PROPAK

Software Tools for Protein Design
Version 2.1
April 1991

PROPAK IS PROVIDED "AS IS" AND WITHOUT ANY WARRANTY EXPRESS OR IMPLIED. THE USER ASSUMES ALL RISKS OF USING THIS SOFTWARE. THERE IS NO CLAIM OF THE MERCHANABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

YOU MAY MAKE COPIES OF PROPAK FOR YOUR OWN USE, AND MODIFY THOSE COPIES. YOU MAY NOT DISTRIBUTE ANY MODIFIED SOURCE CODE OR DOCUMENTATION TO USERS AT ANY SITE OTHER THAN YOU OWN.

v2.1 4/91

PROPAK

Software Tools for Protein Design
Version 2.1   April 1991

Table of Contents | Page

Introduction to the PROPAK System	4
Getting Your Own Version of PROPAK	6
How to Setup a PROPAK Calculation	7
Types of Files Produced by PROPAK	13
What to do if PROPAK Misbehaves	16
Source Code Descriptions	
Programs, Subroutines and Functions	18
Common Block Variables	25
Where to Turn for More Information	31

Introduction to the PROPAK System

Welcome to the PROPAK Users Manual! PROPAK is a computer program of use in the analysis of existing protein structures and the design of new ones. It generates lists of amino acid sequences which are compatible with an input protein backbone. You, the user, are given a great deal of control over the types of sequences that PROPAK will produce. The purpose of this documentation is to aid in the operation and use of the program. Toward this end, it is divided into several specific sections:

(1) acquiring a current version of the program
(2) description of user options, preparation of input files and procedures for running the program
(3) discussion of the contents of the output files
(4) what to do if PROPAK seems to be misbehaving
(5) description of subroutines and common blocks

PROPAK is intended as a first attempt at software of use in protein design work. The package implements an algorithm for the construction and analysis of protein tertiary templates. The program takes as its starting point the atomic coordinates of a known protein structure. It then finds sets of side chains conformations that can pack together to decorate the backbone of the input structure. A number of different filters can be used to restrict the possible side chain repackings. These filters include: (1) avoiding steric overlap, (2) appropriate filling of space, (3) satisfying potential hydrogen bond donors and acceptors, (4) control over use of charged residues, (5) use of residues favorable for local backbone secondary structure.

The lists of side chains compatible with a given protein backbone can be formed into tertiary templates showing the types of amino acid that are allowed at various sequence positions. A rotamer library listing the favorable conformations of each amino acid along with standard deviations is provided. The space filling methods are implemented via analytical molecular volume computations following the methods of Michael Connolly. The program also operates in a survey mode which provides a summary of conformational and structural data for any input structure.

This manual will assume a basic familiarity with the terminology and algorithms used in the packing calculations. Relevant background material can be found in the published papers describing the computer program and its applications; see the last section of this manual.

The PROPAK program is written in a portable Fortran 77 using only very common extensions. The code will run on essentially any of the computer systems in current use. The only portions of the source that may require attention before building PROPAK on your system are the master set of array dimensions, and the system dependent time and date routines. The scripts needed to build the program on various Unix workstations are supplied in subdirectories named for the machine type.

Much of the software you are about to operate is still in the developmental stage.

Questions and comments regarding the PROPAK program, including suggestions for improvements and changes, should be made to the author at the following address:

Professor Jay William Ponder
Biochemistry & Molecular Biophysics
North Building, Room 2811, Box 8231
Washington University School of Medicine
660 South Euclid Avenue
Saint Louis, MO 63110 U.S.A.

phone (314) 362−4195
fax (314) 362−7183
email ponder@comet.wustl.edu

Finally, it should be noted that the concepts embodied in PROPAK originated in the research group of Prof. Frederic M. Richards in the mid−1980's. The author owes a tremendous debt to Fred for his ideas and encouragement during the early stages of this work. Thanks are also due to Michael Connolly for allowing incorporation of a modified version of his AMS analytical molecular surface area program into PROPAK.

Getting Your Own Version of PROPAK

The PROPAK package is distributed via the Internet from an "anonymous ftp" account on the machine dasher.wustl.edu with an IP number of 128.252.162.151. After logging on under ftp (Username: anonymous, Password: "your email address"), the software may be found in the /pub/propak subdirectory. At a minimum, you will need to download the contents of the /doc, /source and /params subdirectories. Also required are the compile/build scripts from the subdirectory named for your machine type. The /example area contains a simple test case where PROPAK is applied to a five residue packing unit in the protein crambin. In addition, compress tar files of complete subdirectories may usually be found at the /pub/propak directory level.

The documentation for the PROPAK programs is located in the /pub/propak/doc subdirectory. The .show files in this area are in the original SGI Showcase format. Portable versions of the documentation are also provided as the .ps Postscript format files. Please read and return the PROPAK license. While the intent is to distribute the PROPAK code to anyone who wants it, the developers would like to keep track of the sites using the package. When new modules and capabilities become available, and when the almost inevitable bugs are uncovered, we will attempt to notify all who have returned a license form. Finally, we remind you that this software is copyrighted, and ask that it not be redistributed in any form.

