

MakeTNT-toolkit (Version 2.4.0)

User Documentation

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2006) M. Brandl

0. What's new in MakeTNT2.4.0?

1. In a Nutshell

2. Installation

- Requirements
- Installation

3. Manuals and Evaluation Runs

- **MakeTNT**
 - General Information
 - Requirements and Installation
 - Command Line Switches and Arguments
 - Conversion Mechanism
 - Output
 - Examples
 - Evaluation Runs MakeTNT
- **PDB2TNT**
 - General Information
 - Requirements and Installation
 - Command Line Switches and Arguments
 - Conversion Mechanism
 - Output
 - Examples
 - Evaluation runs for MDL2TNT/PDB2TNT

- **MDL2TNT**

- General Information
- Requirements and Installation
- Command Line Switches and Arguments
- Conversion Mechanism
- Output
- Examples
- Evaluation runs for MDL2TNT/PDB2TNT

- **MOL22TNT**

- General Information
- Requirements and Installation
- Command Line Switches and Arguments
- Conversion Mechanism
- Output
- Examples
- Evaluation Runs for MOL22TNT

- **EditTNT**

- General
- Requirements and Installation
- How to start EditTNT
- How to use EditTNT
- Output
- Tutorial
 1. Setup
 2. Restraints from Corina
 3. Restraints from xplo2d (XPLOR): adding/removing restraints

- **EditREFMAC**

- General
- Requirements and Installation
- How to start EditREFMAC
- How to use EditREFMAC
- Output
- Example

- **MakeLINK**

- General Information
- Requirements and Installation
- Command Line Switches and Arguments
- Mechanism
- Output
- Example

4. Related Resources and Acknowledgements

- TNT refinement package
 - CORINA
 - MDL file format specification
 - various Rasmol pages:
 - Rasmol at UMass
 - OpenRasMol
 - RasMol at Bernstein + Sons
 - Acknowledgements
-

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modified: Thu Jul 2 14:52:23 BST 2009

MakeTNT in a Nutshell

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

MakeTNT is a toolkit to automatically generate geometrical parameters (MakeTNT, PDB2TNT, MakeLINK, etc.) for modelling and refinement in BUSTER-TNT, autoBUSTER, Refmac, O and Coot.

Restraints for Ligands/Single Residues

Tool	Input Format
MakeTNT	<ul style="list-style-type: none">● XPLOR/CNS/CNX<ul style="list-style-type: none">○ topology (*.top) and parameter (*.par) files produced by Corina○ XPLOR/CNS/CNX○ topology (*.top) and parameter (*.par) files produced by xplo2d● REFMAC library files created by<ul style="list-style-type: none">○ Libcheck (*.lib)○ Corina (*.cif)
PDB2TNT	PDB
MDL2TNT	MDL (3D coordinates)
MOL22TNT	MOL2 (Tripos)
EditTNT	TNT-restraints
EditREFMAC	Refmac-restraints

[IMAGE]

Restraints for Links : MakeLINK/PDB2TNT

MakeLINK

- checks a model for close residue-residue contacts
- adds CONECT cards to the TNT-sequence file
- calls PDB2TNT to generate restraint parameters for links not represented in the TNT-standard library

[IMAGE]

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 20.02.06

Installation

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Content

- Requirements
 - Installation
-

Requirements

- Running **Unix** (Linux, SGI/IRIX or HP/Tru64)

Installation

It is recommended to install **MakeTNT** together with other beta-release software from Global Phasing (see README.GPhL_beta_install for details. If you want to install **MakeTNT** separately, please follow the steps below.

1. create a new directory to install the software in (e.g. `/public/xtal/MakeTNT`) and go there
2. download all necessary files into that directory:
 - `install_MakeTNT.sh` (installation script)
 - `MakeTNT.tar.gz` (system-independent files)
 - `MakeTNT.<HWSP>.tar.gz` (system-dependent files, where `<HWSP>` is one of the supported platforms, e.g. linux, alpha or irix)
3. make the installation script executable:

```
% chmod +x ./install_MakeTNT.sh
```

4. run the installation script:

```
% ./install_MakeTNT.sh
```

This should check that all files are present and that the installation succeeds. Furthermore, it will create two files (`setup.sh` and `setup.csh`) that can be used by users to make the program available

to their environment: please read the message printed at the end of the installation process carefully!

Finally, the provided examples will be run - to test the installation.

