from the Visualize menu allows you to list the members of each cluster at any clustering level. If the data were supplied to XCluster as a file of molecular conformations, you can write out the conformations by choosing Write from the File menu, then selecting Cluster File. The conformations are optimally superimposed, if appropriate, and colored by cluster for viewing.

2.8 Visualizing Representative Structures

You can write a single structure for each cluster in the current clustering to the specified output file by choosing Write from the File menu and then choosing Representative Structures. The structure is chosen which best resembles the “average” geometry for each cluster. See [Writerep](#): on [page 30](#) for details.

Using XCluster

3.1 Cluster and XCluster

There are two interfaces to the clustering program, Cluster and XCluster:

- The command-line interface, Cluster, is a non-interactive (batch) program that operates by reading instructions from a command file and writing output to a log file. This interface is described in [Chapter 4](#).
- The graphical interface, XCluster, is an interactive version of the program that is run from an X terminal or workstation. It provides graphical methods for visualizing the results of clustering operations. XCluster can also be used to visualize the output of Cluster, provided that the necessary commands have been placed into the command file to allow subsequent reading by XCluster.

Both Cluster and XCluster read and write structure files. These files can be in either MacroModel or Maestro format. The program writes its output structure files in whichever format was used to supply the structural input.

In addition to these two interfaces, Maestro's XCluster panel (opened from the Applications menu) provides a very convenient way to select the atoms or torsions used for clustering and run XCluster. See the *MacroModel User Manual* for more information.

3.2 Command-line Options

XCluster can be started from the command line by entering the command `$SCHRODINGER/xcluster`. The syntax of the `xcluster` command is as follows:

```
xcluster [options] [jobname]
```

- *jobname* specifies a command file to be read at start-up. For example, the command `xcluster roseb` starts XCluster, attempts to read the command file `roseb.clu`, and if it finds it, places the user directly into the Command File Editor, whose display is initialized using the contents of this file.

Options

- `-d displayname` specifies that the XCluster window should appear in the location specified by *displayname*.

- `-F` uses full structures in the output structure files. In general, MacroModel uses compressed structures to save disk space. The first structure is always a full structure and the following structures are compressed. In order to work properly, these partial structures must be preceded by a full structure. This option specifies that all structures are full structures.
- `-h` displays help message.
- `-help` displays help message.
- `-HELP` displays help message.
- `-r` reads the *jobname.clu* command file and immediately runs a cluster calculation on the existing input data. This option is used by Maestro to run XCluster and permit analysis of the output using the XCluster GUI.
- `-v` (verbose) generates copious output to the log file; this output is useful mainly for debugging the XCluster program.

3.3 The Main Window

When you type `xcluster`, a window is displayed which contains the major controls for the interactive version of the program. There are three areas in the window.

File Menu. This menu appears at the top of the Main Window, and is used to control the subsequent actions of the program.

Message Area. This is a scrollable text region in the middle of the Main Window. It is used to display error messages and informative output from the program as it executes.

Progress Area. This area displays a dynamic indication of the progress of potentially time-consuming operations, such as the reading and writing of large structure files and the calculation of distance matrices. Occasionally, prompts to the user also appear here.

If your X resources are set up properly, the menu bar is darker in color than the message and progress areas. Throughout the program, areas which are “read-only” to the user are light in color, whereas items which users can alter or interact with, such as push-buttons and editable text windows, are somewhat darker.

3.4 The File Menu

New

Create a new (blank) command file.

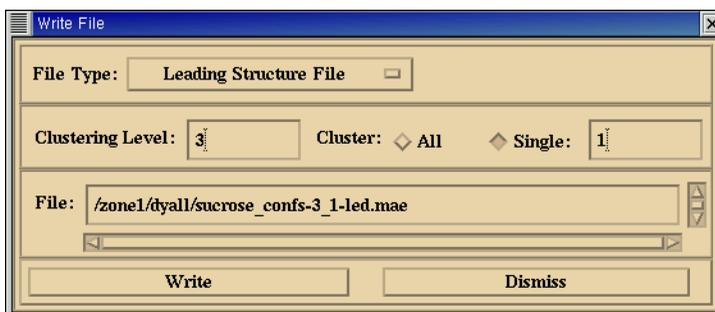


Figure 3.1. The Write File panel.

Open

Open an existing command file for editing or execution. The most common means of initiating an XCluster job is to edit an existing command file and save it under a new name.

Write

Open a panel which allows for writing of a file in one of the six following formats: Cluster File—a file which contains the structures agglomerated into clusters; Representative Structure File—a file containing a single representative structure for each cluster; Average Structure File—a file containing a single structure for each cluster obtained by simple averaging of the coordinates of the cluster; Distance Matrix File—a text file containing the distance matrix values in input order; Generic Distance File—a text file of the distance matrix values in generic order; Leading Structure File—a file containing the structure that appears first in the input file for each cluster (which is the lowest energy member in a cluster if the input came from a conformational search, for example). The file type is determined by the option menu at the top of the panel.

For each of the cluster files, the All and Single buttons can be used to determine whether all or just a single cluster are to be written to the file.

This panel is persistent and its fields can be set either by manual entry or by interaction with the Mosaic, Map or Clustering Statistics displays. This panel may be opened by buttons on these displays as well as by choosing Write from the File menu.

Quit

Leave the XCluster program.

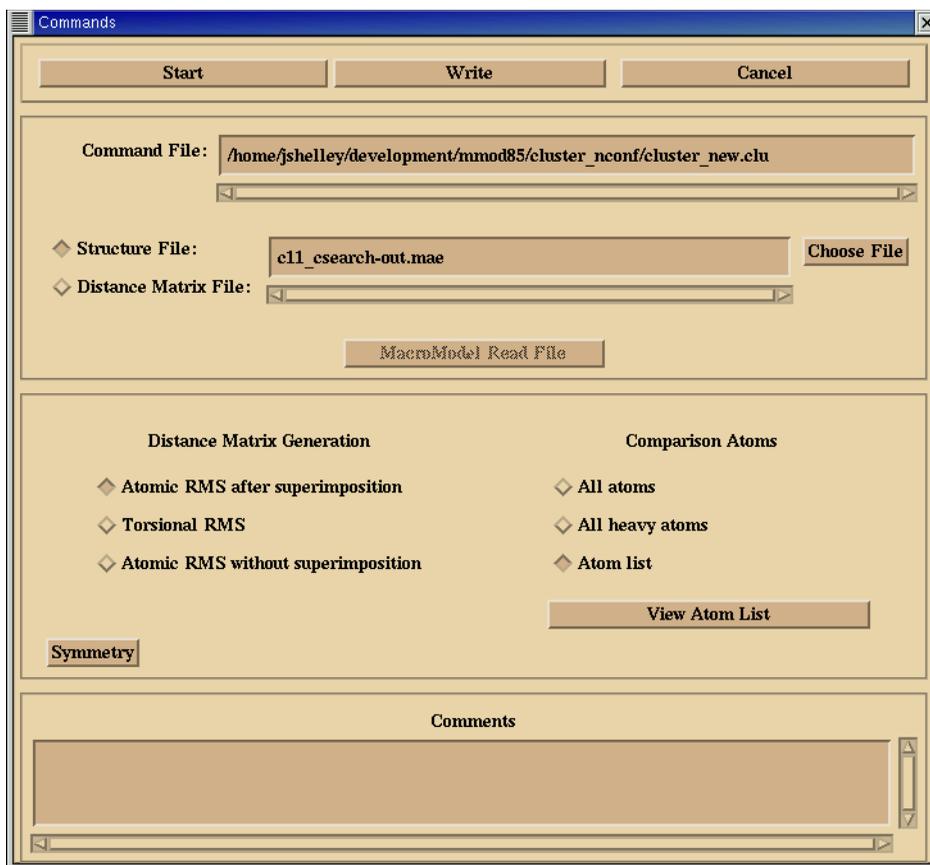


Figure 3.2. The Commands panel.

3.5 The Commands Panel

Choosing the New or Open options from the File menu will bring up the Commands panel. This panel is shown below. If the panel is accessed from the New menu selection it will be displayed initially with all the values set to defaults. If it is accessed from the Open menu selection then the values will be set from the contents of the particular command file selected.

When a set of instructions is given to XCluster in the Commands panel, XCluster writes a command file containing the instructions. The command file can be written and immediately executed by clicking Start in the Commands panel, or it can be saved for later use, perhaps in a Cluster batch job, by clicking Write. In either situation, both the command file and a log file are present after the file is executed. With few exceptions, these files are identical whether the job was performed by XCluster or by Cluster. Thus, there is nearly a one-to-one mapping between

XCluster Commands panel functions, described here, and Cluster batch instructions, described below.

There are four areas in the Commands panel. The Control Region, at the top, contains three buttons which control the execution of the command file. Below this is a File Specification Region, which is used to specify the name of the files which XCluster is to use. When the XCluster graphical user interface is used, the program writes this command file based on user entries in the Commands window. Below this is a Command Specification Region, which is used to specify which sort of molecular comparison is to be used. Finally, there is a Comments region, which allows comments to be inserted into the XCluster command file.

3.5.1 Control Region

Start Button

Write the current settings to the command file and begin the job. The command file is overwritten unless its name is changed; a dialog box appears to warn you about an impending overwrite. The Start button is unavailable if the Structure File or Data File text fields are empty, or if the Atom List or Torsion List panel is active.

Write Button

Write commands corresponding to the current Commands panel settings to the command file. The command file will be overwritten (again, with prior warning by means of a dialog box) if one of the same name already exists. Write will be enabled whenever Start is enabled.

Cancel Button

Close the Command Editor panel without making any changes to the command file.

3.5.2 File Specification Region

Command-File Text Field

This field holds the name of the selected command file.

Input Data File Text Field

This can be either a structure file or a file containing user-supplied distance data (for dfile mode). You must first select either the Structure File or Distance Matrix File radio button. Then you may either type the file name into the text window or select an existing file by clicking the Choose File button, to the right of the text field. This brings up the standard Motif file-selection box. The file names listed are those that have the default suffix for the distance mode chosen—

.mae, .out, or .dat for a structure file and .dst for a distance file—but the user may override these by resetting the search string in the top text field and then clicking the Filter button.

3.5.3 Command Specification Region

This region is active only if the Structure File button is enabled; it has no meaning in dfile mode. It is used to select among the various distance criteria known to the program.

Distance Matrix Generation Buttons

On the left is a set of three Distance Matrix Generation buttons which allow the user to select one of the three distance matrix methods, Atomic RMS after superposition, Torsional RMS and Atomic RMS without superposition.

Comparison Atoms Buttons

To the right of these is a set of Comparison Atoms buttons that are used to specify the atoms used in distance calculations. For the Atomic RMS methods the choices are All atoms, All heavy atoms (i.e., all atoms that are neither hydrogens nor lone pairs), and an explicit Atom list. When the distance matrix method is set to Torsional RMS, the user must specify an explicit list of torsions. If Atom list or Torsional RMS is selected, a View Atom List or View Torsion List button becomes active. These open the RMS Atom Picker and RMS Torsion Picker panels, respectively.

RMS Atom Picker Panel

This panel is used to build a list of comparison atoms. The atom numbers to be added to the list must be entered from the keyboard. To add atoms from the keyboard, click in the text entry region at the top of the panel and enter numbers, pressing the RETURN key after every entry. The numbers appear in the scrollable list area on the right of the panel.



Figure 3.3. The RMS Atom Picker panel.

The atom picker checks for duplicates. If an atom is reselected, a beep sounds and a warning message appears in the XCluster message area. No checking is done, however, to ensure that atom numbers entered are consistent with the current structure.

The Delete button deletes the currently selected entry in the list; you can select an entry by scrolling to it and clicking on it. The Clear button clears all the entries in the list. The Cancel button closes the panel, disregarding any changes made since it was opened. The Done button closes the panel, incorporating the user's changes.

Entering atom numbers from the keyboard can be time consuming or impractical. As an alternative, you can use the XCluster panel in Maestro to set up and run the entire XCluster calculation. This alternative allows you to select the comparison atoms with Maestro's powerful atom selection tools.

RMS Torsion Picker Panel

The operation of the RMS Torsion Picker is similar to that of the RMS Atom Picker. The chief difference, of course, is that torsions are specified by four-atom sets. The atom numbers must be entered four at a time, separated by spaces.

No duplicate torsions are allowed. If a duplicate torsion is specified a beep will sound and a warning message will be placed in the main message area. As with the atom list, no checking is done to ensure that atom numbers or connectivities entered from the keyboard in fact exist in the structure named in the command panel. The four atoms defining a torsion need not correspond to a bonded set of atoms.

The Delete button deletes the currently selected entry in the list, and the Clear button deletes all the entries. The RMS Torsion Picker panel can be dismissed with either the Cancel button, which closes the panel disregarding any changes made since it was last opened, or the Done button, which incorporates the changes.



Figure 3.4. The RMS Torsion Picker panel.

Entering atom numbers from the keyboard can be time consuming or impractical. As an alternative, you can use the XCluster panel in Maestro to set up and run the entire XCluster calculation. This alternative allows you to select the torsions with Maestro's powerful atom selection tools.

Symmetry Panel

Use Enantiomers for Comparison, if selected, makes XCluster treat enantiomers as equivalent.

Automatically perceive symmetry, if selected, turns on the mmsym facility, which recognizes and handles local and global number-order symmetry. If not selected, user options to specify global symmetry (only) become available. These options are described in [Section 4.4 on page 25](#). The Numbering system reflection buttons add the `Reflect:` command if Ring or Chain is chosen. The Rotate numbering system by text window adds the `Rotate:` command if Ring has been selected. The Symmetry Atoms button allows a list of atoms to be specified to the `Symatom:` command.

Comment Window

This is a scrollable and editable text area which displays user comments associated with the current command file. Comments can be of any length and are written to the log file as well as the command file. The comments reappear in the Comment window if the command file is reopened.

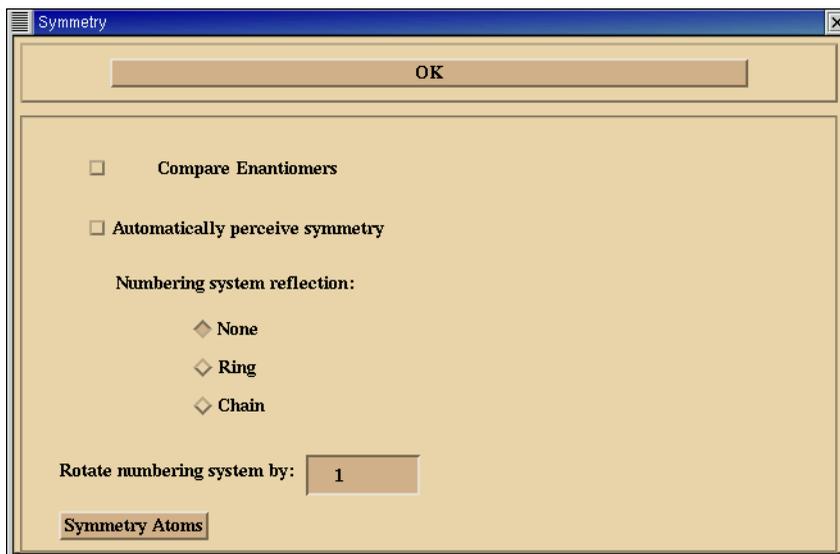


Figure 3.5. The Symmetry panel.

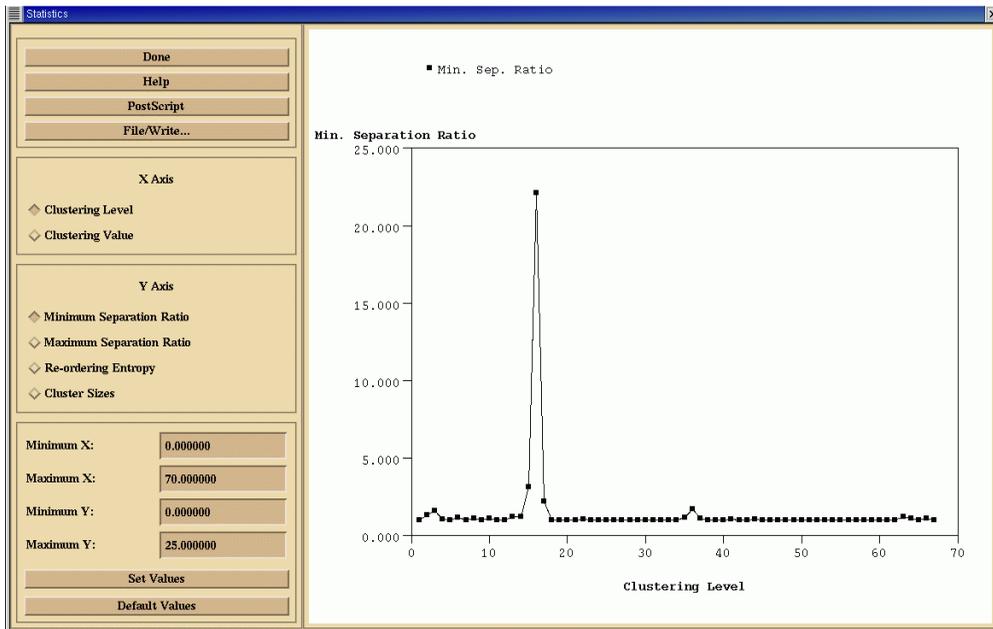


Figure 3.6. The Statistics panel displaying clustering statistics.

3.6 The Visualization Menu

Once clustering is complete, the Visualization menu becomes active. The options in this menu allow the data from the clustering to be visualized in a number of ways.

3.6.1 Clustering Statistics

The clustering statistics panel is used to view line graphs of various statistics versus clustering level or threshold distance. The plot is controlled by two sets of buttons on the left hand side of the panel.

The X-Axis buttons control which variable is to be used for the X-axis of the plot:

- Clustering Level
- Threshold Distance

The Y-axis buttons are used to select a variable for the Y-axis of the plot:

- Minimum Separation Level
- Maximum Separation Level
- Reordering Entropy

- Cluster Sizes (Effective and Actual number of clusters; Minimum, Maximum and Average cluster sizes at each new clustering)

The range of each axis is initially set to encompass the full range of data. You may “zoom in” on one particular region by editing the Axis-Range text fields below the radio buttons. After entering new values for the X and Y minima and maxima, clicking the Set Values button will cause the plot to be redrawn incorporating the new settings. The original scaling can be reinstated by pressing the Restore Defaults button. Axis ranges are also reset to default values when axis modes are changed. No checking is done on the values which are set in the range fields, and entering a “minimum” value which is larger than the “maximum” will effectively reverse the direction of the axis.

