VisTraj User’s Guide

By Howard J Feldman and John J Salama (updated 2011)

What is VisTraj?

Overview

VisTraj is a utility designed to complement our protein conformational space exploration software, Foldtraj, also included in this package. It can be used instead of Maketrj, the command-line equivalent to this program (also included, and normally called from the included scripts InitTraj, Val2Traj or UnfoldTraj). VisTraj adds many new features and capabilities not available with the command-line version. VisTraj is written using the OpenGL 3D Application Programming Interface (API). NCBI's Vibrant windowing API was used for making the GUI, which allows the same code to be built on several flavours of UNIX in addition to Windows.

Trajectory Distributions Files

The Trajectory Directed Ensemble Sampling (TraDES) package consists of VisTraj/Maketrj and Foldtraj. Central to this set of tools is the concept of the trajectory distribution. Simply put, a trajectory distribution is a map of available conformational space at a residue. Thus a protein has a database of trajectory distributions, one representing the conformational space at each residue in the sequence. The database contains additional information about each residue as well (see Residue Properties section). A trajectory distribution file (normally ending in .trj or .trj.bz2) consists of a trajectory distribution database, plus some global parameters describing the space (see Global Properties section). All files loaded by VisTraj should have one of these two extensions. Note that .trj.bz2 files are simply .trj files which have been compressed by the bzip2 program. Once loaded, they will be un-zipped and remain so unless you zip them again to save space. Foldtraj generates 3D protein conformers, sampling the conformational space represented by a trajectory distribution file, using the .trj file as input. Presently, foldtraj must still be run from the command line. However you CAN generate single conformers from within VisTraj. See README.txt that comes with the foldtraj package for details on running command-line foldtraj.

Quickstart or Proteins for Dummies

Making probabilistic protein conformers can be easy. To quickly get started, select New from the File menu, pick "Sequence", then "Amino Acid Input" and the Next. Enter your protein sequence and hit Next. Enter a name for your trajectory file in the "Output File" box and hit Next. In the next window hit Done. In a few moments you'll have your first trajectory distribution built. Now to make protein structures sampled using your new trajectory distribution, select Generate Conformer from the File menu, and pick PDB (if you have Rasmol installed) or MMDB (if you have Cn3D). After a few moments, the chosen molecular viewer will open up with your structure loaded. To create several structures, you need to exit VisTraj. Then at the command prompt, run foldtraj. First check to see that the file you just created is really there, so assuming you saved it in the same directory as VisTraj and Foldtraj, just type DIR (in Windows at the DOS prompt) or ls -l (if using UNIX) and look for your file. It will end in either *.trj or *.trj.bz2; either is OK to give to foldtraj as input. Let us assume your file is named protein.trj.bz2. Now to build 10 conformers of your protein, simply type:

```
foldtraj -i protein -b 10
```

at the DOS or UNIX prompt and hit enter. After a few seconds of initialization, Foldtraj will begin making structures, showing you its progress and dumping them out as .pdb files. In this case the output structures would be called protein_0000001.pdb, protein_0000002.pdb and so on up to 10. More help on using the advanced features of foldtraj can be found in its own README.txt file, in the same directory as the actual Foldtraj program.

More on Trajectory Distributions

Conformational space can be represented in two distinct ways with TraDES. Ramachandran space consists of the F-Y backbone dihedral angles used to describe backbone conformation. a-carbon space uses as its two degrees of freedom the a-carbon "trajectory" at the residue, sometimes referred to as t and a by others. We denote this space by f, the supplement of the angle between the previous two a-carbons and the one in question, and q, the torsional angle between the carbon of interest and the previous three. These two angles happen to correspond to the spherical polar co-ordinates of the i+1 a-carbon if the sphere is oriented such that the ith carbon is placed at the centre of the sphere, the i-1 at the south pole, and the i-2 along the prime meridian (q=0). In both cases, it is necessary to discretize space into finite patches in order that the probability maps may be stored in a matrix. A size of 400 x 400, or approximately 1 resolution, was found to work well, though this can be altered in theory. Thus the entry at any point in the matrix corresponds to the probability that the conformation of a residue falls somewhere in that 0.9 by 0.9 bin, regardless of which of the two spaces we are in.

Initial probability distributions are generated by observing conformations of each of the twenty residue types in a non-redundant set of the PDB, and using the frequency counts in each bin as the default matrix for each of the twenty residues. The counts can be converted to probabilities by simply normalizing (i.e. dividing by the integral of the matrix).

We can further divide our sampling of the PDB into helix, sheet and coil for a total of sixty "dictionary" trajectory distributions as follows: we process the structures in a non-redundant set of the PDB and look at the conformation, amino acid, and secondary structure of each residue, adding a count to the bin corresponding to the conformation in the trajectory distribution corresponding to the amino acid type and secondary structure. This gives us a set of "default" probability distributions to use when no other information is known about the protein. If we know, for example, that a particular Alanine amino acid residue is known to be in a helix, we can assign that residue the dictionary distribution for helical Alanine and ensure that only that region of space will be sampled. There is a small caveat to using a-carbon space, namely that it is inherently uneven, being on the surface of a sphere. Just as a globe is distorted when laid out flat on paper, so is the trajectory distribution. To ensure as even sampling as possible, the 400 x 400 bins all have exactly the same area on the surface of the sphere. This is easily accomplished by using cos f as our ordinate rather than f, hence the appearance of this "strange" unit when viewing unrolled spherical maps in VisTraj.

L-Amino Acids

A
`[Ala] - Alanine`

C
`[Cys] - Cysteine`

D
`[Asp] - Aspartate`

E
`[Glu] - Glutamate`

F
`[Phe] - Phenylalanine`

G
`[Gly] - Glycine`

H
`[His] - Histidine`

I
`[Ile] - Isoleucine`

K
`[Lys] - Lysine`

L
`[Leu] - Leucine`

M
`[Met] - Methionine`

N
`[Asn] - Asparagine`

P
`[Pro] - Proline`

Q

[Gln] - Glutamine

R
[Arg] - Arginine

S
[Ser] - Serine

T
[Thr] - Threonine

V
[Val] - Valine

W
[Trp] - Tryptophan

Y
[Tyr] - Tyrosine

Modified Amino Acids Supported

Acetylated

N-acetyl-L-alanine	[A:ac]
N-acetyl-L-arginine	[R:ac]
N-acetyl-L-asparagine	[N:ac]
N-acetyl-L-aspartic acid	[D:ac]
N-acetyl-L-cysteine	[C:ac]
N-acetyl-L-glutamine	[Q:ac]
N-acetyl-L-glutamic acid	[E:ac]
N-acetylglycine	[G:ac]
N-acetyl-L-histidine	[H:ac]
N-acetyl-L-isoleucine	[I:ac]
N-acetyl-L-leucine	[L:ac]
N2-acetyl-L-lysine	[K:ac]
N6-acetyl-L-lysine	[K:N6ac]
N-acetyl-L-methionine	[M:ac]
N-acetyl-L-phenylalanine	[F:ac]
N-acetyl-L-proline	[P:ac]
N-acetyl-L-serine	[S:ac]
N-acetyl-L-threonine	[T:ac]
N-acetyl-L-tryptophan	[W:ac]
N-acetyl-L-tyrosine	[Y:ac]
N-acetyl-L-valine	[V:ac]