The compilation and building of PROPAK executable should be easy for most of the common workstation and PC class computers. Instead of Makefiles for the Unix versions, we have chosen to provide three separate scripts to compile the source, build an object library, and link binary executables. Compilation on workstations should use the vendor supplied Fortran compiler. For the Macintosh we only have experience with the Language Systems compiler. On IBM PS's, PROPAK can be built under the WATCOM Fortran compiler; versions of the Microsoft and SVS compilers that we tried led to difficulties that may not be trivial to overcome.

Regardless of your target machine, only three small pieces of code can possibly required attention prior to building. The first two are the system dependent date and time routines found in "promo.f", and the CPU routine found in "clock.f". Please uncomment the sections of these routines needed for your computer type. The final set of source alterations are the master array dimensions found the include file "sizes.i". A description of the values is contained in the header of this file. Note that in order to keep the code completely transparent, PROPAK does not implement any sort of virtual memory or heap data structure. This requires that "sizes.i" values be set at least as large as the biggest problem you intend to run. Obviously, you should not set the array sizes to unnecessarily large values, since this can tax your compute resources and may result in performance degradation. The worst case we know of at present is for the DEC Alpha machines, where running a "small" problem using an unnecessarily "large" executable can results in about a 50% CPU time penalty. After compilation using the appropriate "compile" script, you must use a "library" script to create an archive of object code modules. Finally, run a "link" script to produce the complete set of PROPAK executables.

How to Setup a PROPAK Calculation

The PROPAK program can be run either interactively or as a batch job. The EXAMPLES subdirectory contains a sample command file (CRAMBIN.COM) which will perform a complete computation for the test case. A further explanation of this sample procedure is given later. The current section will also explain the various input options and their effects on output and results. After invoking PROPAK, the first question always asks for the name of the input Protein Data Bank (PDB) file. It is assumed that the file contains a protein or peptide structure as the primary data. At present, heme groups and metal ions are supported as auxiliary atoms. The file must follow the PDB format specifications. In particular, the atoms within residues must be in the standard order and all residues should be completely defined. All hydrogen atoms found in the original PDB file will be ignored by PROPAK; the program will assign its own hydrogens at a later point. Once the original protein structure is read in from the PDB file, PROPAK will print a summary of this information and move on to the Major Options menu. This menu contains a set of eight true/false parameters that determine the type of computation PROPAK will perform. A brief description of the effect of each option is given below:

(1) Dihedral Angle Survey Only

If this option is set to YES then PROPAK will calculate the values of the dihedral angles (phi/psi/omega/chi), bond angles (main and side chain), interresidue Cb−Cb distances of less than 6 Ang, and secondary structure for the original PDB file. All of these values will be sent to the master output unit, then the program will stop. If NO, the same angles, distances and secondary structure will be computed and stored internally. PROPAK will then continue by making a user specified packing calculation.

(2) Cycle over All Residues

When set to YES this switch forces PROPAK to perform a series of packing computations in which each residue in turn is treated as a single residue packing unit. This mode can be used to test your input structure. If the structure is of high resolution and well refined, PROPAK should be able to place the native amino acid rotamer at each site.

(3) Stop after Main Chain Check

If set to YES the program will terminate after performing the main chain−main chain check. This option is useful for seeing which rotamers will potentially fit inside of a packing unit. You can also use the result to estimate the size of a subsequent sequence enumeration for the same packing unit (simply take the product of the number of rotamers allowed at each site; a number under about 10 million is a small calculation, over 1 billion is probably prohibatively large). A summary of the check will be placed in a "111" file. If you decide at some later point to make a full packing calculation for the same packing unit, this "111" file will automatically be reused (this saves the CPU time required to perform the molecular volume computations for the original "111" file).

(4) Stop after Side Chain Check

A YES value for this option forces PROPAK to stop after both the main chain check and the side chain check are complete. Results of the two checks are written to the "111" and "222" output files. If set to NO, the computation will continue with combinatiorial sequence enumeration after the side chain check.

(5) Write Rotamer List to Disk

This switch determines whether the allowed rotamer sequences will be written to a "ROT" file during sequence enumeration. The YES value should normally be used for this option, unless an extremely large number of allowed sequences is expected.

(6) Include Charged Residues

If set to YES, the charged amino acid residue types (ASP, GLU, LYS, ARG and perhaps HIS) will be included in the computation. The rotamers for these amino acids will be used just as they were read in from the ROTAMER.LIB library file. When set to NO, charged amino acids are not used (even if they are present in the ROTAMER.LIB file). The status of the HIS residue type is determined by option (7) described below.

(7) Use (+)Charged HIS Residue

This option determines whether HIS residues are treated as charged or neutral throughout any PROPAK calculation.

