gelly Documentation
Version 2.12.1

Contact buster-develop@GlobalPhasing.com

Last modified: Wed Jul 20 17:38:21 BST 2011

Table of Contents

gelly Documentation : Introduction

Contents

What is gelly?

- gelly is the name of the refinement module included as part of the BUSTER program.
- It is also supplied with BUSTER/autoBUSTER as a separate executable that can be used to refine with the GEOMETRY function only (no X-ray term). This is useful for fixing geometry restraint violations caused by manual rebuilding. The regularisation will cause bond lengths, angles, short contacts to be adjusted to the "ideal" values. It is easiest to use this separate executable by using the gelly_refine wrapper as this will do all the restraint file setup, coordinate conversion, weight assignment in an identical manner to autoBUSTER.
- gelly is based on the **geometry** program supplied with TNT, written by Lynn Ten Eyck and Dale Tronrud. It replaces the **geometry**, **shift** and **ncs** programs in BUSTER/autoBUSTER operation.
- gelly enables a much more powerful optimisation method to be used for X-ray crystallographic refinement.
- BUSTER/autoBUSTER using gelly (BUSTER-GELLY-TNT) will perform very much better than the old BUSTER. The R-factor will drop much faster and to lower final values. Feedback from users is very positive.
- If you are refining too many degrees of freedom then you will also notice that the gelly optimiser quickly widens the Rfactor Rfree gap. This is not a problem with the optimiser: instead it indicates that you are trying adjust more parameters than your data/current model justify. We are currently working hard on procedures to help for this case.
- gelly takes a novel approach for rigid body parameters. This makes it possible to mix Cartesian and rigid body parameters during a refinement optimisation. We are currently exploring the potential of this method to help during refinement / model completion.

This manual

- The gelly optimisation module can normally be used without worrying about what is presented here.
- Often autoBUSTER will be using new features that gelly enables in an automated fashion.
- However, you may wish to get access to features that have not yet been scripted or to do things in a way distinct from what a script allows.
- This manual provides the full details of the additional control cards that gelly introduces. If you find anything confusing or need help in other way please let us know.

Contact the authors

- Gelly has been written primarily by Oliver S. Smart, with many contributions from Tom Womack, Claus Flensburg, Wlodek Paciorek, Clemens Vonrhein, Maria Brandl and Gerard Bricogne.
- It is under active development.
- We welcome comments and user feedback both good and bad.
- Please send feedback to *buster-develop@globalphasing.com*

Using gelly within autoBUSTER

The gelly refinement module is automatically run with the autoBUSTER refine command provided refinement has been requested. The autoBUSTER command line options `-Gelly` and/or `-RB` can be used to provide files containing user specified .Gelly cards. Gelly cards all start with `NOTE BUSTER` and are used primarily to control restraint setup and parameter selection. gelly also has some command line options.

Doing geometry-only regularisation with gelly_refine script

The recommended way to run **standalone gelly** is by using the wrapper script **gelly_refine** supplied with autoBUSTER. Separate gelly_refine documentation is available. To see the main options for this type

```
gelly_refine -h
```

gelly_refine takes PDB files as an input, and automatically prepares the `.seq`, `.cor`, `.tnt` files needed to run gelly. As an example, for a simple run from the structure new_test.pdb

```
gelly_refine -p new_test.pdb -d test_dir -o
test_dir/gelly_refine_out.pdb
```

This will run a geometry-only optimisation of the structure and write results to the directory `test_dir`. A geometry report for the initial and final structures will be produced together with a summary of the optimisation.

If you want to control the geometry function and/or parameter selection in **gelly_refine** use the `-l <filename>` option. This can be used to introduce a file of Gelly cards. For example if you want to geometry regularise only part of a molecule that has been rebuilt before full refinement. then produce a file with appropriate `NOTE BUSTER_CONSTANT XYZ ...` cards and use this with the `-l` option.

---

gelly command line options

- gelly command line options are used primarily to control detailed features of a run. The controls can include things like the convergence gradient limit and how frequently pdb files are written.
- autoBUSTER now directly passes to gelly relevant command line options. (In the past the autoBUSTER parameter **BusterExtraArgs** was used - this will still work).
- For the **gelly_refine** geometry regularization script. Command line not recognised by the wrapper script will be passed directly to **gelly**.

  For example, a useful combination of options is

  ```
  gelly_refine -type one -screen_sigma 3.0 -p foo.pdb -d screenit -o
  /dev/null
  ```

  which writes a summary of all the restraint violations worse than 3 sigma for foo.pdb to `screenit/gelly_refine.log`

- The most useful gelly command line options are are

`-autoncs`	automatically detect NCS and impose LSSR restraints for these.
`-autoncs_weight <w>`	use weight `w` for `-autoncs` LSSR restraints.
`-target <file.pdb>`	impose target LSSR restraints to *file.pdb*.
`-target_weight <w>`	use weight `w` for `-target` LSSR restraints.
`-sim_swap_equiv`	Swap equivalent atoms to improve NCS and/or target restraints
`-special_dist <d>`	Distance cutoff `d` for special position handling
`-glim <n>`	Set the gradient convergence limit to terminate a small cycle. Terminate the optimisation when the RMS gradient drops below *n* units
`-dlim <n>`	Stop the optimisation when the RMSd movement of the molecule is more than *n* Å
`-keeppdb <n>`	Write out a pdb file every *n* small cycles

`-verbose <n>`	Set the level of reporting produced by gelly. The default level is `-verbose 2`; verbose level 4 produces enormous lists, for example every inter-atom bond and the full list of atoms in each `COMBINE` and `CONSTANT` set, but is useful for bug reports.
`-screen <NS>`	Adjust screen output so that it report the worst *NS* violations of each kind of restraint (default 100)
`-screen_sigma <s>`	Report up to *NS* restraint violations worse than *s* sigmas (default 3.0)
`-type one`	**gelly_refine** only - Just calculate the initial function values and display a report, rather than optimize
`-max <n>`	**gelly_refine** only - Allow up to *n* optimisation steps (within **autoBUSTER** use `-nsmall` instead).

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 16.02.11

gelly Documentation : The BUSTER geometry function

The geometry function used in BUSTER refinements: TNT and improvements

- The original BUSTER-TNT refinement package combined the geometry function from the TNT refinement package with the maximum-likelihood X-ray objective function from BUSTER.
- The TNT geometry function is still used, but as BUSTER is now used to refine low-resolution structures it has been necessary to supplement the geometry function. These extensions have been implemented in the gelly program.
- If desired it is still possible to use the original TNT geometry function by specifying the '-M Geom2008' macro. The TNT function is potentially superior to the "improved" geometry for high resolutions.
- The improved geometry function (from July 2009 release) has:
  - Bond and Angle geometry restraints taken from the "EH99" set: R. A. Engh and R. Huber (2006) International Tables for Crystallography. Vol. F, Chapter 18.3, pp. 382-392. In particular, cis and trans proline use different sets of restraints. Most other refinement programs currently use the older EH (1991) target parameter sets.
  - An active torsion angle restraint for protein side chain chi angles. A sinusoidal functional form is used by default, to avoid discontinuities in gradients. See Appendix D
  - A restraint which can force chiral centres defined with the 'CHIRAL' card to adopt a given chirality. The functional form is a semiharmonic restraint on the improper torsions. It is only active for distorted chiral centres. It is described in Appendix E
  - Ideal-distance contact restraints replace the TNT "bad contacts". Both these terms are semi-harmonic penalty functions that become active when two atoms that are not bonded come into contact. Although the functional form is the same the TNT bad contact provides a stiff barrier when two atoms get very close (see TNT manual: CONTACT), whilst the ideal-distance contact term, as used in PROLSQ, becomes gently active when two atoms get closer than their ideal distance. For most atom pairs this occurs at the sum of the van der Waals radii. In the case of atoms that can hydrogen bond the ideal distance is reduced from the sum of the van der Waals radii.
  - Ideal-distance contacts are used for 1-4 contacts - that is, atoms that are at the ends of a torsion angle. 1-4 contacts can help prevent eclipsed torsions. and have particular importance in helping to ensure reasonable Ramachandran statistics. Ramachandran used consideration of 1-4 contacts for amino acid models in his original description of the diagram. The TNT geometry function does not consider 1-4 contacts.
  - The ideal-distance contact target parameters used are taken from those used in the refmac package
- The improved geometry is now used by default.
- The use of the improved geometry generally results in slightly lowered Rfree, improved Rfree-Rwork gaps and much improved Molprobity scores (see wiki).
- The improved geometry function has also enabled improvements to be made in the treatment of atoms at special positions. See Appendix F

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 07.06.09

gelly Documentation : Similarity Restraints

Contents

Conventional approaches to NCS restraints

- gelly provides a number of approaches to best exploit similarity information with restraints during refinement.
- Similarity can be between different chains within the structure being refined. Such similarity arises because of Non Crystallographic Symmetry (NCS) and is conventionally exploited using restraints on positions that are based on the superposition of the related chains (or parts of the chain). The restraints are sometimes known as "soft-NCS" but will be called "superposition-based NCS" here.
- BUSTER and autoBUSTER have provided superposition-based NCS restraints from their first releases using the TNT ncs program.
- The gelly module has its own superposition-based NCS restraint routines.
- The gelly code uses RMSD superposition routines from: Coutsias, E.A., Seok, C., Dill, K.A.(2004) "Using quaternions to calculate RMSD", J. Comput. Chem., 25:1849-1857.
- The gelly routines essentially reproduced the TNT approach with some corrections (to derivatives) and improvements (more fine set selection).
- The problem with superposition-based NCS restraints is that in practice their use tends to be involved because it is necessary to adjust restraints to take account of NCS violations and domain type motions.
- NCS violations are where the structures of the NCS partners are clearly distinct. This often arises because of different crystal packing environments for part of the structure. If a conventional superposition-based NCS restraint is used then NCS violations can only arise during refinement if the atoms involved are removed from the restraint ("pruned out"). If distinct parts of the structure are modeled into density then but not pruned out from superposition-based NCS restraints then the next round of refinement will be dominated by moving the structures to agree with one another (not what is wanted!). An alternative approach is to relax the restraints by setting a low weight or high sigma. Although NCS violations are allowed by this approach much of the benefit for genuinely related parts of the structure is lost.
- Domain type motions are also prevented by the naive application of superposition-based NCS restraints. It is necessary to separate the domains into different NCS sets and provide a separate restraint groups.
- Because of the complications in application there is a tendency for NCS information not to be used in refinement, despite the very clear benefits of using NCS restraints in refinement.