Amidated

L-alanine amide	[A:am]
L-arginine amide	[R:am]

Formylated

N-formyl-L-methionine	[M:form]

Hydroxylated

4-hydroxy-L-proline	[P:hy_g]

Lipid Modified

S-farnesyl-L-cysteine	[C:farn]
S-geranylgeranyl-L-cysteine	[C:ger]
N-palmitoyl-L-cysteine	[C:palm_n]
S-palmitoyl-L-cysteine	[C:palm_s]
N-myristoyl-glycine	[G:myr]
N6-myristoyl-L-lysine	[K:myr]

Methylated

N-methyl-L-alanine	[A:meth_n]
N,N,N-trimethyl-L-alanine	[A:meth_n3]
omega-N,omega-N-dimethyl-L-arginine	[R:meth_n7]
L-beta-methylthioaspartic acid	[D:meth_b]
N5-methyl-L-glutamine	[Q:meth_n5]
L-glutamic acid 5-methyl ester	[E:meth_o5]
3'-methyl-L-histidine	[H:meth_n4]
N6-methyl-L-lysine	[K:meth_1]
N6,N6-dimethyl-L-lysine	[K:meth_2]
N6,N6,N6-trimethyl-L-lysine	[K:meth_3]
N-methyl-L-methionine	[M:meth]
N-methyl-L-phenylalanine	[F:meth]

Phosphorylated

omega-N-phospho-L-arginine	[R:po]
L-aspartic 4-phosphoric anhydride	[D:po]
S-phospho-L-cysteine	[C:po]
1'-phospho-L-histidine	[H:po_e]
3'-phospho-L-histidine	[H:po_d]
O-phospho-L-serine	[S:po]
O-phospho-L-threonine	[T:po]
O4'-phospho-L-tyrosine	[Y:po]

Other

L-selenocysteine	[C:sel]
L-selenomethionine	[M:sel]
L-3-oxoalanine	[S:oxal]

2-pyrrolidone-5-carboxylic acid	[E:pyro]
L-glutamyl 5-glycerylphosphorylethanolamine	[E:gpe]
2'-[3-carboxamido-3-(trimethylammonio)propyl]-L-histidine(diphthamide)	[H:diph]
N6-biotinyl-L-lysine	[K:biotin]
N6-(4-amino-2-hydroxybutyl)-L-lysine (hypusine)	[K:hypu]
N6-retinal-L-lysine	[K:retin]

Residue Properties

The following settings pertain to an individual trajectory distribution for one residue in a protein; each residue will have its own unique setting.

Peak

Largest single value in the trajectory distribution. Allows comparison of relative sizes of different trajectory distributions. Units are given below.

Integral

Total area under the trajectory distribution. This is the sum of all the elements in the 400 x 400 matrix. Units are given by the global property Trajectory Distribution Units.

Cis-Peak

Largest single value in a cis-proline trajectory distribution. Allows comparison of relative sizes of different cis-proline trajectory distributions. For non-proline residues, this field is unused.

Cis-Integral

Total area under a cis-proline trajectory distribution. Again, this is only applicable for proline residues.

p(cis)

Percent probability that the bond between residues i and i+1 will be cis (zero unless i+1 is Proline). Affects which of the two above trajectory distributions will be used, as well as other relevant factors.

p(SS)

Percent probability that the residue is involved in a disulfide bridge (zero unless cysteine) (presently unused).

Omega (Mean)

Average peptide dihedral angle (in degrees) to use between residue i-1 and i. When a residue is placed in the cis- conformation, 180 minus this angle is used instead.

Omega (SD)

Standard deviation (in degrees) to use when choosing w randomly.

Chi 1

First sidechain dihedral angle, in degrees, if any. Note the resolution is only 1.4 . Use 0 to choose from rotamer library. If non-zero, foldtraj will try to use this value for the dihedral angle when building this residue, provided steric clashes do not forbid it.

Chi 2

Second sidechain dihedral angle, in degrees, if any. Note the resolution is only 1.4 . Use 0 to choose from rotamer library. If non-zero, foldtraj will try to use this value for the dihedral angle when building this residue, provided steric clashes do not forbid it.

Chi 3

Third sidechain dihedral angle, in degrees, if any. Note the resolution is only 1.4 . Use 0 to choose from rotamer library. If non-zero, foldtraj will try to use this value for the dihedral angle when building this residue, provided steric clashes do not forbid it.

Chi 4

Fourth sidechain dihedral angle, in degrees, if any. Note the resolution is only 1.4 . Use 0 to choose from rotamer library. If non-zero, foldtraj will try to use this value for the dihedral angle when building this residue, provided steric clashes do not forbid it.

Discretization Size

Size of trajectory distribution in both dimensions (normally 400). Each residue can potentially have different resolutions of discretization, though this is presently unimplemented (i.e. they must all have the same value here). See the global property of the same name for more details.

First Row

First row in the trajectory distribution array which contains a non-zero number. The absolute first row is numbered 1. Leading rows containing only zeroes are not stored to save a bit of space.

Number of Rows

The number of rows from the first non-zero row to the last non-zero one. Rows of zeroes may appear amongst the non-zero ones, but this counts up to the last row with any non-zero data in it. Again, a bit of space is saved by not storing trailing rows of zeroes.

# Elements <= 0% peak

The number of entries in the entire trajectory distribution matrix which are zero. This, together with the next three values, gives an indication of the shallowness of the conformational space at the residue in question. A sharp peak will have many zeroes while a broad, uncertain region will have much fewer.

# Elements <= 5% peak

The number of entries in the entire trajectory distribution matrix which are less than or equal to 5% of the peak value. Again, sharp peaks will have a small value here, broad uncertain conformational spaces will result in a larger value.

# Elements <= 10% peak

The number of entries in the entire trajectory distribution matrix which are less than or equal to 10% of the peak value.

# Elements <= 15% peak

The number of entries in the entire trajectory distribution matrix which are less than or equal to 15% of the peak value.

Tries per Residue

Number of times to randomly choose a conformation from the trajectory distribution and attempt to generate a complete residue backbone and sidechain with no steric collisions, before backtracking occurs.

Markov Scale Factor

When non-zero, this adds a chance of rejecting a chosen conformation based on the conformation of the previous residue. See the global property of the same name for more details. Because each residue can have its own value for this scale factor, some residues may be given this bias and others not if desired. This is not used in Ramachandran space.

Global Properties

Trajectory Space

Used to specify which set of trajectory distributions to use for its dictionaries and tells the program what type of data is stored in the dictionary -- Ramachandran data or a-carbon backbone data (see What is VisTraj? for details). Foldtraj builds the actual protein up differently for each selection.
Value is either *RAMACHANDRAN* or *ALPHA-CARBON*.

Backbone Error Tolerance

A measure of the maximum mean squared error (in degrees squared) permitted in the placement of the backbone C and N atoms without rejection. If no solution exists with an error less than this value, the program will choose a new location for the current Ca. The error is calculated as a sum of various parameterized equations under certain mathematical constraints. A high value here will generate the protein very quickly, but with possibly poor bond length and bond angle variability, and with points all over Ramachandran space. A low value will result in a highly accurate backbone, confined to the valid regions of Ramachandran space and with correct bond lengths and angles, however the result will be generated much slower, probably with a lot more backtracking occurring. A value of 50 provides an acceptable backbone with a reasonable rate of roughly 1-2 seconds/residue to generate the whole protein.
Default is *60*.
Minimum allowed is *10.0*.
For Ramachandran space, the generated backbone is always "perfect", however this value then still affects the quality of proline rings built.