Please note that, in order to run the program, you will need a **valid licence key** for each machine (stored in the file `.licence` in the **MakeTNT** installation directory). By default, the installation script will copy over the file `$BDG_home/.licence` if it is found. Otherwise you need to make sure to copy a working `.licence` file into place yourself.

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 15.02.07

MakeTNT

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Contents

- General Information
- Requirements and Installation
- Command Line Switches and Arguments
- Conversion Mechanism
- Output
- Examples
- Evaluation

General Information

MakeTNT is a Java-based command line tool for converting restraints stored in Refmac and XPLOR format to TNT-restraints. From Release 1.5.1 on, it has the options to

- generate REFMAC restraints instead of TNT-restraints
- convert TNT-restraints into Refmac-format

Requirements and Installation

MakeTNT is part of the MakeTNT-toolkit: please check the following links for information on Requirements and Installation.

Command Line Switches and Arguments

- **Usage of MakeTNT:**

```
% MakeTNT -source <program> \
-top <topology-file> -par <parameter-file> \
[-o <filename>] [-n <TNT tag>] \
[-size <length of output>] [-noB] \
[-ccp4][-Odict][-ODB <O-database>][-PDBTOR <pdb-file>]
```

The following table will show more details on how to

- specify input files to MakeTNT
- specify the name (3-letter code) of the ligand in the restraint geometry cards
- specify the name of the output dictionary file

TNT-geometry files generally have the extension ".dic"

Switch	Argument	Description	Possible Values	Compulsory	Note
-source	<program>	source of restraint information to be converted to TNT-format	XPLOR_CORINA, REFMAC, XPLOR_XPLO2D, TNT	yes	
-top	<topology-file>	file defining ligand topology	see: Example Section	yes : XPLOR no : REFMAC	see also: -source
-par	<parameter-file>	file listing ligand restraints	see: Example Section	yes	see also: -source
-n	<tag>	3-(capital)-letter residue name used in TNT-dictionary-file	e.g.: ATP,GMP, ANY,THI,NGY,OUL,IKE	no	default = UNK Use your own!
-o	<filename>	unique filename for TNT-dictionaries	e.g.: kinase_inhibitor986B, ...	no	default = 3-letter code defined by -n Choose something unique!
-size	<length-of-output>	defines the overall size of the resulting dictionary	s (short) m (medium) l (long)	no	default : s (small) see also : Conversion Mechanism
-noB		flag to suppress default BCORREL restraints		no	
-ccp4		flag to put Refmac - restraints instead of TNT-restraints		no	
-Odict		generate O-dictionaries instead of TNT-restraints		no	
-ODB	<O-database>	append non-redundant O-dictionaries to existing O-database		no	only relevant to the generation of O-dictionaries
-PDBTOR	PDB-file	derive value of planar torsions (0/180°) from PDB-file		no	only relevant to the generation of O-dictionaries
-PDBREF	PDB-file	derive element symbols from PDB-file		no	only relevant to the conversion: TNT-dictionary -> *cif-dictionary for Coot

Conversion Mechanism:

This is how **MakeTNT** will generate the various TNT specific geometry definitions (see also the TNT manual)

- **bond and angle restraints:** --> BOND and ANGLE
- **torsion restraints:**
 - planar: --> PLANE (can be controlled by **-size** command line argument)
 - non-planar: ignored
- **improper torsion restraints:**
 - planar: --> TRIGONAL
 - non-planar: --> IMPROPER (order of atoms changed from CABD to ABCD)
- **Libcheck/Refmac plane restraints:** --> PLANE(independent of "-size" command line argument)

Libcheck/Refmac uses definitions similar to **s** (small) and not a complete description like CORINA. In this case **s** and **l** options will result in a very similar output. Planar ring recognition (when using **m**) is not possible. Therefore, for size **m** a size **l** dictionary file is produced.

- **Libcheck/Refmac "chiral volume" restraints:**
 - if the Refmac dictionary-file includes xyz cartesian coordinates: --> IMPROPER calculated
 - direct conversion from chiral volumes can be added on request

Therefore, the different options for the **-size** argument will give:

- **s** (small):

Overlapping planes will give one single TNT-PLANE restraint: this is recommended for visualisation.

- **m** (medium):

Planar 5/6-membered rings give one PLANE restraint each (for Refmac input, size **l** is generated instead).

- **l** (long):

Restraints are generated as specified in input.