Point Picking: A point displayed in a plot can be “picked” by clicking its plot symbol. When a point has been successfully selected a beep sounds. It may be necessary to adjust the axis ranges in order to clearly select points. The effect of picking a point is three-fold. First, the clustering level corresponding to the point is entered in the Clustering Level field of the Cluster Membership dialog box. Second, the Distance Map is set to the specified clustering level. Finally the clustering level corresponding to the point is entered in the File/Write dialog box (see above).

The PostScript button is used to create an encapsulated PostScript representation of the clustering statistics window suitable for printing on a Laser printer. When the PostScript button is selected, a dialog box appears which in turn has a Center on Page button. By default this is off, and in this situation the image will be approximately the size it appears on the screen. If this button is turned on, then the image will be sized to be centered on and fill as much as possible of an 8.5- by 11-inch page.

The Clustering Statistics Menu Selection is designed primarily for data exploration, and the PostScript options are provided primarily for record keeping, rather than for presentation graphics. All the statistics viewable by means of this selection are also printed in tabular form in the log file for the run, and may be extracted from that file and used in conjunction with the data presentation package of the user’s choice.

The File/Write button is used to display the Write File panel (see [page 11](#)) which allows the writing of a number of different types of output files.

3.6.2 Distance Map

This panel displays a pictorial representation of the distance matrix in either the generic or the input ordering, controlled by the Generic or Input Ordering radio buttons. The distance values are represented by a color scale (or gray-scale on a gray-scale monitor) representing values in ten bins of equal width between the minimum and maximum values in the distance matrix.

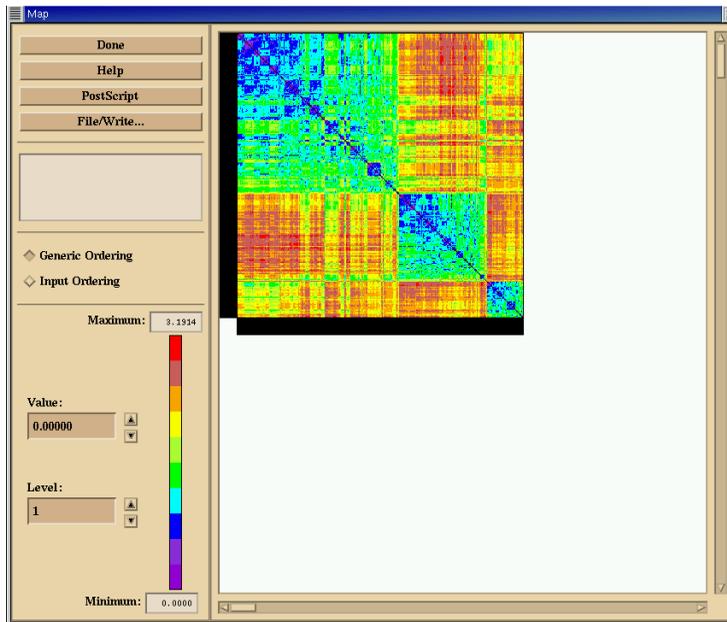


Figure 3.7. The Map panel.

The color black is used to color any matrix element that corresponds to a pairwise distance less than or equal to a given clustering value (default zero). This can be a useful aid for highlighting clusterings. Several methods are available for controlling this clustering level. Note that the on-diagonal matrix elements, which have the value $d_{ii} = 0$, are always black.

Color Bar. The color bar to the left of the matrix display indicates the color scale used; minimum and maximum distance values are displayed in non-editable text windows beneath and above the bar. The ten colors represent ranges obtained from linear interpolation between the minimum and maximum values. The color bar itself is interactive; clicking anywhere along it sets the current threshold distance to the corresponding value.

Threshold Value Field. To the left of the color bar is a text field which displays the current threshold distance. This value can be edited by typing a number into the text field and pressing the RETURN key. The value may be any positive real number less than the maximum value in the distance matrix. To the right of the text field are upward and downward facing arrow buttons. These buttons can be used to step through the unique values on the sorted distance list. Resetting the threshold value also resets the clustering level, where appropriate.

Clustering Level Field. The clustering level can be controlled either by editing this text field or by scrolling through the list of levels using the up and down arrows, next to the field, as for the Value field. In addition, if a plot of some clustering statistic is visible at the same time as is

the distance map, clicking on a plot symbol sets the map's clustering level to that corresponding to the X-axis position of the plot symbol.

Setting the Level always gives a non-negative integer in the text field. Sometimes, however, when the Value Field is set, the Level displays a plus-sign after the integer. This indicates that the distance selected is greater than the critical threshold distance corresponding to the level shown, but less than that corresponding to the next level.

Resetting the clustering level always resets the Threshold Value Field to the corresponding critical threshold value.

Interacting with the Map: Moving the mouse pointer onto the Map display and clicking the left mouse button gives a readout of the original and generic conformation numbers and their pairwise distance value will appear in the Data Text Area on the left of the panel. When the mouse button is released the readout continues and the selected distance matrix element flashes white. If the mouse cursor is still pointing into the display, small adjustments in the selection can be made using the keyboard cursor-control keys. To clear the flashing element and the readout, click the middle mouse button in the display, or click the left or middle button outside the display but within the Map window. Clicking on the map will also update the clustering level and cluster number for writing a single cluster in the Write dialog (see [page 11](#)). The cluster number is taken from the position of the selected cell along the X-axis of the map. The clustering level is set to that displayed in the Clustering Level Field of the map.

When the map is displayed in generic order, “combs” are visible on the left and beneath the display. The “teeth” of the combs represent separators or partitions between clusters. These partitions are successively removed with increasing clustering level. Thus at any setting of the threshold distance or clustering level, the extent of the clusters is immediately visible.

The PostScript button may be used to create an encapsulated PostScript representation of the map. It invokes a dialog box which has options similar to those which appear when the PostScript button is selected while displaying cluster statistics.

The File/Write button is used to display the Write dialog box (see [page 11](#)) which allows the writing of a number of different types of output files.

3.6.3 Clustering Mosaic

The clustering mosaic is a visual representation of the formation of clusters during the clustering process. Recall that the generically ordered conformation list may be viewed as a list of structures separated by partitions. As the clustering level increases, partitions are removed, and at any level a set of conformations without partitions between them represents a cluster. The mosaic contains the same information as the more conventional dendrogram but the information is displayed differently.

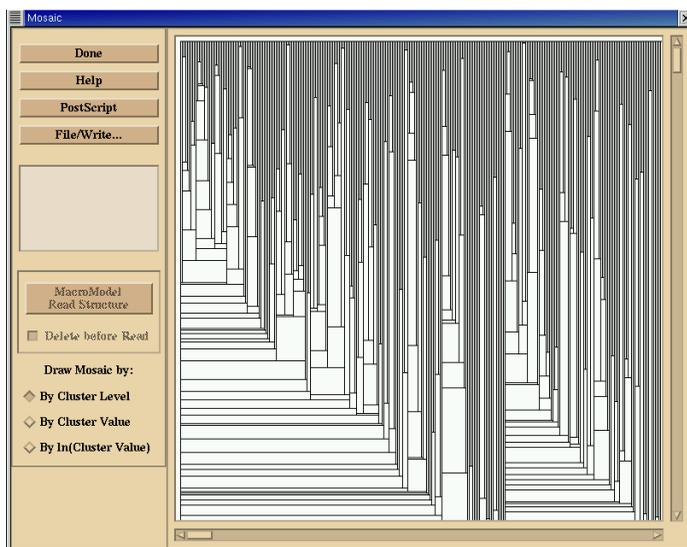


Figure 3.8. The Mosaic panel.

Each row of the mosaic display represents the list of conformations, in the generic order, at a given clustering level; clustering level increases downwards. Vertical lines appearing within each row represent the set of partitions separating the clusters at that row's clustering level. Thus an uninterrupted white region within a row indicates a cluster. The mosaic can be displayed with the vertical spacing between clustering levels represented in three different ways, controlled by options on the left side of the panel. In the By Cluster Level display there is a constant spacing between each clustering level. The By Cluster Value display the spacing between each level is proportional to the threshold value at each clustering level. Finally the By $\ln(\text{Cluster Value})$ display shows the clustering levels drawn proportional the logarithm of the clustering value at each level. Note that both of the displays by cluster value may have overlapping levels; if this is the case a warning is displayed in the data text area on the left. Overlapping values occur when the resolution of the screen is not sufficient to resolve successive levels at the scale of the display. When By Cluster Value is selected no bordering line is drawn below the last clustering; this, for N structures the N horizontal lines are drawn at positions corresponding to the N threshold distances. The first of these lines corresponds to the value of zero.

When By $\ln(\text{Cluster Value})$ is selected, only $(N-1)$ horizontal lines are drawn, that corresponding to the first value being omitted (since $\log(0)$ is undefined). When using the By $\ln(\text{Cluster Value})$ display, the height of any level (distance between two successive tick marks shown on the right) is proportional to the minimum separation ratio.

Interacting with the Mosaic: Each column of the display represents a conformation in the generic ordering going from left to right. If you place the mouse at any position and hold down

the left button, you will see, in the data text area to the left, a display of both the input and the generic sequence number of the conformation corresponding to the column where the mouse cursor is pointing, and also an indication of the threshold value and clustering level corresponding to the row of the cursor. If the mouse button is released the text display will persist, and the cursor will be replaced with a yellow highlighting of the current row and column. As with the Map display, these crosshairs can be repositioned by the use of the cursor keys on the keyboard. The readout and crosshairs can be cleared by clicking the middle mouse button while the pointer is in the display, or by clicking the left or middle mouse button outside the mosaic display, but within the Mosaic window. Clicking in the Mosaic display also updates the clustering level and cluster number fields of the Write dialog (see [page 11](#)).

The PostScript button allows the writing of a PostScript representation of the display, and works as elsewhere in the program.

The File/Write button is used to display the Write dialog (see [page 11](#)) which allows the writing of a number of different types of output files.

3.6.4 Cluster Membership

A description of the composition of the clusters at any level can be obtained by using the Cluster Membership Menu Selection of the Visualization menu. Enter the desired clustering level into the Clustering Level text field and then click the Show Membership button. The membership of the clusters at the level specified are written to the Main Menu Message Window, and to the log file. Only integers in the range 1 to N , the number of conformations which were clustered, are accepted in the text field.

The Clustering Level Text Field can also be filled by clicking a plot symbol in a concurrent display of clustering statistics. Clicking a point in the plot causes the level corresponding to the X-axis position to be entered into the Clustering Level Text Field.

3.7 The Help Menu

About XCluster

This option displays a dialog box with version and copyright information for the program.

Topics

The Topics option is used to display the online help for the XCluster program. Clicking the List All button displays a list of topics for which help is available. Alternatively it is possible to search for a keyword by using the Search and Show button. There are options to make the search case sensitive and for searching only the help topic titles.

Using Cluster

4.1 Batch Interface

4.1.1 Command-line Options

Cluster can be run as a batch program by entering the command `$SCHRODINGER/cluster`. The syntax of the `cluster` command is as follows:

```
cluster [-v] < cmdfile > logfile
```

- `cluster [-v] jobname`-v (verbose) generates copious output to the log file; this output is useful mainly for debugging the Cluster program.
- `jobname` is an optional job name; if specified, input is read from `jobname.clu` and output is placed in `jobname.clg`; for example, `cluster roseb` would use `roseb.clu` as the command file and `roseb.clg` for the log file.

4.1.2 File Conventions

We have found the following file name conventions to be useful. Note that these are simply conventions and may be overridden. The structure file name conventions assume that the files will be written in Maestro format.

- `filename.clu`: the command file from which Cluster reads its input.
- `filename.clg`: the log file into which Cluster places its output.
- `filename_out.mae`: the structure file containing the structures to be analyzed.
- `filename.dst`: a distance file containing pairwise distances to be analyzed by the cluster program using `dfile` mode.
- `filename.map`: a distance file containing pairwise distances for a data set in generic order.
- `filename-n_m_cls.mae`: a cluster file containing molecular conformations, optimally superimposed and colored to exhibit the cluster *m* at clustering level *n*. For example, `roseb-191_cls.mae` would contain all the clusters which appear for the job `roseb` at clustering level 191 and `roseb-191_5_cls.mae` would contain the structures from cluster 5 at clustering level 191. *m* is the leading member of the cluster in question, not an index.

- *filename-n_m_rep.mae*: a cluster file containing a single representative structure from each cluster. For a description of the selection process and the coloring scheme see “Write_{rep}” on [page 30](#).
- *filename-n_m_lead.mae*: a file containing the “lead” structure from each cluster. This structure is the structure that appears first in the input file for a given cluster. For a description of the selection process and the coloring scheme see “Write_{lead}” on [page 31](#).
- *filename-n_m_avg.mae*: a cluster file containing a single average structure from each cluster. For a description of the selection process and the coloring scheme see “Write_{avg}” on [page 32](#).

4.1.3 Command File

The command file is a free-format text file made up of a combination of the following commands. In this chapter the commands are arranged in related groups, and the sequence of presentation roughly reflects the sequence in which the commands must appear.

All commands are recognized by case-sensitive string-matching, and all begin with an upper-case letter and end in a colon.

4.2 Comment Insertion Command

Comment: *comment-text*

Dependencies: None.

Description: Associate the given comment with the command file. The comment may consist of any text. It begins with the line following that on which the Comment: command appears, and is terminated by a line containing only a period.

Arguments: None.

Comment: The comment itself, without the terminating line, will appear in the Comment Text Field, and may be further edited there, if the command file is read into XCluster.

4.3 Input File Specification Commands

sfile: *filename*

Dependencies: None.

Description: Use the input structure file called *filename* as the input to one of the distance-matrix generating commands. This file must be a valid MacroModel or Maestro structure file

and contain structures that are identically numbered; e.g., conformers from a search or sampled structures from a molecular dynamics simulation.

Arguments: The name of input file.

Comment: Necessary if the program is to calculate the distance matrix from a subsequent `Arms:`, `Trms:`, or `Nrms:` command. May not be used in the same command file with `Dfile:`.

Dfile: *filename*

Dependencies: None.

Compatibility: Incompatible with `Sfile:`, or with any of the commands that require `Sfile:`, such as `Arms:`, `Trms:`, `Nrms:`, `Writecls:` and the symmetry commands.

Description: Use the distance list in the file *filename* as input to the program. This implicitly puts the program into `dfile` mode, preparing it to read a user-supplied distance matrix rather than generating its own.

Arguments: The name of the file containing the distance list. The fields in the file can be separated by any combination of white-space: spaces, tabs and newlines. The first field is an integer, N , the number of data points whose mutual distances are to follow. The remaining $N(N-1)/2$ fields in the file consist of these distances, in the sequence, $d_{12}, d_{13}, d_{14}, \dots, d_{1N}, d_{23}, d_{24}, d_{25}, \dots, d_{2N}, \dots, \dots, d_{N-1,N}$. The ordering 1, 2, 3, ..., N can be any arbitrary ordering of the user's choice, but if there is some "natural" ordering of the N points, such as the energy or the time, it should be used.

4.4 Symmetry Specification Commands

With all the symmetry commands, when the distance between a pair of conformations is calculated, the lowest value of all the symmetry-related comparisons is the one used. In the calculation of the distance-matrix element d_{ij} , structure i is held fixed and j is subjected to all allowed symmetry operations. All symmetry commands must be specified before an `Arms:`, `Trms:` or `Nrms:` command.

Mmsym:

Dependencies: Requires a preceding `Sfile:` command. Must precede any `Arms:`, `Nrms:` or `Trms:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Performs automatic recognition of local and global molecular symmetry.

Arguments: None.

Comments: Must appear before `Rotate:` and `Reflect:`, which, however, are ignored if `Mmsym:` is present. May be used with `Enant:`.

Symatom: *atom-list*

Dependencies: Requires a preceding `Sfile:` command. Must precede any `Arms:`, `Nrms:` or `Trms:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Specifies the list of atoms for which symmetry actions will be applied. As an example, in a cluster analysis of structures resulting from a conformational search of cyclononane, the symmetry atoms would be the numbers of the ring atoms, in sequence.

Arguments: The list of symmetry-related atoms must be specified. This is a list of atom numbers, one per line, starting on the line following that on which the command itself appears, and terminated by a blank line.

Comments: Ignored if preceded by `Mmsym:`.

Rotate: *fold*

Dependencies: Requires a preceding `Sfile:` command. Must precede any `Arms:`, `Nrms:` or `Trms:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Instruct the program to perform number-system rotation of the symmetry atoms during distance calculations. This command is usually used for simple cyclic compounds.

Arguments: An integer which indicates the “fold” of the rotation to be applied to the numbering system. For example, for cyclononane, the “fold” of the rotation is 9.

Comments: Functionality replaced by `Mmsym:`, but retained for backward compatibility. Ignored if preceded by `Mmsym:`.

Enant:

Dependencies: Requires a preceding `Sfile:` command. Must precede any `Arms:`, `Nrms:` or `Trms:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Instruct the program to compare enantiomers as well as original structures.

Arguments: None.

Reflect: chain | ring

Dependencies: Requires a preceding `Sfile:` command. Must precede any `Arms:`, `Nrms:` or `Trms:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Instruct the program to perform numbering system reflection of the symmetry atoms during the comparison procedure.

Arguments: One of the following keywords must be included.

- `chain:` reflect as follows:



- `ring:` reflect as follows:



The ring mode of reflection allows cyclic conformations with heteroatoms to be compared; for example, atom 1 in this cyclic system could have been a nitrogen instead of a carbon.

Comments: Functionality replaced by `Mmsym:`, but retained for backward compatibility. Ignored if preceded by `Mmsym:`.

4.5 Distance Matrix Generation Commands

These commands are used to select from among the distance criteria known to the program.

Arms: `heavy` | `all` | *atom list*

Dependencies: Requires a preceding `Sfile:` command. Must follow any symmetry command.

Compatibility: Incompatible with `Dfile:`, `Trms:`, and `Nrms:`.

Description: Use pairwise R.M.S. atomic displacement following best rigid-body superposition as the definition of distance between a pair of structures.⁴

Arguments: Exactly one of the following must be specified:

heavy: the rigid body superposition and the R.M.S. calculation will use all atoms, that are neither hydrogens nor lone pairs.

all: the rigid body superposition and the R.M.S. calculation will use all atoms in the structure.

atom list: a list of atom numbers to be used for both superposition and distance calculations. The first atom number must appear on the line following the *Arms*: command. The atom numbers must be specified one per line, and the list is terminated by a blank line. There must be at least three atoms for the comparison procedure to work.