Local Structure Similarity Restraints (LSSR)

- To facilitate the use of NCS (and other similarity) information in refinement, Local Structure Similarity Restraints (LSSR) have been developed.
- LSSR restraints are briefly described in an abstract number TP193 for the ACA Meeting 2008 June
- Publications describing the method in full are in preparation.
- This documentation provides information about how to use the restraints in practice with this release of autoBUSTER (May 2008).
- LSSR restraints reduce the need for manual intervention in using NCS restraints during refinement. Because of this simple command line arguments for their activation have been introduced.
- The automated options provide a simple approach that should satisfy many requirements. However, they may need some manual adjustment. Because of this it is also possible for users to precisely tailor the restraints in a detailed manner using .Gelly type cards .

- Practical examples demonstrating the use of LSSR are maintained on the Global Phasing BUSTER Wiki.
- It should be noted that if users prefer to use conventional superposition-based NCS restraints these are still available for both positions and temperature factors. NCS setups from previous releases will work in exactly the same manner as before.

Target Restraints: Exploiting structural similarity to an already known structure

- NCS restraints allow the exploitation of the similarity between two or more chains within a structure.
- But suppose instead of having NCS we know that the structure being refined is closely related to a structure that has already been solved.
- We call the known structure a "target structure". The target structure is provided by the user as a pdb-format file. It remains fixed during the refinement.
- Restraints between the refined structure (or part of it) to the fixed target structure are very much the same as NCS restraints between two or more chains in the refined structure.
- LSSR restraints can be used for target restraints. A command line option facilitates the most common type of application.
- It is also possible to use superposition-based restraints to a target structure. There is no need to superpose the target structure onto the refined structure.

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 10.07.08

Appendix A: Defining atom sets in gelly

Contents

The `NOTE BUSTER_SET` card

BUSTER_SET cards manipulate 'set's. A set is a group of atoms. The idea is that they can be used to easily produce complex selections for CONSTANT parameter selection, COMBINE groupings and NCS groups.

Using copy to create a set

To copy an existing set "Chain_A" into a new set "NcsAts"

```
NOTE BUSTER_SET NcsAts =
Chain_A
```

Alternatively the following creates a new set from a selection of atoms

```
NOTE BUSTER_SET newSet = { <atom_selection> }
```

The **<atom_selection>** must be surrounded by curly brackets. There are currently two forms for the specifications:

1. original TNT syntax as used on COMBINE and CONSTANT cards, surrounded by curly brackets. This is explained in the TNT Users Manual Refinement Data Statements section. It can be understood by considering these examples:
   - To define a set of all atoms in the molecule with the name CA (normally alpha Carbons):

     ```
     NOTE BUSTER_SET CAs = { *|*:CA }
     ```
   - To define a set of all atoms from chain P with the name CA :

     ```
     NOTE BUSTER_SET CAs = { P|*:CA }
     ```
   - To define a set of all atoms in the molecule with the name CA or CB :

     ```
     NOTE BUSTER_SET CAs = { *|*:CA *|*:CB }
     ```
   - To define a set of all atoms in chain A with residue number in range 1 to 23

     ```
     NOTE BUSTER_SET A1to23 = { A|1 - A|23 }
     ```
   - To define a set of all atoms in chain A with residue number in range 1 to 6 and range 9 to 12 and residue number 18

     ```
     NOTE BUSTER_SET A1to6and9to12and18 = { A|1 - A|6 A|9 - A|12 A|18
     }
     ```
   - To define a set of all atoms in chain A - note that gelly automatically defines a set 'Chain_x' for each named chain in the PDB file:

     ```
     NOTE BUSTER_SET MychainA = { A|* }
     ```
   - To define a set of the specified atoms from residue B|57:

     ```
     NOTE BUSTER_SET Back57 = { B|57:CA B|57:C B|57:N B|57:O }
     ```

     In this case it would probably be easier to use the intersection operator - see below for alternative definition of "Back57".

- To define a set containing the atoms of residue C|61 marked as from alternate conformation A (useful for occupancy refinements):

  ```
  NOTE BUSTER_SET C61a = { C|61:*.A }
  ```

2. RESTYPE this selects all atoms in one of the residue types listed. For example:
   - To define a set of all atoms residue type `WAT`:

     ```
     NOTE BUSTER_SET WATs = { RESTYPE WAT
     }
     ```

   - To define a set of all atoms residue type `PHE`, `TYR` or `TRP`:

     ```
     NOTE BUSTER_SET Aromatic = { RESTYPE PHE TYR TRP
     }
     ```

   This kind of selection is often useful in combination with other sets.

The {}'s produce a set definition directly containing an atom selection. This "creates a set on the fly". There are a number of examples below. As well as being used in `NOTE BUSTER_SET` they can also be used directly in the specifications of: `NOTE BUSTER_CONSTANT` , `NOTE BUSTER_COMBINE` , `NOTE BUSTER_COMBINE_RES` , `NOTE BUSTER_NCS_...` cards in place of buster set names

The intersect operator: &

The intersect operator can be used to define a set C that contains only atoms that are members of set A and are members of set B:

```
NOTE BUSTER_SET C = A & B
```

For example, to make a set "BackA" that is the intersection of two existing sets "Back" and "Chain_A". The new set will contain atoms that are members of both sets: (backbone atoms in chain A).

```
NOTE BUSTER_SET BackA = Back &
Chain_A
```

Note that you can use { } in either set selection (as explained in copy section above). For example to define a set containing the CA atoms of ALA residues from chain B:

```
NOTE BUSTER_SET CaALA_B = { B|*:CA } & {RESTYPE
ALA}
```

The union operator: +

The union operator can be used to define a set C that contains atoms that are members of either set A or set B:

```
NOTE BUSTER_SET C = A + B
```

For example, to make a set ChainAB that contains atoms from both chain A and chain B:

```
NOTE BUSTER_SET ChainAB = Chain_A +
Chain_B
```

To make a set ChainABC that contains chains A, B and C

```
NOTE BUSTER_SET ChainABC = { A|*} + { B|*}
NOTE BUSTER_SET ChainABC = { C|*} + ChainABC
```

Note that only single operators are allowed on each line, so the above specification has to be given on two lines. The union operator is useful in adding groups of atoms to existing sets. For instance, to add a range of residues B|25-B27 to a set called Rigid:

```
NOTE BUSTER_SET Rigid = Rigid + { B|25 - B|27 }
```

The complement operator: \ absolute mode

Given a set A the complement operator can be used to find the set B that consists of all atoms that are not members of A.

```
NOTE BUSTER_SET B = \ A
```

For instance to get the set of atoms that are not Water:

```
NOTE BUSTER_SET NotWater = \ Water
```

or to define a set that consists of everything that is not in chain A or chain B

```
NOTE BUSTER_SET NotAB = \ { A|* B|* }
```

The complement operator: \ relative mode

Given two sets A and B the complement operator can be used to find the set C that consists of all atoms that in set B but are not members of A.

```
NOTE BUSTER_SET C = B \ A
```

For instance you want to define two ncs sets: Nterm having all residues between 11 and 59 of chain A. Set Cterm should be the rest of chain A. This can be done by:

```
NOTE BUSTER_SET Nterm = { A|11 - A|59 }
NOTE BUSTER_SET Cterm = { A|* } \ Nterm
```

Now suppose you want to remove the residue A|60 from set Cterm. Do this by:

```
NOTE BUSTER_SET Cterm = Cterm \ { A|60 }
```

Furthermore you want to remove the side chains only of residues 123, 137 and 162, as these residues have different contacts and distinct conformations in the different NCS copies. This can be done by using the automatic defined set Side:

```
NOTE BUSTER_SET XludeS = Side & {A|123 A|137 A|162}
NOTE BUSTER_SET Cterm = Cterm \ XludeS
```

listing the atoms in a set to output log

This lists every atom that is in a set. It is useful for testing.

```
NOTE BUSTER_SET Existing_set_name LIST
```

As an example we are confused about whether the definition for Cterm in the last section is correct. We will list it after each operation that modifies it.

```
NOTE BUSTER_SET Nterm = { A|11 - A|59 }
NOTE BUSTER_SET Cterm = { A|* } \ Nterm
NOTE BUSTER_SET Cterm LIST
NOTE BUSTER_SET Cterm = Cterm \ { A|60 }
NOTE BUSTER_SET Cterm LIST
NOTE BUSTER_SET XludeS = Side & {A|123 A|137 A|162}
NOTE BUSTER_SET Cterm = Cterm \ XludeS
NOTE BUSTER_SET Cterm LIST
```

Automatic/dictionary defined atom sets

To make life easier gelly will generate some sets automatically. Others are defined by the dictionary.