Output:

Using PDB2TNT in its default mode, the following output files are written into the working directory:

- TNT-dictionary file <filename>.dic (e.g. staurosporine.dic)

- for Libcheck/Refmac-input including coordinates: PDB-file

With the -CCP4 flag, you will obtain:

- CCP4-library file (e.g. LIG.GPhL.cif)

If you choose the -Odict option, you will produce:

- dictionary file: e.g.: LIG_O_refi_dict.txt
- dictionary header-file: TXL_O_refi_dict.hd (.bond_angles count line)

If you choose -Odict, -ODB <O-database>:

- the new dictionary will be appended to the existing O-database
- if the database does not yet include an entry for a residue of the 3-letter code specified

Submitting a PDB-file (-PDBTOR <pdb-file>) in addition to the dictionary file will enable

- to specify planar torsions in O-dictionaries (0/180°).
 - The PDB-file is only used to derive the direction of planar torsions. Everything else is derived from the restraint files.
-

Examples:

Example input files are included in the `samples/maketnt` subdirectory of your `MakeTNT-toolkit` distribution. Also included are TNT-dictionary files converted by **MakeTNT**. To test the commands listed below, enter the respective subdirectory and run the following commands:

- **CORINA:**

- run the command

```
% MakeTNT -source XPLOE_CORINA \  
-top nsc628503.top -par nsc628503.par \  
-n TXL -o taxotere
```

- for atom numbers > 100 CORINA atom names consist of the chemical element only
 - for those atoms **MakeTNT** creates numbers based on two digits and 26-symbols (0+A-F,H-Z), each , e.g.
`100 = 00; 101 = 0A, 102 = 0B, 126 = A0, ...`
 - to label your PDB-file in the same way, please use **MakePDB** (a small additional application constructed for this purpose):

```
% MakePDB -source PDB -input nsc628503.pdb
```

In this example, H-atoms are included for demonstration purposes. At the moment, hydrogen atoms are not used during refinement with BUSTER-TNT and should be removed from the coordinate files.

- **Libcheck/Refmac:**

- please check MakeTNT's current format requirements for Refmac restraint dictionaries
- run the command

```
% MakeTNT -source REFMAC \  
-par ATP.lib \  
-n ATP -o adenosinetriphosphate
```

- **XPLOR:**

- run the command

```
% MakeTNT -source XPLOR_XPLO2D \  
-top brl_xplor_top.txt -par brl_xplor_par.txt \  
-n BRL -size s -o rosiglitazone
```

- **XPLOR to Refmac:**

- run the command

```
% MakeTNT -source XPLOR_XPLO2D \  
-top brl_xplor_top.txt -par brl_xplor_par.txt \  
-n BRL -size s -o rosiglitazone -ccp4
```

- **TNT to CCP4:**

- go back to the CORINA-directory and run the command

```
MakeTNT -source TNT -par taxotere.dic -ccp4 -n TXL -PDBREF  
TXL_tnt.pdb
```

- **TNT to O**

- go back to the PDB2TNT-directory and create a local version of the O-library a

```
cp $ODAT/stero_chem.odb .
```

then run the command:

```
MakeTNT -source TNT -par taxotere.dic -n TXL -Odict -ODB
stereo_chem.odb
```

In the above run, planar torsions for your dictionary have been read from the comments in taxotere.dic (NOTE - tag).

- If your TNT-dictionary file does not have planar torsions included as comments (any TNT-dictionary file produced by MakeTNT, or by the other tools prior to version 1.9.2), you can specify a PDB-file which will be used to assign the directions of planar torsions, please use:

```
MakeTNT -source TNT -par taxotere.dic -n TXL -Odict -ODB
stereo_chem.odb -PDBTOR TXL_tnt.pdb
```

If a PDB-file is submitted by -PDBTOR, commented planar torsions in the TNT-dictionary file will be ignored.

- A few words of caution:
 - All input available to MakeTNT - tools can be converted into O-dictionaries, yet O has problems processing residues with more than 100 atoms (as in the H-atom including taxotere structures in this documentation)
 - Upon loading a PDB-structure, by default O would omit H-atom positions. O-dictionaries at the moment do include H-atom positions, thus hydrogens need to be included when loading a ligand into O.