Backbone Precision

Determines the degree of precision various "virtual" and real backbone angles are determined with, in degrees. Unless you need higher precision than 0.05 , don't change this number as making it larger won't significantly speed up the program and making it smaller is pointless as any errors incurred less than 0.05 should have little effect on the overall resulting structure.
Valid range is *[0.0005, 5.0]*.
For Ramachandran space, this only affects precision in Proline ring angles.

Rotamer Tries

Affects the number of attempts to place a sidechain rotamer before aborting and backtracking. Normally, this only fails if a sidechain-backbone or sidechain-sidechain collision occurs, which tend to occur more frequently than backbone-backbone collisions due to the larger number of atoms involved. We have found that speed is improved by trying several rotamers at a spot before backtracking. Since some sidechains are more complex than others, this number is multiplied by the number of c angles a given residue has to determine the number of actual tries, so with the default of 7, big residues like Arg and Lys will have 28 attempts made, while smaller ones like Ser will have only 7 tries, and Ala and Gly will always have just one attempt made.
Valid range is *[1, 100]*.

Backbone Atom Bounciness

Used for collisions between backbone atoms (defined as N, H, C, O, Ca, Ha, Cb and any hydrogens on the Cb). It determines whether atoms are modelled as soft or hard spheres. A value of zero corresponds to the hard sphere model. When this has a value between zero and one, then whenever a collision occurs, the percent by which the atoms are too close is calculated. For example, if the van der Waals radii indicate two atoms should be 3.0 A apart, and they are found to be 2.7 A apart, then they are 10% too close. Then a random number between zero and Bounciness*100% is chosen. If this number exceeds the closeness (10% in this case) then the collision is "allowed" as a soft-sphere collision. This acts like a linear van der Waals energy cutoff, so with the default of 0.25, no atoms are ever allowed closer that 75% of the sum of their radii, with a linearly increasing acceptance probability from zero to one for distances of 75% to 100% of their normal minimum separation. The default value works well, producing relatively few collisions and tighter structures than those obtained with hard atoms. NOTE: if reconstructing backbones from a-carbons, you may wish to try a high setting (e.g. 25000.0) to turn off all collision testing.
Values must be *non-negative*.

Sidechain Atom Bounciness

Used for collisions involving at least one sidechain atom. Since placement of sidechains is given much less freedom by the program compared to the backbone atoms, you may wish to set this value higher to compensate. It determines whether atoms are modelled as soft or hard spheres, just as described for backbone atoms above.
Values must be *non-negative*.

Bumpcheck Hydrogen

If you do not want collisions involving at least one hydrogen atom to "count" then change this to FALSE, otherwise set it to TRUE. Turning this off will result in less collisions and thus faster

structure generation. When two heavy atoms are close and can form a hydrogen bond, this setting has no bearing on the outcome, it only determines whether backtracking will occur when a given hydrogen atom experiences a van der Waals collision.
Valid values: *TRUE, FALSE*

Increment Size

You only need to change this if you get a lot of "Avoiding endless loop" error messages in the foldtraj error log, in which case fiddling with this value may help. Changing this and/or Backbone Start should not affect the resulting structures in any way whatsoever, they only affect the speed with which the backbone is reconstructed between a-carbons.
Default is *20*.
Valid range is *[1.0, 89.0]*.
Not used for Ramachandran space.

Backbone Start

Do not change Backbone Start unless you get a lot of "Avoiding endless loop" errors, in which case changing this number by small amounts may help (its value is in degrees, default 250).
Valid range is *(0.0, 360.0]*.
Not used for Ramachandran space.

Mean Tries per Residue

The number of tries Foldtraj will make at placing the next residue before giving up and backtracking one residue. A high value may slow down the program but generally results in less backtracking, while a low value will result in much backtracking but at a faster rate. The default is 100 which provides a good compromise. This is only an approximate value and will automatically be decreased by the program in areas where less conformational freedom exists, for example. Think of this setting as the average default value to use - don't change unless you have a good reason to.
Valid range is *[4, 250]*.

Markov Scale Factor

The Markov scale factor affects how Markovian information will be incorporated into the random walk. For now, this is only used for walks in a-carbon space and is ignored for Ramachandran walks. The default of zero means no Markovian information is used. Other values in *(0.0, 1.0]* cause the algorithm to bias the choice of trajectory based on the trajectory chosen at the previous residue. Trajectories are more likely to remain similar to that of the previous residue, so for example, while in the middle of a helix, there is a lower probability than normal of suddenly changing to a sheet, which is located far away from helix in trajectory space. A value of 0.0 corresponds to no Markovian bias while values closer to 1.0 indicate a large bias and tend to create very large amounts of secondary structural elements. This value must be set with caution, or undesirably long helices or extended regions may result.

Distribution Units

Describes the meaning, if any, of the values stored in the trajectory distribution. This is informational only, as all trajectory distributions are normalized to a volume of one before being used as a probability distribution anyhow. Nevertheless, this can give some insight into the source of the raw numbers in the matrices and what their meanings are, if any.

Constraint File

Indicates the file containing geometric constraints that was used, if any, at the time of the trajectory distribution creation. Constraints are stored in the trajectory .trj file itself, so the file is no longer needed and this is merely for your convenience.

Trajectory Distribution Creation Method

Indicates the method used to generate the trajectory distribution. Again, this is only informational. See What is VisTraj? section for more details. The possible values and their meanings are:

N\A - unknown, possibly taken from backbone of known structure.

UNIFORM - all areas of space were assigned equal probability.

STANDARD - the standard amino acid dictionaries were used to obtain initial trajectory distributions. In this case the "dictionary" distributions for helix, sheet and coil are added back together, so as not to distinguish between them. Thus for a given Asp residue, the conformational space used corresponds to the observed frequencies of Asp in the PDB, regardless of secondary structure.

ONE STATE SECONDARY STRUCTURE - a one-state secondary structure prediction was made for the sequence, and then the standard dictionary trajectory distribution for the corresponding amino acid and secondary structure at each residue was used.

THREE STATE SECONDARY STRUCTURE - a three-state secondary structure prediction was made for the sequence, and then the three standard dictionary trajectory distributions for a given residue in the sequence were added together in proportion to the three probabilities assigned by the secondary structure prediction. For example, suppose an Ala residue was predicted as 75% helix, 5% sheet, 20% coil by some method. Then the three dictionary distributions for Ala are first normalized to have the same integral, then we take 0.75*(Ala Helix distribution) + 0.05*(Ala Sheet) + 0.20*(Ala Coil) and use this as the trajectory distribution for this residue. In this way, conformational space is biased by the secondary structure prediction.