Nrms: *heavy* | *all* | *atom-list*

Dependencies: Requires a preceding *Sfile*: command. Must follow any symmetry command.

Compatibility: Incompatible with *Dfile*:, *Arms*:, and *Trms*:

Description: Calculate a distance matrix using the pairwise R.M.S. atomic displacement of the comparison atoms in place—i.e., without first attempting rigid-body superposition—as the distance measure.

Arguments: As for *Arms*:

Comments: This command has not been extensively tested.

Trms: *torsion-list*

Dependencies: Requires a preceding *Sfile*: command. Must follow any symmetry command.

Compatibility: Incompatible with *Dfile*:, *Arms*:, and *Nrms*:

Description: Calculate the distance matrix using the pairwise R.M.S. difference between corresponding torsions as the definition of distance. The difference between torsion angles is measured the “shortest way around.” For example, the difference between the angles of 175 and -175 is 10.

Arguments: The torsion list must be specified. This is a list of atoms, four to a line, each set of four defining a torsion angle to be used for comparison. The first torsion appears on the line following the *Trms*: command. The list is terminated by a blank line.

4. Kabsch, W. *Acta Cryst.* **1976**, A32, 922, and Kabsch, W. *Acta Cryst.* **1978**, A34, 827.

4.6 Clustering Commands

Cluster:

Dependencies: None.

Description: Builds clusters at all clustering levels from the values in the distance matrix as described in [Chapter 6](#). Also calculates simple statistics about the clusterings.

Arguments: None.

Comments: The `Cluster:` command can only be used after a valid distance matrix has been generated or read into the program. The statistics generated by the command appear in the log file.

Thresh: *level*

Dependencies: Requires a preceding `Cluster:` command.

Description: Write out the membership of all clusters at a given clustering level. The output also includes the value of the critical threshold distance for the clustering, and the values of the separation ratio for the individual clusters.

Arguments: An integer indicating the desired clustering level.

Comments: Ignored by XCluster. The functionality of this command is available interactively from XCluster.

4.7 File Output Commands

XCluster always generates a log file, but in addition certain other files can be created using special commands.

writecls: *level filename clust_num*

Dependencies: Requires a preceding `Cluster:` command.

Compatibility: Incompatible with `Dfile:.`

Description: Write a MacroModel or Maestro format file of conformations reflecting the clusters at a given threshold level or for a given number of clusters. The format used is whichever format the structural input (`Sfile:`) was supplied in. This command operates to create “super-molecules,” so that when reading this file each cluster will be read as a single structure. The structures in the output file are written from coordinates which have been superimposed to give

the best fit with their conformational near-neighbors, after applying any symmetry operations. Each cluster is written in a different color, using a cyclic scheme of 20 colors.

Arguments:

- *level*: If *level* is positive, it is the threshold level defining the clustering, and must be an integer between 1 and *N* (the number of structures), inclusive. If *level* is negative, *|level|* is the number of clusters to use, which determines the threshold level; *|level|* must be an integer between 1 and *N*, inclusive.
- *filename*: the name of the structure file to be created. We have found it convenient to use the following naming convention: *jobname-n_cls.mae*, where *n* is the threshold level for the clustering. If a cluster number is specified, we use the convention *jobname-n_m_cls.mae*, where *m* is the cluster number. This limits output to a single cluster *m* of the clustering at threshold level *n*.
- *clust_num*: the number of the structure which is the leading member of the cluster to be written. If *clust_num* is not specified or given as *all* then structures from all clusters at the given level will be written to the file.

Comment: Ignored by XCluster. The functionality of `Writecls:` is available interactively in XCluster. `Writecls:` always writes out the conformations in the generic ordering; thus

```
Writecls: 1 filename all
```

writes out all the structures as individual structures, but in the generic ordering, and possibly symmetry-transformed and moved to new positions in space. Note that cluster files may be very large. For large structures it is probably better to either sample some of the structures listed from the “Cluster Membership” or `Thresh:` commands, or use the `Writerep:` or `Writeavg:` command described below to write the representative or average structure for each cluster.

Writerep: *level filename clust_num*

Dependencies: Requires a preceding `Cluster:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Write a structure file containing a single representative structure for each cluster at a given threshold level or for a given number of clusters. The file is in MacroModel or Maestro format, depending on the format of the (`Sfile:`) structural input. The structures in the output file are written from coordinates which have been superimposed to give the best fit with their conformational near-neighbors, after applying any symmetry operations. The way the most representative cluster is chosen is as follows. For each cluster, the centroid of each of the comparison atoms is determined, after each structure has had its generic transform applied to

it. Then the structure within the cluster that has the smallest r.m.s. interatomic distance from these centroids is selected as the most representative structure.

In the output file, non-comparison atoms are colored gray. The color of each comparison atom is determined by the radius of gyration (R_g) of the point-cloud of all the positions for that atom in the entire ensemble (not just the cluster the structure in question represents). Thus, the color scheme is based on the variability of atomic position in the superimposed structures. The range zero to the maximum R_g observed is divided into equal intervals, and the atoms are colored blue, green, yellow, orange or red, depending upon which interval they fall into. Red represents the highest- R_g interval. In those situations in which the positions of only a few atoms define a strong clustering, as in our roseotoxin-b example, these atoms are colored red, and atoms whose positions change little are colored blue. When, on the other hand, all comparison atoms contribute strongly to the clustering, as in our pentane example, only a small range of R_g will be observed (i.e., there may be no blue atoms), and the red atoms have no special significance.

This command also causes tables to be written to the log file giving statistics on distance to the centroid and R_g for the most representative structures, the clusters, and the entire ensemble.

Arguments:

- `level`: If *level* is positive, it is the threshold level defining the clustering, and must be an integer between 1 and *N* (the number of structures), inclusive. If *level* is negative, *llevel* is the number of clusters to use, which determines the threshold level; *llevel* must be an integer between 1 and *N*, inclusive.
- `filename`: the name of the structure file to be created. We have found it convenient to use the following naming convention: *jobname-n_m_rep.mae*, where *n* is the threshold level for the clustering and *m* is the number of the cluster for which a representative structure is to be calculated.
- `clust_num`: the number of the structure which is the leading member of the cluster to be written. If `clust_num` is not specified or given as “all” then a representative structure from all clusters at the given level will be written to the file.

Comment: Ignored by XCluster. The functionality of `write_rep` is available interactively in XCluster.

writelead: *level filename clust_num*

Dependencies: Requires a preceding `Cluster:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Write a structure file containing a single structure for each cluster at a given threshold level or for a given number of clusters. The file is in MacroModel or Maestro format,

depending on the format of the (`Sfile:`) structural input. The structures in the output file are written from coordinates which have been superimposed to give the best fit with their conformational near-neighbors, after applying any symmetry operations. The structure written for each cluster is the first or “lead” structure in the cluster, which is the lowest-numbered structure in the input ordering. Thus, if the input structures are in increasing order of energy (or time), the lead structure will be the lowest-energy structure (or the first structure in the trajectory) in the cluster.

Arguments:

- `level`: If *level* is positive, it is the threshold level defining the clustering, and must be an integer between 1 and *N* (the number of structures), inclusive. If *level* is negative, *llevel* is the number of clusters to use, which determines the threshold level; *llevel* must be an integer between 1 and *N*, inclusive.
- `filename`: the name of the structure file to be created. We have found it convenient to use the following naming convention: *jobname-n_m_led.mae*, where *n* is the threshold level for the clustering and *m* is the number of the cluster to be written.
- `clust_num`: the number of the structure which is the leading member of the cluster to be written. If `clust_num` is not specified or given as “all” then the leading member structure from all clusters at the given level will be written to the file.

Comment: Ignored by XCluster. The functionality of `writelead`: is available interactively in XCluster.

Writeavg: *level filename clust_num*

Dependencies: Requires a preceding `Cluster:` command.

Compatibility: Incompatible with `Dfile:`.

Description: Write a structure file containing a single average structure for each cluster at a given threshold level or for a given number of clusters. The file is in MacroModel or Maestro format, depending on the format of the (`Sfile:`) structural input. The structures in the output file are written from coordinates that are superimposed to give the best fit with their conformational near-neighbors, after applying any symmetry operations. For comparison atoms the average structure is calculated as the average of the cartesian coordinates of all the structures in the cluster. Non-comparison atoms have their coordinates taken from the representative structure. In the output file atoms are colored as for the representative structure file (see above).

Arguments:

- `level`: If *level* is positive, it is the threshold level defining the clustering, and must be an integer between 1 and *N* (the number of structures), inclusive. If *level* is negative, *llevel* is

the number of clusters to use, which determines the threshold level; *level* must be an integer between 1 and *N*, inclusive.

- *filename*: the name of the structure file to be created. We have found it convenient to use the following naming convention: *jobname-n_m_avg.mae*, where *n* is the threshold level for the clustering and *m* is the number of the cluster to be written.
- *clust_num*: the number of the structure which is the leading member of the cluster to be written. If *clust_num* is not specified or given as “all” then an average structure from all clusters at the given level will be written to the file.

Comment: Ignored by XCluster. The functionality of *Writeavg*: is available interactively in XCluster.

Writedst: *filename*

Dependencies: Requires a preceding *Cluster*: command.

Description: Write out the current distance matrix to a file. The file is written as a list of the $N(N-1)/2$ unique values, in the input ordering. The format is such that the written file will be readable using the *Dfile*: command.

Arguments: The name of the file. We have found it convenient to use the *.dst* extension for the distance matrix files.

Comment: Ignored by XCluster. The functionality of *Writedst*: is available interactively in XCluster.

Writemap: *filename*

Dependencies: Requires a preceding *Cluster*: command.

Description: Write out the distance matrix in the generic conformation order (see [Chapter 6](#)); otherwise, the format is as for *Writedst*:.

Arguments: The name of the file. We have found it convenient to use the *.map* extension for the generic-order map files.

Comment: Ignored by XCluster. The functionality of *Writemap*: is available interactively in XCluster.

Figures of Merit

This section describes several statistics which are functions of the clustering and which are printed into the log file as a result of the Cluster: command. From XCluster, these statistics can be displayed on the screen versus either critical threshold distance or clustering level, using the Cluster Statistics selection from the Visualize menu of the main window.

We are convinced of the utility of the *minimum separation ratio*, R , in the analysis of clusters; the *effective cluster number*, $N_{\text{cl,eff}}$, also appears to be useful. The *reordering entropy* S_{re} , though it is an intriguing concept, has yet to prove its utility in practice.

In [Chapter 7](#) we show how these statistics can be used to understand the clustering taking place in several ensembles of molecular structures.

5.1 Separation Ratio

Recall that a critical threshold distance, T , is a value of a distance-matrix element that creates a new clustering as we ascend the sorted list of d_{ij} values. Recall also that there are N clusterings, given N data points; therefore there will also be N values of T . We now index these by clustering level, using the symbol T_i , where i ranges from 1 through N . Now consider two successive values of the critical threshold distance, T_i and T_{i+1} . We define the *minimum separation ratio* of a clustering, R_i , by $R_i = F(T_{i+1}, T_i)$. Let us consider the meaning of R_i .

T_{i+1} is the distance between some pair of items which, at clustering i , must be in different clusters. Since T_{i+1} is the next value of d_{ij} , in the sorted distance list after T_i that creates a new clustering, T_{i+1} must be the shortest distance between any two items not in the same cluster at clustering level i . R_i is then the ratio of the shortest distance between two items not in the same cluster (i.e., the shortest distance between clusters) to the current clustering's critical threshold distance. The current clustering's critical threshold distance will also be the greatest nearest neighbor distance over all pairs of points in the same cluster.

If R_i is large, all clusters must be well separated, for in this situation the shortest distance between pairs not in the same cluster is large compared to the threshold distance defining the clustering.

It sometimes happens that for some clustering level, i , a particular value of R_i is well in excess of the others (see [Chapter 7](#)). If this occurs at a high clustering level, T_{i+1} is almost certainly an indication of "good" clustering. At clustering levels where clustering is poor, R_i tends to be

close to unity; a value of R_i greater than about 2 and appearing at a high clustering level is usually, in our finite but limited experience, interesting.

We can also define the **separation ratio** of a single cluster. At a given clustering level, i , this is equal to the ratio of the shortest distance between a structure in this cluster and one in any other cluster to T_i . In this context, the minimum separation ratio, discussed above, is the minimum of the separation ratios of the individual clusters. Even when the minimum separation ratio is low, some of the clusters may have high separation ratios. This means that these clusters, only, are well separated from their neighbors. A plot of the maximum separation ratio is available from XCluster. The individual separation ratios are printed along with cluster memberships defined by the `Thresh:` command in Cluster or by the Cluster Membership selection from the main window's Visualize menu.

5.2 Effective Number of Clusters

Recall that the molar entropy of mixing of ideal gases is given by:

$$\frac{\bar{S}}{R} = - \sum_{i=1}^k x_i \ln x_i$$

where the x_i are the mole fractions of the k species present. Similar formulas are used in statistics and in information theory to describe how “spread out” a distribution is.

In the ideal-gas example, if the k gases occur with equal mole fractions, then:

$$\frac{\bar{S}}{R} = -k \left[\left(\frac{1}{k} \right) \ln \left(\frac{1}{k} \right) \right] = \ln k$$

and $\exp(\bar{S}/R)$ is the number of species present. Thus $\exp(\bar{S}/R)$ can be thought of as the “effective number” of species present. If one of the gases is slowly removed until only $(k-1)$ species are present, $\exp(\bar{S}/R)$ decreases continuously from k to $(k-1)$.

For any mixture of k gases, \bar{S}/R and $\exp(\bar{S}/R)$ are maximal when the x_i are all equal. Also, if a mixture contains more than k species, but all the mole fractions but k of them are small, and these k mole fractions are approximately equal, then $\exp(\bar{S}/R)$ is approximately k , since the contribution of the others to \bar{S}/R then approaches the value zero, since by l’Hopital’s rule,

$$\lim_{x \rightarrow 0} x \ln x = 0$$

Suppose that at some clustering we observe k clusters with populations n_1, n_2, \dots, n_k . We use the symbol x_i for $F(n_i, N)$, the fraction of items in cluster i . We then define the clustering entropy by:

$$S_{cl} = - \sum_{i=1}^k x_i \ln x_i$$

and the effective number of clusters by $N_{cl,eff} = \exp(S_{cl})$. At clustering level 1 and N we have, respectively, N clusters of one item each and one cluster of N items. At these levels, $N_{cl,eff}$ is equal to the actual number of clusters, N and one, respectively. Between these limits, $N_{cl,eff}$ generally lies below the actual number of clusters, since it is rare for the clusters to be of equal size except at the endpoints of the clustering-level range. Thus a plot of $N_{cl,eff}$ versus clustering level is generally concave when viewed from above.⁵

In certain situations (see Chapter 7), a few large clusters form early, and then either grow slowly by accretion as clustering index increases or else remain approximately constant in size as outlying items agglomerate to form new clusters. In this situation a plot of $N_{cl,eff}$ versus clustering level tends to exhibit a broad, nearly flat region whose ordinate is roughly indicative of the number of large clusters present.

5.3 Reordering Entropy

As discussed elsewhere, there is a (non-unique) “generic ordering” of the items such that at any clustering level, all items in the same cluster will be adjacent on the list. Thus, one can represent a given clustering as a set of dividers or partitions placed in the generically ordered list, separating the clusters. At the first clustering level, where all items are in separate clusters, there is a divider after each item on the list. At each clustering level, a single divider is removed.

We now consider the question: at a given clustering level, how many ways can the list be reordered without destroying the contiguity of elements belonging to the same cluster? At the first level, where each item is in a cluster by itself, there are $N!$ ways of ordering the list, since the first item can be in any of N positions, the second can be in any of $(N-1)$ positions, etc. At the N th level, where all items are present in a single cluster, the ordering is again irrelevant, and we

5. In any event, $N_{cl,eff}$ is guaranteed to decrease monotonically as cluster level increases. Suppose two clusters, I and J , with fractional populations p_I and p_J , agglomerate with an increase in clustering level; recall that this is the only type of event that can occur with such an increase. Prior to the agglomeration, the contribution of I and J to S_{cl} is $S_{before} = -p_I \ln p_I - p_J \ln p_J$; afterwards the contribution of the combined cluster is $S_{after} = -(p_I + p_J) \ln (p_I + p_J) = -p_I \ln(p_I + p_J) - p_J \ln(p_I + p_J)$. p_I and p_J are positive, by their definition; therefore $\ln(p_I + p_J) > \ln(p_I)$, $\ln(p_J)$, since the logarithm increases monotonically with its argument. It follows that $S_{after} < S_{before}$.

again have $N!$ possibilities. But between these bounding levels some reorderings break up clusters, and are forbidden. If some clustering exhibits k clusters having $n_1, n_2, n_3, \dots, n_k$ members, then within each cluster the members can be freely reordered, giving

$$\prod_{i=1}^k n_i!$$

possibilities. In addition, the k clusters themselves can be reordered on the list, giving $k!$ additional possibilities. The total number of allowed reorderings is then given by:

$$W = k! \prod_{i=1}^k n_i!$$

In analogy to Boltzmann's $S = k_B \ln W$, we define the reordering entropy as $S_{re} = \ln W$. S_{re} goes through a minimum as clustering level or threshold increases. It can be shown that there is a tendency for S_{re} to be minimal when one has a large number of small clusters, and in fact we have observed this in chemical examples. The minimum value of S_{re} generally defines a unique clustering level, and this seems interesting by itself; however, it is not clear whether the minimum- S_{re} clustering is interesting in any other way.

How It Works

6.1 Generic Conformation Order

The concept of the generic conformation order is central to an understanding of how XCluster works. As mentioned earlier, it is possible to reorder the original data items (conformations, if molecular structures) so that, at any clustering level, conformations in the same cluster are in an adjacent block in the reordered list. We call such an ordering “generic.” Again, this implies that at any clustering level, the generically ordered list can be thought of as a linear list of conformations with partitions or dividers between some of the list members. The partitions separate the clusters that are formed at that level. At threshold level 1, where each conformation is in a cluster by itself, there is a divider between each pair of adjacent conformations. At each ascending threshold level, a single divider is removed. At threshold level N , given N data items, all the dividers have been removed, and there is a single cluster, consisting of all the items. In this section we describe how a generic ordering is assembled.

Let us suppose we have N data items, originally numbered from 1 through N . After the distance matrix is created or read in, the $N(N-1)/2$ pairwise distance values are sorted into ascending order, and a record is kept of which data items i and j ($i < j$) each distance value pertains to. In this discussion, i and j will always refer to the numbering of data items in the original (input) ordering.

