

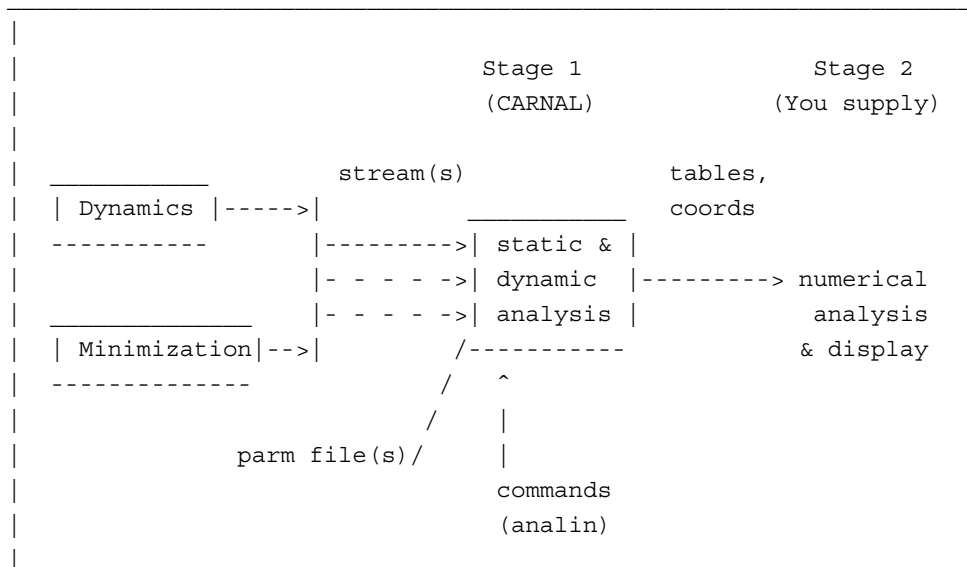
1. Carnal

Usage:

```
carnal [-O] < analin > analout
carnal [-O] -i analin -o analout -p parm
```

-O Overwrite output files.

CARNAL is a coordinate analysis program that uses a flexible command language. In addition to conventional trajectory measurements, it allows comparisons between multiple streams of coordinates. It has many of the capabilities of ANAL and MDANAL, which it was intended to eventually replace. It also provides set-theoretic group specification, cartesian vector oriented measurements, hbond analysis, output of distributions (including radial), selection of coordinate sets from streams, interpretations of md streams in terms of windows, and format conversion.



1.1. Introduction

CARNAL input is essentially a programming language that lets one specify variables and perform operations on them. The control input file for CARNAL is referred to as *analin*. The commands in this file name the other input and output files as well as the measurements to be performed. These commands are described in detail in the syntax specification and examples below. The -o file or standard output contains carnal's interpretation of the analin input and summary data for the run.

Note that periodic boundary conditions are only applied to DISTRIBUTION DIST measurements. This is because CARNAL allows measurements across multiple coordinate sets, but could be enabled in the normal case of measuring within a single set.

1.1.1. Input

CARNAL takes as input one or more "streams" of input coordinates, which are lists of restart, mdcrd, or Amber pdb format files. It detects formats transparently. The formats can be mixed in a stream, but the files must have the same number of atoms in the same order defined by the parm file for that stream. Each stream may have a different parm file. The `-p` argument or the first parm file defined in analin is used as the default parm file if no parm file is specified for a stream. At least one stream must be specified; the first one is used as the default when a stream reference is expected. CARNAL will also load static coordinate sets for comparison with the individual sets in the streams. NOTE: periodic boundary conditions are only applied to DISTRIBUTION DIST measurements; this is because of difficulties in measuring across 2 streams, but could (and at some point will) be enabled in the normal case of measuring within a stream. Compressed input files ending in `.Z` or `.gz` are uncompressed transparently, allowing disk space to be saved (the disk version remains compressed).

1.1.2. Output

CARNAL writes as output tables of measurements (scalar measurements or 0/1 hydrogen bond occupancies), distributions (including radial) and coordinate sets in mdcrd, restrt or pdb format. Summary data is also written to the file named with the `-o` argument on the command line, or standard output if no `-o` argument is given.

1.2. Analin introduction

This introduction is intended to give the feel of the analin language via an overview of the syntax and a simple example. A complete syntax definition and more complex examples are given below.

1.2.1. Summary of Analin Sections

These are the required sections in the analin input file syntax. Comments follow `'!'`s. In actual analin files, a `'#'` at the beginning of a line turns it into a comment. There are 4 main sections, each begun by a keyword in this order:

```
FILES_IN      ! name parm, coord/restrt/pdb files
FILES_OUT     ! name output tables, coord/restrt/pdb dumps
DECLARE       ! describe items to be measured
OUTPUT        ! direct declared stuff to output files
END           ! end input, start execution: STOP may be
              ! substituted for debugging: program stops
```

Things are declared in the first 3 sections and referenced in the last 2 sections. When something is declared it is given an id for referencing it later.