(8) Repack only Native Sequence

When set to YES only the native amino acid type from the original PDB file will be tested at a given packing unit position. This option is used to find any possible repackings of the native amino acid sequence. It can also be used during the refinement of X–ray data to explore alternative side chain conformations. Since the number of rotamers allowed at a given site will usually be quite restricted, large packing units of 20 or more residues can be accomodated under this option.

Operation of the Program

After the Major Options have been set, PROPAK will embark upon the specified computation and will ask any necessary questions as they arise. These questions can involve: location of the library files, type of interatomic contact distances to use, residues included in the packing unit, status of residues outside the packing unit, or the value of the space filling criterion. A discussion of the questions and possible answers is presented below:

Location of the Library Files

During most calculations, two separate data files are needed to describe the amino acid rotamer library (ROTAMER.LIB) and the secondary structure preferences of the amino acids (ROBSON.LIB). When these files are needed, PROPAK will automatically look for them in the current directory. If either file is not found, the program will ask for the location and name of the library file. The ROTAMER.LIB file contains the complete list of possible side chain rotamers with their chi angle values and standard deviations. The program will read rotamers from this file until a blank line is encountered. This allows "unused" rotamers to be stored at the bottom of the ROTAMER.LIB file beneath a blank line. The standard ROTAMER.LIB file supplied with PROPAK contains all the common rotamers for each amino acid. Rotamers may be added to or deleted from the file to enable special purpose calculations. The ROBSON.LIB file contains the propensity of each amino acid type to appear in helix, sheet, turn or coil structures. It is in a coded format due to Garnier, Osguthorpe and Robson (*Journal of Molecular Biology*, 1978 & 1988). The ROBSON.LIB file should not be altered unless the user fully understands its contents.

Interatomic Contact Distances

PROPAK can use either of two internally stored sets of minimum interatomic contact distances (the closest distance allowed between two atoms without a severe steric "bump"). These values are used to detect unacceptable steric interactions within the original protein structure and during the introduction of new side chain rotamers. The sets of distances are based upon the Ramachandran "normal" and "extreme" limit distances. The "normal" distances correspond to a set of larger atomic radii and are the suggested values for use with a well refined protein structure. The "extreme" distances correspond to smaller atomic radii and can be used when the input structure contains many close contacts. Until further experimental data allows the development of better parameters for the PROPAK procedure, we recommend use of the largest distances consistent with the placement of native−like amino acid rotamers at all packing unit sites.

Residues Included in the Packing Unit

If a packing calculation is to be performed, PROPAK will ask for a list of the "variable" residues; i.e., the residue sites which are to be included in the packing unit. The side chains of packing unit residues should pack against each other and point toward a common region of space. If the full rotamer library is searched, not more than about a seven residue unit should be subjected to sequence enumeration (the size of the enumeration increases roughly combinatorially). If only the rotamers for the native residue are used at each site, then twenty or more sites can be included in the packing unit.

Status of Residues Outside the Packing Unit

After the packing unit is specified, PROPAK will decide upon a treatment for the residues outside the packing unit. The user is given the option of "fixing" any or all residues not contained in the packing unit. A fixed residue will have its native side chain conformation assigned and fully active throughout the main chain−side chain check. If a residue is not fixed, only the main chain and Cb atoms will be permanently assigned. Fixing the side chains adjacent to the packing unit can greatly restrict the sequences that will fit into the unit. However, the type and number of sequences enumerated with non–packing unit side chains fixed are obviously dependent on the presence of this "fixed" structure.

Value of the Space Filling Criterion

When sequence enumeration is desired, the program will ask for a volume filling criterion. In order to become an "allowed" sequence, any trial sequence generated during the enumeration phase must meet the volume criterion. The criterion is the minimum amount of overlap that a trial sequence needs to have with the native packing unit structure file. Its value can be expressed either as a specific number in Ang^3 or as a percentage of the volume of the native packing unit (by entering the fraction as a number between 0 and 1, i.e., 0.9 means a trial sequence must overlap at least 90% of the native packing unit).

Postprocessing of the Sequence Results

After PROPAK has completed a sequence enumeration computation, the "allowed" rotamer sequences will be available for inspection. If the rotamer sequences were written to a disk file, this list can be further condensed to provide results in terms of amino acid sequences and compositions. A series of three small auxiliary programs (BUBBLE, SORTER and COMPOS) coupled with appropriate operating system commands will perform these postprocessing operations. The steps involved are given below in a command procedure for Open VMS systems called CRAMBIN.COM. This procedure runs a full packing computation with sequence enumeration and post–processing to give amino acid sequence and composition lists. Essentially any PROPAK calculation can be made using simple modifications of this command file. A similar script file for use with Unix systems is included in the /example subdirectory of the PROPAK distribution.