Discretization Size

The trajectory distributions are discretized probability density functions in a-carbon or Ramachandran space, and this indicates the number of divisions in each dimension. The overall range for cos f is -1 to 1 and for q is -180 to 180 , while for Ramachandran space, both dimensions range from -180 to 180 . The default choice of 400 allows for very accurate reconstruction of arbitrary protein backbones and so is the reccommended choice. If a different resolution is desired, the standard "dictionary" trajectory distributions (see What is VisTraj?) must be regenerated (presently not supported). This setting is only really used when generating the trajectory distribution dictionary - after this, it is not used but stored in the trajectory files/ dictionaries themselves so changing it will have no effect on the results once the initial dictionary

is created. Thus for now you are limited to 400x400 trajectory distributions, which were included with this program, and you cannot change this. Note also that in a-carbon space, the use of cos f and q as the two independent axes ensures that each element in the trajectory distribution represents a patch on the unit sphere around the atom of equal area (namely 4p/size^2).
Value must be *100 or greater.*

Creating New Trajectory Distributions

NOTE: Please read the entire "What is VisTraj?" section before attempting to read this section.

Trajectory distributions can be created by choosing New from the File Menu. This will initiate the New Trajectory Distribution Wizard which will walk you through the creation of your file in four simple steps:

Step 1 - Choose the source

You have three options here. If you know just the sequence of your protein of interest, choose input type Sequence and specify whether the sequence is stored in a file (in plain text or FASTA format) or whether you will enter it manually. If you wish to create a trajectory distribution using the backbone conformations from a known structure, select MMDB structure as your input type. Either way, hit Next to proceed to step two, or Cancel to abort the whole procedure.

Step 2 - Specify the protein

This step differs depending on what you chose in step one:

a. If you chose the default in step one, to enter the protein sequence manually, you may now enter it (or cut and paste from a web browser, for example). The sequence is not case sensitive, and non-standard amino acids can be inserted using the Insert A.A. button (or typing their codes directly if you know them). Note that some non-standard amino acids may only be placed at the N- or C-terminus, but you do not have to worry about this, since you the program will ensure everything is in order before letting you proceed. The length of the sequence entered so far is also shown for your convenience. Your sequence must be at least three residues long. Hit Next when you're ready to continue, or Back or Cancel if you change your mind.

b. If you chose a sequence file as your input type, you will be asked now enter the file name (or Browse for it). When you find it, press Next to continue, or Back or Cancel if you change your mind. "seq.in" is the default filename. Your sequence must contain at least three residues.

c. If you chose MMDB Structure as your input type, now browse for the structure file (usually with a .val extension) or type in its name with full path. Note that this format has changed several times so if you have problems, be sure to get a recent structure file off the NCBI site `http://www.ncbi.nlm.nih.gov/Structure/` and save the file in Cn3D (Asn.1) format. VisTraj should be able to read both the older-style structure file, an Asn.1 Biostruc, and the newer-style, a "MIME-wrapped" Biostruc-seq. A Biostruc contains a complete description of the chemical graphs and co-ordinates of one or more biomolecular structures, similar to a PDB file but with bonds explicitly represented. A Biostruc-seq in addition contains a Bioseq, an Asn.1 description of the amino acid sequence of the protein chain(s) in the corresponding Biostruc. The MIME-wrapping eases integration with web browsers and allows dragging and dropping to open files. As both file types have the .val extension, it is not trivial to distinguish the two file types, though normally you will obtain Biostruc-seqs if downloaded directly from NCBI. For the purposes of VisTraj the two are equivalent, so do not worry too much about which type you have.

Pick Next to continue or Back or Cancel to change your mind.

Step 3 - Specify Trajectory Distribution Parameters

This step again differs depending on what you chose in step one:

a. If you have entered the sequence manually or from a file (steps 2 a) or b)) you will be presented with several options on how to generate your trajectory distribution.

   Data Compression Type affects the file size and loading speed. Do not adjust this unless you really know what you are doing. Picking BZip here reduces the file size but slows down loading, sometimes by a LOT. Picking None increases the file size (a LOT) and loads the fastest. The default of RLE provides a good comprimise of file size and loading speed.

   Trajectory Distribution Creation Method affects where the initial trajectory distributions for each residue come from (see What is VisTraj? section for more info).

   - **Uniform** distributions are just that - a value of one is assigned to every bin in the matrix at every residue.
   - **Standard** distributions use the dictionary of trajectory distributions based on the PDB and amino acid type to make the inital trajectory distribution at each residue. Thus the distribution at each residue depends only on residue type and there are only twenty possibilities.
   - **One State Secondary Structure** distributions are generated much like standard ones except that instead of choosing one of twenty possible distributions, they choose from one of sixty based on amino acid type and secondary structure prediction (helix, sheet and coil). By default the GOR secondary structure prediction method is used though any prediction may be supplied (see Secondary Structure Prediction File below).
   - **Three State Secondary Structure** distributions are generated by adding together the three dictionary distributions for a given amino acid type in the proportions give by a three-state secondary structure prediction. For example, if an Ala residue was predicted 75% helix, 20% sheet, 5% coil, we would add 0.75 * (Ala Helix) + 0.20 * (Ala Sheet) + 0.05 * (Ala Coil) to give our trajectory distribution for that residue, where Ala Helix, Ala Sheet and Ala Coil are our three dictionary distributions for Ala and have all been normalized to the same integral. Thus unlike the other methods, there are an infinite number of possible resulting distributions. Again the

GOR method is used by default though any secondary structure prediction may be supplied by the user.

Call RPSBlast determines whether Reverse PSIBlast (RPSBlast) will be called in order to identify fragments or domains of the sequence of interest which have similarity to domains of known structure in the PFAM or SMART databases. When a hit is found, the sequence is added to the pre-calculated multiple alignment, and the backbone trajectory, and in some cases sidechain dihedrals, in the aligned region will be taken from the known structures, with some distance contraints added near insertions or deletions. Note that this feature requires that you have the entire MMDB structural database as well as the CDD database locally. These are available for download on our ftp site, `ftp://ftp.mshri.on.ca/pub/TraDES/rps`, along with a document describing how to set this up to work.

Constraint File is used to specify a text file containing distance constraints to apply during the buildup procedure of foldtraj. The format of the file is similar to that used by X-PLOR, and a sample one, sample.constr, is included in this package for your convenience. More about distance constraints can be found in its own section of this manual.

Secondary Structure Prediction File is normally used to specify a filename in which to save the GOR secondary structure prediction for the sequence of interest. Note that this options is not applicable for the Uniform or Standard creation methods. The default is "ssout" in the current directory so be sure to change this if you do not want to overwrite and earlier prediction. Alternatively, by selecting Read rather that Write here, the file specified will be used as input rather than output, allowing the user to supply an arbitrary secondary structure prediction instead of using the GOR. The format of the supplied prediction file is simple and illustrated with sample.ss included in the downloadable package. The short C program "phd2ss.c", also included, shows how the output from an arbitrary secondary structure prediction algorithm (in this case PHD) can be parsed and output in the correct format for VisTraj. Feel free to modify and use phd2ss.c to suit your needs (but please do not redistribute it in altered format without the authors' permission).

Output File is where you should enter the name of the trajectory distribution file you wish to create. The .trj extension will automatically be appended to the name so you need not do this yourself. If no name is given, the default of "protein.trj" will be used - overwriting any existing file of the same name.