1.2.2. A Simple Analin Example

Select some coord sets from mdcrd files and output them in pdb format:

```
FILES_IN
```

```

    PARM  p1 ketop;           ! keyword, id, filename
    STREAM s1 kecrd kfcrd;   ! keyword, id, 2 filenames
FILES_OUT
    COORD  c1 /tmp/ke.p PDB; ! keyword, id, filename, output format
DECLARE
                                ! no declarations for this simple case
OUTPUT
    COORD  c1 s1 SELECT (1 3 5 200);
                                ! keyword, files_out id, files_in id:
                                ! command to select sets 1, 3, 5, 200
END

```

In this case, coord sets 1, 3, 5, and 200 from the concatenated stream of mdcoord files kecrd and kfcrd will be output to pdb files /tmp/ke.p.1 /tmp/ke.p.3 /tmp/ke.p.5 /tmp/ke.p.200.

1.3. Analin Syntax Specification

Notes

Aspects to be changed are indicated by 'CHANGE:'. Definitions must precede references: you cannot refer to something defined later in the file. Characters reserved for explicit purposes are:

```
# - . % ( ) & | , ! ?
```

A '#' as the first character of a line makes the line a comment. The format is entirely free, i.e. statements can be spread across multiple lines with any indentation and with comment lines embedded. Lines may not exceed 80 characters – re-read the preceding sentence if you think that causes a problem. In the syntax below, items in [] brackets are optional and items within {} braces are descriptions rather than token-by-token matchings.

---FIRST SECTION = "FILES_IN"

Input coordinates may be MD crd dump, inpcrd/restrt or Amber output pdb format.

FILES_IN

PARM id filename;

Amber Parm file. Multiple parms can be defined; the 1st one (or one defined by the [-p parm] argument) becomes the default for STREAMS that don't specify a parm.

STREAM strid [parmid]

[NOBOX] [ATOM n] [WIN x y] file1 file2 ... ;

At least 1 STREAM must be specified. STREAM files are read sequentially at each step. If > 1 STREAMS are named, they can be compared at each step. If no parmid is given, the first one defined is used by default. The NOBOX and ATOM options allow CARNAL to handle certain

discrepancies between the parm topology and the input stream. If these options are inaccurate, synchronization may be lost, resulting in garbage. ATOM is for reading in a stream that has *fewer atoms than the prmtop* - such a stream might have been created earlier using the ATOM option in the COORD section. All coordinate sets in a stream must have the same number of atoms.

NOBOX

Mdcrd files for periodic simulations have box coordinates after each coordinate set. Carnal automatically detects the presence of periodic conditions from the parm topology and allows for reading the box coordinates in mdcrd. However, minimization restrt files (as well as constant volume mdcrd files previous to 4.1) do not include the box coordinates. The NOBOX option allows carnal to read min.rst and old constant volume mdcrd files correctly.

ATOM n

Read only n atoms (more may be defined in the parm file). I.e. coordinates for only n atoms are in the stream. Implies NOBOX, *i.e.* if a box is specified in prmtop, it is ignored.

WIN

Means, "skip x sets, use y sets" repeatedly. This is for analysis of periodic equilibration / data collection runs such as gibbs.

STATIC statid [parmid]

[NOBOX] [ATOM n] file1 file2 ... ;

STATIC files are read at the beginning and remain in memory for comparison with STREAM coordinates. Each static set in an id can be referenced by 'id%1', 'id%2' etc. See STREAM above for "NOBOX" and "ATOM n" descriptions.

---SECOND SECTION = "FILES_OUT"

FILES_OUT

TABLE tabid filename ;

In the tables, there is one "logical row" per coordinate set measured, so a given measurement over a trajectory occupies a column. For example, the Nth item directed to the table (in the OUTPUT section, below) would form the Nth numerical column and the Ith measurement of that value would be in the Ith logical row. The logical rows are wrapped so that a row continues through a series of lines in a single file beginning with keys of columns by grepping for the key letter. If there is demand, rows can instead be spread across multiple files (filename.0 filename.1 ...) or just tabbed continuously within a file (harder to read visually). Thus, the format is:

```
L0    m1    m2    m3    m4    . . . |
```

```

L1  m11  m12  m13  m14  ...  |  1st coord set
L2  m21  m22  |
L0  m1   m2   m3   m4   ...  |
L1  m11  m12  m13  m14  ...  |  2nd coord set
L2  m21  m22  |

```

where the measurements are m1, m2, ... m22 extending over a logical line consisting of lines 'L0', 'L1', and 'L2'.

COORD crdid file [APPEND] [BLANK] [format] ;

APPEND

Add to end of named file if it exists.

BLANK

Write a blank line after each set.

Format symbols are 'PDB' 'RST' and 'CRD'. The default format is CRD.