Contents of the CRAMBIN.COM File

```
$!
$!  ********************************************
$!  **                                        **
$!  ** crambin.com -- PROPAK on crambin example **
$!  **                                        **
$!  ********************************************
$!
$!  first, run PROPAK to generate allowed sequences
$!
$ assign/user   crambin.out    for006
$ run/nodebug   propak
crambin
0
e
4
2,13,26,30,32
0
2
```

```
(19)  300
(20)  $!
(21)  $!  next, sort and analyze the sequences generated
(22)  $!
(23)  $ sort/nodup    crambin.seq    crambin.se2
(24)  $ assign/user   crambin.tmp    for006
(25)  $ run/nodebug   sorter
(26)  crambin
(27)  $ append/nolog  crambin.tmp    crambin.out
(28)  $ delete/nolog  crambin.tmp;*
(29)  $ assign/user   crambin.tmp    for006
(30)  $ run/nodebug   bubble
(31)  crambin
(32)  $ append/nolog  crambin.tmp    crambin.out
(33)  $ delete/nolog  crambin.tmp;*
(34)  $ sort          crambin.cmp    crambin.cm2
(35)  $ assign/user   crambin.tmp
(36)  $ run/nodebug   compos
(37)  crambin
(38)  $ append/nolog  crambin.tmp    crambin.out
(39)  $ delete/nolog  crambin.tmp;*
(40)  $!
(41)  $!  finally, clean up the mess we have made
(42)  $!
(43)  $ delete/nolog  crambin.seq;*
(44)  $ delete/nolog  crambin.cmp;*
(45)  $ delete/nolog  crambin.cm2;*
(46)  $ copy/nolog    crambin.se2    crambin.seq
(47)  $ copy/nolog    crambin.cm3    crambin.cmp
(48)  $ delete/nolog  crambin.se2;*
(49)  $ delete/nolog  crambin.cm3;*
```

Line 10	Assign the Master Output (Fortran I/O Unit 6) to be placed in the file CRAMBIN.OUT
Line 11	Start the PROPAK program
Line 12	Name of original PDB file (assumes PDB extension)
Line 13	Number of a Major Option to change (in this case, use 0 or blank line since no options are changed)
Line 14	Use the "Extreme" limit contact distances
Line 15	Manual input of "variable" (packing unit) residues
Line 16	List of the five packing unit residues
Line 17	A blank line (or 0) to end the residue list

Line 18	All non−packing unit residues are to be "fixed"
Line 19	Require minimum overlap of 300 Ang3 between native and trial rotamer sequences
Line 23	Sort to eliminate duplicate amino acid sequences
Line 24−28	Use SORTER to talley the allowed sequences over amino acids (instead of over rotamers)
Line 29−33	Use BUBBLE to alphabetize within each amino acid sequence to give the amino acid composition
Line 34	Sort to group identical amino acid compositions
Line 35−39	Use COMPOS to make a final list of the number of occurences of each amino acid composition
Line 43−49	Rename output files and delete temporary files

Types of Files Produced by PROPAK

The PROPAK program produces several different types of output files. Taken together, these files allow complete analysis and reconstruction of the computation. The following is a brief description of each output file associated with the CRAMBIN test case found in the [PROPAK.EXAMPLE] subdirectory:

CRAMBIN.OUT

This file contains the master output from the PROPAK calculation. Unless redirected prior to a run (Fortran I/O Unit 6), this output will appear on the screen for interactive jobs or in the log file for batch jobs. Note that if unit 6 is explicitly assigned, you will not be able to view the master output while PROPAK is running (in either interactive or batch mode). It contains summaries of information about the original protein structure, the main chain check, the side chain check and sequence enumeration.

CRAMBIN.PDQ

The "PDQ" file contains sets of coordinates for each side chain rotamer that is allowed at each position of the "packing unit". Coordinates for all allowed "wobbles" of each rotamer are included. The coordinates are in Protein Data Bank (PDB) format so that they can be used to directly replace the corresponding coordinates in the original PDB file. This allows the creation of "mutant" proteins with alternate sequences replacing the native sequence of the packing unit.

CRAMBIN.111

The complete results of the main chain−side chain check are recorded in a coded format in the "111" file. It contains a section for each variable residue site (ie, each residue in the "packing unit"). For each residue, there is one line for each rotamer library member that passes the check. This line contains the rotamer number and residue type of the library member, a string of 19 coded numbers, and the overlap volume of the library member with the packing unit. The 19 coded numbers represent the status of each possible wobble of the rotamer; 0=the wobble is sterically disallowed, 1=the wobble is allowed, 2=the wobble forms a polar/polar H−bond, 3=the wobble forms a polar/charge H−bond, 4=the wobble forms a charge/charge H−bond, 5=the wobble forms a disulfide link.

CRAMBIN.222

The complete results of the side chain−side chain check are recorded in a coded format in the "222" file. The output consists of one matrix for each pairwise combination of variable residues. The rows and columns of a matrix represent the rotamers allowed at the two residue sites. An element of the matrix marks the status of one particular combination of rotamers; 0=the pair of rotamers have a steric clash, 1=the pair is allowed, 2=the pair form a polar/polar H−bond, 3=the pair form a polar/charge H−bond, 4=the pair form a charge/charge H−bond, 5=the pair form a disulfide link.