As an example these are the sets defined for pdb structure 150l:

```
 Have defined    14 sets:
      set: 'All' has    5327 atoms in    815 residues - automatically defined
      set: 'Empty' has      0 atoms in      0 residues - automatically defined
      set: 'Chain_null' has    164 atoms in    164 residues - automatically defined
      set: 'Chain_A' has   1303 atoms in    165 residues - automatically defined
      set: 'Chain_B' has   1284 atoms in    162 residues - automatically defined
      set: 'Chain_C' has   1288 atoms in    162 residues - automatically defined
      set: 'Chain_D' has   1288 atoms in    162 residues - automatically defined
      set: 'Water' has    164 atoms in    164 residues - defined by dictionary file
      set: 'StdProtein' has   5163 atoms in    651 residues - defined by dictionary file
      set: 'ExoticAA' has      0 atoms in      0 residues - defined by dictionary file
      set: 'Protein' has   5163 atoms in    651 residues - defined by dictionary file
      set: 'Back' has   2601 atoms in    651 residues - defined by dictionary file
      set: 'Side' has   2562 atoms in    606 residues - defined by dictionary file
      set: 'Other' has      0 atoms in      0 residues - defined by dictionary file
```

Sets 'All' and 'Empty' are self explanatory. Gelly will create a separate set for each distinct chain in the structure. 'Chain_null' contains atoms whose chain id is blank. The set 'Water' contains atoms from residue types normally defined to be water (HOH, WAT ...). The set 'StdProtein' contains atoms from the normal coded amino acids in proteins. 'ExoticAA' are residues recognized in csdx_protgeo.dat as amino acids. 'Protein' is the union of the two sets 'StdProtein' and 'ExoticAA'. 'Other' are residues that are neither 'Protein' or 'Water'.

The two most useful dictionary defined sets are 'Back' (the set of backbone atoms in 'Protein') and 'Side' (all non-'Back' atoms in 'Protein').

It is intended to add more automatically defined sets in future, for instance sets of atoms with alternate location codes. If you have a suggestion for automatic set definition to make life easier then please contact the author.

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modified: Thu Mar 24 13:11:05 GMT 2011

Appendix B: Parameter selection in gelly

Contents

Parameter selection overview

Parameter selection means choosing which atomic parameters (XYZ, B and OCC) are held constant or combined. gelly understands normal TNT **CONSTANT** and **COMBINE** cards. However, to take advantage of the more elaborate atom selection available in gelly the **NOTE BUSTER_CONSTANT** or **NOTE BUSTER_COMBINE** cards should be used. These are explained below.

Holding parameters fixed with the NOTE BUSTER_CONSTANT card

```
NOTE BUSTER_CONSTANT <Parameter_code> <Set selection>
```

This is similar to a standard TNT CONSTANT card but uses a set definition.
<Parameter_code> code can be **'XYZ'** or **'B'** or **'OCC'** or **'ALL'**
<Set selection> is either the name of a BUSTER_SET or an atom selection in curly brackets (explanation).

For example you want to hold the positions of main chain atoms fixed. For this the dictionary defined set 'Back' can be used:

```
NOTE BUSTER_CONSTANT XYZ Back
```

or you have a ligand with a residue number of A|500. You wish to fix the occupancy of all atoms that are not in the ligand

```
NOTE BUSTER_SET Ligand = { A|500 }
NOTE BUSTER_SET NotLigand = \ Ligand
NOTE BUSTER_CONSTANT OCC NotLigand
```

The 'NotLigand' set is everything that is not in the 'Ligand' set (protein, water, cofactors .....). The complement operator: \ absolute mode is used in the definition.

Grouping parameters together with the NOTE BUSTER_COMBINE card

```
NOTE BUSTER_COMBINE <Parameter_code> <Set selection>
```

This is similar to a standard TNT COMBINE card but uses a set definition.
<Parameter_code> code can be **'XYZ'** or **'B'** or **'OCC'** (but not **'ALL'**)
<Set selection> is either the name of a BUSTER_SET or an atom selection in curly brackets (explanation).

For example you have a helix from A|50 to A|65 and you wish for the backbone of this to move as a single rigid body. BUSTER_SET cards are first used with the intersect operator: & to define the selection.

```
NOTE BUSTER_SET Helix1 = { A|50 - A|65 }
NOTE BUSTER_SET Helix1Back = Helix1 & Back
NOTE BUSTER_COMBINE XYZ Helix1Back
```

Note that it is possible to define multiple rigid bodies in gelly and mix these with normal Cartesian parameters for other atoms.

As another example let us continue developing the occupancy refinement example above. It would be sensible to combine the occupancies for all atoms of the ligand. This means that the occupancy of all atoms in the ligand will be changed by the same amount during refinement. If all the occupancies start at the same value (e.g. 1.0) then in effect a single occupancy is used for all ligand atoms.

```
NOTE BUSTER_SET Ligand = { A|500 }
NOTE BUSTER_SET NotLigand = \ Ligand
NOTE BUSTER_CONSTANT OCC NotLigand
NOTE BUSTER_COMBINE OCC Ligand
NOTE BUSTER_COMBINE B Ligand
```

The last line also combines the temperature factors for the ligand. If these all start at the same value this means that a single temperature factor will be defined for all atoms. If not then all values will be incremented up or down by a single value.

Defining groups for each residue with the `NOTE BUSTER_COMBINE_RES` card

Suppose you wish to define a grouped B factor for each side chain group in a protein. It would be possible to have a separate `NOTE BUSTER_COMBINE` cards for each residue but this would be really long winded. Instead use a:

```
NOTE BUSTER_COMBINE_RES <Parameter_code> <Set
selection>
```

This takes a set of atoms divides them into individual residues and then defines a separate combine group for each. the set. `<Parameter_code>` code can be `'XYZ'` or `'B'` or `'OCC'` (but not `'ALL'`)
`<Set selection>` is either the name of a BUSTER_SET or an atom selection in curly brackets (explanation).

Example - you want to use a separate COMBINE B group for the side chain atoms from each residue. You also want the same for the main chain atoms for each residue. This can be done by specifying:

```
NOTE BUSTER_COMBINE_RES B Side
NOTE BUSTER_COMBINE_RES B Back
```

Note that the BUSTER sets Back and Side are automatically defined. Also if a particular residue only has one atom in the side chain (eg CB of an ALA) then no COMBINE group would be defined - as a minimum of two atoms are required. It is also possible to use `NOTE BUSTER_COMBINE_RES` to define individual rigid bodies for each residue. In this case a minimum of 3 atoms is required - otherwise no group is defined.

Resetting previous parameter selection with the `NOTE BUSTER_RESET_CONSTANT_COMBINE` card

```
NOTE
BUSTER_RESET_CONSTANT_COMBINE
```

It can sometimes be useful to disregard CONSTANT, COMBINE, NOTE BUSTER_CONSTANT and NOTE BUSTER_COMBINE cards that have been produced by automatic scripts. This card can be used to get gelly to ignore all such cards that precede it. Other definitions can be specified after this.

As an example let's continue developing the occupancy refinement example above.

```
NOTE BUSTER_RESET_CONSTANT_COMBINE
NOTE BUSTER_SET Ligand = { A|500 }
NOTE BUSTER_SET NotLigand = \ Ligand
NOTE BUSTER_CONSTANT OCC NotLigand
NOTE BUSTER_CONSTANT XYZ NotLigand
NOTE BUSTER_CONSTANT B NotLigand
NOTE BUSTER_COMBINE OCC Ligand
NOTE BUSTER_COMBINE B Ligand
```

The first command specifies that gelly should disregard all previous constant and combines found. It then defines the ligand to be residue 500 from chain A. The xyz coordinates, temperature factors and occupancies of all non-Ligand atoms are held fixed. A single occupancy and temperature factor is used for the ligand. The net result is that the ligand will be able to refine its atomic positions, a temperature factor and an occupancy within a rigid protein. This means that a single temperature factor will be defined for all atoms.

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 10.07.08

Appendix C1: Similarity restraints: command line arguments

Contents

Command line options overview

It is our intention that it should be easy for the user to apply LSSR similarity restraints. Accordingly command line options are provided for the most common applications. For expert users and sophisticated treatments .Gelly cards can be used. The process of switching from the automated options to a .Gelly card treatment is described below. The .Gelly cards themselves are described in the next appendix.

Automated setup of LSSR for NCS command line option **`-autoncs`**

The **`-autoncs`** command line option will attempt to identify NCS within a structure and apply LSSR similarity restraints on positions. The **`-autoncs`** command line option can be used on the autoBUSTER command line or the gelly_refine command line,

The identification compares all pairs of chains within in a structure. For each pair it will assess the proportion of residues that has the same residue number and residue type (e.g., "ALA"). If this is above 80% then **`NOTE BUSTER`** type cards are written to activate LSSR restraints for the NCS identified. Note that water molecules are ignored by **`-autoncs`**. If no NCS is identified the **`-autoncs`** will report this and the run will continue.

Note that no restraint is placed on the temperature factors by the **`-autoncs`** option. If you wish to place harmonic restraints on temperature factors then this can be done by using the **`NOTE BUSTER_SIM_RESTRAIN_B`** .Gelly card. The process of switching from the automated options to a .Gelly card treatment is described below.

A practical example demonstrating the use of the **`-autoncs`** can be found in the 002 run of the basic NCS setup example found at the Global Phasing BUSTER Wiki.

autoncs and pruning, the **`-autoncs_noprune`** option

LSSR restraints remove much of the need to "prune out" parts of the structure that are NCS violators. This is because markedly distinct parts of the structure will result in different interatomic contact distances (so LSSR restraints will plateau). A problem arises where there are distinct but rather similar side chain conformations in different NCS copies. Here LSSR restraints will work hard to pull the conformations together. The **`-autoncs`** procedure includes a simple routine designed to identify such regions and automatically prune them from the LSSR restraints applied. The procedure is conservative taking the attitude "if in doubt prune it out". It will therefore remove some parts of the structure that could happily be left in NCS restraints. It is also not perfect and you may need to manually adjust the pruning, as described below.