HBOND hbid base_file [TABLE] [LIST];

TABLE

Output table of occupancies in base_file.tab. This table has two sections. The first part is a key that lists the possible hbonds in order. The format is:

```

# 1  (ADE  2 O5' )--(HB  1 H   ) .. (ADE  2 O1' )
# 2  (ADE  2 O5' )--(HB  1 H   ) .. (ADE  2 N7' )

```

The second part consists of a matrix of 0's and 1's. Each column is for a given hbond pair according to the numbering in the key section, and each row (line) is for a coordinate set. The format is '0' if no hbond is happening, '1' if it is. The Unix 'awk' utility can be used to extract column(s) of interest for further occupancy analysis or plotting, e.g. "egrep -v '^#' | awk '{print \$2, \$5, \$8}' base_file.tab" where the 'egrep' command strips out the key section and the 'awk' command selects the columns of interest. Note that if there are too many columns for awk to handle, the 'perl' utility may be needed.

LIST

Output list of per-hbond-per-set to base_file.lis for extensive analysis. The format is:

```

1 ( ADE  2 N6   )( THY  5 O4   ) 2.930768 9.125721
2 ( ADE  2 N6   )( THY  5 O4   ) 2.957820 3.151730

```

where the 1st number is the number of the coordinate set (starting with 1), followed by donor, acceptor, distance and angle (in

radians). Atoms are specified by residue name, residue number, and atom name. (See OUTPUT HBOND STAT for summary hbond info, including fraction of occupancy.)

HBOND specifications are given in the OUTPUT section.

Summary info is printed to standard output.

CHANGE someday: at least one of {TABLE, LIST} must be given, and the OUTPUT HBOND statement is req'd to do any hbond analysis.

DISTRIBUTION dbid filename [DAP][MIN];

DAP

Put number of intervals on 1st line.

MIN

For DIST option, below. For each 'solvent' atom, write out min distance to 'solute' for the run (multiple lists separated by a '%' are output if WIN is chosen with DIST). This is to figure out which waters to keep in a second pass dumping COORDs. List output goes to filename.min.

The definition of contents of the file is described in the OUTPUT section.

---THIRD SECTION = "DECLARE"

Each object is bound to a crd set; if not bound explicitly, the default is stream 0. References to that object inherit the binding of the object, except for within a GROUP statement (GROUP (GROUP id)...). I.e. in general an optional [crdset] is not allowed after a declared id is used (binding at reference rather than creation), for now.

"Points" can be atoms (specified by "number [crdset]" or "atom_name residue_number [crdset]") or centers of geometry or mass of groups of atoms (see GROUP definition below). For example, "PLANE id 12 34 58;" specifies the plane formed by atoms 12, 34 and 58 in the default stream, and "PLANE id OD1 2 ND1 4 OD1 6;" specifies the plane formed by atom name OD1 in residue 2, etc. "PLANE id gid1 gid2 gid3;" specifies the plane formed by the centers of geometry of three groups.

DECLARE

----Group is defined by set theoretic operations. Group attributes include center of geometry or mass, moment of inertia, and radius of gyration.

GROUP id [crdset] (((set OP set) OP set) ...);

The group is defined on the default stream unless [crdset] is given. The nesting in parentheses determines the order of evaluation.

OP can be either '&' or '|'

where '&' = intersection, '|' = union

and set can be any one of: { (ATOM numlist),
(ATOM [NAME|TYPE] namelist),

```
(RES numlist),
(RES NAME namelist),
(SOLUTE)
(GROUP namelist_of_groupids),
!set }
```

In the (GROUP) set, the groups are OR'd together. The NAME and TYPE options allow the use of '?' as a wild card matching any single character.

Allowing expressions:

```
groupid%center
    center of geometry - default if groupid is used as a point
groupid%cmass
    center of mass
groupid%momin
    moment of inertia
groupid%radgyr
    radius of gyration
```

For example, groupid%radgyr could be included in an OUTPUT / TABLE list (see below) and thus form a column of a table.

CUTRES id x y z cut;

CUTRES id groupid cut;

CUTRES produces a list in AMBER GROUP format (Appendix B) of all residues with any atom within *cut* of the given center point, or if a *groupid* is given, within *cut* of the group, including the residues in the group itself. It is intended primarily for analyzing a single coordinate set to generate a group of atoms within an area of interest that will be allowed to move in a belly dynamics or perturbation simulation.

AXIS id {2 points} ;

AXIS axid1 1 2 ;

Atoms 1 -> 2 in stream 0 by default.

AXIS axid1 1 st1id 2 st2id ;

Atom 1 in stream/static st1 -> atom 2 in stream/static st2.

AXIS axid2 grp1id grp2id%cmass;

grp1 center of geometry to grp2 center of mass: note that the groupids may be the same or groups may be defined on different streams.

PLANE plid { 3 points or 2 axes } ;

A plane is treated as its normal vector where appropriate.

ANGLE angid { 3 points or 2 axes/planes };

Planes are interpreted as normal vectors.

TORSION tid { 4 points or 3 axes/planes };

Planes are interpreted as normal vectors. Note that in the averages printed to standard output, carnal section).

TORSION tid BACKBONE [residue1 [residue2]] [crdset] ;

Find all torsions involving backbone bonds (between Amber main chain atoms), starting with residue1 (default: 1st residue) and ending with residue2 (default: last residue). If first and last residues' terminal

backbone atoms are bonded to each other, torsions involving them are included.