CRAMBIN.ROT

The "ROT" file lists each of the allowed amino acid rotamer sequences which can replace the original packing unit side chains. Each line of the file contains a sequence number followed by the rotamer number and residue type of each rotamer in the particular sequence. The order of residues sites within the sequence is simply the expected numerical order (ie, 2−13−26−30−32 for the CRAMBIN example).

CRAMBIN.SUM

Each allowed rotamer sequence will generally differ from the native packing unit seuqence in a variety of physical chemical characteristics; a summary is provided in the "SUM" file. The file contains contains one line for each allowed rotamer sequence (in the same order as the "ROT" file). Each line has the sequence number followed by a set of 13 additional numbers which provide information about the particular rotamer sequence:

<u>Column</u>	<u>Description</u> <u>of</u> <u>Contents</u>
1	Sequence number (same as "ROT" file"
2	Number of polar residues in this packing unit
3	Number of charged residues in this packing unit
4	Number of CYS residues in this packing unit
5	Number of polar/polar H−bonds between this packing unit sequence and the rest of the protein structure
6	Same as 5, only for polar/charge H−bonds
7	Same as 5, only for charge/charge H−bonds
8	Same as 5, only for disulfide links
9	Number of polar/polar H−bonds between two groups contained in this packing unit sequence
10	Same as 9, only for polar/charge H−bonds
11	Same as 9, only for charge/charge H−bonds
12	Same as 9, only for disulfide links
13	Overlap volume (intersection) of this packing unit with the native packing unit in Ang3
14	Inverse secondary structure prediction, a large positive number indicates that this packing unit sequence is compatible with the secondary structure of the residue sites in which it resides, a large negative indicates an incompatiblity between sequence and structure

CRAMBIN.SEQ

The allowed amino acid sequences that can replace the original packing unit are listed in the "SEQ" file. Each line of this file contains a single amino acid sequence in one letter code. The order of the residues in this sequence is simply the expected numerical order (ie, 2−13−26−30−32 for the CRAMBIN case).

CRAMBIN.CMP

The allowed amino acid compositions that can be placed into the original packing unit are stored in the "CMP" output file. Each line of this file contains an amino acid composition (in one letter code), the number of allowed amino acid sequences that have this composition, and the sum of the volumes (average Voronoi volumes) of the amino acids making up the composition.

What to do if PROPAK Misbehaves

Several types of problems or errors can crop up during the use of the PROPAK software. Some of the more common of these are described in this section.

A VMS Error is Generated upon trying to Run PROPAK

PROPAK is a big program that can require a large share of system resources. If VMS indicates that you have insufficient room of various kinds upon trying to run the program, then some system parameters may need to be changed. See the DISTRIBUTION section of this manual.

The Input PDB File is Obviously Read in Incorrectly

Check the original PDB file. Some other programs (FRODO, for example) write "PDB" files with the atom records in a non−standard order or format. Your file must conform to the Brookhaven guidelines to be read correctly. In addition, PROPAK assumes that the protein is a continuous polypeptide chain with no missing or duplicate residues. Alternative residue numbering schemes (as often used for serine proteases, for example) are handled correctly by the program. Only a few types of auxiliary groups are interpreted by the current version of PROPAK (heme, in particular). Any water molecules in the PDB file are ignored.

PROPAK Asks You to Increase some Parameter then Stops

If your version of PROPAK was not dimensioned with large enough arrays to handle the problem you attempt, the program will indicate which dimensioning parameter needs to be increased. These parameters are in the SIZES.FOR source code. After making the change, you must recompile and relink the program. Please see the DISTRIBUTION section for further details.

Error Message from Routine ZMATRIX, then Divide by Zero

PROPAK uses the original side chain conformations from the input PDB file during hydrogen atom addition and when testing new side chain rotamers. If your original structure is idealized in certain non−physical ways, errors will result when new atoms are positioned. A common error of this sort occurs when a crystallographer has placed side chain atoms in a straight line (an ill−determined LYS, for example). If you suspect such a problem, make a Dihedral Angle Survey Only computation and examine the output for "odd" side chain geometry.

PROPAK seems to be taking "Forever" to Finish a Run

The sequence enumeration phase of a PROPAK computation grows combinatorially with each additional residue. It is very easy to give PROPAK a calculation that cannot be completed in a reasonable length of time. When in doubt, request that the program Stop after Main Chain Check. Then the product of the number of allowed rotamers at each packing unit site is equal to the number of trial sequences that must then be tested in a full computation. If this number is too large (greater than about 1 billion) the packing unit will have to be divided and separate computations run.

Programs, Subroutines and Functions

The PROPAK software consists of the main program, several smaller programs and over 60 subroutines and functions. The complete Fortran source contains nearly 15,000 lines of code. Data is passed between routines both in calling arguments and in approximately 30 common blocks. This section contains a brief description of each program, subroutine and function. The following section provides a listing of the variables found in each Fortran common block.