The process of building a generic ordering is carried out by a successive reordering of the original list. This is done as we ascend the sorted distance list. Originally, each data item is in a cluster of its own. We give each cluster a name, or label, which is the number of the lowest-numbered conformation in it; thus, at the start, each structure, i , is in a cluster labeled i . This is threshold level 1.

The first distance on the sorted list d_{ij} , for some i and j , is guaranteed to create a new clustering (level 2), since it brings structures i and j , which were formerly in separate clusters, into a single cluster. Thus, at level 1, point i was in cluster i and point j was in cluster j . At level 2, we perform a partial reordering of the list by removing item j from its original position in the list and inserting it after item i . Thus, data points i and j are now contiguous in the reordered list. At the same time, we maintain a record of the fact that at level 2, items i and j both belong to cluster i .

For each new value on the sorted distance list, we first check to see whether the two data items separated by this distance are already in the same cluster; if so, this distance is not a critical

threshold distance, and we move on. If the two data items are not in the same cluster, then the current distance is a critical threshold distance—i.e., it is associated with a new clustering level—and we do another partial reordering of the list. Let us again call the two data items i and j . Let these elements be members of clusters I and J , where I and J are the lowest-numbered elements of the respective clusters. Let us suppose for the sake of discussion that $I < J$. We remove from their place in the list the contiguous items in cluster J , and insert them immediately after the last member of cluster I . We also assign all the items previously belonging to cluster J to cluster I .

Thus at each critical threshold distance—i.e., at each clustering level—we take an entire contiguous cluster of data points and move it to a new position immediately following another such contiguous cluster, and join the two groups together into a single, larger cluster. At no point do we break up a contiguous cluster previously formed, and at no point do we reorder the data points within a cluster previously formed. Thus, at the end of the process, at threshold level N , where only a single cluster remains, we have created an ordering which preserves the contiguity of clusters at all previous levels.

We mentioned earlier that a generic ordering is not unique. The above process always keeps the first data item in the input sequence first in the generic ordering. If the input sequence were shuffled, the first member of the reordered sequence would differ, but it would still be generic. Also, at any clustering level, the pair of clusters brought together could be exchanged without destroying the generic property. The algorithm described above does neither of these things, and thus produces the particular generic reordering which best preserves the input ordering of the data points.⁶

In the generic ordering, items which are close to each other in distance tend, in a statistical sense, to be close to each other in the list. Because in building the list pairs that are separated by short distances are brought together first, the tendency for closeness in distance to be correlated with closeness on the list is strongest for the shortest spatial distances. When the generic conformational ordering is used to create the distance matrix, the smallest matrix elements lie closest to the main diagonal. When the matrix is visualized at increasing clustering levels, the clusters thus appear to “grow out of” the main diagonal.

6.2 Three-dimensional Superposition

The rigid-body transforms that best superimpose the structures upon each other are calculated as clustering is performed. Regardless of the distance criterion used for clustering, the transforms are calculated so as to minimize the R.M.S. deviation of the distances between corre-

6. How many generic orderings are there? There appear to be 2^{N-1} of them. At each clustering level beyond the first, the pair of clusters brought together could have their locations exchanged, giving two possibilities for each of $N-1$ levels.

sponding atoms in a pair of conformations, following transformation. The “comparison atoms” used in the calculation of the transforms are those used in the calculation of the pairwise distances, if the distance comparison is done using either of the pairwise atom distance paradigms. When the R.M.S. of the torsional differences is used to calculate conformational distances, the atom set used for superposition is the union of the atoms used to specify the torsions.

Suppose d_{ij} ($i < j$) is a critical threshold distance, and let I and J be the labels of the clusters to which i and j belong. Assume for the sake of this discussion that $I < J$. If it is, we transform j so as to best superimpose upon i ; if not, we transform i so as to best superimpose upon j . If symmetry operations have been specified, the transform will include a symmetry part as well as a rigid-body translation and rotation.

Once the transform is calculated, it is applied to all elements of cluster J . Thus, an entire cluster is transformed spatially whenever it is joined to another cluster. Once the clustering process is complete, each conformation has been assigned a “generic transform,” appropriate to all the clusterings. Note that the first conformation in the input order remains stationary, since it is the lowest-numbered input structure. This is analogous to its always being first in the generic ordering which XCluster produces.

When structures are written out, the generic transforms are applied to the coordinates of the corresponding structures. The coloring of the structures, however, varies with the clustering; at any clustering level, structures in the same cluster are given the same color.

Examples

7.1 Two-dimensional Examples

These didactic examples provide several systems whose clustering, unlike that within the conformational space of any interesting molecule, can be directly visualized. We present several ensembles of points in two dimensions which exhibit different degrees of clustering, and discuss what inferences can be drawn from the output that XCluster gives when presented with these systems. Some of the characteristics we observe in the output will recur in our molecular examples, and we will be able to draw similar inferences there, despite the fact that the conformational space of an “interesting” flexible molecule is of far too high a dimensionality to permit the ensembles of points to be directly visualized.

All four examples are derived from the same data: 150 points in three sets of 50 each, the sets having been drawn from two-dimensional gaussian distributions with standard deviations of 0.2, 0.6 and 1.0. The examples differ in where the three sets are centered with respect to each other. As [Figure 7.1 on page 44](#), [Figure 7.6 on page 49](#), [Figure 7.15 on page 56](#), and [Figure 7.20 on page 59](#) show, the sets are placed more and more closely to each other in this sequence. Set A has a standard deviation of 0.2 and in Example 2D-1 is centered at (4,0); set B has a standard deviation of 0.6 and in Example 2D-1 is centered at (0,0); set C has a standard deviation of 1.0 and in Example 2D-1 is centered at (0,8). In Example 2D-2, set A moves to (2,0) and C moves to (0,4). In Example 2D-3, set A moves to (1,0) and C moves to (0,2). In Example 2D-4, set A moves to (1/2,0) and C moves to (0,1). For all these examples, the same random scrambling is applied to all 150 points to define the input sequence in the .out file that XCluster reads.

Some points are hidden under others in all of these figures; thus one cannot actually count 150 distinct points in any of them. The input files for the examples are supplied with the XCluster program, so you can experiment with the program yourself as you go through the examples.

We have given a precise definition for the term “cluster,” but disregarding this for the moment, and using the word in a strictly intuitive sense, most would agree that Example 2D-1 presents three clusters, Example 2D-2 presents three clusters with perhaps some ambiguity about several points, Example 2D-3 presents probably two clusters with some ambiguity, and Example 2D-4 presents a single cluster. We show XCluster plot, map and mosaic displays for these examples, and describe how these displays can be interpreted.

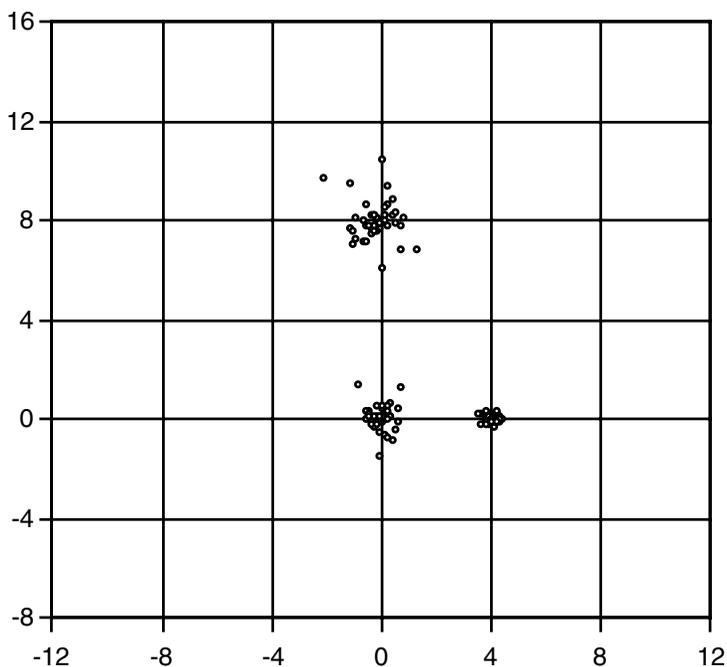


Figure 7.1. Two-dimensional point distribution, Example 2D-1.

Map and mosaic figures were created using the PostScript option of the respective XCluster commands. The plots were created by editing the .c1g file and passing the output of the cluster command to the plotting program KaleidaGraph on a Macintosh.

7.1.1 Example 2D-1

Minimum separation ratio

The minimum separation ratio is discussed in [Chapter 5](#). The minimum separation ratio is high at a clustering level at which all clusters are well separated from other clusters; that is, whenever the minimum distance between a pair of points not in the same cluster is high compared to the current critical threshold value, which is the longest nearest-neighbor distance within a cluster.

If we confine our view to the higher clustering levels, where few clusters appear, the minimum separation ratio, shown in [Figure 7.2](#), is high at clustering levels 148 and 149. At these levels we have three and two clusters respectively; note that if there are N points, then at clustering level n , we will have $N-n+1$ clusters.

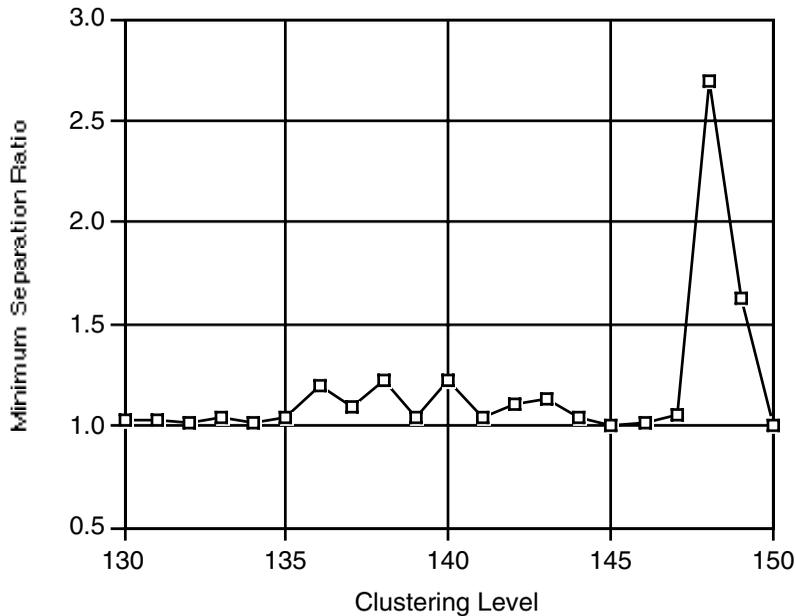


Figure 7.2. Minimum separation ratio, Example 2D-1.

Level 148 is the one at which sets A, B and C appear as separate clusters, and level 149 is the one at which sets A and B have joined into a single cluster and set C remains separate. The fact that the minimum separation ratio is higher for level 148 than for level 149 accords with our intuition that the agglomeration of the complete set of points into three clusters “makes more sense” than the agglomeration into two clusters; the fact that the minimum separation ratio at clustering level 149 is about 1.6 means that even after sets A and B have been joined into one cluster, the ratio of the shortest pairwise distance between this cluster and the remaining cluster, set C, to the longest distance within a cluster, which will be a distance between one point in A and one point in B, is about 1.6.

If you run XCluster on data set 2D-1 and look at the plot of minimum separation ratio over the entire range of threshold levels, you will observe that level 3 exhibits a minimum separation ratio of about 2. At this level there are 148 clusters, most of which consist of single points. The critical distance is small, about 0.005. What this means is that the next pair of points to agglomerate will be about 0.01 apart. Since there are so many clusters at low threshold levels, the chances are high of there being a high minimum separation ratio at some low threshold level. A statistical analysis would probably show that it takes a higher minimum separation ratio to be significant at low than at high threshold levels; in any event, it is our experience that values of the minimum separation ratio of, say, two or more are “interesting” at high, but not at low threshold levels.

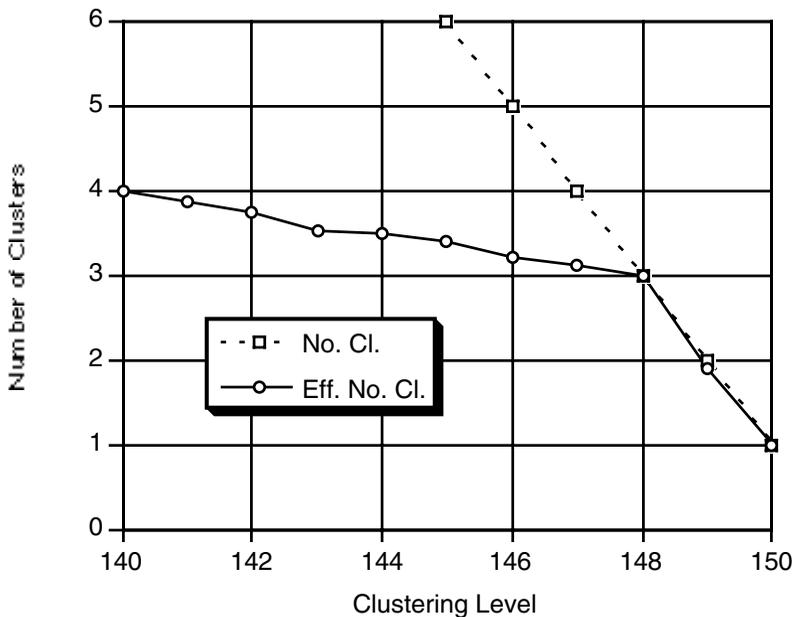


Figure 7.3. Actual and effective cluster number, Example 2D-1.

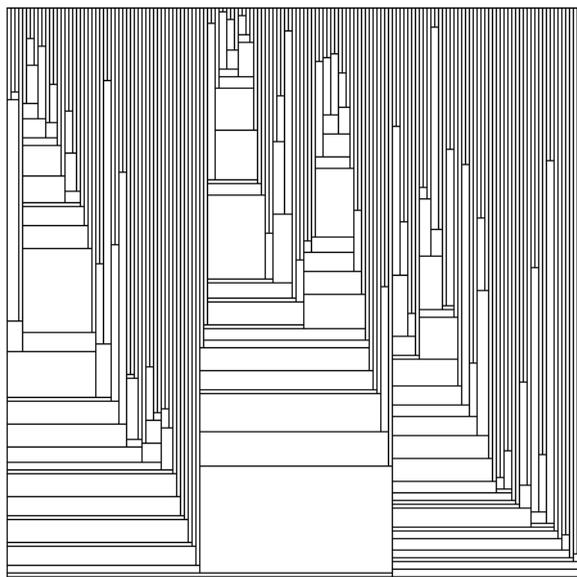


Figure 7.4. Mosaic display, Example 2D-1.

Actual and effective cluster number

With N data points the number of clusters at level n is equal to $N-n+1$; therefore a plot of number of clusters versus n gives a straight line with a slope of -1 . This is the dotted line in [Figure 7.3](#). The “effective number of clusters,” defined in [Chapter 5](#), reflects both the actual number of clusters and their relative sizes. The effective number is always less than or equal to the real number of clusters, and is equal to the actual number only when the sizes are equal. The fact that at level 148 both numbers are equal to 3 means is consistent with the existence of three clusters of equal or nearly equal size. At level 149 we have two clusters, one of 100 points (consisting of sets A and B) and one of 50 points (set C), but 100 and 50 are close enough that the effective number is close to the actual number, two.

In contrast, observe the small change in effective cluster number that occurs between levels 147 and 148. The actual number of clusters decreases by one, as it must. The fact that the effective number of clusters decreases by much less means that the two clusters that are being joined here are of very different sizes. The mosaic will show this in detail.

Mosaic

In a mosaic display, a given horizontal row represents a given clustering level, and vertical columns represent data points, ordered from left to right in the generic ordering. Holding down any mouse button on the display itself exhibits the clustering level, cluster and data point indices corresponding to the location on the display beneath the cursor; this labeling information is absent from [Figure 7.4](#).

The bottom row represents clustering level 150. The fact that it goes all the way across the bottom of the diagram without vertical interruptions means that all the conformations are joined into a single cluster. Going up one level, to 149, there is a single vertical “divider” two thirds of the way across the diagram. This separates the two clusters forming here, the left side consisting of sets A and B, and the right side consisting of set C. Going up one more level, to 148, we see that there are three clusters of equal size. At this level, as at level 150, the effective and actual numbers of clusters were equal, as described above and in [Chapter 5](#).

Going up still one more level, to 147, we see that there are four clusters present, one of which consists of a single data point. We noted earlier that the effective cluster number changes little going from level 147 to 148, and that this means that the two clusters agglomerating at level 148 are of very different sizes. The mosaic makes it clear that in fact a cluster of 49 data points is joining one containing a single data point at this level.

Whenever a single vertical column extends nearly all the way to the bottom of the mosaic, as the rightmost column does, it means that this data item is in some sense an outlier; it remains isolated from the other clusters as they are assembled. Mouse clicks reveal that the column on the far right corresponds to the 90th data point in the input file; examining the coordinates in

the input file, we see that this data point is the one with the largest y-coordinate on the diagram (i.e., the one closest to the top of Figure 7.1). This data point joins the adjacent cluster (set C) just before sets A and B agglomerate.

To what extent is data-point 90 an outlier? The mosaic doesn't tell us this, but the plot of minimum separation ratio does. Figure 7.2 shows that this value is close to unity at level 147, which means that this outlier will join an existing cluster at a threshold distance close to the current threshold distance. In other words, the critical threshold value defining the current clustering is only a little bit smaller than the one which will join this outlier to an existing cluster. Thus this point is not very far from a previously existing cluster. If this point had a y-coordinate of, say, 14, rather than about 10.5, then the minimum separation ratio at level 147 would have been high, and we would have concluded that this point was quite isolated; we might also have inferred that the entire ensemble consisted of four, rather than three clusters, the fourth being this isolated point.

Generic-order map

The map (Figure 7.5) is simply a visual representation of the distance matrix, with the darker shading representing the shorter distances. On a color display, the shorter distances are represented by the blue end of the spectrum. This figure exhibits the matrix in the generic ordering; note that the dark regions form three well-defined on-diagonal squares. This reflects the three well-defined clusters in the data. If you operate the program interactively, you can also examine the map with the data points in the input order, which was random for this data set.