DIST dsid { 2 of: points, axes, planes };
 Select 2 points, 2 axes, or point and axis or plane. [planes and axes are not supported yet]

IMAGE imageid groupid ;

IMAGE imageid groupid%cmass ;

Only for periodic systems. Place the system so that groupid (center of geometry or center of mass) is at the center of the box, and image all residues accordingly. Cleans up Ewald runs. Uses groupid's stream. Imageid can be used as a streamid in measurements. NOTE: this is not guaranteed to give the desired, intact system on the first try – it depends on the transformations that Ewald has made and the size of the box. For example, the center of geometry of a DNA duplex could be in the center of the box, but the strands could be on the edges. Successive transformations using trial and error may be required to restore an Ewald trajectory to normal appearance, and different transformations may be required for different frames. Also, intact molecules can be broken up by this option if centering makes some residues project out of the box (possibly an indication that the box size used was too small).

RMS rmsid [FIT] groupid ;

RMS rmsid [FIT] groupid streamid ;

RMS rmsid [FIT] groupid streamid refercid ;

RMS rmsid [FIT] groupid staticid [ATOM] [RES];

RMS rmsid groupid2 prevrmsid ;

Using atoms in groupid, measure rms of one coordinate set to another. If FIT is selected, the current coordinate set is first positioned for minimum mass-weighted rms of the group on the reference coordinates; this also allows the rmsid to be used to determine the resulting (non-fitted) RMS measurement on other groups (as in the prevrmsid case above), and the FITted rmsid can be used like a stream for other measurements, as well as output via a COORD statement. Per-residue and per-atom RMS values within the group can be output in a TABLE by rmsid%residues or rmsid%atoms.

The first and simplest case above uses groupid to compare the default stream to its first set. The second case compares a named stream (rather than the default) to its first set. The third case specifies both the stream and a reference set for comparison; this reference set could be a static (single) set or another stream (comparing successive sets in each stream).

The fourth example, using a STATIC id, produces a (triangular) matrix of RMS values, one for each pair of coordinate sets in the STATIC id. The ATOM and RES options additionally cause per-atom and per-residue RMS to be reported. All output is calculated and printed immediately, since it does not depend on reading a serial stream. The matrix format for a 3-coordinate-set STATIC would be:

```

---RMS MATRIX
set0 set1 value0
set0 set2 value1
set1 set2 value2
---
```

NOTE: use of FIT with this option leaves all coordinate sets in the STATIC set with the group center of mass placed at the origin and each successive coordinate set rotated to fit its predecessor.

The final case measures the rms of a second group on a pair of sets that were positioned by a previous RMS FIT statement. See OUTPUT TABLE for instructions on obtaining per-residue and per-atom rms for streams. Any number of any of these types of RMS measurements can be used.

DME dmeid groupid ;

DME dmeid groupid streamid ;

DME dmeid groupid streamid refcrdid ;

DME dmeid groupid staticid;

Using atoms in groupid, measure Distance Matrix Error between one coordinate set and another. DME compares intra-group distances in one conformation to those in another conformation.

$$DME = \sqrt{2/(N * (N - 1))} * \frac{\sum}{atom\ pairs} (distance - refdistance)$$

In proteins, the convention is to use DME for groups defined on *C-alpha* atoms. The first and simplest case above uses groupid to compare the default stream to its first set. The second case compares a named stream (rather than the default) to its first set. The third case specifies both the stream and a reference set for comparison; this reference set could be a static set or another stream. The final example produces a (triangular) matrix of DME values, one for each pair of coordinate sets in the STATIC id. The format for a 3-coordinate-set STATIC would be:

```

---DME MATRIX
set0 set1 value0
set0 set2 value1
set1 set2 value2
---
```

PUCKER pukid NUCLEIC [streamid] [residue_names|residue_numbers] ;

PUCKER pukid number_of_points points ;

Measure pucker using algorithm of D. Cremer and J. A. Pople (JACS 96:6 pp 1354-1358, 1975). For the NUCLEIC options, the Altona and Sundaralingham convention (JACS 94 pp 8205-8212, 1972 or p. 20 of Saenger's "Principles of Nucleic Acid Structure", Springer-Verlag, 1983) is approximated by adding 90 degrees to the phase angle, the puckers are

always ordered according to residue order in the parm file, and the standard atom names (O4'/O1', C1', C2', C3', C4') are used to determine the points. (For a comparison of different nucleic acid pucker conventions, see S. C. Harvey and M. Prabhakaran, JACS 108:20, pp 6128-6136, 1986.) In the general case (specifying points explicitly), a ring of N points can be parameterized by N-3 alternating amplitudes and phase angles. Note that the Cremer/Pople algorithm finds a mean plane based on the assumption that successive points in the ring have the same angle between them with respect to the center of geometry, so for kinky rings this may not work. Note that in the averages for angles printed to standard output, carnal section).

PUCKER pukid NUCLEIC;

Measure pucker of all standard residues ('94 force field: G5, G, G3, GN, *etc.*; '91 force field: GUA, *etc.*) in default stream using standard atom names (O4'/O1', C1', C2', C3', C4').