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modified: Fri Jul 17 16:06:00 BST 2009

MakeTNT Test Runs

Evaluation : Generation of TNT-Restraints by MakeTNT

No. Comp.	Restraint Type	Restraint Source	Successful Conversion	Conversion Failed	Reason for Failure	Remarks
49967	XPLOR_CORINA	NCI open source database (249071 3D structures) / Corina ^[1]	49967	-	-	-
2426	REFMAC	Refmac standard library ^[2]	2426	-	-	-
3289	XPLOR_XPLO2D	NCI open source database / xplo2d ^[3]	3288	1	input: 2 molecules	[4]
1000	REFMAC (written by Corina)	NCI open source database / Corina ^[5]	1000	-	-	-

- [1] From the first 50000 compounds of the NCI open source database, **49967** structures with 3D coordinates have been selected. 10 of the structures have kindly been provided by Dr. Armin Ruf, Roche, Basel. XPLOR-type *.top and *.par files have been obtained from Corina.
- [2] All 2457 cif monomer descriptions (ccp4-4.2.2) have been run through Libcheck, Vers 4.0.1. **2426** lib-files could be obtained. For 31 files Libcheck failed.
- [3] From the first 20000 compounds of the NCI open source database, **3289** compounds have been selected if they had 3D coordinates and fulfilled at least one of the following criteria: > 1 4-atom ring, > 2 5-atom rings, > 2 6-atom rings, > 4 (4-6)-atom rings.
- [4] Compounds including metal atoms can cause trouble, as metal coordination is treated as a covalent bond by xplo2d, which increases the number of ring systems to be processed by MakeTNT. In order to cope with this problem, MakeTNT has been configured to process generous numbers of ring systems. In case of problems, please make sure, that the metal coordination listed in the *.top files generated by xplo2d is correct. If it still does not work, let us know.
- [5] The test was run on Refmac-dictionaries generated by Corina for the first 1000 coordinates of the NCI-dataset.

Evaluation : Small Molecule Refinements Using Corina-Derived Restraints Converted to TNT-Format by MakeTNT

Each cell in the table below represents one optimisation run, using the TNT-GEOM module without X-ray terms and a fast optimization routine kindly provided by Dr. Oliver Smart, Globalphasing. Corina generated coordinates have been randomized to a RMS deviations of 0.25, 0.5 and 1.0 Ångströms and optimized with TNT-dictionaries, sizes s, m and l. Different coordinates have been generated for each run. Each resulting structure has been manually inspected for integrity. Optimizations producing apparent errors in the geometry have been marked in red. The results indicate that:

- Corina-generated XPLOR-restraints converted by MakeTNT reproduce correct structures, even after severe distortion.
- Clustering of planar restraints (restraint size s(default), m) improves optimization.
- -size "s"(default) produces the best results, especially if the starting model is very distorted.
- -size "l" should not be used for restraints derived from XPLOR-type formats.

compound number of atoms	structure	optimisation: jiggle[Å]/Size of MakeTNT-Restraint file no. refinement cycles/successful?								
		0.25/s	0.25/m	0.25/l	0.5/s	0.5/m	0.5/l	1.0/s	1.0/m	1.0/l
NCI124652/28	[IMAGE]	126 yes ^[a]	57 yes ^[a]	74 yes ^[a]	49 yes ^[a]	45 yes ^[a]	167 yes ^[a]	73 yes ^[a]	108 yes ^[a]	55 no
NIST04051410392D/10	[IMAGE]	16 yes	36 yes	17 yes	17 yes	81 yes	14 yes	36 yes	32 yes	29 yes
NCI16498/25	[IMAGE]	13 yes	16 yes	184 yes	17 yes	25 yes	86 yes	60 yes	88 yes	114 yes
NCI367798/48	[IMAGE]	116 yes	231 yes	494 yes	284 yes	219 yes	>2001 no	98 yes	416 no	415 no
NCI376722/29	[IMAGE]	26 yes ^[a]	21 yes ^[a]	144 yes ^[a]	18 yes ^[a]	26 yes ^[a]	62 yes ^[a]	158 yes ^[a]	205 yes ^[a]	415 no
NCI46/20	[IMAGE]	46 yes	21 yes	21 yes	59 yes	117 yes	215 yes	72 yes	136 yes	243 yes
NCI56940/28	[IMAGE]	21 yes	134 yes	173 yes	21 yes	74 yes	173 yes	73 yes	258 yes	115 yes
NCI628503/58	[IMAGE]	147 yes	127 yes	140 yes	190 yes	151 yes	396 no	238 yes	223 no	624 no
NCI631303/29	[IMAGE]	21 yes	46 yes	88 yes	199 yes	25 yes	189 no	35 yes	131 yes	303 no
NCI640266/26	[IMAGE]	61 yes	72 yes	157 yes	31 yes	75 yes	125 yes	112 yes	127 yes	266 yes

[a]: structure of good quality, except for positioning of metal ions(e.g. Cu, Co), where currently no parameters for non-bonded interactions are available in Buster/TNT