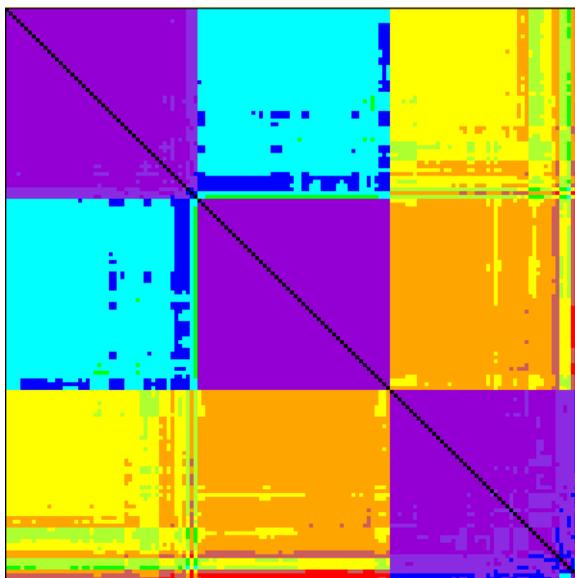


Figure 7.5. Generic-order map display, Example 2D-1.

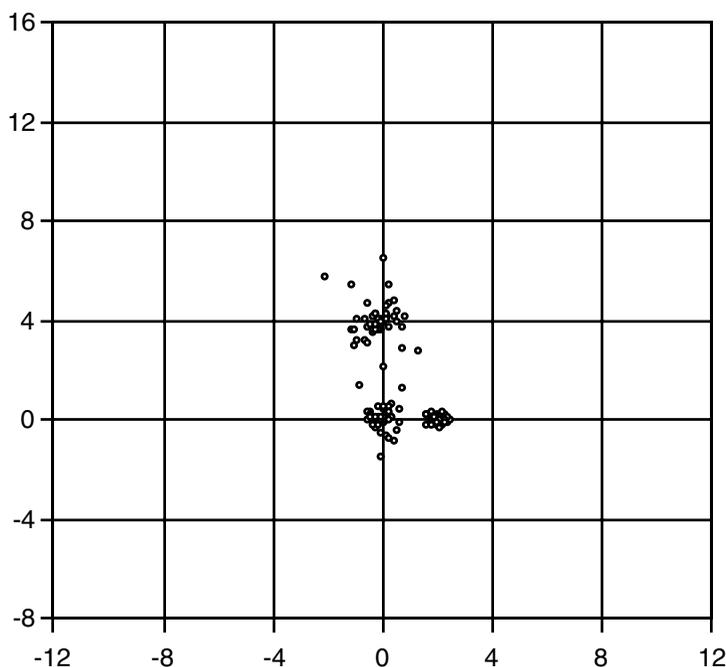


Figure 7.6. Two-dimensional point distribution, Example 2D-2.

The input-ordered map appears random, meaning that in the input ordering there is no tendency for data points close to each other in space to be close to each other in the input file; as we shall see, in real chemical examples, there may be some tendency for the input file to already be partially clustered.

It turns out that in our first three examples, set A is in the middle of the map, set B is at the lower left and set C is at the upper right.

7.1.2 Example 2D-2

Minimum separation ratio

There are no significant peaks in this statistic (Figure 7.7) at high clustering level, and indeed there are no well-defined clusters, even in the intuitive sense, as Figure 7.6 makes clear. However, intuitively, there seems to be some clustering going on. For example, by removal of just a few points, distinct clusters would appear. So the question is whether any of the displays that XCluster exhibits can be used to detect this fact, given that the minimum separation ratio does not. The behavior of the effective number of clusters and the appearance of the mosaic does provide such insight, but it is worth noting at this point that high values of the minimum

separation ratio at high clustering levels unequivocally indicate the presence of distinct clusters, but that the absence of such an indication does not rule out the possibility of the more subtle sort of clustering (using the word in its intuitive sense) that this example exhibits. Work is in progress to better identify the three-fold clustering in situations like this one.

Actual and effective cluster number and mosaic

The effective cluster number, shown in Figure 7.8, has a “staircase” appearance at high clustering level; there is a steep drop from a value of about 3.5 to about 2.2 at level 145, where there are actually 6 clusters present, then a drop from about 2.0 to about 1.1 at level 148, where 3 clusters are present. Recall that if there are n large clusters of approximately equal size present, plus a few additional clusters, each containing only a few points, the effective cluster number will be a small amount larger than n .

We know that at level 150 only a single cluster is present; reasoning from right to left in the figure, the slight increases in effective number at levels 149 and 148 mean that we have one large cluster present, together with some small ones. The mosaic (Figure 7.9) shows that at level 148 we have two clusters of one point each, together with a large cluster of the remaining points, and that at level 149 we have a large cluster and a single outlying point.

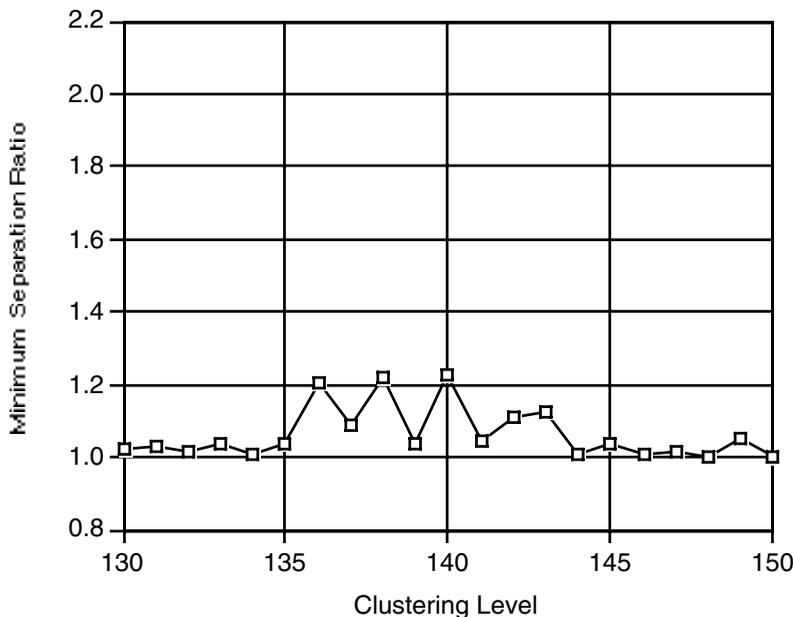


Figure 7.7. Minimum separation ratio, Example 2D-2.

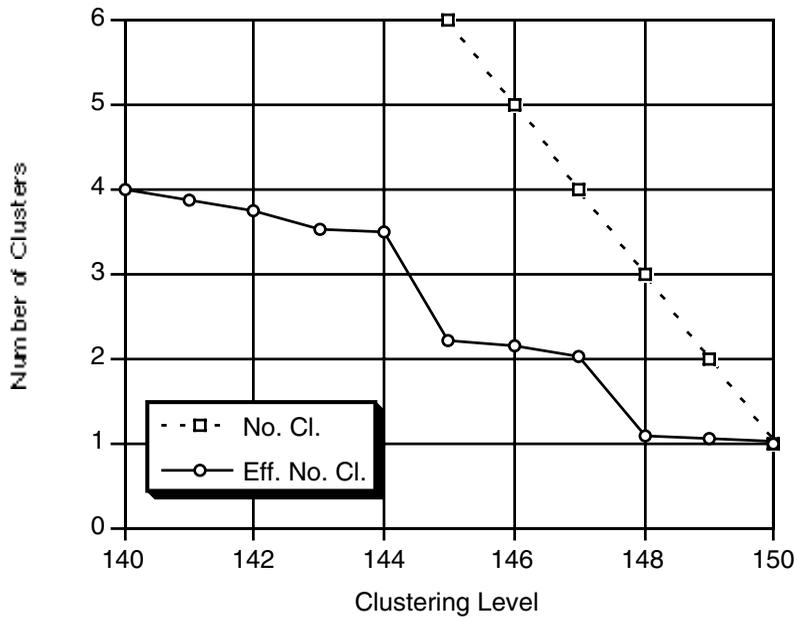


Figure 7.8. Actual and effective cluster number, Example 2D-2.

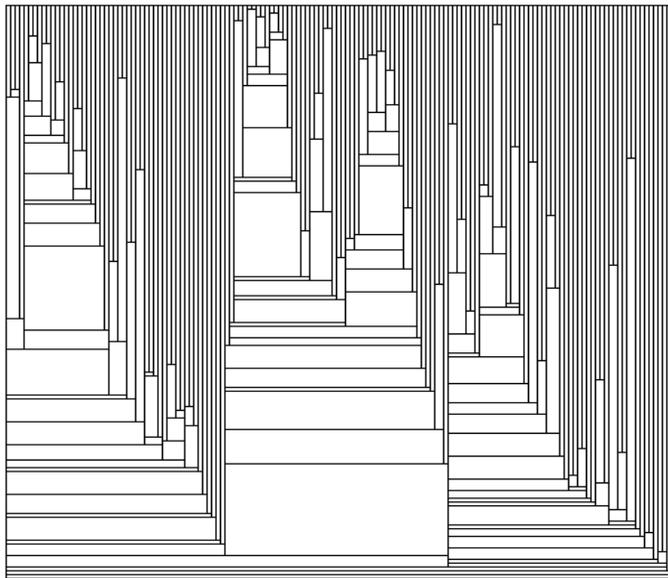


Figure 7.9. Mosaic display, Example 2D-2.

The same reasoning process can be continued moving to the left in [Figure 7.8](#). The effective number of clusters increases by about one going to level 147. This indicates the breaking up of the large cluster into two clusters whose sizes are of the same order of magnitude. From the mosaic, one of them contains 100 data points and the other 48 points.

Continuing to the left, we see another rise of about one in effective cluster number from level 145 to 144. The mosaic shows that this is associated with the breakup of the 100-member cluster into two clusters of 50 each. Proceeding to level 140, we see a gradual rise in N_{eff} to about 4, corresponding to a “chipping away” of data points from the large clusters. If we had been presented with an effective cluster number of 4 without having gone through this reasoning process, we could not have known whether this value resulted from the presence of four large clusters, with perhaps a few outliers, or from some other combination of cluster sizes. The reasoning process demonstrates that at level 140 there are three large clusters, together with some small ones. This is confirmed by the mosaic.

Thus the staircase appearance of N_{eff} at high clustering level has informed us that our ensemble of points consists of three large clusters, together with additional points which lie between the clusters, keeping the minimum separation ratio low.

Generic-order map

The map ([Figure 7.10](#)), like the mosaic, has an overall appearance of “threeness;” however, in comparison with [Figure 7.5](#), some of the on-diagonal blocks have lost areas of high density, and some of the off-diagonal blocks have gained areas of high density. A high density point in an off-diagonal block means that there are two structures, one in the “row” on-diagonal block and one in the “column” on-diagonal block, which are close in distance. By our definition of a cluster, such a pair will cause the two on-diagonal blocks, if they are clusters already, to agglomerate into a single cluster.

It is also interesting that the block representing set C has a dark “sub-block” in its lower left corner. The sets were sampled from Gaussian distributions, so their respective point densities are greatest at their centers. In [Figure 7.10](#) the distance delimiting the two darkest shadings happens to fall within the distances sampled within set C, and the dark sub-block represents a somewhat arbitrary demarcation of that set’s “inner core.”

The shadings (or colors, on a color display) are chosen to evenly span the distances exhibited within the entire ensemble of points. Since the overall distance range is smaller in Example 2D-2 than in Example 2D-1, point set C, for example, looks different in the two maps, despite the fact that this set’s list of internal distances is identical in these two examples, and also for Examples 2D-3 and 2D-4.

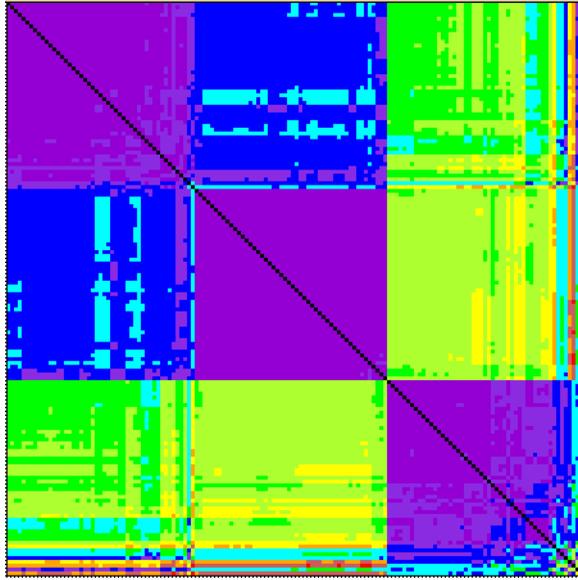


Figure 7.10. Generic-order map display, Example 2D-2.

7.1.3 Example 2D-3

Minimum separation ratio

Again, this statistic (Figure 7.11) exhibits no significantly separated clusters at high clustering levels.

Actual and effective cluster number and mosaic

An analysis similar to that given for Example 2D-2 shows that at high clustering levels this ensemble consists of two large clusters with some outliers and connecting points. This is clear both in the single-step staircase appearance of the N_{eff} plot (Figure 7.12) and in the mosaic (Figure 7.13).

Generic-order map

The generic-order map (Figure 7.14) has significant darkly shaded area in the off-diagonal area connecting the lower-left on-diagonal block (set B) and the central on-diagonal block (set A). That these two blocks form a single cluster at high clustering level is apparent in the mosaic; map and mosaic are both assembled using the same generic clustering order. The point distribution (Figure 7.15) bears out this interpretation.

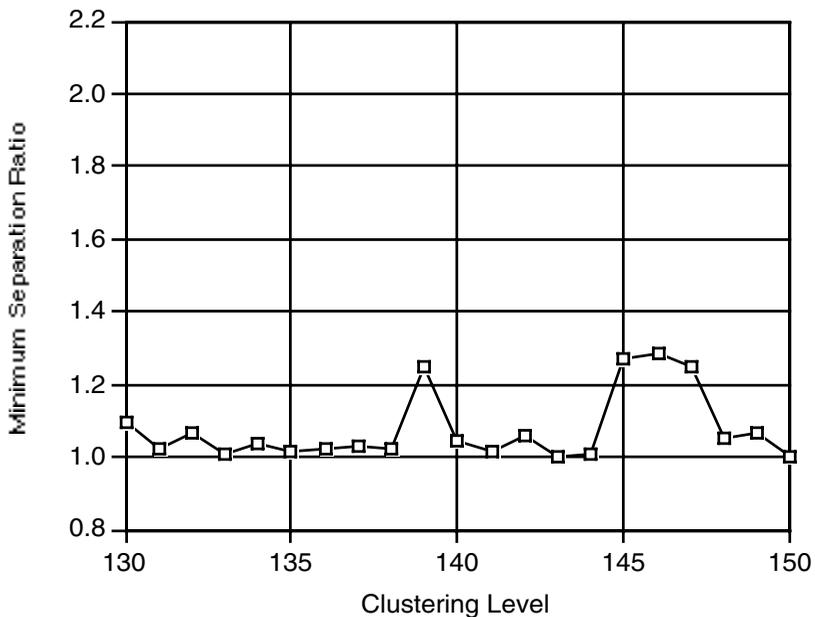


Figure 7.11. Minimum separation ratio, Example 2D-3.

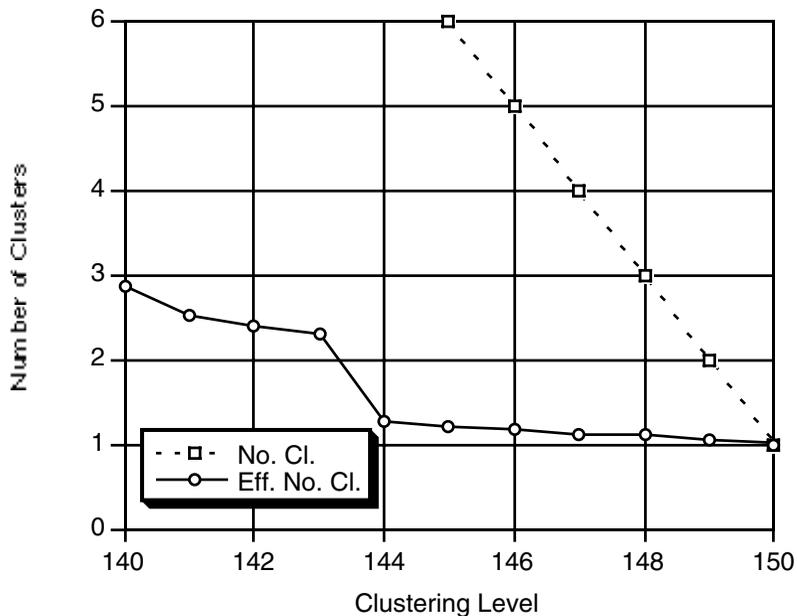


Figure 7.12. Actual and effective cluster number, Example 2D-3.

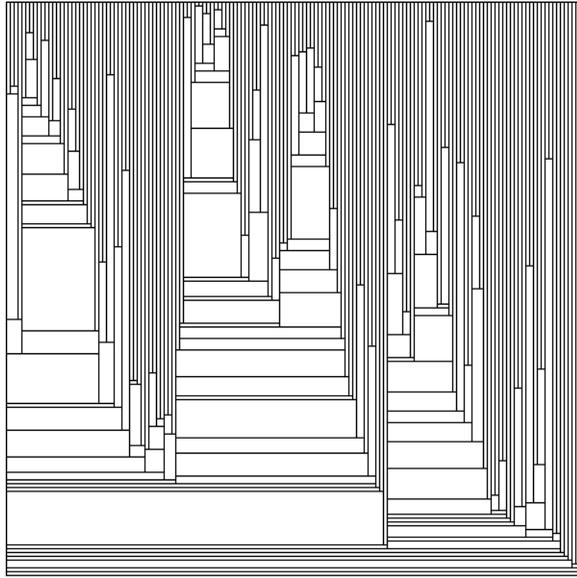


Figure 7.13. Mosaic display, Example 2D-3.

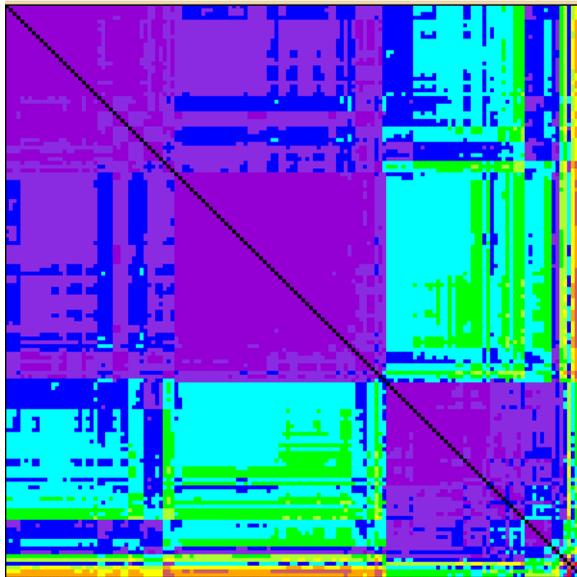


Figure 7.14. Generic-order map display, Example 2D-3.