If you want the automated NCS but do want the automated pruning option to used then use the `-autoncs_noprune` option rather than the `-autoncs` option. This can be a good idea at the start of a refinement where it can be expected that minor differences between NCS copies are yet to be of importance. The basic NCS setup example found at the Global Phasing BUSTER Wiki. shows how the `-autoncs_noprune` option can be usefully used (003) following on from an initial run with `-autoncs_noprune`. The pruning is shown to result in both a slight improvement in Rfree as well as getting rid of some difference map features caused by LSSR pulling together residues that are conformationally close but distinct in the different NCS copies.

autoncs* setting the weight with `-autoncs_weight` *`value`*

The default weight for autoncs LSSR restraints is `2.0/(Number_of_chain_in_NCS-1)`. This means that 2 fold NCS is weighted 2.0, 3 fold 1.0, 4 fold 0.66, 5 fold 0.5, 6 fold 0.40. The value used is reported to `LIST.html`.. It should be noted that each NCS relation between two chains creates a separate set of LSSR restraints. So this automatic weighting compensates for this and ensures that high order NCS is not overweighted.

To change the weight for **`-autoncs`** (or **`-autoncs_noprune`**) specify the **`-autoncs_weight`** ***`value`*** option. Note that the value is absolute. So to if you have 6 fold ncs and you want to double the default weight specify **`-autoncs_weight 0.80`**.

Automated LSSR setup to an external target pdb structure with `-target` *`filename.pdb`*

- Target restraints enable the exploitation of the similarity of the structure under refinement to an already known structure.
- The command line option `-target` *`filename.pdb`* sets up basic LSSR similarity restraints to a target structure.
- The `-target` option will set up a similarity group matching all atoms in the refined structure to all atoms in the target structure. This means that the chain identifiers (and residue numbers) must correspond between the two. If the chain ids do not match then you should be able to use a `-Gelly` file option instead of the `-target` option. See Specifying target type restraints with gelly cards for help with this.
- No automated "pruning" of the target restraints is performed. To prune out atom/residues from the target restraint sets then it is necessary to manually adjust the conditions used. For help in doing this see section Manual adjustment of LSSR restraint conditions starting from the automated options .
- It is not necessary or desirable to displace the coordinates into a common frame to the structure being refined. LSSR restraints are based on close interatomic distances with a structure so superposition is irrelevant.
- The using target restraints example found at the Global Phasing BUSTER Wiki demonstrates basic usage of target restraints and the `-target` option.

Setting the weight for automated target restraints with `-target_weight` *`value`*

By default `-target` will use LSSR restraints with a weight of 1.0. Use the `-target_weight` *`value`* option to change the weight.

Swapping equivalent atoms to improve NCS and/or target restraints with the `-sim_swap_equiv` or `-sim_swap_equiv_plus` options

Consider the side chain of a PHE residue:

```
       CD1-CE1
      /       \
-CB-CG         CZ
      \       /
       CD2-CE2
```

it is common that two different NCS copies can be very similar other than atoms CD1 and CE1 match CD2 and CE2 and vice versa. The **`-sim_swap_equiv`** option works through the residues in the molecule checking whether the NCS (and/or target) restraint function is improved by swapping the coordinates, temperature factors and occupancies of relevant pairs of atoms. The check is done at the beginning of every big cycle and details of any residues swapped are written to the relevant `LIST.html` file. The use of this procedure will mean that IUPAB naming conventions for sidechains are likely to be broken. It is intended to address this in a future release.

The **`-sim_swap_equiv_plus`** option extends the swapping to residues that are quasi-symmetric such as ASN, GLN and HIS. Using this option may well disrupt hydrogen bonding patterns and it is advised to check its result. The option is dumb - if a residue has good hydrogen bonding in one copy but bad in the other it is equally likely the good will be swapped to the bad!

In general, for a given refinement the procedure needs only be applied once and should then not be used (it is slow and could disrupt sensible hydrogen bonding patterns).

The basic NCS setup example found at the Global Phasing BUSTER Wiki shows how the **`-sim_swap_equiv_plus`** option swaps 65 residues to improve NCS. This results in improved Rfree, Rfree-Rwork gap and geometry scores compared to leaving equivalent atoms to remain different.

Note that **`-sim_swap_equiv`** and **`-sim_swap_equiv_plus`** procedures will work both for LSSR restraints, superposition-based (RMSD) and/or temperature factor restraints. The restraints can be produced by command line options or in a .Gelly type file.

Manual adjustment of LSSR restraint conditions starting from the automated options

- The `-autoncs` and `-target` options produce basic LSSR similarity restraints. In the final stages of refinement it is likely that you will need to adapt the restraint conditions for your needs.
- This section briefly explains how to do so.
- For a practical tutorial see the Manual adjustment of LSSR restraint conditions starting from the automated options example found at the Global Phasing BUSTER Wiki.
- The first thing to do is locate the `LIST.html` output for the final cycle of the refinement made with the `-autoncs` and/or `-target` options. The location of this file will be listed in the "top" log written by autoBUSTER to standard output. In general it will something like: `foo/01-BUSTER/Cycle-5/LIST.html` where `foo` is the directory specified in the autoBUSTER `-d` option. Open this file with a web browser (firefox). You will find log output from the automated restaint options that should include lines with a hyperlink to the file `auto_gelly_cards.txt`

  ```
  -autoncs option .Gelly cards also written to file  auto_gelly_cards.txt
  ```

- The file `auto_gelly_cards.txt` contains the .Gelly cards produce by the the `-autoncs` and/or `-target` options. (You could also find it with the linux **find** command).
- Copy the contents of this file to a file - lets call it `manual_gelly_cards.txt`
- Use your favourite editor to edit `manual_gelly_cards.txt` to reflect what you want. To do this you will need to understand the .Gelly cards used. These are described in the next appendix.
- For the next run drop the `-autoncs` and/or `-target` options. Instead specify `-Gelly manual_gelly_cards.txt`

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 16.02.11

Appendix C2: Similarity restraint: .Gelly card specifications

Contents

Overview

The use of NCS restraints in refinement can be complex. Because of this we have introduce LSSR similarity restraints and simple to use command line arguments to use these. These should cover many common applications. However, the automated setup methods may need manual tweaking. In addition users may want to use sophisticated treatments. This appendix describes the .Gelly cards allow the application of different kinds of restraints for both NCS and to target structures. See also, how to use .gelly cards in autoBUSTER .

Defining similarity groups for NCS with the `NOTE BUSTER_SIM_DEFINE` card

This card is used to define a "similarity group". The card syntax is:

```
NOTE BUSTER_SIM_DEFINE <Group_name> <Set_specifier> <Related_chain_list> ...
```

where:
`<Group_name>`: this is an arbitary user defined group name. It is usually best to make this informative. For instance in define an NCS relation between chain A and B a good group name would be "**`ncsAB`**", but if you wanted you could call it "**`Peter_Rabbit`**".
`<Set_specifier>`: This is used to define the template atoms for the NCS relation. The template must be from a single chain but can be all or any part of this. Set_specifier can either be (a) a buster set name - for instance 'Chain_D' or (b) a direct atom selection within curly brackets, for instance { D|*:CA } for just CA atoms in chain D. (Further explanation)
`<Related_chain_list>`: this is a list of chains whose atoms are to be restrained to the position of the corresponding atom in the template after superposition. Note that the template chain name must not be specified. If more than one related chain is given restraints are used between all pairs of chains but only for atoms whose equivalents are in the template. (The word 'null' should be used to specify the blank chain.)

- The match used requires that related atoms have a different chain identifier but the same atom name and residue number/insertion code.
- If you want to define a similarity group that relates two chains to another two chains then this can be done with a **`NOTE BUSTER_SIM_RESTRAIN_JOINT`** card.
- Note that the definition of a group does not in itself turn on any restraints, this is done independently, for example with a **`NOTE BUSTER_SIM_RESTRAIN_LSSR`** card.
- The same `NOTE BUSTER_SIM_DEFINE` card is used for both NCS and target type relations. The use of the `NOTE BUSTER_SIM_DEFINE` card for target restraint definition is dealt with below.
- As part of setting up a similarity group the RMSD for a superposition between the different chains and and analysis of the difference in temperature factors is listed.

Examples

- To define NCS between chain A and B use the card:

  ```
  NOTE BUSTER_SIM_DEFINE ncsAB Chain_A B
  ```

- Suppose you have six-fold NCS between chain A, B, C, D, E and F. You want to define a NCS group for this but for the protein main chain only:

```
NOTE BUSTER_SET Chain_A_NCS = Chain_A & Back
NOTE BUSTER_SIM_DEFINE ncsABCDEF Chain_A_NCS B C D E F
```

- Following on from the previous example. You can define a tight superposition based restraint on positions for the group with a separate NOTE BUSTER_SIM_RESTRAIN_RMSD card. A sigma 0.05 Angstroms is used here so the coupling between the chains is very tight.

```
NOTE BUSTER_SET Chain_A_NCS = Chain_A & Back
NOTE BUSTER_SIM_DEFINE ncsABCDEF Chain_A_NCS B C D E F
NOTE BUSTER_SIM_RESTRAIN_RMSD ncsABCDEF 0.05
```

- Following on with this example. Suppose you found that the loop between residue numbers 107 and 112 had distinct conformations. You can "prune" these out from the NCS restraint by adjusting the set of atoms used from the template. You may want to look at help for NOTE BUSTER_SET card syntax.

```
NOTE BUSTER_SET Chain_A_NCS = Chain_A & Back
NOTE BUSTER_SET Chain_A_NCS = Chain_A_NCS \ { A|107 - A|112 }
NOTE BUSTER_SIM_DEFINE ncsABCDEF Chain_A_NCS B C D E F
NOTE BUSTER_SIM_RESTRAIN_RMSD ncsABCDEF 0.1
```

Using LSSR restraints for a similarity group with NOTE BUSTER_SIM_RESTRAIN_LSSR card

Once a similarity group has been defined (with a NOTE BUSTER_SIM_DEFINE card) LSSR restraints can be activated for this group with a **NOTE BUSTER_SIM_RESTRAIN_LSSR** card. The theory of LSSR restraints is detailed elsewhere. The card syntax is:

```
NOTE BUSTER_SIM_RESTRAIN_LSSR <Group_name> [Weight]
```

where:

<Group_name>: this is a user defined similarity group name that has already defined with a NOTE BUSTER_SIM_DEFINE card or a NOTE BUSTER_SIM_RESTRAIN_JOINT card.