Evaluation : Small Molecule Refinements Using Refmac-Derived Restraints Converted to TNT-Format by MakeTNT

Each cell in the table below represents one optimisation run, using the TNT-GEOM module as described in the previous section. 10 structures have been chosen from the Refmac (ccp4-5.0.2) standard library. Starting coordinates and library description files have been obtained by Libcheck (Vers 4.1.3). The coordinates have been randomized to RMS deviations of 0.25, 0.5 and 1.0 Ångströms and optimized with TNT-dictionaries, sizes s and l. Different coordinates have been generated for each run. Each resulting structure has been manually inspected for integrity. Optimizations producing apparent errors in the geometry have been marked in red. We observed that:

- Refmac restraints not as good Corina restraints.
- The two dictionary sizes l,s perform similarly well.
- For the steroid thrombin-inhibitor compound (GR3), the problems are due to a the way planarity restraints are defined for the double bond in the six-membered ring. In this case, the problem can be spotted by manual inspection of the 3D coordinates generated by Refmac.

compound number of atoms	structure	optimisation: jiggle [Å] / Size of MakeTNT-Restraint file no. refinement cycles / successful?					
		0.25/s	0.25/l	0.5/s	0.5/l	1.0/s	1.0/l
AQS 28	[IMAGE]	16 yes	13 yes	30 yes	40 yes	30 yes	46 yes
BRL 25	[IMAGE]	28 yes	55 yes	36 yes	34 yes	45 yes	36 yes
DPS 36	[IMAGE]	55 yes	93 yes	24 yes	106 yes	100 yes	291 yes
EST 20	[IMAGE]	36 yes	24 yes	53 yes	29 yes	90 no	86 yes
GR3 39	[IMAGE]	73 no	96 no	73 no	294 no	169 no	529 no
MTB 22	[IMAGE]	no atom labels in Refmac input (e.g.: 'C5') currently not recognized by TNT					
NEV 20	[IMAGE]	80 yes	34 yes	102 yes	71 yes	106 yes	29 yes
TMF 33	[IMAGE]	40 yes	105 yes	72 no	56 yes	99 yes	136 no
TXL 58	[IMAGE]	225 yes	257 yes	206 yes	155 no	395 yes	273 no
ZYA 28	[IMAGE]	56 yes	22 yes	95 yes	44 yes	61 yes	71 yes

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 13.12.05

PDB2TNT

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Contents

- General Information
 - Requirements and Installation
 - Command Line Switches and Arguments
 - Conversion Mechanism
 - Output
 - Examples
-

General Information

PDB2TNT is a Java-based command line tool to generate TNT-restraints from a PDB-file. In more recent versions, PDB2TNT can also be used to generate dictionaries for Refmac and O.

Requirements and Installation

PDB2TNT is part of the MakeTNT-toolkit: please check the following links for information on Requirements and Installation.

Command Line Switches and Arguments

- Usage of PDB2TNT:

```
% PDB2TNT -i <input-file> [-n <tag>] [-o <filename>] \  
[-noB] [-FLEXTOR] [-size <length of output>] \  
[-HPOS] [-NOMDL] [-ccp4] [-ccp4_tnt] [-Odict] [-ODB  
<O-database>] [-OnoH] [-notor] [-TightRestrains]
```

Switch	Argument	Description	Possible Values	Compulsory	Note
-i	<input-file>	name of PDB-file submitted	no format restrictions, any extension	yes	
-n	<tag>	3-Letter code to be used in TNT-dictionary file	e.g.: ATP,GMP,ANY,THI,NGY,OUL,IKE	no	default = 3-letter ID from PDB-file if absent: UNK Use your own!
-o	<filename>	unique filename for TNT-dictionaries	e.g.: kinase_inhibitor986B, ...	no	default = 3-letter code defined by -outname Choose something unique!
-noB		flag to suppress default BCORREL restraints		no	
-FLEXTOR		output of default parameters for non-planar torsion angles		no	
-size	<length of output>	defines the overall size of the resulting dictionary	s (short) m (medium) l (long)	no	default : m (medium) see also : Conversion Mechanism
-HPOS		Switch to ignore H-atom positions		no	Not recommended ! Use only if H-atoms are unreliable
-NOMDL		Switch off generation of pseudo-MDL file		no	Recommended if the quality of your PDB-file is good
-ccp4		flag to generate CCP4-restraints instead of TNT-restraints		no	
-ccp4_tnt		flag to generate CCP4-restraints AND TNT-restraints		no	
-Odict		flag to generate dictionaries for O instead of TNT-restraints		no	
-ODB	<O-database>	append non-redundant O-dictionaries to existing O-database		no	only relevant to the generation of O-dictionaries
-OnoH		exclude hydrogen atoms from O-dictionary entries		no	only relevant to the generation of O-dictionaries
-notor		flag to suppress torsion restraints for non-planar groups in O-dictionaries		no	only relevant in combination with -Odict
-TightRestrains		sigma for bond lengths: 0.020Å, sigma for bond angles: 2.0deg		no	default: 0.040Å, 5.0deg