Click Next to proceed to the final step, or Back or Cancel if you have second thoughts.

b. If you used an MMDB structure as your input source (step 2 c)), you will receive a different set of options:

Chain Name can be used to specify which molecule in the MMDB file you selected you wish to make the trajectory distribution file for (since a trajectory distribution file corresponds to one molecule or 'chain' and not necessarily a whole structure). If left blank, the first chain of the structure will be used (which is often what you want).

Model Number - some structure files contain multiple models, usually when solved by NMR, each with slightly different co-ordinates. Here you may specify which model to get the backbone conformations from. Usually model 1 is the "best" to use, so do not change this unless you are sure you need to.

Call RPSBlast is presently ignored when using a structure as the trajectory distribution source.

Residue Start # and Finish # allow you to specify precisely which part of a protein chain you want to make a corresponding trajectory distribution for. This is useful, for example, in a large multi-domain protein chain where you may only want to model one domain. Note that leaving Finish # set to zero tells VisTraj to go from the start residue to the end of the chain, without you needing to know exactly how long the sequence actually is.

Standard Deviations for x dir. and for y dir. determine the exact shape of the peaks in trajectory space placed at each residue. When both are zero, a single sharp peak (d function) is placed in trajectory at each residue, corresponding to the conformation of that residue in the known MMDB structure. When non-zero, the sharp peak is replaced by a smooth Gaussian function with the given standard deviations in the x and y directions (in degrees). Note that for Ramachandran space, 'x' is F and 'y' is Y while for a-carbon space, 'x' is q, 'y' is f.

Gaussian Peak Height is simply the raw number that will be the highest point on each trajectory distribution. Since the units are relative and arbitrary, the default of 100 should normally not be changed. Making this too big (> 10000) could cause overflows.

Save Sidechain Chi Angles is false by default. When set to true, VisTraj will record not only the backbone conformation of the MMDB structure you gave it, but also the sidechain dihedral (c) angles at each residue in the structure (these can be viewed in Residue Properties, see that manual section for more info).

Data Compression Type behaves as described in step 3 a).

Time Step and Temperature provide an alternate way of specifying the standard deviations of the Gaussians placed in trajectory space. If you fill in values here, the standard deviations will be computed from these. While far from a true molecular dynamics simualtion, the standard deviations will be computed to simulate movement of the backbone (due to the uncertainty caused by the Gaussians) at the stated temperature (in K) for the stated time (in fs). Typically 50 fs. at 273K works well. This method is still being tested, and further details on how it works will be published in an upcoming paper.

Below this is where you select whether you want to specify the standard deviation explicitly or give the temperature and time step instead.

Constraint File is the same as for step 3 a).

Output File works the same as in step 3 a) except that if no name is specified, the default will be the name of the MMDB structure being used, but with its .val extension replace with .trj.

Click Next to proceed to the final step, or Back or Cancel if you have second thoughts.

Step 4 - Specify Global Parameters

Here you can specify the general parameters which affect the making of the trajectory distribution, and how the conformers will be built up by foldtraj. Click on a parameter to see a description of it in the window below and its value in the box in the middle. To change a value simply select a new one or type in a new value. If an invalid value is entered it will not be accepted. Hit Apply after making your changes to save them. Note these values are actually stored in your .foldtrajrc (for UNIX people) or foldtraj.ini (for Windows folk, in your C:\WINDOWS or equivalent directory) file, and may be altered there directly with a text editor if you prefer. The parameters are also described in the Global Properties section of this help file. To make any changes, click Back or Cancel, otherwise hit Begin to start making your trajectory distribution file.

Step 5 - Wait

Depending on your system, a progress bar will pop up as the trajectory distribution file is built, or a simple message will appear. When it is done, your file will automatically be loaded into VisTraj and be ready to manipulate. If you didn't specify a name for your file and it got the default name of 'protein', you will be warned to rename it to be sure it does not get overwritten the next time you do that. If any errors occur, you probably entered an invalid parameter so try again and check the settings you give carefully. Do not forget to check the secondary structure prediction output file ('ssout' by default) if applicable and if you are interested in this.

Main View Window

Aside from the menus, covered in the following sections, there are several other ways to manipulate a loaded trajectory distribution.

Left and Right Arrows

The small triangular arrows just below the File menu allow you to navigate from one residue to the next. Clicking the left arrow will take you to the trajectory distribution for the previous residue, while clicking the right arrow takes you to the next residue. The left and right arrow keys on your keyboard will accomplish the same tasks.

Residue Hopping

To the right of the navigation arrows, the current residue being viewed is indicated, below this is a "Jump to Residue" box. Enter any valid residue number and click Go to be immediately transported to the desired residue without having to pass through the intervening residues.

Sequence

The amino acid sequence of the loaded protein is shown in the upper middle area below the menus. The residue currently being viewed is highlighted in red. Note that an X here corresponds to a post- translationally modified amino acid. When you are viewing such a residue (i.e. it is selected in red) then the code for the modified amino acid will be shown just below the sequence. If you do not know what the code means, please see the "Modified Amino Acids Supported" section of this manual.

Main Display Area

In the main display area, you will see the trajectory distribution for the residue indicated in red in the sequence window. For an explanation of the colouring scheme, please see the Visible Height section of this help file under "Draw Menu".

Rotation

Click your mouse in the main display window and drag to rotate the trajectory distribution in 3-D space. This is generally loads of fun. In case you care, moving the mouse horizontally rotates the distribution about the north pole (for spherical view) or an imaginary axis perpendicular to the plane (planar view). Moving the mouse vertically rotates about an imaginary horizontal line cutting the display window into two equal parts. For more precise rotation, you may also use the scrollbars located on the right and bottom of the main display window. Move the scrollbars by dragging with the mouse or clicking on the arrows at either end. The bottom scrollbar is equivalent to horizontal mouse movement while the right scrollbar is equivalent to vertical mouse movement.

Zoom

You may zoom in or out on the trajectory distribution by holding down the Ctrl key and dragging the mouse in the display window to the left (to zoom in) or to the right (to zoom out). Tip: zoom in really close to see the fine detail stored in the distributions.

Maximize, Minimize, Close and Resize

Depending on your windowing system, you may have up to three buttons in the upper right corner to minimize, maximize and close the program. These all work as expected. Clicking the X to close is equivalent to picking Exit from the File Menu (see File Menu below). You may also resize the window if your windowing system allows it, and the trajectory distribution will grow or shrink accordingly.

File Menu

New

This option is covered extensively under "Creating New Trajectory Distributions".

Open

Use this to load up a trajectory distribution file for your perusal. If you have modified a currently loaded file, you will be given a chance to save your changes before closing the current file or to change your mind and not open a new file. Otherwise, browse to the file and it will load up. Note that trajectory distributions normally should end in the .trj extension. However, sometimes the final .trj files may be bzipped to save space and make transport quicker, so if you see a file ending in .trj.bz2, this should work fine with VisTraj and can be directly loaded. The file will automatically be uncompressed and then loaded if this is the case (but it will not, however, be recompressed when you close it, you must do this yourself if desired). Also note that if, for example, you have two files named 'protein.trj' and 'protein.trj.bz2' in the save directory, then regardless of which you choose, 'protein.trj.bz2' will be decompressed, overwriting 'protein.trj', and then opened, so do be careful in how you name your files.

Save

Saves to disk any unsaved changes to the trajectory distribution file currently being edited.