[Weight]: this is a optional weight. The weight must be a positive real number. If no weight is given the default weight of 1.0 is used.

- Example: Suppose you have two chains A and B that are related by NCS. To use LSSR restraints for these you could you the .gelly cards:

```
NOTE BUSTER_SIM_DEFINE ncsAB Chain_A B
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsAB
```

  This will mean that close interatomic contact distances within the A chain will be restrained to be close to the corresponding distances within the B chain. The plateauing LSSR restraint function will be used. Note that LSSR restraints are solely on position no restraint on temperature factors is produced (this can be done separately).

- Example where weights are set, you have two homo trimers in the asu. One trimer has chains A B C and the other D E F. A 2-fold axis means relates A to D, B to E and C to F. Each chain is conformational closer to its 2-fold partner than the other chains. You decide you want to restrain the 2 fold relations with a weight of 2.0 compared to the intertrimer at half weight:

```
# double weight for NCS across 2 fold
NOTE BUSTER_SIM_DEFINE ncsAD Chain_A D
NOTE BUSTER_SIM_DEFINE ncsBE Chain_B E
NOTE BUSTER_SIM_DEFINE ncsCF Chain_C F
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsAD 2.0
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsBE 2.0
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsCF 2.0
# half weight intertrimer
NOTE BUSTER_SIM_DEFINE ncsABC Chain_A B C
NOTE BUSTER_SIM_DEFINE ncsDEF Chain_D E F
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsABC 0.5
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsDEF 0.5
```

- Note the TNT "WEIGHT NCS" (set in autoBUSTER by parameter **wncs**) has no effect on LSSR restraints, only influencing harmonic restraints.

Using RMSD based positional restraints for a similarity group with NOTE BUSTER_SIM_RESTRAIN_RMSD card

Once a similarity group has been defined (with a NOTE BUSTER_SIM_DEFINE card) superposition based RMSD restraints can be activated for this group with a **NOTE BUSTER_SIM_RESTRAIN_RMSD** card:

```
NOTE BUSTER_SIM_RESTRAIN_RMSD <Group_name> [sigma_XYZ]
```

where:

<Group_name>: this is a user defined group name that has already defined with a NOTE BUSTER_SIM_DEFINE card or a NOTE BUSTER_SIM_RESTRAIN_JOINT card.

[Sigma_XYZ]: is a optional positive real number specifying the sigma to be used on the restraint in Å. Each atom pair in the restraint will contribute separately with this sigma. If no **Sigma_XYZ** is specified then a sigma will be obtained from the "WEIGHT NCS" that is current. The TNT "WEIGHT NCS" value is normally set by the autoBUSTER parameter **wncs**. The default value for **wncs** is 50). Following the practice of the TNT **ncs** program **Sigma_XYZ** is set to **1/sqrt(wncs)** (if not specified on the card). The default **Sigma_XYZ** is accordingly 0.14Å.

- These restraints use RMSD superposition routines from: Coutsias, E.A., Seok, C., Dill, K.A.(2004) "Using quaternions to calculate RMSD", J. Comput. Chem., 25:1849-1857
- See also TNT Users' Guide section on Noncrystallographic Restraints
- We used to call these restraints "soft-NCS". However, because the restraints are fully harmonic and superposition is inflexible for large chain they are not really soft! LSSR restraints are normally much softer. If you want to ensure that two chains are kept pretty much identical then use RMSD based restraints with a sigma of 0.05Å. Differences will then not be allowed during refinement.
- Restraints on temperature factors can be separately specified with a `NOTE BUSTER_SIM_RESTRAIN_B` card
- For example, you have two chains A and B and want tight RMSD restraints for all atoms, use a .gelly file containing:

```
NOTE BUSTER_SIM_DEFINE ncsAB Chain_A B
NOTE BUSTER_SIM_RESTRAIN_RMSD ncsAB 0.05
```

- For another example suppose you had 4 chains A, B, C and D. You wish to have a soft NCS restraint on positions only for backbone atoms for all the chain except residue 23 to 27 where there is a loop with distinct conformations. This can be achieved by using the dictionary provided sets 'Back' for backbone atoms and automatic set 'Chain_A'. The sigma used is left under the control of `wncs` (defaults to 0.14Å)

```
NOTE BUSTER_SET Chain_A_NCS = Chain_A & Back
NOTE BUSTER_SET Chain_A_NCS = Chain_A_NCS \ { A|23 - A|27}
NOTE BUSTER_SIM_DEFINE ncsABCD Chain_A_NCS B C D
NOTE BUSTER_SIM_RESTRAIN_RMSD ncsABCD
```

Using temperature factor restraints for a similarity group with `NOTE BUSTER_SIM_RESTRAIN_B` card

Once a similarity group has been defined (with a `NOTE BUSTER_SIM_DEFINE` card) harmonic restraints coupling the temperature factors can be activated for this group with a `NOTE BUSTER_SIM_RESTRAIN_B` card:

```
NOTE BUSTER_SIM_RESTRAIN_B <Group_name> [sigma_B]
```

where:
`<Group_name>`: this is a user defined group name that has already defined with a `NOTE BUSTER_SIM_DEFINE` card or a `NOTE BUSTER_SIM_RESTRAIN_JOINT` card.

`<Sigma_B>`: is a positive real number specifying the sigma to be used on the restraint in Å^2. Each atom pair in the restraint will contribute separately with this sigma. `(Sigma_B)` : is a optional positive real number specifying the sigma to be used on the restraint in Å. Each atom pair in the restraint will contribute separately with this sigma. If no `Sigma_B` is specified then a sigma will be obtained from the "WEIGHT NCS" that is current. The TNT "WEIGHT NCS" value is normally set by the autoBUSTER parameter **wncs**. The default value for `wncs` is 50). Following the practice of the TNT **ncs** program `Sigma_B` is set to `(5/0.3)*1/sqrt(wncs)` (if not specified on the card). The default `Sigma_B` is accordingly 2.36Å^2.

- The restraint is based on that used in TNT this couples individual temperature factor but allows the average temperature for each chain to vary without penalty.
- In practice use of NCS restraints on temperature factors usually only produces a marginal improvement in Rfree within autoBUSTER. For this reason the automated options do not use temperature factor restraints.
- For example, you have two chains A and B and want to use LSSR for positions and to couple temperature factors with the default moderate coupling, use a .gelly file, containing:

```
NOTE BUSTER_SIM_DEFINE ncsAB Chain_A
B
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsAB
NOTE BUSTER_SIM_RESTRAIN_B ncsAB
```

Joining together different similarity groups with the `NOTE BUSTER_SIM_JOINT` card (useful for including water in NCS)

The `NOTE BUSTER_SIM_JOINT` card allows the joining together of two or more similarity groups to form a new group. The card syntax is:

```
NOTE BUSTER_SIM_JOINT <New_group_name> <Existing_group1> <Existing_group2> ...
```

where:
`<New_group_name>`: this is an arbitary user name for the new group. It is usually best to make this informative.
`<Existing_group1>`: this is a user defined group name that has already defined with a `NOTE BUSTER_SIM_DEFINE` card
`<Existing_group2>`: this is a user defined group name that has already defined with a `NOTE BUSTER_SIM_DEFINE` card
`<Existing_group3>`: a JOINT card must have at least two groups but can be composed of as many as you want.

- The main practical use of a joint card is to include water molecules into NCS restraints. This is demonstrated in the 4cha including water in to NCS example found at the Global Phasing BUSTER Wiki. The ccp4 tool sortwater (http://www.ccp4.ac.uk/dist/html/sortwater.html) can be used to sort waters by the protein chain to which they "belong". If this has been done for a protein with chain id's A and B producing waters in chains U and V then the .Gelly file would be useful:

```
NOTE BUSTER_SIM_DEFINE ncsA-B Chain_A B
NOTE BUSTER_SIM_DEFINE ncsU-V Chain_U V
NOTE BUSTER_SIM_JOINT ncsAU-BV ncsA-B ncsU-V
NOTE BUSTER_SIM_RESTRAIN_LSSR ncsAU-BV
```

the joint similarity group means that the A/U hybrid is related to the B/V hybrid. LSSR restraints are used. This means that the contact distances between corresponding water molecules and their proteins will be restrained to be similar.

Support for old-style methods of defining NCS: `CLUSTER` and `NOTE BUSTER_NCS_SOFT` cards

TNT uses a CLUSTER card with CHAINS in it to specify harmonic restraints on positions and temperature factors. **gelly** will interpret **`CLUSTER`** cards by 'translating' them into its own format and should produce identical function values. Details of the translation will be listed in `LIST.html`. For instance the CLUSTER card:

```
CLUSTER NTERM RESIDUE 11 - 59 CHAINS A B C
D
```

is translated by gelly:

```
Translating card 'CLUSTER NTERM RESIDUE 11 - 59 CHAINS A B C D' into gelly format:
NOTE BUSTER_SIM_DEFINE         cluster0001_NTERM { A|11 - A|59 } B C D    # card added - translated CLUSTER
NOTE BUSTER_SIM_RESTRAIN_RMSD  cluster0001_NTERM    # card added - translated CLUSTER
NOTE BUSTER_SIM_RESTRAIN_B     cluster0001_NTERM    # card added - translated CLUSTER
```

The previous release of **gelly** used a **`NOTE BUSTER_NCS_SOFT`** card to allow more flexible specification of harmonic NCS restraints. These cards continue to be read and will produce identical results. They are interpreted by 'translating' them into the new format. Details of the translation will be listed in `LIST.html`. It is advised to switch to the new format as this allows the use of new features such as LSSR.

Specifying target type restraints with gelly cards

Target restraints enable the exploitation of the similarity of the structure under refinement to an already known structure. A command line shortcut sets up a basic default approach. This section explains the gelly cards for target restraints. The cards are both used by the shortcut routine and can be user specified for finer control.