ADDHYDRO

adds amide nitrogen and alpha carbon hydrogen atoms to the peptide backbone structure

ADDRES

general routine for attaching a specified side chain rotamer to a residue site

AHELIX

finds pieces of alpha helical secondary structure by comparison to idealized helical Ca–Ca distances

ANGLE

computes the angle between three specified atoms

ATOMLIST

makes room for a new atom and sets information about its name, radius and hydrogen bonding capabilities

ATOMSIZE

assigns atomic radii to each atom in the structure

BHAVEN

reads in the original structure from a standard Protein Data Bank (PDB) file format

BONDLIST

forms a bond between two atoms by adding their attachment to the appropriate arrays

BSTRAND

finds pieces of beta strand secondary structure by comparison to idealized strand

Ca–Ca distances

BUBBLE

a bubble sort routine which alphabetizes each allowed amino acid sequence

BUMPER

checks the seriousness of the van der Waals overlap between two specified atoms and prints a message listing the degree and type of overlap

CISPEP

locates cis–peptide units by comparing actual Ca–Ca distances to idealized values

CLOCK

determines the elapsed CPU time in seconds since the start of the job

COMPOS

compresses the list of allowed, alphabetized amino acid sequences into a list of allowed amino acid compositions and their frequencies

CONNOLLY

a collection of routines which perform the geometric calculations needed to compute the van der Waals or contact–reentrant volume of a group of atoms

CONTROL

prints a menu of major program options and asks the user for any desired changes

DENSITY

analyzes the packing density near a specified residue; not implemented in the current program

DIHDRL

computes the dihedral angle between four specified atoms

DISTMAP

computes the distance matrices containing all of the interresidue Ca–Ca and Cb–Cb distances

FIXRES

adds the permanently fixed side chains for all residues outside the packing unit

GETIME

gets the elapsed CPU time in seconds for an interval

GRIDCLR

removes all atoms from the 3–D grid used to check for interatomic contacts

GRIDLIST

adds a specified to the current 3–D grid used to check for interatomic contacts

GRIDOFF

removes a single specified atom from the 3–D grid used to check for interatomic contacts

GRIDVIEW

prints out in line printer format a collection of views of slices through the protein structure

HEMHYDRO

adds hydrogen atoms to a heme group found in the original PDB file

HOMOLOGY

uses Pascal's triangle to compute the frequency of amino acid sequences with a given amount of homology

INITIAL

contains a data block which initializes several of the variables and arrays used by the program

INTERIOR

tests for and prints a list of the residues buried in the interior of the protein using an algorithm based only on the coordinates of the protein backbone atoms

LIST2ND

organizes the prior secondary structure assignment by sequence segments

MAINGRID

assigns the main chain atoms of the protein to the 3–D grid used to check for interatomic contacts

MAINVDW

sets the covalent connectivity relationships between the main chain atoms (1−2 is a covalent bond, 1−3 is two atoms separated by one atom, etc.)

METRIC

computes the distance between two specified atoms

METRIC2

computes the square of the distance between two specified atoms

PHIPSI

finds the values of the phi, psi, omega and chi angles for the structure in the original PDB file

PROMO

writes a short message containing info about the PROPAK program package and the copyright notice

PROPAK

the top level program; calls second level routines in the user defined order to perform the desired protein packing computation

QUERY

asks the user to provide lists of variable (packing unit) residues and fixed residues (ie, side chains present)

RANKING

performs an inverse secondary structure computation on the current trial sequence using the Robson statistical data base

RICHARDS

routines to compute the secondary structural units of a protein from Ca−Ca postions using Fred Richards' distance matrix method

SECOND

gets the secondary structure by calling routines to implement Fred Richards' distance matrix secondary structure method

SEQUENCE

prints out the sequence of the protein

SETIME

initailizes the elapsed CPU timer

SETMESH

sets the spacing between points in the 3–D grid and translates the structure to place it at the center of the grid

SETORIG

resets the current structure to contain only the backbone (main chain) atoms and their hydrogens

SETRANK

reads in the Robson secondary structure propensity library file

SETROT

reads in the list of amino acid side chain structures and standard deviations for the rotamer library file

SHARPTRN

finds pieces of beta turn secondary structure by comparison to idealized strand Ca–Ca distances

SIZES

contains the master array dimensioning parameters; used as an "include" file an most other source files

SORTER

takes a sorted list of allowed amino acid sequences and talleys the occurence of each amino acid type

TAUVAL

finds the main chain and side chain bond angles (tau values) for the structure in the original PDB file

TEMPLATE

contains a stand–alone program to read/write, make and edit compositional tertiary templates from data generated by PROPAK

TESTCB

checks variable residue sites (ie, the packing unit) to see if Cb atoms will fit at those sites; if not the site is marked as either a probable or a possible GLY residue

TESTCG

checks variable residue sites (ie, the packing unit) to see if Cg atoms will fit at those sites

TESTPAIR

tests all possible combinations of rotamers at pairs of variable residue sites for steric clashes (ie, the "side chain−side chain" check)

TESTRES

tests each rotamer in turn at each of the variable residue sites for steric clashes (ie, the "main chain−side chain" check)

TREESRCH

performs a combinatorial search for allowed rotamer sequences; operates by first generating a sequence or partial sequence, then testing for steric clashes or inappropriate space filling