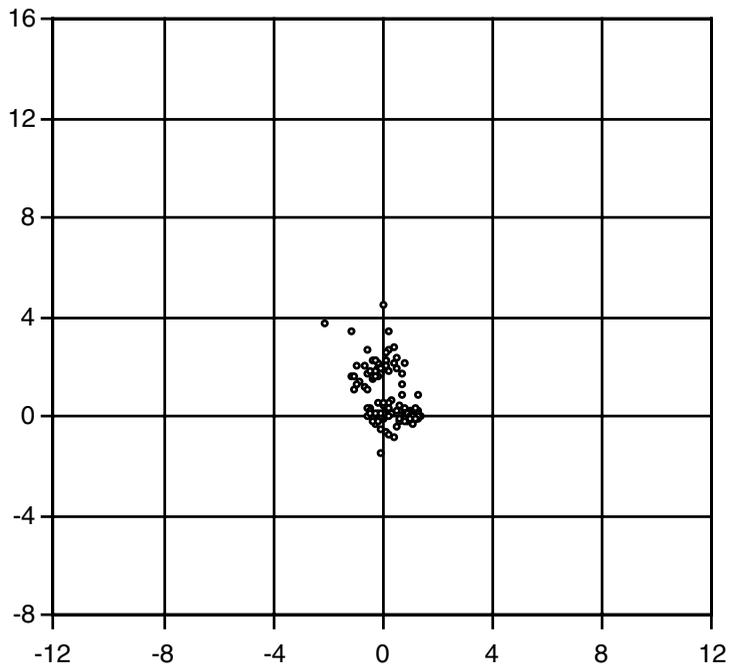


Figure 7.15. Two-dimensional point distribution, Example 2D-3.

7.1.4 Example 2D-4

Based on what has been said about the preceding three examples, readers should be able to discern that at high clustering levels, there is only a single cluster. This is tantamount to saying that there is not a great tendency for the data to cluster at all, as the point distribution (Figure 7.20) shows. This is especially clear from the mosaic (Figure 7.18), but note also that N_{eff} (Figure 7.17) has no abrupt “jumps” in the region of high clustering level. This indicates that in this region a large cluster is growing by accretion of small ones.

Some subclustering is still apparent in the map (Figure 7.19); however, the map presents a more chaotic appearance than the maps of the earlier examples, and at high clustering levels the off-diagonal areas connect most of the on-diagonal blocks into single clusters.

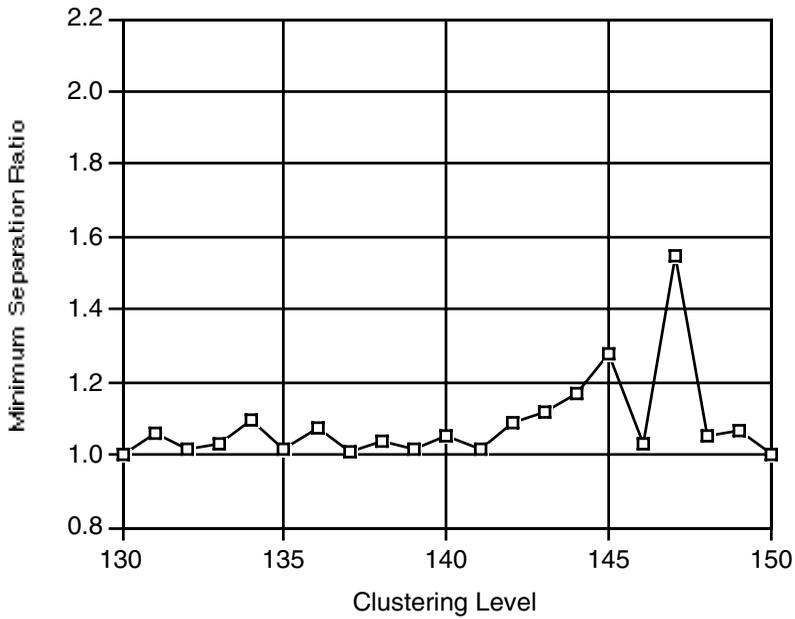


Figure 7.16. Minimum separation ratio, Example 2D-4.

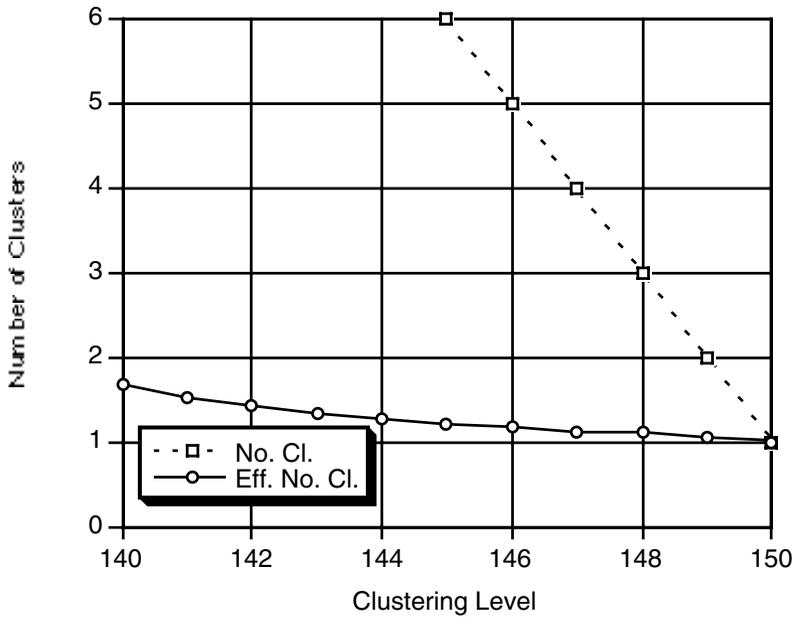


Figure 7.17. Actual and effective cluster number, Example 2D-4.

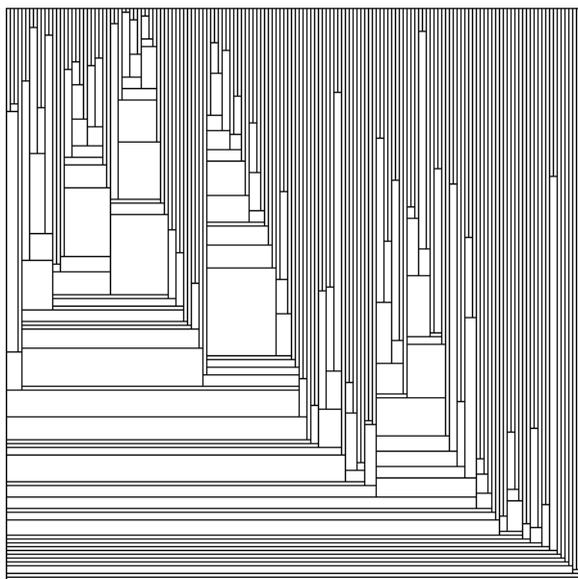


Figure 7.18. Mosaic display, Example 2D-4.

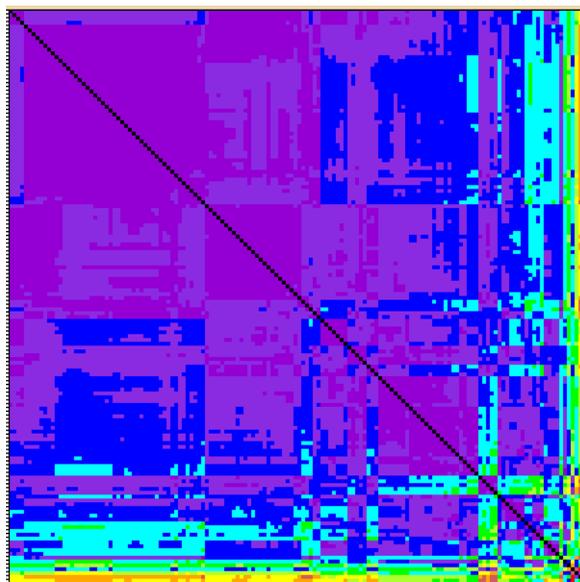


Figure 7.19. Generic-order map display, Example 2D-4.

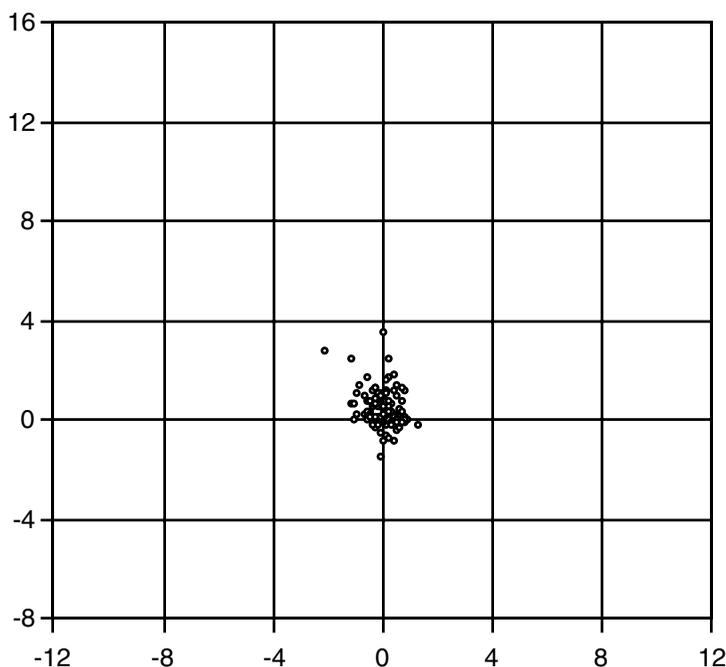


Figure 7.20. Two-dimensional point distribution, Example 2D-4.

7.2 Conformational Search Examples

7.2.1 Roseotoxin-B, a Cyclic Peptide

This peanut mold toxin contains a 19-membered ring. The ring contains five peptide bonds, and these as well as the other torsions were allowed to vary during a conformational search. Since energetic factors constrain the peptide bonds to torsional angles close to 0 and 180 degrees, the ring exhibits less conformational flexibility than, for example, a 19-membered alkyl ring would exhibit. MacroModel (BatchMin) found 192 unique local minima, using the Amber* force-field with GBSA/chloroform solvation.

Because of the stiffness induced by the peptide bonds, the ring conformations cluster strongly into two groups. These groups are characterized by canonical values of the C-alpha—C-beta torsion of the sole beta-alanine in the structure.

Here we exhibit the results of an XCluster run employing the Trms: option applied to this dihedral angle. In a separate run, not exhibited here, Trms: was applied to the full set of ring torsions. In that run, the plot of minimum separation ratio against clustering level was similar to that exhibited in [Figure 7.21](#), except that the peak value at clustering level 191, where two

clusters are present, was about 3.7 instead of the value 34.0 exhibited in the figure. Bearing example 2D-1 in mind, a value of 3.7 at high clustering level is significant; a value of 34.0 is remarkable.

The membership of the two clusters at clustering level 191 was identical for the two runs, which demonstrates that the C-alpha—C-beta torsion indeed characterizes the clusters. Figure 7.22 shows the structures in the cluster file generated by XCluster at clustering level 191. XCluster has superimposed the structures as well as it can based on the distance-matrix option it has been given. For Trms:, XCluster uses the defining atoms of the torsion(s) as the basis for superposition. Since we have specified only this single torsion, it is especially clear in the illustration that the two clusters are characterized by gauche-plus and gauche-minus values of this dihedral angle. When we ran XCluster using all ring torsions, all ring atoms were used for the superposition. In visualizing the resulting two-cluster .cls file, the defining nature of the C-alpha—C-beta torsion was still apparent, but not as clearly as in Figure 7.22.

The two clusters at level 191 have memberships of 85 and 107 structures. Since these numbers are of the same order of magnitude, the effective number of clusters is very close to two, as Figure 7.24 demonstrates. The mosaic, Figure 7.23, shows that at the penultimate clustering level there are two clusters of approximately equal size, which then join at the highest level. This is particularly clear in the mosaics which are drawn by cluster threshold value and the logarithm of the cluster threshold value (Figure 7.25), and is what we expect, based on the previous discussion.

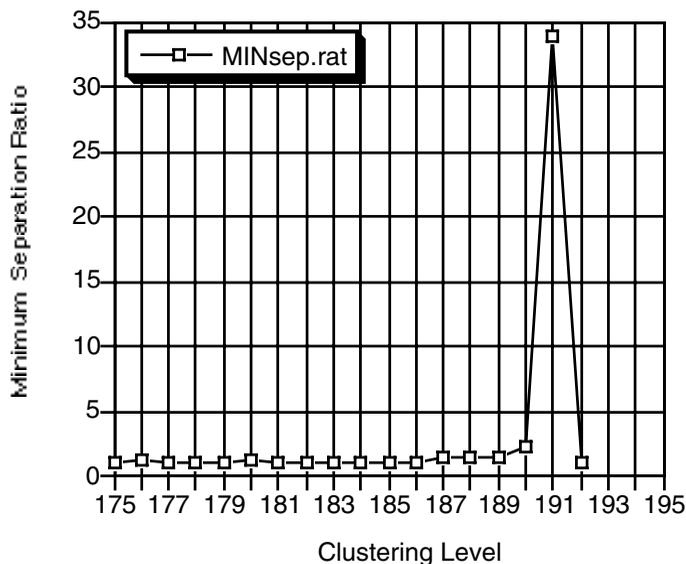


Figure 7.21. Minimum separation ratio, roseotoxin-B search.



Figure 7.22. Clustered molecular display, roseotoxin-B search.

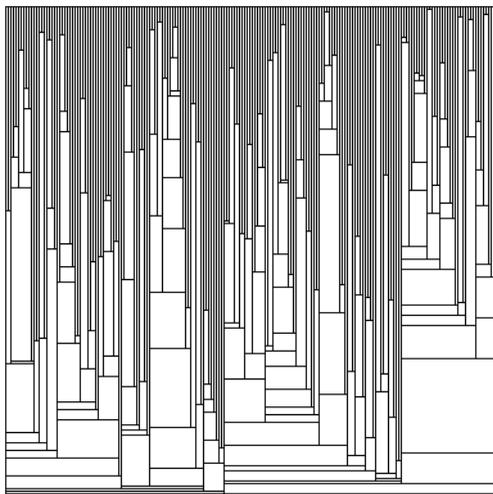


Figure 7.23. Mosaic display by clustering level, roseotoxin-B search.

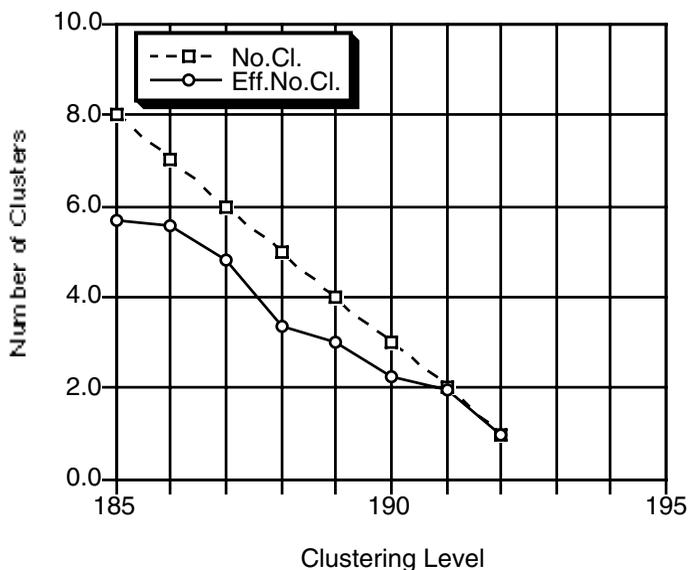


Figure 7.24. Actual and effective cluster number, roseotoxin-B search.

The scale of Figure 7.25(a) is such that many levels are compressed into single horizontal rows. Note the deletion of the bottom bounding horizontal line as discussed in Chapter 3. In Figure 7.25(b) note that the upper and lower bounding horizontal lines are absent as discussed in Chapter 3. The spacing between adjacent tick marks on the right is proportional to the height of the graph in Figure 7.21; the greatest value of the minimum separation ratio occurs at the penultimate level, where two clusters are present.

The generic-ordered map Figure 7.26(a) exhibits two well-separated on-diagonal blocks, which is what we expect from strong two-fold clustering. Figure 7.26(b) illustrates the map in the input-ordered sequence. On a large scale, this map appears chaotic, and this tells us something about the data. The .out file from a MacroModel conformational search lists the conformations in order of increasing energy. The fact that in the input sequence the map does not exhibit strong block-diagonal form indicates that for roseotoxin-B, structures that are similar in energy do not have similar ring conformations.

A closer examination of the input-ordered map, on a smaller scale, reveals that the map exhibits small on-diagonal blocks containing two to four structures each. Using mouse-clicks to identify the members of such groups, then examining the structures, reveals that such structures differ significantly only in torsions about side-chain bonds. That such conformations occur in adjacent groups in the .out file tells us that altering the conformation about these bonds affects the overall energy very little. The fact that the adjacent groups form on-diagonal low-distance blocks in the input-ordered map implies that the torsion about which we are clus-

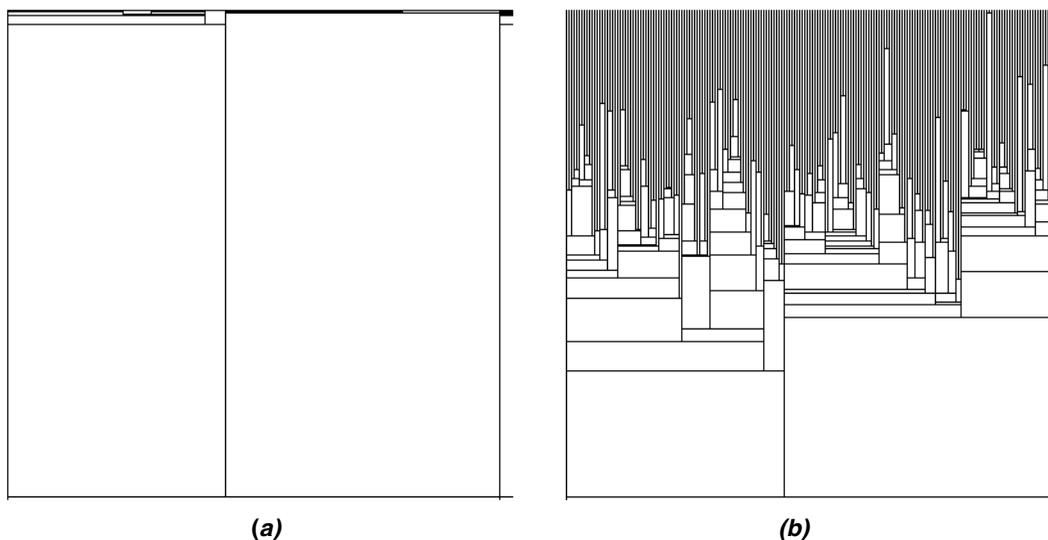


Figure 7.25. (a) Mosaic display by cluster value, and (b) Mosaic display by the logarithm of the cluster value, roseotoxin-B search.

tering is not much affected by these off-ring torsions. From this, together with the fact that the generic-ordered map appears substantially the same when all ring torsions are compared, it can be inferred that the off-ring torsions are only weakly coupled to the ring energetics.