Specifying target structure coordinate files with the `NOTE BUSTER_TARGET01` card

The first thing that is necessary for target restraints is load an external target structure to be used. The

```
NOTE BUSTER_TARGET01 full_path_to_file.pdb
```

card allows the specification of the coordinates to be used for a target coordinate set.

- Note that the target coordinates are kept fixed during each refinement.
- It is necessary to specify the full unix pathname for the coordinate file
- It is normal to use .pdb coordinates for the target file but the routines also allow TNT type .cor records to be used.
- All records other than ATOM/HETATM are ignored.
- It is not necessary or desirable to displace the coordinates into a common frame to the structure being refined. LSSR restraints are based on close interatomic distances with a structure so superposition is irrelevant. RMSD based restraints will work out the optimal superposition for the similarity group(s) during the calculation.
- If more than one target coordinate set is required then use

  ```
  NOTE BUSTER_TARGET02 full_path_to_file2.pdb
  ```

  for the next. Up to 99 coordinate sets can be defined.
- Note that the *`-target filename.pdb`* command line option will produce "on-the-fly" additional gelly cards including **`NOTE BUSTER_TARGET01`** to activate default (all atom) LSSR target restraints to a given structure.

Defining similarity groups for target restraints with the `NOTE BUSTER_SIM_DEFINE` card

- Target similarity groups are defined by a special form of the **`NOTE BUSTER_SIM_DEFINE`** card. The special form has a single a "Related_chain" is specified starting with "TARGET01_".
- For example

  ```
  NOTE BUSTER_TARGET01 full_path_to_file.pdb
  NOTE BUSTER_SIM_DEFINE AtargetY Chain_A TARGET01_Y
  NOTE BUSTER_SIM_RESTRAIN_LSSR AtargetY
  ```

  will define a similarity group between the predefined BUSTER_SET "Chain_A" (that is composed of the A chain) and the Y chain from the TARGET_01 structure. LSSR restraints are used for the similarity group.
- For target similarity groups only, it is allowed to use a wildcard for the target chain, for example

```
NOTE BUSTER_TARGET01 full_path_to_file.pdb
NOTE BUSTER_SIM_DEFINE AllTarget All TARGET01_*
NOTE BUSTER_SIM_RESTRAIN_LSSR AllTarget
```

This will match all atoms from all chains to the corresponding atom from the target structure (matching atom name, residue number and chain identity).

- If more than one target structure is used then the "Related_chain" chain specifier should be start to "TARGET01_", "TARGET02_" etc. For example, consider a structure under refinement with two chains A and B. The A chain is based on a high resolution structure "foo.pdb". The B chain is based on separate high resolution structure "another.pdb". Target restraints could be defined:

```
NOTE BUSTER_TARGET01 full_path/foo.pdb
NOTE BUSTER_SIM_DEFINE Afoo Chain_A TARGET01_A
NOTE BUSTER_SIM_RESTRAIN_LSSR Afoo
NOTE BUSTER_TARGET02 full_path/another.pdb
NOTE BUSTER_SIM_DEFINE Banother Chain_B TARGET02_A
NOTE BUSTER_SIM_RESTRAIN_LSSR Banother
```

- Once a similarity group has been defined to a target structure, it can be used for LSSR restraints, and/or harmonic RMSD based restraints and/or temperature factor restraints. The same cards as for NCS restraints are used. See `NOTE BUSTER_SIM_RESTRAIN_LSSR`, `NOTE BUSTER_SIM_RESTRAIN_RMSD` and `NOTE BUSTER_SIM_RESTRAIN_B` card discussion.

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 23.11.10

Appendix D: Utility restraint terms in gelly

Contents

As well as all the restraints that are defined by TNT-type restraint dictionaries for input molecules **gelly** allows the definition of a number of "utility" restraints that can be used a user to allow finer control of refinement. It is intended to add more of these terms in the future. If you would like a restraint on any particular feature then let us know.

Restraint on the distance between two given atoms: `NOTE BUSTER_DISTANCE`

It can be useful to be able to restrain the distance between two given atoms. For instance such restraints can be used to ensure secondary structure is maintained in the initial cycles of refinement. The `NOTE BUSTER_DISTANCE` allows the specification of a harmonic restraint on the distance between any two atoms to a given target with a given sigma. The restraint can be half-harmonic, that is only apply if the distance strays above (or alternatively below) the target. It can be specified:

```
NOTE BUSTER_DISTANCE +3.0 0.02 A|10:O A|14:N
```

this card means that a large penalty will be applied if the distance between mainchain O of residues A|10 and N of A|14 goes ABOVE 3.0 angstroms. the sigma for the restraint is 0.02 angstroms (as strong as a typical bond). Such restraints can be used to maintain expected helical secondary structure

To restrain a distance to be equal to a target value use an = in place of + for example:

```
NOTE BUSTER_DISTANCE =3.0 0.02 A|10:O A|14:N
```

To have apply a restraint only if two atoms come below a certain distance then use - instead

Note that using a distance restraint between two atoms does not effect any short contacts between them - this can be controlled using the TNT EXCLUDE card.

A practical example demonstrating the use of `NOTE BUSTER_DISTANCE` cards can be found in the forcing a zinc ion site to be tetrahedral example found at the Global Phasing BUSTER Wiki.

Using utility distance restraints across a cell symmetry operation (for special positions)

A new feature introduced in the November 2011 release of BUSTER is to be able to specify a utility distance restraint to an atom across a cell/symmetry operation. The main use for this kind of restraint is to prevent an atom drifting from a special position or axis. To specify that a restraint is across a symmetry operator add `SYMM` to the end of the `NOTE BUSTER_DISTANCE`. The `SYMM` means that the distance restraint will be to a symmetry copy of the 2nd atom. All symmetry operations including lattice translations are checked (other than the identity) and which ever gives the smallest distance is chosen (if two or more operations give the same distance the first one found is chosen). As an example

```
NOTE BUSTER_DISTANCE +0.001 0.02 A|457:O A|457:O SYMM
```

This would be used for a water molecule on a special position. The restraint is on the distance between the water molecule and its closest symmetry equivalent. If this distance is above 1/1000 th Angstrom then a harmonic restraint is applied reducing the distance with a sigma of 0.02 Angstroms. The effect of this will be to strongly pull the atom and its symmetry equivalent together. Note that in the case of a symmetry axis the atom will still be able to move along the axis. For further details on the handling of special position atoms and ions see Appendix F: Handling special position atoms and ions .

A practical example demonstrating the use of `NOTE BUSTER_DISTANCE` cards can be found in the Handling Special Position atoms and ions example found at the Global Phasing BUSTER Wiki.

Details as to the operator setup by `SYMM` can be found in the `LIST.html` output file in a section like:

```
Setup symmetry for utility distance restraints (NOTE BUSTER_DISTANCE)
  A    457 O       to symmetry copy of A   457 O       SYMM specified so found operator   3555    gave minimum distance=   1.129
  D    581 O       to symmetry copy of D   581 O       SYMM specified so found operator   3545    gave minimum distance=   0.332
  E    1244O       to symmetry copy of E   1244O       SYMM specified so found operator   8555    gave minimum distance=   0.129
```

Although it is seldom necessary it is also possible to use a specified symmetry operator for a restraint. The operator wanted is specified in "PDB convention" (where `1555` is the identity operator) the end of the line. For instance:

```
NOTE BUSTER_DISTANCE +0.001 0.02 A|457:O1 A|457:O2 3665
```

This would apply a restraint reducing the distance between atom `O1` of residue 457 and atom `O2` of the same residue but through the specific 3rd symmetry operator (see output PDB file) and the translation vector `110` . Fortunately is it seldom necessary to use a specific operator - try `SYMM` in the first instance.

Restraint on the angle between three given atoms: `NOTE BUSTER_UTILANGLE`

This restraint is on the angle between any three atoms. This could be a bond angle but need not be. For example:

```
NOTE BUSTER_UTILANGLE 109.5 4.0 B|109:SG B|154:ZN B|114:SG
```

This places a harmonic restraint on the angle between the three atoms from the B chain SG from residue 109, ZN residue 154 and SG from residue 114. The target value of the restraint is 109.5 degrees and the sigma is 4 degrees.

Note that using angle restraint does not directly effect any short contacts - this can be controlled using the TNT EXCLUDE card.

LIMITATION: currently cell/symmetry operations are ignored in UTILANGLE calculations A practical example demonstrating the use of `NOTE BUSTER_UTILANGLE` cards can be found in the forcing a zinc ion site to be tetrahedral example found at the Global Phasing BUSTER Wiki.

Restraint on the torsion angle between 4 given atoms: `NOTE BUSTER_UTILTOR`

- UTILTOR provides a sinusoidal penalty on the torsion angle between four specific atoms.
- The syntax of the card is

  NOTE BUSTER_UTILTOR $<N_{minima}>$ $<\varphi_{zero}>$ $<\sigma_{\varphi}>$ <Atspec1> <Atspec2> <Atspec3> <Atspec4>

  where:
  $<N_{minima}>$: this is number of minima in the sinusoidal penalty function.
  $<\varphi_{zero}>$: this is the angle in degrees of one of the minima in the sinusoidal penalty function.
  $<\sigma_{\varphi}>$: is the sigma for the restraint in degrees.
  <Atspec1> <Atspec2> <Atspec3> <Atspec4>: are the four atom specifiers.