UPCASE

converts a text string to all upper case letters

VDWCLR

removes all covalent bonds and other connectivity information for the current set of atoms

VDWLIST

finds and stores interatomic connectivity information (1−2 through 1−5 relationships) for a specified atom starting from the list of all covalent bonds

VDWOFF

removes all interatomic connectivity information for a single specified atom

VECNORM

computes the unit vector in the direction between two specified atoms

VOLUME

figures out which atoms to include in a molecular volume computation and calls the appropriate routines to do the geometric calculations

ZMATRIX

finds the Cartesian coordinates for an atom from its defining internal coordinates

Common Blocks and Variables

AMINOS	**amino acid names and statistics**
amino3	three letter code for each amino acid type
amino1	one letter code for each amino acid type
amino	full text name for each amino acid type
aasize	average Voronoi volume in Ang3 for each amino acid type
aavolume	average molecular volume in Ang3 for each amino acid side chain type
polar	1=polar residue type; 0=nonpolar residue
ionize	1=charged residue type; 0=not charged
ANGLES	**angles from the original PDB structure**
phi	phi angle value for each residue
psi	psi angle value for each residue
omega	omega angle value for each residue
chi	chi angles (up to 4) for each residue
cnang	C−Ca−N angle value for each residue
cbnang	N−Ca−Cb angle value for each residue
cbcang	C−Ca−Cb angle value for each residue
tau	side chain bond angles for each residue
ATMINF	**currently defined atoms**
natom	current number of atoms
atmnam	abbreviated name of each atom
atmres	number of residue to which each atom belongs
atmoff	1=atom currently active; 0=atom turned off
ATTACH	**interatomic connectivity relationships**
nbond	number of covalent bonds to each atom
bonds	atom numbers of the atoms bonded to each atom
vdw	matrix of atom/atom relations; 2=atoms covalently bonded, 3=atoms are 1−3 related, 4=atoms are 1−4 related, 5=atoms are 1−5 related, 0=atoms are greater than 1−5 from each other
AUXINF	**auxiliary atoms**
naux	number of auxiliary atoms (heme atoms, metal ions, substrate, etc.)
auxfirst	atom number of the first auxiliary atom
auxlast	atom number of the last auxiliary atom
BUMPED	**steric clashes ("bumps")**

nbump	number of times a test group has been bumped
bumped	atom numbers of the atoms that bump a test group
status	text phrase to describe the type of bump
overlap	distance by which the atom overlaps a test group atom
BURIED	**buried (interior) residue sites**
buried	1=the residue is buried in the structure interior; 0=the residue is exposed on the surface
CHRONO	**CPU timing information**
cputim	total CPU time used as of last call to "setime"
COORDS	**Cartesian coordinates of the atoms**
x	the X−coordinate of each current atom
y	the Y−coordinate of each current atom
z	the Z−coordinate of each current atom
DISULF	**interatomic relations in potential disulfide links**
disulf	marks the interatomic relationship to CYS S atoms potentially involved in disulfide links; 1=CYS S, 2=CYS Cb, 5=greater than 1/5 to CYS S, etc.
DOUBLE	**allowed pairs of rotamers**
pair	four dimensional matrix that contains the results of the side chain−side chain check; a non−zero value means that the rotamer combination is sterically allowed
DSTMAP	**interresidue distances**
distca	distance matrix containing the Ca−Ca distances
distcb	distance matrix containing the Cb−Cb distances
angcacb	value of angle between the Ca−Cb vectors of two residues
FILES	**name of the input PDB file**
brookhaven	name of the input PDB file without any directory path or extension
leng	length of the text string in brookhaven
pdbfile	full name of the input file as entered by the user
FOLDS	**secondary structure assignment**
fold	secondary structure type for each residue site
GRID	**3−D grid of molecular structure**

grid	3−dimensional grid to which atoms are assigned; 0=no atom overlaps the grid point, otherwise contains atom number of the atom that most overlaps the grid point

HBONDS	**hydrogen bonding potential of atoms**
hbond	ability of each atom to participate in hydrogen bonding; 0=either donor or acceptor, −1=acceptor only, 1=donor only, 3=neither donor nor acceptor
chrg	formal charge of each atom (charges are assigned to attached hydrogens as well)

HISTYP	**ionization state of histidine**
histyp	selects neutral or positively charged histidine

INFORM	**control printing of extra information**
verbose	if true, turns on extra printing during execution of the "connolly" routines

IOUNIT	**Fortran unit numbers for input and output**
iout	unit number for the main output
inp	unit number for the user's input
ipdb	unit number for the original PDB file
igrid	unit number for the grid views
iseq	unit number for the amino acid sequences
irot	unit number for the rotamer sequences
isum	unit number for rotamer sequence statistics
ipdq	unit number for mutant PDB coordinates
imono	unit number for main chain check info
ipair	unit number for side chain check info
isph	unit number for artificial sphere file