Save As

Works like save but allows you to enter a new name and/or path for your file. If the given name does not already end in .trj, this extension will automatically be added for you. Be careful, for if a file by that name already exists, it will be overwritten without notice.

Close

Releases the current file from memory so that you can make or load a new one. If you have made changes which have not been saved yet, you will be asked if you wish to save them before closing the file. If you say no, all changes made since you last saved will be discarded.

Generate Conformer

Selecting an option from this sub-menu invokes Foldtraj to generate a probabilistic conformer of the protein represented by the currently loaded trajectory distribution. Any changes made to the currently viewed residue will be saved by this action. This allows you to make changes to trajectory distributions or their parameters and observe the effects on the resultant structures. Even unsaved changes you may have made to any Global Properties will be used for the run. Simply choose whether you want a PDB (Rasmol) or MMDB (Cn3D) file and the rest is automatic. The actual file will be stored as "last_generated.pdb" or "last_generated.val" in the VisTraj directory in case you want to view it later. The first time you use this command, you will get a brief message indicating that the Rotamer Library is being loaded. After this, it stays in memory until you exit VisTraj to save time when making the structures. Then, depending on your machine, you will either get a static message or a progress bar as your protein folds. The progress bar can go both forwards AND backwards since the algorithm backtracks when collisions occur. Hit cancel at any time, if you get impatient, to abort (not available on all platforms). Normally, expect it to take about 5-20 seconds for a 150 residue protein, but it could take much longer depending on your parameters and trajectory distributions. After a few minutes with no success, it will bail out anyways and give you a partial structure, as far as it got (anything's better than nothing). Once finished, the structure viewer program you selected will launch auto-magically, with your structure loaded. Note that you must have either Rasmol or Cn3D on your computer in order for them to open up with your structure loaded. On UNIX machines, the programs must be in your PATH for VisTraj to find them, and must be called "rasmol" or "Cn3D". For Windows users, you must have Rasmol properly associated with .PDB files and/or Cn3D associated with .VAL files. This is usually done for you when you install the programs but if you should need to do it manually, see you Windows documentation for details. If you wish to keep a generated structure, it is recommended that you use the "Save As.." feature of Cn3D or Rasmol, otherwise the file will be overwritten the next time you use this menu option. Lastly, it is important to note that this runs exactly the same code as running Foldtraj on the command-line, except with a lot less (in fact no..) flexibility. Lastly, note that in order to allow Cn3D to display proteins containing non-standard amino acids, you must overwrite its *bstdt.val* (in the *Data* subdirectory where Cn3D is installed to) with the newer one which comes with VisTraj.

Structure Info

Displays information about the protein sequence stored in the trajectory distribution file. This is often taken from the PDB header information if an MMDB structure was used to make the trajectory distribution file, if applicable. Click on Close when you are done viewing this information.

Exit

Exits the program. If a file is loaded and has unsaved changes, you will be given a chance to save them before exiting. If you answer no, all changes will be discarded and the program will terminate.

Edit Menu

Global Properties

This brings up a dialog box to edit the global properties of the current trajectory distribution. Note that this may not necessarily match what is stored in your .foldtrajrc (UNIX) or foldtraj.ini (Windows) configuration file. This file is read only when making new distributions, and once created, the parameters are stored within the .trj file itself, allowing each file to have its own associated parameter set. Click on a setting to get a description in the large window at the bottom (or see Global Properties section of this file) and the current value is displayed in the middle. If it is not greyed out, you may change its value by typing or selecting a new one. In most cases the range of valid values is given in the description. You will be told if you try to assign an invalid value. Only change these if you understand what they do. Hit Apply to enact the changes, followed by Close when you are done.

Residue Properties

This brings up a dialog box allowing you to modify the properties of the current residue. As for the global properties, a description of each appears at the bottom as you select the various settings (or see the Residue Properties section of this manual) and its value appears in the middle. If it is not greyed out you may change it by selecting or entering a new value. Do not change these unless you understand what they do. You will be told if you try to set something to an invalid value. Hit Apply to enact the changes, followed by Close when you are done. Note that there is a special button "Mutate Residue" for changing the residue. This opens up the Amino Acid Selection Window (see Mutate Residue below for more details).

Distance Constraints

Opens the Distance Constraint Editor. In the left pane, distance constraints, if any, will be listed. This summary window indicates the numbers of the two residues and their respective atoms that are involved in the constraint. Click on one to select it and get a more detailed view in the right pane, which will show the residue numbers and atom names again, as well as their mean distance, minimum and maximum deviation from this mean, two angles, three dihedrals and the probability associated with the constraint. See "What are Distance Constraints?" for details on what these values mean. Along the bottom are four options: Edit, Add, Delete and Close. Hitting Edit will open up another window allowing you to modify the values for the distance constraint. Simply click in a field and type a new value to change it. If you enter an invalid value in a field, you will be told when you hit Apply. Distances are in Å and all angles are in degrees. Hit Apply when you are done to save your changes, or Cancel to exit without saving your changes. Note that if you wish to change the residue numbers, you must delete and then add a new constraint. The Add function works just like the Edit function except that you can enter values in all fields now, and they start out empty. Note leaving the default value of 0.0 for probability will assign an actual probability of 95% to your constraint. This may seem a little confusing but if this default ever changes, 0.0 will always result in the default being assigned. When done click Add to add it or Cancel if you change your mind. If you added it, your new constraint will now show up in the left pane, sorted numerically by first and then second residue. To Delete a constraint, select it in the left pane and click Delete. You will be asked to confirm and if you say Yes, the constraint will be permanently removed. Finally, click Close when you are done editing your constraints.

Residue Fragments

Opens the Residue Fragment Editor. In the left pane is the list of fragments attached at the current residue (see "What are fragments?" for more details). Fragments are listed as the range of residues which they span. All fragments will start at the current residue of course. If you click on a fragment to select it, the probability associated with it is shown on the right. Along the bottom are four options: Edit, Add, Delete and Close. Close will close the window when you are done editing fragments. Delete allows you to erase unwanted fragments. You will be asked to confirm before the fragment is actually deleted. Clicking Add will bring up the Add New Residue Fragment window, in which you can fill in information about the first residue in the fragment. All newly created fragments are one residue long, and they can be extended later by Editing them. Fill in the dihedral angles and the standard deviations you want for the first residue of the fragment, and put 0 for sidechain dihedrals if you wish them to be chosen probabilistically from the backbone-dependent rotamer library. You also specify the probability associated with the new fragment here. When you are done click on Add, or if you change your mind choose Cancel. If any information you enter is invalid, you will get a message to this effect when you try to Add it, and be given a chance to correct it. Lastly, choosing the Edit option from the main fragment window brings up the Edit Fragment window. In the left pane are listed the residue numbers, one by one, comprising the fragment. Click on one to select it, and its properties will appear to the right. This includes the residue number, amino acid, and all the dihedral angles and their standard deviations. You will also see the number of tries that will be made before backtracking at that residue (a function of the backbone dihedral standard deviations that you choose) and p(SS) which is for future use in relation to disulphide bridges. Along the bottom you again have four options: Edit, Append, Delete and Close. Close will bring you back to the fragment list window where you may edit a different fragment. Edit allows you to change the dihedrals for the currently selected residue in the fragment. Make your changes and hit Apply to save them, or Cancel if you change your mind. Append allows you to extend the fragment by one residue. Enter the dihedrals for the new residue and hit Add, or choose Cancel if you change your mind. Note you can see but not change the probability at this point, as the probability must be constant throughout the entire fragment. You may delete the last residue in a fragment with the Delete button, and after confirming your choice the fragment will be one residue shorter. If you have only one residue left in the fragment, this will remove the entire fragment. Note that if you wish to delete a residue in the middle of the fragment, you must delete all residues following it first, and then add back what you want.