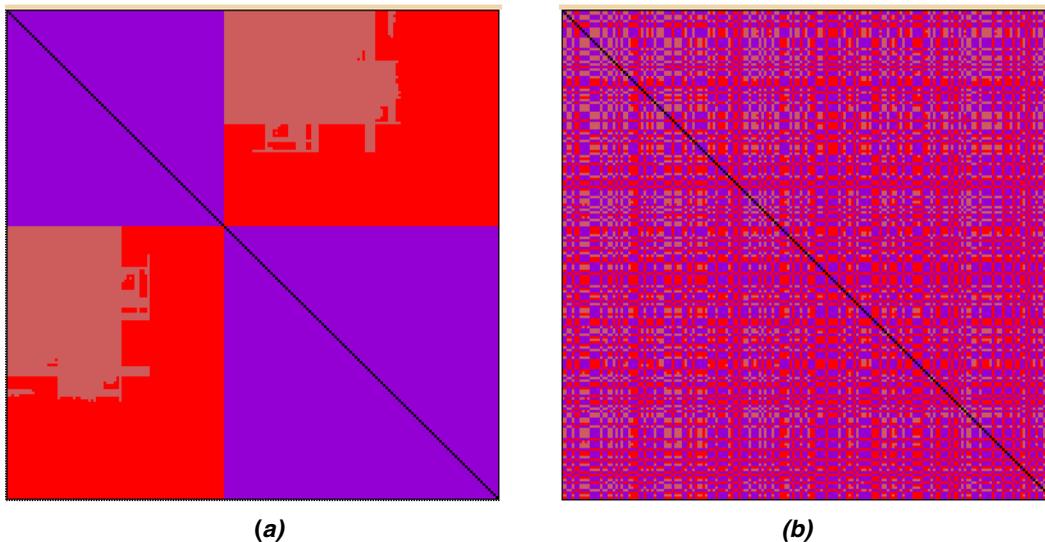


Figure 7.26. (a) Generic-order map display, and (b) Input-order map display, roseotoxin-B search.

7.2.2 Cycloheptadecane

Here is an example of an ensemble of conformations that exhibits little or no clustering.

A conformational search was performed on cycloheptadecane in MacroModel, allowing all symmetry operations to be used in structure comparison, and using the MM3 force-field without solvation for minimizations. 132 unique minima were found, and are shown in [Figure 7.27](#). These were processed using XCluster, applying the `Trms:` command to the set of all ring torsions. In the cluster analysis, as in the search, all symmetry operations were performed in comparing conformations, specifying all ring atoms using the `Symatom:` command, number-order rotation using `Rotate: 17`, number-order reflection using `Reflect: ring` and enantiomerization using `Enant:.` One effect of the symmetry operations is that each of the $132 \times 131/2 = 8646$ pairs of conformations has to be compared $4 \times 17 = 68$ ways; thus the construction of the distance matrix is considerably more time-consuming than it was for roseotoxin-B, which had 192 conformations, but no symmetry.

In contrast to roseotoxin-B, the cycloheptadecane conformations exhibit little or no clustering. This is in accord with the well-known fact that a 17-membered alkane ring is “floppy.” The lack of clustering is apparent everywhere: [Figure 7.31](#) exhibits no significant peaks in the minimum separation ratio at high clustering levels; the effective number of clusters ([Figure 7.28](#)) changes only slowly in the same region; at high clustering levels, the mosaic ([Figure 7.29](#)) consists of a single large block which grows by accretion; and no large on-diagonal blocks appear on the map ([Figure 7.30](#)).

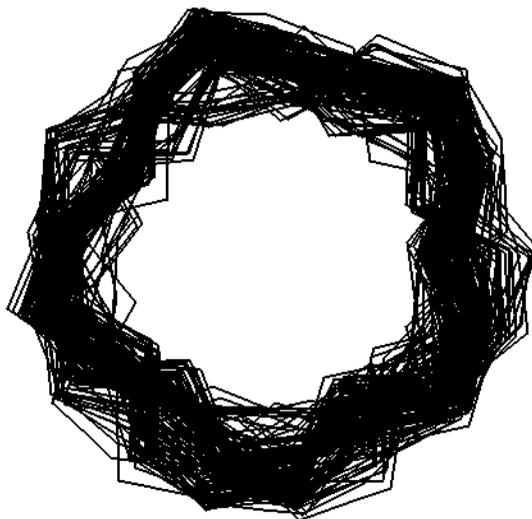


Figure 7.27. Clustered molecular display, cycloheptadecane search.

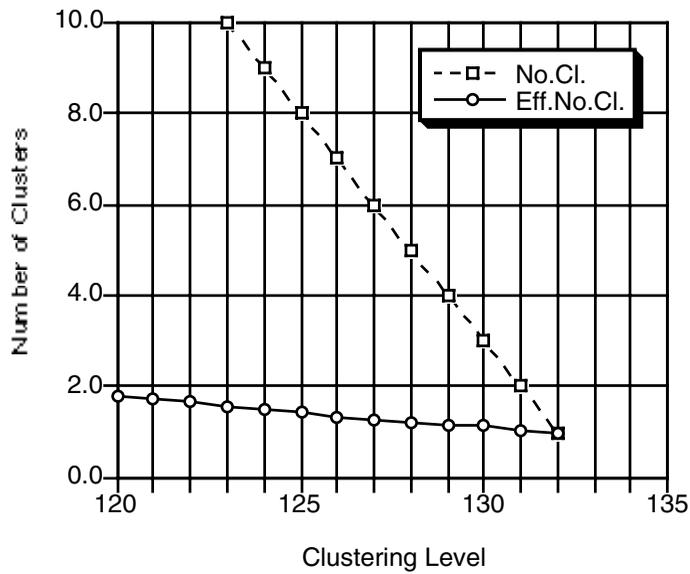


Figure 7.28. Actual and effective cluster number, cycloheptadecane search.

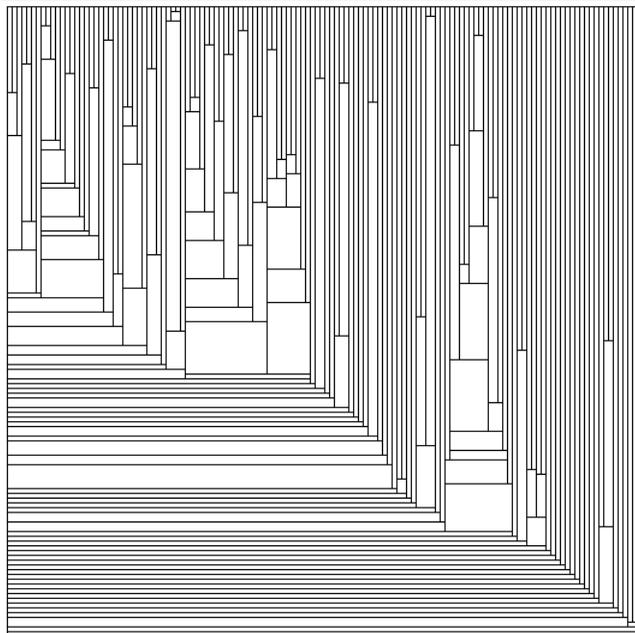


Figure 7.29. Mosaic display, cycloheptadecane search.

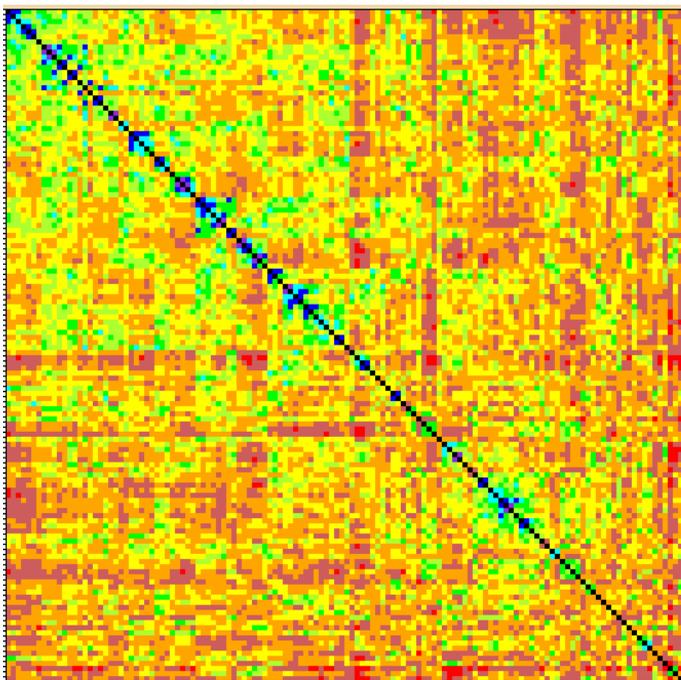


Figure 7.30. Generic-order map display, cycloheptadecane search.

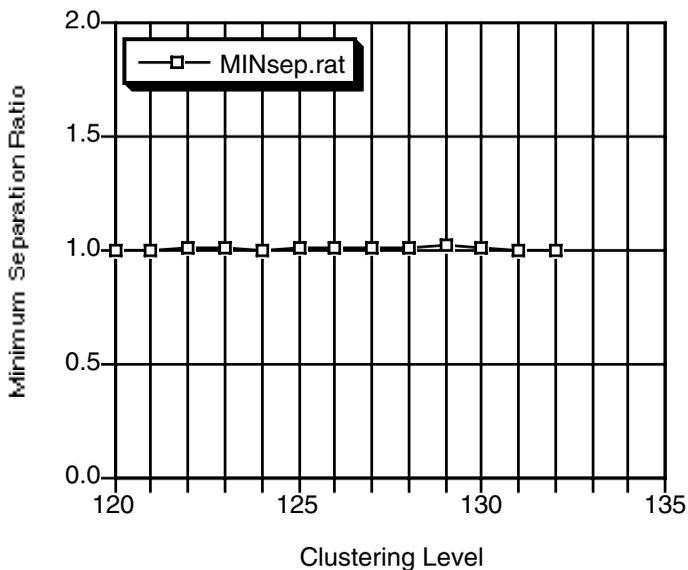


Figure 7.31. Minimum separation ratio, cycloheptadecane search.

7.3 Molecular Dynamics of Pentane

In this discussion we use the symbols “t” for the trans conformation of a torsion, “+” for gauche-plus and “-” for gauche-minus.

Stochastic dynamics was run on n-pentane, initially in the tt conformation. We used a united-atom model with the Amber* force-field, a time-step of 1 fs, a temperature of 350 K, and a total run time of 200 ps, saving a structure every picosecond. No solvation was model was employed. We chose the force field and atomic representation not for perfect suitability to this system, but rather to generate a didactic example rapidly.

XCluster was run specifying symmetry operations of `Reflect: chain` and `Enant:.` At this level, the nine conformational minima obtained by allowing each torsion to take on its three canonical values coalesce into four groups, namely: (++, --), (+-, -+), (tt), and (t+, t-, +t, -t). We would expect a fully convergent dynamics run to explore the basins associated with all four of these minima. The energies of the minima are 4.09, 10.22, -0.83 and 1.77 kJ/mol.

Figure 7.32 indicates good clustering at level 198, where three clusters appear. The generic map (Figure 7.36 on page 70) also clearly indicates three clusters, and the plot of effective cluster number versus clustering level (Figure 7.33) indicates that when three clusters coalesce to two, between levels 198 and 199, the two coalescing clusters are large and of similar size, whereas when this combined cluster joins the remaining cluster, this remaining cluster is small in comparison. This interpretation is confirmed by the mosaic (Figure 7.34).

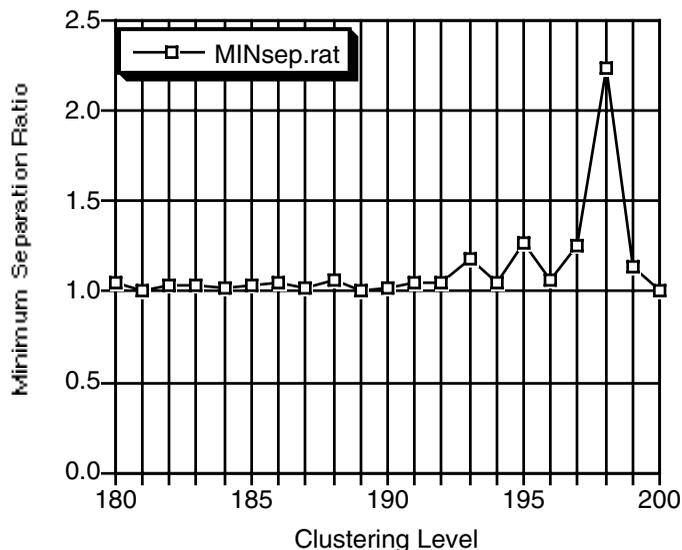


Figure 7.32. Minimum separation ratio, pentane dynamics.

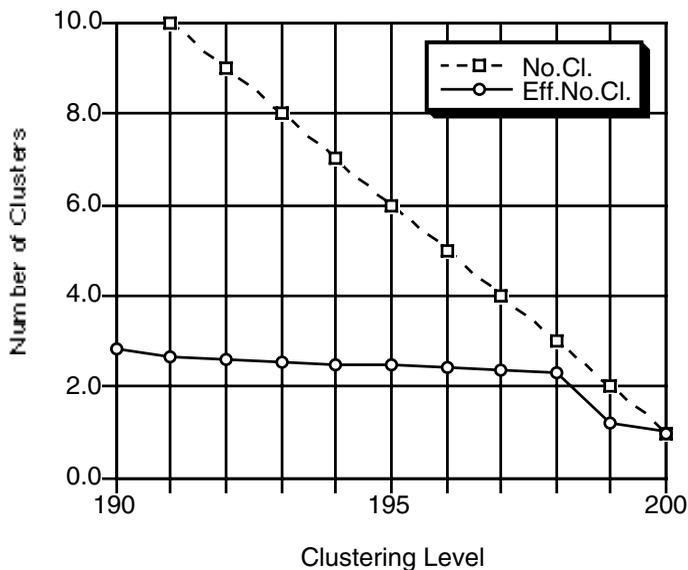


Figure 7.33. Actual and effective cluster number, pentane dynamics.

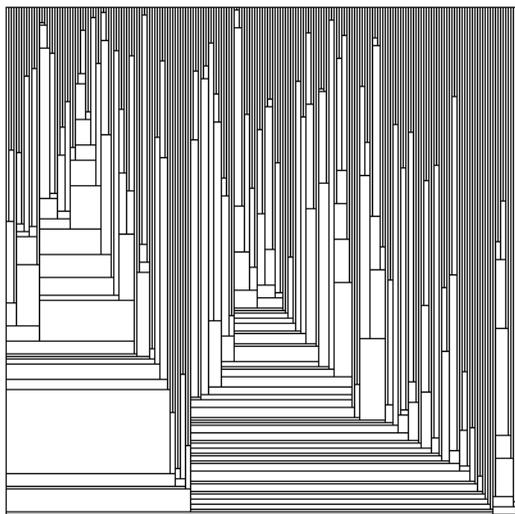


Figure 7.34. Mosaic display, pentane dynamics.

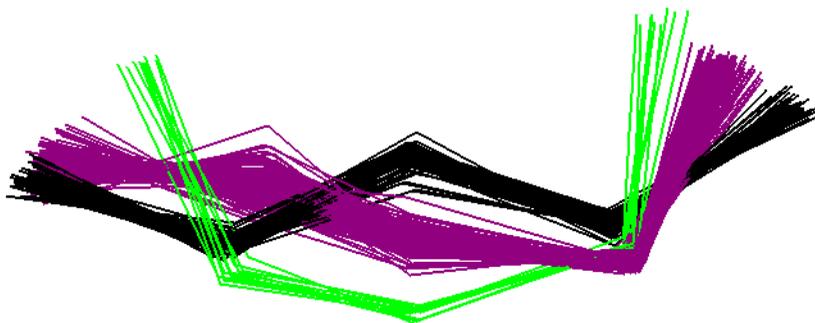


Figure 7.35. Clustered molecular display, pentane dynamics.

We used XCluster to write out a file of clustered conformations at level 198; we then generated PostScript output, shown in Figure 7.35. The three clusters are clearly visible. The superposition performed by XCluster when it writes a cluster file is identical at all clustering levels; what varies is the color scheme, which is translated into a shading scheme in print. The overall appearance of Figure 7.35 is in accord with a division of the ensemble into three clusters, and the shading scheme demonstrates that XCluster’s selection of the three clusters is in accord with that of our intuition.

Close examination of the structures found demonstrates that of the four conformational classes we predict, the one corresponding to $(+, +, -)$ does not occur in the output, despite the fact that $(+, -, -)$, which is higher in energy at the minimum, does occur. This is simply an accident of the random numbers thrown in the course of this particular run.

It is instructive to examine the map in the input-ordered sequence (Figure 7.36(b)). The structures in the .out file that MacroModel produces in a dynamics run are, of course, in time-ordered sequence. Note the large on-diagonal blocks in this figure. These indicate that the molecule spends some time exploring the basin of attraction associated with a single minimum before “flipping” into another basin. This is a well-known property of dynamics on complicated energy surfaces.

Several other well-known dynamical phenomena can be seen in the input-ordered map. First, there are many off-diagonal blocks, in addition to the on-diagonal blocks. Each off-diagonal block lines up with two on-diagonal blocks: one in the horizontal direction and one in the vertical direction. What this tells us is that the structures in these two on-diagonal blocks are similar; in fact, the shading (color coding) in the figure tells us that the structures within each of these two on-diagonal blocks are as similar to those within the other block as they are to those within the same block. We conclude that the two on-diagonal blocks explore the same basin.