PUCKER pukid NUCLEIC streamid;

Same as above, but uses specific stream rather than default.

PUCKER pukid NUCLEIC GUA;

Measure pucker of all residues named 'GUA' using standard atom names.

PUCKER pukid NUCLEIC 2,4,6,8 ;

Measure pucker of residues 2,4,6, and 8 using standard atom names.

PUCKER pukid 5 O1' 2 C1' 2 C2' 2 C3' 2 C4' 2 ;

Measure pucker of 5 points: O1' (residue 2), C1' (residue 2), etc.

---FOURTH SECTION = "OUTPUT"

OUTPUT

TABLE tbid { column_list } ;

At least one column must be specified. Columns are printed in their order in the list. Column_list may include ids, classes of measurement (e.g. DIST) which print in the order declared, MEAS which prints all scalar measurements, or ALL which prints everything. AXIS ids result in vectors, PLANE ids in normals, and GROUP ids default to center of geometry unless attributes are specified, such as grpид%cmass and grpид%radgyr. RMS ids default to the rms of the group, while rmsid%residues and rmsid%atoms give per-residue and per-atom rms respectively. For per-residue rms, the group must not have any partially-included residues. If either per-residue or per-atom options are used, the statistics are printed in the summary with the residue and atom names.

COORD crdid streamid

[SELECT (-i j k,l m-p q-)] [MOD h]

[AVERAGE] [ATOM n] [EXH2O m [GROUP gid]] [INH2O gid] ;
SELECT (-5 7 8,10 100-105 200-)

Select certain sets from the stream by order. Numbers are separated by spaces or commas, and '-' is used to indicate ranges. In this example, select sets 1 through 5, then sets 7, 8, and 10, then sets 100 through 105, then sets 200 through the end. Note that this option selects files for output only, and does not affect measurements on the stream, as opposed to the STREAM WIN option, which pre-selects sets for all the other commands.

MOD h

Select every h'th set from the stream. Note that this option selects files for output only, and does not affect measurements on the stream, as opposed to the STREAM WIN option, which pre-selects sets for all the other commands.

AVERAGE

Average the coordinates. Not compatible with EXH2O or INH2O, but ok with ATOM. Produces a single set, so PDB or RST format is advised for the corresponding FILES_OUT COORD declaration. Suggested that this be applied to an RMS FIT streamid so that the area of interest has minimal distortion from drift or pressure scaling of the box. Note: The averaged coordinates may be a chemically unrealistic hybrid of different regions of phase space, so visual inspection, energy analysis, and perhaps energy minimization may be in order, depending on the purpose. As a simple example, the methane C-H bond length shortens due to H rotation. Note 2: a measurement on averaged coordinates is not the same as the average of the same measurement over the same trajectory.

ATOM n

Output only the first n atoms. ATOM, GROUP, INH2O and EXH2O are mutually exclusive options.

GROUP id

Output only atoms in the group. ATOM, GROUP, INH2O and EXH2O are mutually exclusive options.

EXH2O m [GROUP gid]

Omit all but m waters from the set, retaining either those closest to the non-waters, or those closest to the atoms specified by GROUP. Distance is measured from water oxygen. Waters are printed in order of closeness to the solute, i.e. the order varies from set to set, so identity-based dynamic graphics smoothing schemes will fail. ATOM, GROUP, INH2O and EXH2O are mutually exclusive options.

INH2O gid

Omit all waters from the set except those with atom type OW in group gid, where gid contains only OWs. This group is intended to be built with the output of a previous pass using DISTRIBUTION MIN which informs the user how close each water came to the area of interest during the run. See example below. ATOM, GROUP, INH2O and EXH2O are mutually exclusive options.

HBOND hbid [DONOR [EXACT] g1] [ACCEPTOR [EXACT] g2]
[DISTANCE x] [ANGLE y] [STATS];

DONOR and ACCEPTOR indicate group ids for searching for the appropriate atoms (note: donor is the heavy atom to which the hydrogen is attached). The default for either is all atoms. A single group id may be given in place of separate definitions. If DONOR and/or ACCEPTOR are specified, the EXACT option forces carnal to use all heavy atoms that may apply, instead of just the 'classic' ones such as oxygen and nitrogen.

DISTANCE

Cutoff distance in angstroms between the heavy atoms: default is 4.0.

ANGLE

Cutoff H-donor-acceptor angle in degrees (0 is linear): default 1 radian \approx 60 degrees.

STATS

Directs printing of per-hbond summaries to standard output. The format is:

```

HBOND h1 stats:
# 19 (ADE 2 N6 )_(ADE 2 HN6A)..(THY 5 O4 ) % 64.400000
      distance      avg: 2.909575   max 2.961379   min 0.000000
      angle(deg)    avg: 7.241544   max 15.219878  min 0.000000

```

where the # refers to the column of file.tab and the '64.400000' gives the percentage of occupancy. The other statistics are only for the "occupied" states. The distance is between donor and acceptor atoms.