LIMITS	**interatomic contact distances**
normal	minimum contact distances for each pair of atom types; this is a "large" set of values based upon the Ramachandran "normal" limit
normal2	square of the normal values for each atom type
extreme	minimum contact distances for each pair of atom types; this is a "small" set of values based upon the Ramachandran "extreme" limit
extreme2	square of the extreme values for each atom type

MCHAIN	**lists of peptide unit atoms**
n	atom number of amide nitrogen for each residue

ca	atom number of alpha carbon for each residue
c	atom number of carbonyl carbon for each residue
o	atom number of carbonyl oxygen for each residue
oxt	atom number of C−terminal residue carboxylate oxygen
hn	atom number of amide hydrogen for each residue
hca	atom number of alpha hydrogen for each residue

PACK **native packing unit volume**

packvol volume of the native packing unit or other target area

PARAMS **major program options**

charge	inclusion of charged residue types
dopair	performance of side chain−side chain check
dotree	enumeration of allowed rotamer sequences
dorot	writing of rotamer sequences to a file
survey	angle computation and distance matrices only
cycle	treat each residue in turn as a packing unit
native	fill packing unit with native sequence only

PI **value of "pi"**

pi double precision value of the irrational number "pi"

RADIUS **atom types and atomic radii**

radtyp	atom type to be used for each current atom
radius	atomic radius for each atom type

RESET **original atoms used in resets**

norig number of backbone (main chain) atoms in the original PDB structure, including hydrogens

RESINF **residue site in the current structure**

nres	number of residue sites in the structure
resnam	three letter name for each residue
restyp	residue type number for each residue
reslib	rotamer library member closest to the original PDB structure for each residue

RESULT **residues in and out of packing unit**

fixed	1=residue side chain is fixed as per original structure; 0=residue is present only through Cb
varied	1=residue side chain is allowed to vary (ie, part of the "packing unit"); 0=residue is not variable
known	type of residue at each site, if determined

ROBSON	**Robson secondary structure statistics**
hinfo	helix propensity data according to Robson
sinfo	sheet propensity data according to Robson
tinfo	turn propensity data according to Robson
cinfo	random coil propensity data according to Robson
ROTINF	**side chain rotamer library values**
nrot	number of amino acid rotamers in library
rotres	residue type number for each rotamer
rotnam	residue name for each rotamer
rotchi	chi angle values for each rotamer
rotsig	chi angle standard deviations for each rotamer
wobble	standard deviations in each rotamer chi angle to be tested during the main chain−side chain check
SCALE	**translation and scaling for 3−D grid**
meshsize	distance between adjacent points on 3−D grid
gridsize	number of points in each 3−D grid dimension
xtrans	translation of original structure in X−direction
ytrans	translation of original structure in Y−direction
ztrans	translation of original structure in Z−direction
SCHAIN	**lists of common side chain atoms**
cb	atom number of beta carbon for each residue
hgl	atom number of potential GLY hydrogen for each residue
hcb	atom numbers of beta hydrogens for each residue
SEQUEN	**variable residues and their allowed rotamers**
nvary	number of variable residues (ie, "packing unit")
vary	residue number of each variable residue site
order	index to the order in which the variable residues are searched during sequence enumeration
nposs	number of rotamers that pass the main chain−side chain check at each variable residue site
poss	rotamer number of each rotamer that is allowed at each variable residue site
size	overlap volume with the native packing unit of each rotamer that is allowed at each variable residue
SINGLE	**allowed rotamers at single sites**
nmono	number of rotamers that pass the main chain−side chain check for each residue
mono	status of each wobble for each side chain rotamer at

each residue; 0=the rotamer is sterically disallowed, 1=rotamer allowed, 2=rotamer forms polar/polar H−bond, 3=rotamer forms polar/charge H−bond, 4=rotamer forms charge/charge H−bond, 5=rotamer forms disulfide link

monomax — largest value (over all wobbles) in the mono array for each rotamer at each residue

monosize — overlap volume with the native packing unit of each rotamer at each residue

monolist — rotamer number of each rotamer that is allowed at each residue

Where to Turn for More Information

(1) Jay W. Ponder and Frederic M. Richards, Tertiary Templates for Proteins: Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes, *Journal of Molecular Biology*, **193**, 775–791 (1987).

(2) Jay W. Ponder and Frederic M. Richards, Internal Packing and Protein Structural Classes, *Cold Spring Harbor Symposia on Quantitative Biology*, **52**, 421–428 (1987).

(3) Frederic M. Richards and Craig E. Kundrot, Identification of Structural Motifs from Protein Coordinate Data: Secondary Structure and first–Level Supersecondary Structure, *Proteins*, **3**, 71–84 (1988).

(4) J. Garnier, D. J. Osguthorpe and B. Robson, Analysis of the Accuracy and Implications of Simple Methods for Predicting the Secondary Structure of Globular Proteins, *Journal of Molecular Biology*, **120**, 97–120 (1978).

(5) Michael L. Connolly, Computation of Molecular Volume, *Journal of the American Chemical Society*, **107**, 1118–1124 (1985).