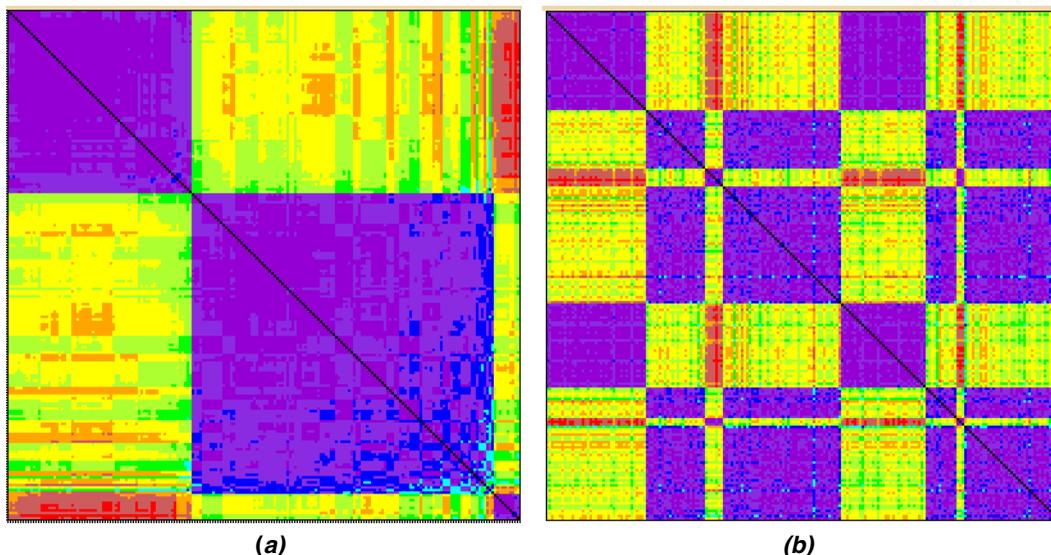


Figure 7.36. (a) Generic-order map display, and (b) Input-order map display, pentane dynamics.

To discuss one particular instance of this phenomenon, the off-diagonal block at the bottom of Figure 7.36(b) connects the first and fifth on-diagonal block. We initialized the molecule in the *tt* conformation: this figure tells us that after exploring several other basins, the molecule returns to the basin surrounding the *tt* minimum and stays there for some time, before flipping into the region surrounding another local minimum.

Every on-diagonal block is associated with at least one off-diagonal block. This means that the basin of every local minimum encountered during the run is encountered more than once. If all basins are revisited many times, we can have some confidence that our run has converged—i.e., that it has explored the conformational space sufficiently completely to give us faith in ensemble properties computed from the run's statistics. As mentioned earlier, however, our a priori understanding of this simple system is sufficient to allow us to conclude that this run has not, in fact converged; the basin surrounding the $(+, -)$ minimum has not been encountered. See, however, the remark at the end of this section.

Another well-known dynamical property is evident in the third and seventh (second-to-last) on-diagonal blocks. These blocks are very small, implying that the molecule does not spend much time there. These blocks are also correlated, through off-diagonal elements, only with each other, and represent the highest-energy basin encountered, the $(+, -)$ basin. The low population of this species is presumably explicable on the basis of its energetics, but note that both of these blocks are embedded within larger blocks. This implies that while the molecule is exploring some other basin, it visits this high-energy basin for a short period of time, then

returns to the basin from whence it came. This is a common observation in molecular dynamics; it is as if the molecule, while in a high-energy form, retains some “memory” of its recent journeys.

We can also say a little bit more about this specific instance. The large blocks in which these small blocks are embedded are themselves correlated through an off-diagonal block, showing that the high (+-, -+) region is encountered only as an interlude during the exploration of a particular other basin. This basin happens to be that associated with the minimum (+t, -t, t+, t-). Examination of all the transitions in [Figure 7.36\(b\)](#) reveals that each transition involves the flip of only a single torsion from one class to another, which is the situation we expect to observe most often. In order to demonstrate this rigorously for the run, we would need to carefully examine the input-ordered map that XCluster would produce from this same data set without the specification of symmetry equivalencies. This map would be both richer and more complex than [Figure 7.36\(b\)](#), and would likely demonstrate the lack of convergence noted earlier, by means of the appearance of some on-diagonal blocks only once.

Command Reference

Comment:

Insert user comments into the command file.

Sfile: *filename*

Use the MacroModel or Maestro molecular structure file called *filename* as the input to a subsequent distance matrix generating command. Any output files created by later commands will be in whichever format the Sfile: file was in.

Dfile: *filename*

Use the distance list in the file *filename* as input to the program, which will then be run in dfile mode.

Arms: heavy | all | *atom-list*

Calculate the distance matrix using pairwise atomic R.M.S. displacement between pairs of conformations as the distance criterion. The R.M.S. displacement is calculated after optimal rigid body superposition.

Nrms: heavy | all | *atom-list*

Calculate the distance matrix using pairwise atomic R.M.S. displacement between pairs of conformations as the distance criterion. The R.M.S. displacement is calculated “in-place;” i.e., without rigid body superposition.

Trms: *torsion-list*

Calculate the distance matrix using as the distance criterion the R.M.S. of the differences between corresponding torsion angles in a pair of structures. Calculate a difference between torsion angles “the short way around;” e.g., the difference between +170° and -170° is 20°.

Symatom: *atoms-list*

Specifies the list of atoms to which symmetry operations will be applied.

Mmsym:

Instructs the program to automatically perceive local and global number-order symmetry.

Enant:

Enantiomerize coordinates during comparison.

Rotate: *fold*

Perform a *fold*-fold numbering system rotation during the comparison procedure in the Arms:, Trms: or Nrms: command.

Reflect: chain | ring

Perform a numbering system reflection during the comparison procedure in the Arms:, Trms: or Nrms: command.

Cluster:

Create clusters at *N* clustering levels using the values in the distance matrix; calculate figures of merit for the clusterings, and print these to the log file.

Thresh: *level*

Write out the membership of the a “cluster” at a given clustering level. Also print statistics for each cluster at this level.

Writecls: *level filename cluster_num*

Write a structure file containing molecular conformations superimposed and colored according to cluster membership. If *level* is negative, use the clustering level that has *llevel* clusters. *llevel* should not be greater than the number of structures. If *cluster_num* is missing or specified as “all” then all clusters at clustering level *level* are written. Otherwise *cluster_num* specifies the number of a single cluster to write to the file. This command also prints the same information as the Thresh: command.

Writerep: *level filename cluster_num*

Write a structure file containing a single representative molecular conformation for each cluster. If *level* is negative, use the clustering level that has *llevel* clusters. *llevel* should not be greater than the number of structures. If *cluster_num* is missing or specified as “all” then all clusters at clustering level *level* will be written. Otherwise *cluster_num* specifies the number of a single cluster to write to the file. This command also prints certain statistical data to the .clg file and to the message window.

Writeavg: *level filename cluster_num*

Write a structure file containing a single average molecular conformation for each cluster. If *level* is positive, use the specified clustering level. If *level* is negative, use the clustering level that has *llevel* clusters. *llevel* should not be greater than the number of structures. If

cluster_num is missing or specified as “all” then all clusters at clustering level *level* are written. Otherwise *cluster_num* specifies the number of a single cluster to write to the file.

Writedst: *filename*

Write out the current distance matrix in the input order to file *filename*.

Writelead: *level filename clust_num*

Write a structure file containing the first molecular conformation from the input structure file for each cluster. For files containing conformers produced by a MacroModel conformational search this conformation is usually the lowest energy conformation for that cluster. If *level* is positive, use the specified clustering level. If *level* is negative, use the clustering level that has *llevel* clusters. *llevel* should not be greater than the number of structures. If *clust_num* is missing or specified as all then all clusters at clustering level *level* are written. Otherwise *clust_num* specifies the index of a single cluster to write to the file. This command also prints certain statistical data to the .clg file and to the message window.

Writemap: *filename*

Write out the distance matrix in the generic order to file *filename*.

X Resources

If your computer is equipped to run X windows, there is a hidden file in your home directory called `.xdefaults`. This file is read by the X server and is used to override system defaults with personal preferences for fonts and other resources to be used by X-based applications.

Our suggested fonts and color schemes for XCluster can be overridden by adding lines to the `.xdefaults` file in your home directory. The fonts and colors we specify may not be available on all machines, so you may have to experiment a bit with doing this. To see a list of fonts available on your machine, issue the command `xlsfonts` from the UNIX prompt. To see a list of available colors, look through the file called `rgb.txt`, probably located in the directory `/usr/lib/X11`.

The following list of resources is a useful subset of those which may be specified. A more complete list appears in a file called `XCluster` which is supplied with the distribution. Entries in `.xdefaults` in your home directory will override the default values. Note that such entries must be free of trailing blanks.

In this description the definition of a resource is sometimes spread over two lines because of the length of the description; however, in the files `XCluster` and `.XDefaults` each description must appear on a single line.

Background Colors:

```
XCluster*background: wheat2
```

This is the background color for the XCluster program. It can be changed to any one of the valid color names found in the file `/usr/lib/X11/rgb.txt`. We have found that `wheat2` is a good choice for both color and grayscale terminals. It is also possible to change the foreground color, but the default of `black` is recommended.

The following color scheme is used for a long list of resources:

- All pushbuttons and editable text: `tan`
- All non-editable text and scrolling lists: `antiquewhite2`

The easiest way to alter these is to simply change all occurrences of “`tan`” and “`antiquewhite2`” in the file to your desired values. It is these values that differentiate the appearance of window regions which are interactive from those that are not.

Map Colors:

```
XCluster*Col0: DarkViolet
XCluster*Col1: BlueViolet
XCluster*Col2: Blue
XCluster*Col3: cyan
XCluster*Col4: green
XCluster*Col5: GreenYellow
XCluster*Col6: yellow
XCluster*Col7: Orange
XCluster*Col8: IndianRed
XCluster*Col9: Red
```

These resources control the colors in the Map display. Col0 represents the lowest distance values in the display and Col9 the highest.

Mosaic Colors:

```
XCluster*MosaicDA*foreground: yellow
```

This resource specifies the color of the cross-hairs in the mosaic display. It can be set to any color, but light shades are best for maximum clarity.

Fonts:

```
XCluster*fontList: -adobe-helvetica-bold-r-normal--14-*
```

This is the default font for all buttons and labels. Making this font smaller (e.g., changing the 14 to “12” or “10”) will reduce the screen area used by the program.

```
XCluster*MessageText.fontList:-adobe-courier-medium-r-normal--12-*
XCluster*MapLabel.fontList:-adobe-courier-medium-r-normal--12-*
```

These are the fonts for the main message window and for the read-out on the Map display, respectively. They should be set to a non-proportionally spaced font such as Courier.

```
XCluster*MacroModel Help*scrolled help text.fontList:
-adobe-courier-medium-r-normal--14-*
```

This resource specifies the text font in the help window.

Getting Help

Schrödinger software is distributed with documentation in PDF format. If the documentation is not installed in `$(SCHRODINGER)/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*.

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

- Check the Auto-Help text box, which is located at the foot of the main window. If help is available for the task you are performing, it is automatically displayed there. Auto-Help contains a single line of information. For more detailed information, use the online help.
- If you want information about a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Maestro menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- For information about a panel or the tab that is displayed in a panel, click the Help button in the panel, or press F1. The help topic is displayed in your browser.
- For other information in the online help, open the default help topic by choosing Online Help from the Help menu on the main menu bar or by pressing CTRL+H. This topic is displayed in your browser. You can navigate to topics in the navigation bar.

The Help menu also provides access to the manuals (including a full text search), the FAQ pages, the New Features pages, and several other topics.

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

- *Maestro User Manual*, for detailed information on using Maestro
- *Maestro Command Reference Manual*, for information on Maestro commands
- *Maestro Overview*, for an overview of the main features of Maestro
- *Maestro Tutorial*, for a tutorial introduction to basic Maestro features
- *MacroModel User Manual*, for detailed information on using MacroModel
- *MacroModel Quick Start Guide*, for a tutorial guide to using MacroModel
- *MacroModel Reference Manual*, for information on MacroModel commands

- MacroModel Frequently Asked Questions pages, at https://www.schrodinger.com/MacroModel_FAQ.html
- Known Issues pages, available on the [Support Center](#).

The manuals are also available in PDF format from the Schrödinger [Support Center](#). Local copies of the FAQs and Known Issues pages can be viewed by opening the file `Suite_2009_Index.html`, which is in the `docs` directory of the software installation, and following the links to the relevant index pages.

Information on available scripts can be found on the [Script Center](#). Information on available software updates can be obtained by choosing Check for Updates from the Maestro menu.

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail: help@schrodinger.com
USPS: Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone: (503) 299-1150
Fax: (503) 299-4532
WWW: <http://www.schrodinger.com>
FTP: `ftp://ftp.schrodinger.com`

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

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

On UNIX you can obtain the machine and system information listed above by entering the following command at a shell prompt:

```
$SCHRODINGER/utilities/postmortem
```

This command generates a file named `username-host-schrodinger.tar.gz`, which you should send to help@schrodinger.com. If you have a job that failed, enter the following command:

```
$SCHRODINGER/utilities/postmortem jobid
```

where *jobid* is the job ID of the failed job, which you can find in the Monitor panel. This command archives job information as well as the machine and system information, and includes input and output files (but not structure files). If you have sensitive data in the job launch directory, you should move those files to another location first. The archive is named *jobid-archive.tar.gz*, and should be sent to help@schrodinger.com instead.

If Maestro fails, an error report that contains the relevant information is written to the current working directory. The report is named *maestro_error.txt*, and should be sent to help@schrodinger.com. A message giving the location of this file is written to the terminal window.

More information on the `postmortem` command can be found in [Appendix A](#) of the *Job Control Guide*.

A	
agglomeration	
relation to critical threshold distance.....	6
successive	5
algorithm, clustering	5
Arms : command.....	27
atom list panel.....	14
axes, units for.....	5
B	
batch interface.....	9, 23
C	
chain, reflection of numbering	27
cluster membership	
display of	22
writing out	29
cluster size, displaying plot of	18
cluster, definition.....	5
Cluster: command.....	29
clustering	
algorithms used.....	5
commands.....	29
definition.....	2, 5
entropy.....	37
hierarchical	5
scheme	5
clustering level	
control of in visualization.....	19
definition.....	2, 7
in mosaic display	21
plot abscissa.....	17
use with point picking	18
clustering statistics	
plots of.....	17
PostScript output	18
color	
by cluster	8, 30, 41
comparison atoms.....	31
command file	
description	24
name convention.....	23
specifying on Cluster startup.....	23
specifying on XCluster startup.....	9
written by XCluster	12
Commands panel	12
Comment : command.....	24
comments, inserting into command file.....	13, 24
comparison atoms	
centroid of.....	30
use in rigid-body transforms.....	41
conformational distance, definition.....	2
conformational search, clustering example.....	59
conventions, document.....	vii
critical threshold distance	
defined	6
in distance map	20
output of.....	29
smallest	7
D	
dendrogram.....	8
dfile mode	
defined	25
use of	2
Dfile: command	25
directory	
installation	3
Maestro working.....	3
display, setting.....	9
distance criteria	
description	2
selection in Cluster	27
selection in XCluster	14
distance file	
generic order.....	23, 33
input order	23, 33
naming conventions	23
reading	13, 25
distance list, sorted.....	6
distance map	
displaying	18
use of	20
distance matrix	
analysis of user-supplied	2
calculation of	2, 6
display of	18
input of.....	13, 25
selected element in distance map	20
E	
effective cluster size, displaying plot of.....	18
effective number of clusters	36

- Enant: command 26
- enantiomers, comparison of 26
- entropy, reordering 37
- environment variable, SCHRODINGER 3
- F**
- figures of merit 2, 8
- File menu 10
- file name conventions 23
- filtering 1
- fonts 78
- G**
- generic order
- definition 7, 39
 - distance file in 23
 - in mosaic display 21
- generic transform
- application of 30
 - definition 41
- graphical interface 9
- H**
- hierarchical clustering 5
- I**
- interfaces for XCluster 9
- L**
- log file
- generation of 29
 - name convention 23
 - statistics recorded in 8, 18
 - verbose output to 10, 23
- M**
- Maestro, starting 3
- Map panel 19
- map, distance, display of 18
- message area 10
- mmsym facility, activated from GUI 16
- Mmsym: command 25
- molecular dynamics, clustering example 67
- mosaic
- description 21
 - displaying 8
 - use of 21
- Mosaic panel 21
- N**
- Nrms: command 28
- P**
- partitions
- change with clustering level 7
 - representation in distance map 20
- Plot panel 17
- PostScript output
- clustering statistics 18
 - map image 20
 - mosaic display 22
- product installation 79
- progress area 10
- R**
- Reflect: command 27
- reflection, of numbering system 27
- reordering entropy
- as plot ordinate 17
 - defined 37
 - definition 37
- rescaling 5
- resources 77
- rigid-body superposition 1, 2
- rings, reflection of numbering 27
- RMS Atom Picker panel 14
- RMS Torsion Picker panel 15
- Rotate: command 26
- S**
- Schrödinger contact information 80
- separation level, as plot ordinate 17
- separation ratio
- definition 35
 - output of 29
- Sfile: command 24
- structure files
- formats recognized 9
 - name conventions 23

-
- superposition 32
- 3-D 5, 40
 - of coordinates in output 29, 30, 32
 - rigid-body 1, 2
- Symatom: command 26
- symmetry 5, 25
- Symmetry panel 16
- T**
- Thresh: command 29
- threshold distance
- critical 6, 7
 - definition 6
 - display in Map panel 19
 - effect on map display 20
 - plot abscissa 17
- torsion list panel 15, 16
- Trms: command 28
- U**
- units, for distances 5
- V**
- visualization of clusters 8
- W**
- Write File panel 11
- Writeavg: command 32
- Writecls: command 29
- Writedst: command 33
- Writelead: command 31
- Writemap: command 33
- Writerep: command 30
- X**
- X defaults 77
- X resources 77

120 West 45th Street, 29th Floor
New York, NY 10036

Zeppelinstraße 13
81669 München, Germany

101 SW Main Street, Suite 1300
Portland, OR 97204

Dynamostraße 13
68165 Mannheim, Germany

8910 University Center Lane, Suite 270
San Diego, CA 92122

Quatro House, Frimley Road
Camberley GU16 7ER, United Kingdom

SCHRÖDINGER.