DISTRIBUTION dsid

```
{ RAW | min max nboxes [WIN nsets] }
{ measid | DIST group1 [ group2 [ALL]] [NORM] } ;
```

Distribution output can be either RAW (a long line of ascii floating point numbers per coordinate set) or binned and normalized. If the latter, the WINDOW option causes the distribution for each nsets supplied by the STREAM to be written, with a '%' line to separate each window.

RAW is the recommended option for large data sets, since the proper range and number of bins are hard to guess at: the rdis program can be used on the raw output to quickly try various min/max/nboxes numbers on the raw file. Note, however, that measurements including many terms can generate files larger than the original trajectory, so the RAW option may not be appropriate in such cases. For example, when measuring O-O distributions in a system of N waters, there are 9N numbers per coordinate set, but $N(N-1)/2$ distances to write out if RAW is used. For N=1000 this amounts to a RAW file 55 times larger than the trajectory.

Either an id for a scalar measurement or a radial distance distribution (DIST) may be specified. In the latter case, one or two groups can be specified. A single group may be given, in which case all intra-group

distances are used (this would be appropriate for e.g. water O-O); otherwise, two groups are required. When just two groups are given (without the ALL option), the groups are treated as "solute" and "solvent" respectively: for each "solvent" atom, the distance to the closest "solute" atom is applied. The ALL option includes all group1-group2 distances rather than the "solvent" to closest "solute" atom. Note that when two groups overlap, distances of 0 would be obtained for the atoms that are in both groups, so groups should be disjoint. The MIN option from the FILES_OUT section above is only valid for the plain, two-group mode.

Volumetric NORMAlization is optional when radial DISTribution is selected. This only makes sense for measuring the distribution around a single atom or a set of chemically identical atoms, since the normalization is done by dividing the count for each interval by the volume of a spherical shell having radii equal to the shell boundaries. (To normalize the distribution around a functional group, for example, would require calculating the volumes of the non-spherical shells around the group.)

The output format is: "bin_center value smoothed_value integral," where bin_center is the center of the interval of the bin (the first is $\min + 0.5 * (\max - \min) / \text{nboxes}$), the value is the distribution for that bin, and the integral is the cumulative sum at the current bin. The integral is not affected by the NORM option.

1.4. Examples

1.4.1. Simple coordinate averaging

```
#Simple Coordinate Averaging
FILES_IN
  PARM  p1 ketop;           ! keyword, id, filename
  STREAM s1 kecrd kfcrd;   ! keyword, id, 2 filenames
FILES_OUT
  COORD  c1 /tmp/ke.p PDB;  ! keyword, id, filename, output format
DECLARE
                                ! no declarations for this simple case
OUTPUT
  COORD  c1 s1 AVERAGE;
                                ! keyword, files_out id, files_in id:
                                ! command to average sets
END
```

1.4.2. Simple distance, angle, and torsion measurements

```
# Plain measurements involving points (atoms, centers of mass)
FILES_IN
  PARM  p1 prm.top;
  STREAM s1 a1.trj a2.trj a3.trj;
FILES_OUT
```

```
TABLE tab1 meas.tab;
DECLARE
#
#   First, some measurements using atoms only: format is:
#       FUNC_NAME ID atom_specs ;
#   each atom_spec can be either:
#       ATNAME RESNUM
#   or
#       ATNUM
#
DIST dist1 O1' 2 O1' 7;
ANGLE ang1 2 12 13;
TORSION tor1 C1 4 C2 4 C3 4 C4 4;
#
#   A special case for torsions:
#
TORSION tor2 BACKBONE;
#       ^ find all torsions consisting completely of
#       main chain atoms
#
#   Now some more geometrical stuff: the angle between
#   the normal vectors of two planes:
#
PLANE pla1  C1 4 C2 4 C3 4;
PLANE pla2  C1 5 C2 5 C3 5;
ANGLE ang2  pla1 pla2;
#
#   Now some measurements using composite points:
#   format is the same, except for atom_specs.
#   First we'll define a group consisting of the non-waters,
#   and a group consisting of atoms in 3 numerically adjacent
#   residues:
#
GROUP g1 (SOLUTE);
GROUP g2 (RES 5,6,7);
#
#   And now to measure the distance between the centers of
#   mass of each group to see how that pair of residues
#   fluctuates from the center:
#
DIST dist2 g1%cmass g2%cmass ;
#
#   Now let's define 2 more residue-based groups:
#
GROUP g3 (RES 21,22,23);
GROUP g4 (RES 44,45,46);
#
#   And we'll measure the angular fluctuations of the 3
#   residue-based centers of mass:
```

```

ANGLE ang3 g2%cmass g3%cmass g4%cmass ;

OUTPUT
#   Now direct all the measurements to the table defined above
#   MEAS refers to all scalar measurements; each 1 will be a
#   column in the order defined (dist1 ang1 tor1 tor2 dist2 ang2).
#   Alternatively, the ids could be given explicitly in any order.
#
TABLE tab1 MEAS;
END