- The actual penalty function applied is

  $V_{utiltor} = \{V_{barrier}/2\}*[1-\cos(N_{minima}(\varphi-\varphi_{zero}))]$

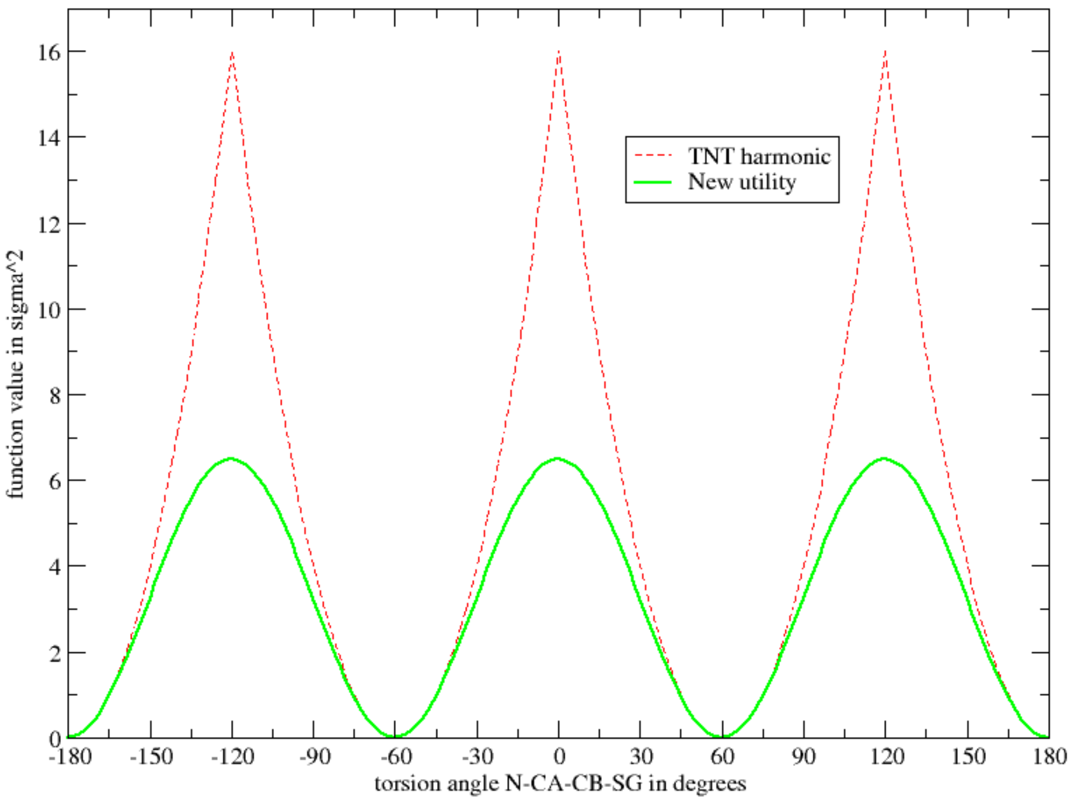
  The barrier height is set so that curvature around each minima is equal to that for a harmonic term with this sigma. This can be found by the relations

  $V_{barrier} = K/(\sigma_{\varphi}.N_{minima})^2$
  $K = 4*(180/\pi)^2 = 13131$

- For example if residue A|3 is a cysteine the card:

  ```
  NOTE BUSTER_UTILTOR 3 60.0 15.0 A|3:N A|3:CA A|3:CB A|3:SG
  ```

  will produce a penalty on the χ_1 of the residue. The penalty with have 3 minima, one a 60°. (The other two minima will consequently be at ±120° from this, that is at -60° and 180°). The function value will be around 1.0 at 60± 15° . The barrier height will be 6.48 units (see above).



The graph compares the utiltor penalty with the relevant TNT csdx dictionary torsion term:

```
egrep "CYS.*TORSION" ~/autobuster/tnt/data/csdx_protgeo.dat
GEOMETRY        CYS  TORSION  3060      15    N    CA    CB    SG
```

autoBUSTER weight these torsion terms zero. A future release will implement dictionary sinusoidal penalty functions on the torsion angles. In the meantime the UTILTOR provides (a slightly inconvenient) way to allow torsion control for selected important parts such as ligands.

LIMITATION: currently cell/symmetry operations are ignored in UTILTOR calculations

---

Restraint on the sum of occupancies for two atoms: `NOTE BUSTER_OCCSUM`

When refining occupancies it can be useful to ensure that the occupancies of alternative conformations are restrained such that their sum is equal to a given value (normally 1.0). The `NOTE BUSTER_OCCSUM` card imposes a restraint:

```
V_occsum = K [ (Sum_i occ_i) - 1.0]^2 where
```

where

```
K= 1/sigma^2
```

Example:

```
NOTE BUSTER_OCCSUM 1.0 0.005 H|1:CB.A
H|1:CB.B
```

This specifies that atoms CB.A and CB.B from residue 1 in chain H have a restraint on the sum of their occupancies to equal 1.0. The sigma for the restraint is 0.005

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 25.04.2014

Appendix E: CHIRAL Restraints in gelly

Contents

CHIRAL Restraints: Summary

A restraint has been added to the TNT **geometry** function that restrains the improper torsions of chiral centres. The restraint only becomes active when atoms start to approach planarity. The restraint should "flip" the chirality of any incorrect centres. It will be turned on by default in **autoBUSTER** and **gelly_refine** but can be zero weighted if wished.

The gelly CHIRAL Restraint term

The TNT **geometry** program includes analysis that the chirality of atoms, such as the α carbon in amino acids is correct. For example, our standard dictionary for proteins `csdx_protgeo.dat` includes the term, for the CA atom from alanine

```
GEOMETRY ALA CHIRAL 1 1 CA N CB
C
```

This in the TNT **geometry** program will result in a report of bad chirality if an alanine CA atom flips into a D rather than the L enantiomer. But no actual restraint penalty is produced to prevent such a flip. The TNT geometry program does provide an harmonic IMPROPER torsion term but this must be weighted very highly to force incorrect chiral centres to flip to correct values.

To remedy this deficiency, in the release of Feb 2007, we introduced a new restraint to prevent chiral atoms approaching planarity and flipping to the incorrect enantiomer. This restraint takes a half-harmonic form:

$$V_{chiral} = \begin{cases} 0.0 & \textit{if } \Omega_{model} < \Omega_{cutin} \\ (W_{chiral}/\sigma_{chiral}^{2})*(\Omega_{model}-\Omega_{cutin})^{2} & \textit{otherwise} \end{cases}$$

- Here Ω_{model} is the improper torsion angle as defined in the TNT `GEOMETRY *** CHIRAL` term. For example the `ALA` Calpha term quote above will mean that the Ω_{model} is the improper torsion angle `CA-N-CB-C`. For an ideal geometry this angle is around -34 degrees. The angle increases to zero degrees as the chiral center becomes planar. Positive angles mean that the alanine has flipped to a D enantiomer - most undesirable.
- Ω_{cutin} is the angle above which the restraint becomes active. In practice an angle of -25 degrees is found to be optimal - this is the hardcoded default. It is possible to override this by using a **`NOTE BUSTER_CHIRAL_ANGLECUT`** card, although in practice it is unlikely to be necessary.
- The weight W_{chiral} and "sigma" σ_{chiral} control how step the penalty preventing inversion is. Conveniently **autoBUSTER** already sets the WEIGHT for the TNT CHIRAL term (despite this term originally only producing a report). The default WEIGHT is set to 5.0 and this should normally be left unaltered. With this weight and Ω_{cutin}=-25 degrees it was found that a "sigma" σ_{chiral}=3 degrees produced a penalty that could in practice flip incorrect chiral centres. If it is desired to turn off the new penalty this can be done by setting the weight of the term to zero. To do this within **autoBUSTER** by setting the command line option **`wchiral="0.0"`** - see autoBUSTER documentation. For **gelly_refine** use a card:.

  ```
  WEIGHT CHIRAL 0.0
  ```

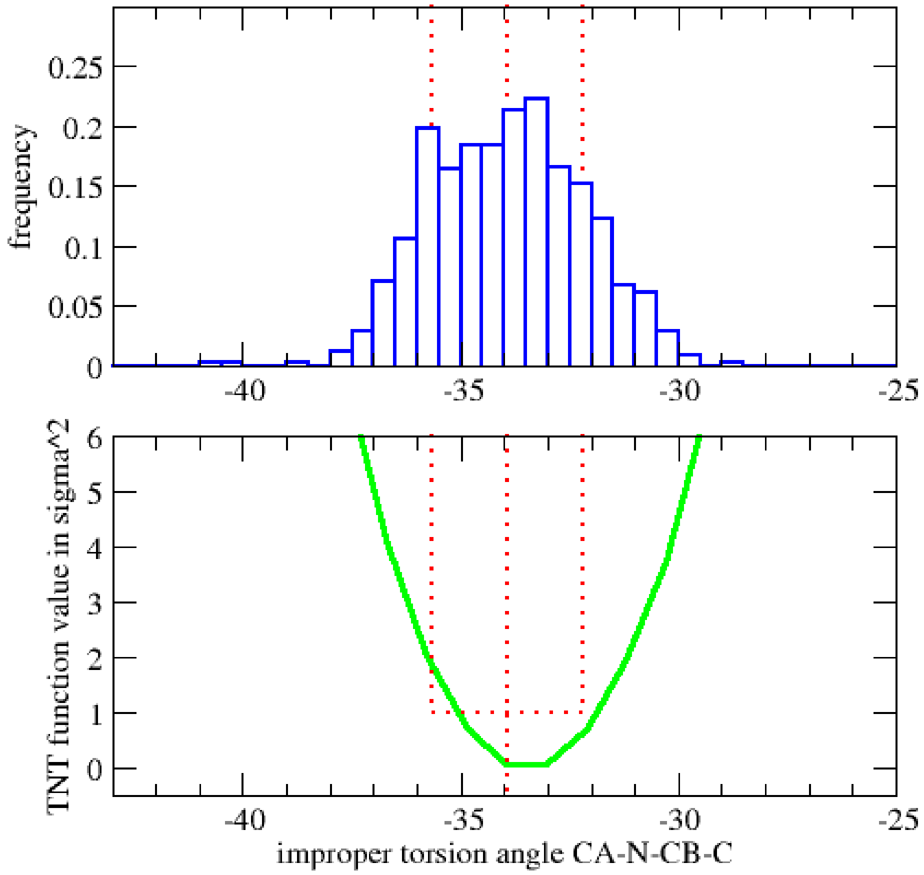
To restrain the chirality of a D amino acid it is simply necessary to change the order of the atoms in the TNT GEOMETRY restraint term. For instance D-alanine (residue type DAL) has a chiral term for its CA atom (from `csdx_protgeo.dat`)

```
GEOMETRY DAL CHIRAL 1 1 CA CB N C ! D-amino acid
Calpha
```

Compared to normal ALA the 2nd and 3rd atoms in the improper torsion are swapped. The improper torsion angle consequently has an opposite sign. The restraint will therefore favour the D-form and prevent flipping to the L.