Add\Replace Noise

This function allows you to modify the data stored in the currently viewed trajectory distribution. In the dialog that opens you have several options. Most important is on the right, where you must choose either Add or Replace. If Add is selected (the default), then whatever type of "noise" you choose will be added or "superimposed" to the current data in the trajectory distribution. On the other hand if you choose Replace, the current data will be overwritten for that residue. Next choose the type of "noise" you wish to add on the left. Four of the options correspond to the four trajectory creation methods and the fifth lets you add single peaks. We shall discuss each briefly here. see "Trajectory Creation Method" under "Global Properties" for more details. Uniform noise adds/replaces with a flat distribution with a value of 1 across all of trajectory space. Standard noise uses one of the twenty amino acid-based dictionary distributions used for creating new trajectory distributions. Which of the twenty is chosen depends on the amino acid, or for modified amino acids, on its unmodified parent. One State Secondary Structure noise works like Standard except one of the sixty dictionary distributions is used based on amino acid and secondary structure, the latter of which you choose from a drop down box. For example, if you know a particular residue is in a helix, you might want to Replace with a One State of Helix. Three State Secondary Structure allows you to enter percentages for helix, sheet and coil (be sure these add up to 100 or you will be scolded). The three dictionary distributions for the particular current amino acid will then be added in these proportions and used as the "noise". Finally, if you choose Single Peak, you may specify the exact trajectory space co-ordinates and magnitude to place a single peak into conformational space. When you have selected exactly what you want to do, hit OK to make the change or Cancel to abort.

Insert Residue

Brings up the Amino Acid Selection dialog, allowing you to insert new residues into the trajectory distribution. Simply select the residue you wish to insert and hit the Insert button, or Cancel if you change your mind. The residue will be inserted before the current residue (so you cannot add to the very end... sorry) and filled with the standard dictionary trajectory distribution for that amino acid (or for modified amino acids, its unmodified parent). You can always change this afterwards by using Add\Replace Noise (see above). Note that certain modified amino acids may only be placed at the N- or C- terminus while others can be placed anywhere; the list only shows amino acids you may insert at your current location.

Remove Residue

Use this to completely erase a residue from the sequence and the trajectory distribution database. You will be asked to confirm so pick No if you are not sure, otherwise choose Yes and it will be removed. You must have at least three residues in your protein so you will not be able to delete residues if you reach this point.

Mutate Residue

This opens the Amino Acid Selection dialog and allows you to choose a new amino acid for the current residue. Only amino acids which may be placed at your current location will be shown in the list, so choose one and then hit the Mutate button, or hit Cancel to Abort the change. Once you mutate a residue the display in the sequence window will change accordingly, however nothing else will change. If you were now to Add\Replace Noise with Standard noise though, for example, the trajectory distribution would be filled with the dictionary distribution for the new mutated residue. This is not done automatically because usually you will want the original preserved. This can be useful for homology modelling, starting with the trajectory distribution for a known structure and then mutating its residues to your target sequence, but keeping the conformation the same.

Filter Menu

This menu allows you to apply digital smoothing filters to your trajectory distributions. This helps to compensate for the slight inaccuracies in sampling caused by the discretization of conformational space into 400 x 400 bins. It also makes them look better for screenshots . Note you may filter the same distribution repeatedly, with the same or different filters, and the results will be cumulative.

Smooth

A smooth filter evenly spreads out all peaks of the distribution evenly and is usually the best one to use. Only try the others if this does not give you the desired effect.

Gaussian

For the Gaussian filter you must supply the magnitude and standard deviation of the kernel function. The default should usually work best but change them if you wish to see the difference. Magnitude has very little effect on the result, with a bigger value making the filter slightly more precise. Choosing a large standard deviation will perform greater smoothing.

Low Pass

The Low Pass filter works similar to the Gaussian, also requiring a magnitude and standard deviation for the kernel. Again, magnitude only affects the precision of the filter while standard deviation affects its shape. The low pass filter should, in theory, produce the best looking results of the filters offered, but practically often is very similar to the other ones.

Revert to Original

As soon as you change residues (or save), all filtering will become permanent. Until this point however, you may undo all filters applied to the current residue by choosing this menu option.

View Menu

Sphere

Available only for a-carbon trajectory space, this plots the trajectory distribution on the surface of a sphere. An arrow indicates the North Pole (f = 0) and an arc indicates the "prime meridian" where q = 0 . Using this view, imagine that Ca_i is at the centre of the sphere, Ca_{i-1} is at the south pole and Ca_{i-2} is outside the sphere along the prime meridian. Then the height at any point on the sphere is proportional to the probability that Ca_{i+1} is located at that point. Hence the fourth a-carbon is specified easily and uniquely by its position relative to the previous three backbone a-carbons.

Plane

For Ramachandran trajectory space, this displays the standard Ramachandran map, with the height at any point giving the probability that the residue will be found in the corresponding backbone conformation. For a-carbon space, this shows the same information as the spherical view but in "unrolled map" form, much like in an atlas of the Earth. The sphere is "projected" onto a surrounding cylinder which is subsequently cut and unrolled to produce the flat representation. See Discretization Size under Global Properties for more information on the choice of units for the unrolled map.

Trans

Normally you will be viewing the trans trajectory distribution for each residue. Use this option to switch back to it if you were viewing the cis one.

Cis

Proline residues have a separate trajectory distribution stored for cis residues. Whenever a proline residue is placed during the random walk of foldtraj, the conformation is chosen from either the cis or the trans trajectory distribution depending on whether the proline itself is placed cis or trans. Use this option to view the cis trajectory distribution if one exists for the current residue. It may be modified and viewed just like a trans distribution. Note that filters affect both cis and trans distributions when present, while adding or replacing with noise affect only the currently selected one. Mutating a proline residue will permanently destroy its cis trajectory distribution, while mutating to or adding a proline will use the dictionary cis proline distribution by default.

Draw Menu

Rendering

Choose between solid and wireframe rendering for the main view window. Wireframe may be slightly faster and/or look cooler.

Shading

Turn lighting on for more realistic 3-D images with this menu option. By default shading is off.

Visible Height

If you wish to see only a portion of the distribution, use this option to limit which points are drawn. Drawing less peaks will also speed up the rendering. Points are normally color-coded with those having a height of at least half the peak value being coloured red. Those with heights less that a quarter of the peak value are green. Points between one quarter and one half the peak value are yellow. Thus selecting, for example, 25%-50% on this menu will turn off all but the yellow points in the distribution. Perhaps more useful would be picking 50%-100% to highlight the peaks. This option is probably most useful when making screenshots.

Camera Position

This item allows you to choose between several predetermined camera locations to give you a good top or side view of your distribution. This is handy if you want to take several screenshots of different residues and want to ensure all are in precisely the same orientation, for example.