```

1.4.3. RMS deviation

You want a measurement of the minimum RMS deviation of a group of atoms as a measure of how disordered some structures are relative to another structure. Given the best fit on that group of atoms, you also want to know how much another subgroup differs and how much all the atoms outside the fit group differ. You also want to save the fit structures for viewing.

```

#
# RMS example: fit central 8 bases of a G4 DNA quadruplex
#
FILES_IN
  PARM p1 p524.top;
  STREAM s1
    sm4.pdb
    sm9.pdb
    sm17.pdb;
  STATIC ref_set sm3.rst;
#           ^^^^^^^ this file will be used as one
#           reference set for the comparison;
#           no pdb file around, but that's ok
FILES_OUT
  TABLE tbl sm.rms;
#           ^^^^^^^ this is a table for the per-set rms values
  COORD crd fit.p PDB;
#           ^^^ save some structure(s) in PDB format
#           ^^^^^ name of file
DECLARE
#
#   Now let's get down to business.
#   Declare a group of atoms to fit on - all the
#   non-sugars in the central 8 GUAs:
#
GROUP grp1 ( (ATOM NAME N9 C8 H8 N7 C5 C6 O6 N1 H1
              C2 N2 HN2A HN2B N3 C4)
             & (RES 4,6, 13,15, 22,24, 31,33) );
#           ^ boolean for "all of the atom names in
#           the 1st part that occur in the following

```

```

#           residue numbers"
#
# RMS fit the structures in the stream to the reference
# set using the group of atoms just defined (fitting is
# mass-weighted). This creates a new thing that can be
# treated as a stream.
#
RMS fit1 FIT grp1 s1 ref_set;
#           ^^^^^^^ reference structure id -
#                   if not given, the first
#                   structure in the stream
#                   would be used; or this id
#                   could be for a different
#                   stream instead of the static
#                   coord set used here
#           ^^ stream id - what to fit
#           ^^^ group of atoms to use for fitting
#           ^^^ fit (position) the stream set to minimize rms
#           ^^^ 'fit1' is the new, streamlike thing
#
# Specify the group of atoms to measure a secondary
# deviation - the terminal bases on each strand, w/out
# the sugars:
#
GROUP g2 ((RES 2,8, 11,17, 20,26, 29,35) &
          (ATOM NAME N9 C8 H8 N7 C5 C6 O6 N1 H1
            C2 N2 HN2A HN2B N3 C4));
#
# Measure the RMS on that group resulting from the fit
# of the other bases, i.e. between the target structure
# and the new, fitted structure. Just measuring this
# time, not creating a new set.
#
RMS fit2 g2 fit1 ref_set;
#
# Let's see what that fit does for all the atoms outside
# of the fit, not just the end bases.
# Specify the group of all atoms not in the original group
# used for the fitting:
#
GROUP g3 (!grp1);
#
# Measure the RMS of that group on the fitted structures
# as before.
#
RMS fit3 g3 fit1 ref_set;
#
# Just for fun, create a new fitted set using the 1st group
# but using the 1st set in the stream as reference.

```

```

#
RMS fit4  FIT  grp1  s1;
#           ^^ just specifying the stream with no
#           reference defaults the reference to
#           the 1st set in the stream
#           ^^^^ use our "central bases" group again
#           ^^^ FIT the thing
#           ^^^^ name of a new, stream-like thing starting with the
#           second crd set in the stream
#
OUTPUT
#
#   Write the RMS values to the table:
#
TABLE tbl  fit1  fit2  fit3  fit4;
#           ^^^^  ^^^^  ^^^^  ^^^^ output the measured rms
#                               values as columns in the
#                               table declared as a file
#                               above
#           ^^^ to table 'tbl'
#
#   Average the the structures fitted using the 1st group
#   on the reference set and print them to the coordinate
#   file defined above. Perhaps we will then energy min
#   this structure and claim it means something.
#
COORD c1  fit1  AVERAGE;
#
END

```

1.4.4. Coordinate selection: waters

You want coordinate dump of solute with selected waters. Not the closest waters at each step: you *know_exactly* which waters you want: the same ones in every set, maybe so that when you smooth the trajectory, the Nth water won't change its identity at each step. This is a 2-pass procedure: first you need to get a list of the waters: you need the atom number of the OW (atom type) in each water. If you want those waters to be all those that came within a given distance of the solute, have I got an option for you. The thing to have is, for each water, the closest it came to whatever you want to call the solute. DISTRIBUTION MIN will give you a list of atom number, distance pairs that you can sort to generate the list of favored waters you need. With this list of OW atom numbers, you can define a group which, in another pass, can be used to filter waters.

```

# use of INH2O with DISTRIBUTION MIN - 1st pass
FILES_IN
  PARM p1 prm.top;
  STREAM s1 a1.trj a2.trj a3.trj;
FILES_OUT
  DISTRIBUTION d1 file MIN;

```

```

#                               ^this is the key: creates "file.min"
DECLARE
  GROUP g1 (SOLUTE);
  GROUP g2 (ATOM TYPE OW);
#                               ^^ have to use OW
OUTPUT
  DISTRIBUTION d1 0.0 10.0 10 DIST g1 g2;
#                               ^^^^^^^^^^^^^ you'll also get the
#                               net curve; RAW ok
#                               ^^^^^ this is the 2nd key
#                               ^^^^^ order is solute then
#                               solvent
END