Background work for the chiral restraint term

The main purpose of the term would be to restrain the chirality of chiral centres in amino acids (and other molecules) to correct enantiomeric form. The restraint should flip incorrect centres but should not interfere with other restraints for good geometry. For the α carbon of normal L-amino acids, the restraint would apply on the improper torsion `CA-N-CB-C`. It was decided initially to make a survey of the value found for this improper torsion in 5 well refined very high resolution (< 0.81 Å) structures, from the pdb (1ucs 1us0 1yk4 1fn8 1gci). Results are shown in the histogram with blue bars:



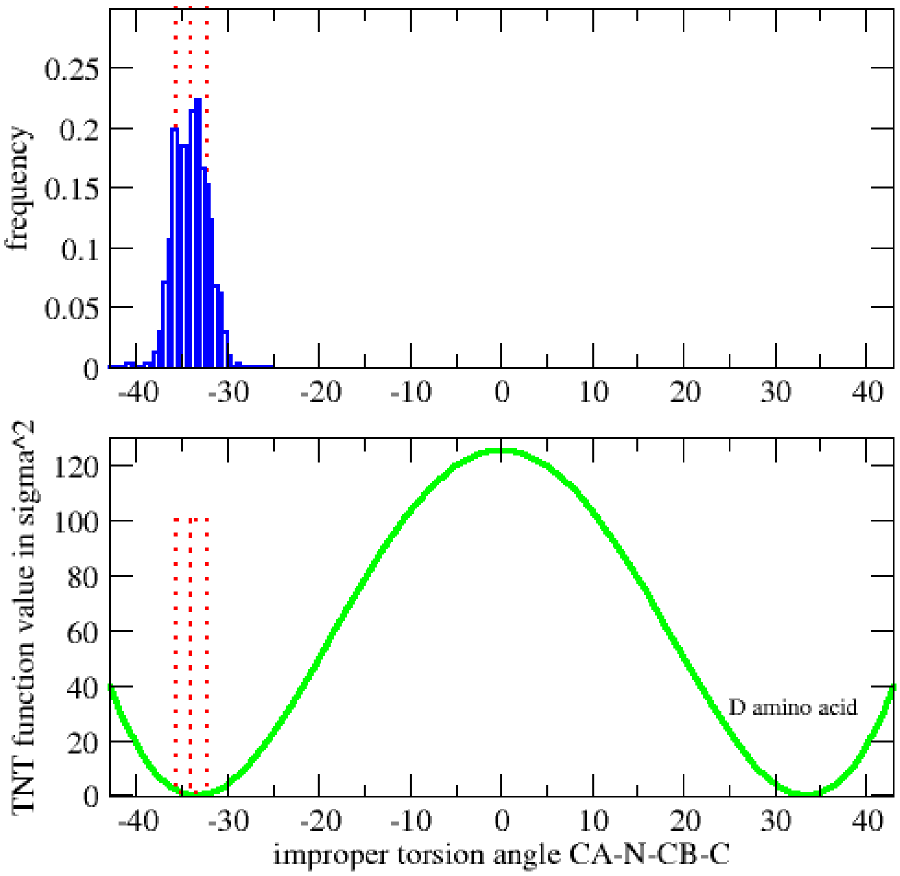
The average improper torsion was found to be -33.9 degrees with a std deviation of 1.7 degree - shown by the red dotted lines.

If a normal L-amino acid is distorted so that its chirality is pushed first towards planar and then inverted the bond angles around the CA are strained (the angles are CB-CA-N, CB-CA-C and C-CA-CB). This means that the TNT geometry function already has restraints that would act to prevent an alpha carbon inverting. To judge whether these restraints produce reasonable behaviour the TNT geometry function value cost of the chiral inversion of a single alanine residue was studied. The method used was adiabatic mapping, otherwise known as co-ordinate driving. The weights used for each term were set to the **autoBUSTER** default. A graph of function value found for each improper torsion is shown above (green line). It should be noted that no explicit restraint on the improper torsion (or any other chiral measure) was used.

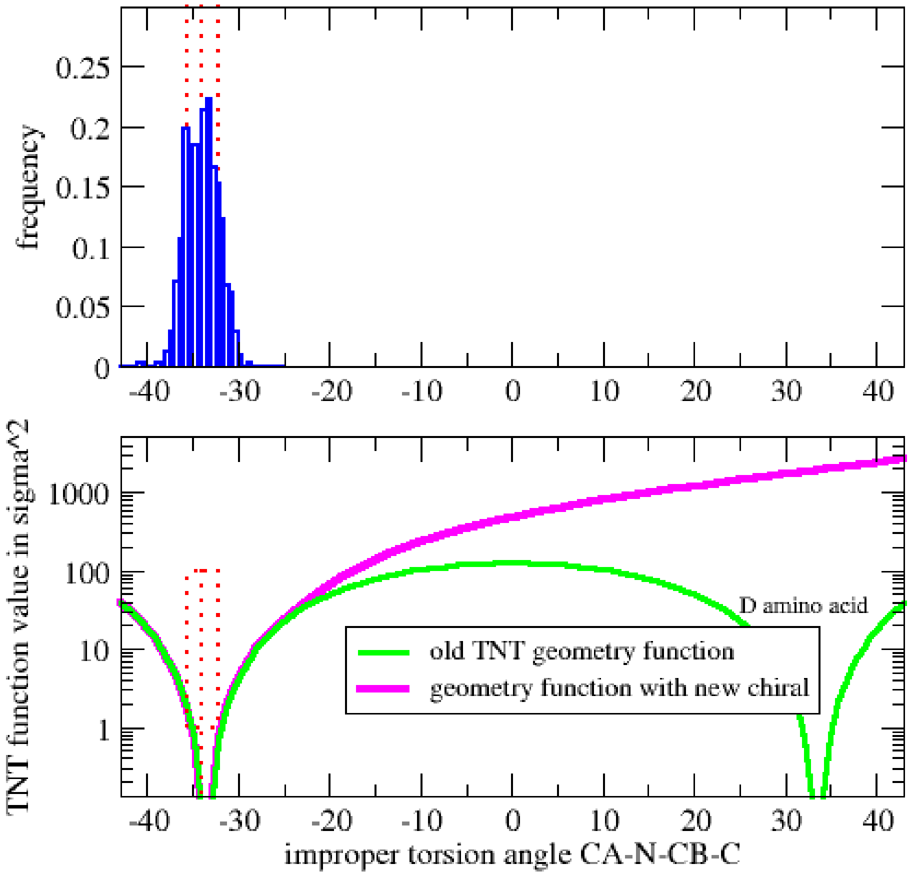
The minimum in the geometry value accords well with the mean from the distribution (central red dotted line). Furthermore, a function value of 1.0 is reached close to plus/minus sigma from the mean.

It can be concluded that the Engh and Huber parameters for the three bond angles N-CA-CB, C-CA-CB and N-CA-C are very well suited to describe behaviour near the minimum. Introducing a restraint on the improper torsion that was active near the minimum would produce "double counting" and be unwise.

If the atom is distorted to planarity and beyond the angle restraints do not favour the L-form over D-form. This can be shown by plotting the graphs on a larger scale:



The penalty for inversion is just over 120 sigma squared. It was concluded what was required was additional restraint that only operated when the chiral improper torsion was above -25 degrees and was sufficiently large to wipe out the D minimum. The gelly CHIRAL Restraint term does exactly this:



Note the function value now uses a log scale. The new chiral penalty wipes out the second minimum but leaves behavour near the normal L-minimum unaltered.

Trials where undertaken on structures with incorrect flipped chiral centres. Refinement using **autoBUSTER** using **gelly** and the new restraint term invariably resulted in the chiral centre being flipped to the correct chirality. In some cases where the model was very poorly placed into density occasionally the centre would remain with a near planar improper torsion (-15 degrees). Even if this was the case the new restraint ensures that centre has the correct chirality.

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 10.07.08

Appendix F: Handling special position atoms and ions

- The non-bonded contact term of the TNT geometry function ignored any interaction between an atom and any of its symmetry copies. This is correct for single atoms in special positions, but leads to odd results for larger groups: protein side-chains could interpenetrate, and sulfate ions would be pushed from special positions and axes. To avoid this, it was necessary manually to create EXCLUDE cards.
- The July 2009 release introduced a distance cutoff of 0.5 Å. This means that an atom must be within 0.5 Å of its symmetry copy to be regarded as special. This distance can be adjusted with the **`-special_dist`** command line argument: for instance **`-special_dist 1.5`** would treat any atom within 1.5 Å of its symmetry copies as special.
- All contacts between any atoms from a residue containing a special atom are ignored.
- A WARNING is produced if the occupancy of any atom regarded as special is 1.0.
- The November 2011 introduced the ability to manually setup utility distance restraints to prevent atoms drifting from special positions or axes. For further details see Appendix D: Using utility distance restraints across a cell symmetry operation (for special positions).
- It is planned to further improve the treatment of atoms at special positions in a future release. In particular, automated adjustment of occupancies and distance restraints to symmetry copies would be desirable.

Page Author: Oliver S. Smart
Please send feedback to: *buster-develop@globalphasing.com*
Last modification: 25.04.2014