Quick Rotation

This is off by default. Turn quick rotation on for slower machines, and nothing will be redrawn as you rotate or zoom with your mouse until the mouse button is released, allowing for much faster movement. Quick rotation does not affect rotation via the scrollbars.

Background Color

Choose between a black or white background. Axes labels in the planar view will be coloured appropriately as well.

Help Menu

Contents

Opens up the help viewer (what you are presumably using right now!). Note that although the help window may be open when other sub-windows are open, you must close all other windows except the main one in order to navigate the help file.

About

Displays names and contact information of the authors of VisTraj.

TRADES Website

Opens a Netscape (or Internet Explorer) browser window and points it to the official TRADES homepage (a browser must be installed and in your path or not much will happen).

What are Distance Constraints?

Summary

Distance constraints are a way to tell foldtraj that two pieces of a protein are close together spatially, though they may be far apart in sequence. Possible sources of such information are from an NMR experiment (NOEs), fluoresence resonance energy transfer (FRET), disulphide bridges, and so on. Foldtraj will attempt to satisfy all the constraints you give it in the structures that it generates from your trajectory distribution, but after a certain number of failed attempts it will eventually give up and violate a constraint. It will try to satisfy it again the next time it builds a structure (hence you should tell foldtraj to build many structures and hope that at least some satisfy all your constraints). Constraints can be added to a trajectory distribution either one at a time through the Edit -> Distance Constraints menu option, or by specifying a constraint file when making a new trajectory distribution (see the appropriate manual sections for details).

Details

A proper constraint between two atoms requires six degrees of freedom to uniquely specify. First however, the atoms themselves must be specified. The residue numbers and atom names for each of the two atoms must be given. For obvious reasons, the atoms must be in different residues. Either X-PLOR or PDB atom naming conventions may be used when specifying the atom name so use whichever you are most comfortable with. The six degrees of freedom can be interpreted with the following diagram as an aid:

```
   \           /
    \ 1     2 /
    /         \
   /           \
  /             \
0 \             / 3
```

Thus assume the chain proceeds in the direction from residue 0 to 1 to 2 to 3 in the above figure. The residues involved in the constraint are those labelled 1 and 2 above, and an arbitrary number of residues may lie in sequence space between 1 and 2. Then we can completely specify the positions of 2 and 3 relative to 0 and 1 with a distance, two angles and three dihedrals. Namely, the distance between 1 and 2, the angle at 1 between 0 and 2, the angle at 2 between 1 and 3, the torsion about 0-1 of the atom preceding 0 and 2, the torsion about the virtual bond between 1 and 2, with respect to 0 and 3, and the torsion about 2-3 with 1 and the residue following 3. Note that when computing angles in the above picture, co-ordinates for '1' and '2' correspond to those of the atoms which are specified to exist in the distance constraint, while co-ordinates for '0', '3' and their preceding and following residues respectively are taken to mean their a-carbon co-ordinates, which seems the most logical choice. Sometimes it will only be possible to know the distance and the angles will be unknown, so the latter are optional. Because there is often uncertainty in the distance, you may indicate this with the "Minimum Delta" and "Maximum Delta" fields. These values are subtracted from and added to the Mean Distance respectively, to give the overall allowed distance range. There is no sharp cutoff and something which is just outside this distance range may also be accepted by foldtraj. The angles have a fixed tolerance range which may not be adjusted by the user but is set to some reasonable value.

Probabilities

Lastly, each constraint has a probability associated with it. Just before structure generation begins, the constraint list is pruned. For example, a constraint assigned a 50% probability will be removed half the time by this pruning process and ignored for the remainder of that run of foldtraj. The next time you run foldtraj, a new set of constraints may end up being used. If you do not like this idea, simply give all your constraints a probability of 1.0.

What are Fragments?

Summary

Fragments can be used to add probabilistic tangents to the random walk taken by foldtraj. That is to say, when fragments are attached to a given residue, there is a chance that the entire fragment will be placed, as one big "chunk", rather than using the trajectory distributions to choose the conformations of those residues. For example, suppose residue 6 of a protein has a five-residue fragment associated with it, with an associated probability of 0.5. That means that when foldtraj gets to residue 6, there is a 50% chance it will use the trajectory distributions to build residues 6 through 10, and a 50% chance it will use the dihedrals stored in the fragment for residues 6 through 10. Residue 11's conformation will be chosen using the trajectory distribution for that residue, unless residue 11 has fragments attached to it as well. Once a fragment is placed, it may still be subsequently removed through backtracking of course, if atomic collisions or other such problems occur. In this case, the fragment(s) will have the same probabilities of being chosen again the next time they are encountered. If more than one fragment is attached to a residue, each will be tested in turn. For example if fragment 1 has a probability of 0.9 and fragment 2 has a probability of 0.8, then there is a 90% chance of placing fragment 1, and a 10% chance of not. Out of this remaining 10%, there is then an 8% (= 80% x 10%) chance of choosing fragment 2, and a 2% chance (the left over) of using the trajectory distributions as they are, ignoring the fragments completely. Keep this is mind when choosing probabilities for fragments. When a fragment is placed, the dihedral angles are chosen according to the mean values and standard deviations you specify when creating the fragment (see the appropriate section under Edit Menu for details). If you want angles to be precise, be sure to use zero for the angle standard deviations, but be warned that atomic collisions may then prevent the fragment from ever being placed if angles are not chosen carefully.

Suggested Use

This feature is mainly implemented for future use, where fragments will automatically be added to your trajectory distribution file based on motif detection algorithms, similar to the Baker lab's I-sites library. Fragments allow you to perform "mini-threading", so that if you recognize that part of your sequence contains a well-known turn, or structural motif, you can give VisTraj that information by attaching a fragment at the appropriate location, with a probability dependent on your confidence in your assignment of that motif. Future versions may have a complete motif library incorporated into VisTraj for your convenience.

Credits

Authors

VisTraj was programmed by John J. Salama as a summer project with help from Howard Feldman. Both authors along with Dr. Christopher W.V. Hogue designed the software. This documentation was put together by Howard Feldman. The TraDES package is now maintained by Christopher Hogue.

Toolkits

VisTraj and foldtraj combine many independent software libraries. These include, but are not limited to: the NCBI toolkit including the MMDBAPI and Vibrant, Reverse PSI-BLAST, CodeBase database, Bzip2 compression, DSSP secondary structure assignment, GOR secondary structure prediction, OpenGL and glut. It is coded entirely in C and is platform independent. It is dependent on a stable implementation of OpenMotif and GLUT, which excludes certain Linux environments and the Mac OS X. We are working towards a QT version of VisTraj to overcome this problem.

Availability

VisTraj can be downloaded in binary form, along with the rest of the TraDES package at `ftp://ftp.blueprint.org/pub/TraDES/`.
Available platforms are limited by backwards compatibility of the OpenMotif library. See the FTP site for more information. See the file `CompatibilityNotes.txt` for more info on platform-related issues. For more general information about foldtraj and the TraDES algorithm, visit the TraDES website at `http://trades.blueprint.org/`

Contact Information

Please address any correspondence via email to:
Christopher W.V. Hogue Ph.D.
National University of Singapore
hogue@nus.edu.sg