```

--Now the critical step: filtering the water. First we use 'awk' to see what waters we want to keep, e.g.:

```
awk '$2 < 3.0 {print $1}' file.min | wc -l
```

This tells you how many water oxygens came within the cutoff (3.0 in this example). Choose a cutoff such that the resulting list of type OW atoms is the right size for you and:

```
awk '$2 < 3.0 {print $1}' file.min > temp
```

Now you need to take the list of OWs in the temp file and include it in a GROUP ATOM statement as in the following example in order to select the waters into a coordinate dump.

```

# use of INH2O with DISTRIBUTION MIN - 2nd pass
FILES_IN
  PARM p1 prm.top;
  STREAM s1 a1.trj a2.trj a3.trj;
FILES_OUT
  COORD c1 filtered.trj;
DECLARE
  GROUP g1 (ATOM 2,11,26,29);
#                               ^^^^^^^^^^^^^ the type OW atom numbers of your choice
OUTPUT
  COORD c1 INH2O g1;
END

```

1.4.5. Radial distance distributions

```

# DISTRIBUTION EXAMPLE: ONE GROUP to itself (OW-OW)
FILES_IN
  PARM p1 prmtop;
  STREAM s1 crd;
FILES_OUT
  DISTRIBUTION d1 xdb ;

```

```

DECLARE
  GROUP g1 (ATOM TYPE OW);
OUTPUT
  DISTRIBUTION d1 RAW DIST g1 ;
#           ^ group id from DECLARE GROUP
#           ^ distance macro
#           ^ dump all distances to file for use w/ rdis
#           program (i.e. don't bin measurements or
#           output bins)
#           ^ id
#           For each 'solvent' group atom, the nearest 'solute' atom
#           is found and binned if it satisfies the min, max criterion.
END

# DISTRIBUTION EXAMPLE: TWO GROUPS using closest atom in 1st to
#                       each of 2nd
FILES_IN
  PARM p1 prmtop;
  STREAM s1 crd;
FILES_OUT
  DISTRIBUTION d1 xdb ;
DECLARE
  GROUP g1 (RES NAME ADE);
  GROUP g2 (RES NAME THY);
OUTPUT
  DISTRIBUTION d1 RAW DIST g1 g2 ;
#           ^ 2nd group is the 'solvent'
#           ^ 1st group is the 'solute'
#           ^ distance macro
#           ^ dump all distances to file (don't bin)
#           ^ id
#           For each 'solvent' group atom, the nearest 'solute' atom
#           is found and binned if it satisfies the min, max criterion.
END

# DISTRIBUTION EXAMPLE: TWO GROUPS using all inter-group pairs
FILES_IN
  PARM p1 prmtop;
  STREAM s1 crd;
FILES_OUT
  DISTRIBUTION d1 xdb ;
DECLARE
  GROUP g1 (RES NAME ADE);
  GROUP g2 (RES NAME THY);
OUTPUT
  DISTRIBUTION d1 RAW DIST g1 g2 ALL;
#           ^ consider all group1-group2
#           interactions

```

```

#           ^ 2nd group id from DECLARE GROUP
#           ^ 1st group id from DECLARE GROUP
#           ^ distance macro
#           ^ dump all distances to file (don't bin)
#           ^ id
END

```

1.4.6. Hbond examples

You want a occupancies for all possible hbonds at each step. This file will consist of a line for each coordinate set in the stream with a '0' or '1' followed by a space for each possible hbond. You also want the percentage occupancy of each hbond over the run, and the average distance and angle when occupied. And while you're at it, you want to print the distance and angle of each possible hbond.

You also want to specify the maximum distance and angle that qualify an hbond. The percentages and averages are written to the main output at the end of the run.

```

FILES_IN
    PARM p1 hbtop;
    STREAM s1 hbmd;
FILES_OUT
    HBOND h1 xhb TABLE LIST;
#           ^ write a list of distances/angles
#           to "xhb.lis"
#           ^ write occupancies to "xhb.tab"
#           ^ use "xhb" as the basis for filenames
#
#
DECLARE
OUTPUT
    HBOND h1 DISTANCE 3.3 ANGLE 20.0 STATS;
#           ^ print the averages
#           of the occupied cases
#           ^ limit the angle;
#           default 1 radian ~= 60 degrees
#           ^ limit the distance between heavy atoms;
#           default 4 Angstroms
END

```

Perhaps you want to specify the donor and acceptor groups, if only to limit the number of columns in the table. This time, we'll also just use the default criteria for hbonds.

```

# HBOND ANALYSIS EXAMPLE: USING GROUPS FOR DONOR/ACCEPTOR
FILES_IN
    PARM p1 hbtop;
    STREAM s1 hbmd;
FILES_OUT
    HBOND h1 xhb TABLE LIST;
DECLARE

```

```
DECLARE
  GROUP g1 (ATOM TYPE N2 NA);
  GROUP g2 (ATOM TYPE NC O);
OUTPUT
  HBOND h1 DONOR g1 ACCEPTOR g2 STATS;
END
```