Conversion Mechanism:

Atom-specific covalent radii are used to convert the PDB-file into a graph, which is the basis for deriving TNT - Restraints. Target values for restraints are taken from the 3D-coordinates. If present the positions of H-atoms are used to improve the quality of the restraints (this can be suppressed by the -HPOS option). As Buster-TNT presently does not support the refinement of hydrogen atoms, they are removed from the final restraint file.

If an atom has three neighbours bonded to it, an improper torsion is measured between those four atoms and if its value is close to 0 or 180 degrees a TRIGONAL restraint is created, otherwise an IMPROPER torsion restraint is generated. IMPROPER torsion restraints are also generated for atoms surrounded by four (or more) neighbours.

Planar torsion restraints are put out if two bonded atoms (of elements with double/triple bond potential) are in planar coordination (based on their bond angles) and all torsion angles around this bond are planar as well. If the bonded atoms are in planar conformation, but at least one torsion angle is not, no restraints are not formed and a message to the user is written instead.

The different options for the **-size** argument are:

- **s** (small):

Overlapping planes will produce one single TNT-PLANE restraints; produces large, rigid planes; recommended for visualization; if necessary planes can be broken down to smaller fragments using size m(default);

- **m** (medium, default):

Planar 5/6-membered rings give one PLANE restraint each. Recommended for large planar systems and in structures of high resolution.

- **l** (long):

Each planar torsion angle will result in a separate PLANE restraint.

Output:

Using PDB2TNT in its default mode, the following output files are written into the working directory:

- TNT-dictionary file: `<.dic` (e.g. `taxotere.dic`)
- if original PDB-file included hydrogens: PDB-file without H-pos for refinement: `<tag>_tnt.pdb` (e.g. `TXL_tnt.pdb`)
- pseudo-MDL-file: `<filename>.fromPDB.mdl` (e.g. `taxotere.fromPDB.mdl`)
 - bonds as perceived by PDB2TNT
 - non-terminal bonds perceived as rigid are represented as triple bonds
 - can be visualized in rasmol: **rasmol -mdl taxotere.fromPDB.mdl**
 - suitable for processing by MDL2TNT
 - can be suppressed by -NOMDL option

With the `-CCP4` flag, two files are generated:

- dictionary file for Refmac (e.g: `taxotere.GPhL.cif`)
- pseudo-MDL-file: `<filename>.fromPDB.mdl` (e.g. `taxotere.fromPDB.mdl`)

If you choose the `-Odict` option, you will obtain:

- dictionary file: e.g.: `TXL_O_refi_dict.txt`
- dictionary header-file: `TXL_O_refi_dict.hd` (.bond_angles count line)
- pseudo-MDL-file: `<filename>.fromPDB.mdl` (e.g. `taxotere.fromPDB.mdl`)

If you choose `-Odict, -ODB <O-database>`:

- the new dictionary will be appended to the existing O-database
 - if the database does not yet include an entry for a residue of the 3-letter code specified
-

Example:

Example input files (`TXL.pdb_tnt`, `TXL.pdb`) can be found in the `samples/maketnt/PDB2TNT` subdirectory of your MakeTNT-toolkit installation.

- In the easiest case (as in `taxotere_tnt.pdb`), all atoms have unique labels and hydrogen atoms are removed.

To produce a TNT-dictionary file **taxotere.dic**, run:

```
% PDB2TNT -i TXL_tnt.pdb -n TXL -o taxotere
```

- If PDB2TNT detects hydrogen atoms and/or non-unique atom naming (example: `TXL.pdb`), a new PDB-file `taxotere_tnt.pdb` is generated. If present, multiple conformations are removed based on their occupancies.

To test this functionality, run **PDB2TNT** on `TXL.pdb` :

```
% PDB2TNT -i TXL.pdb -n TXL -o taxotere
```

Please use `taxotere_tnt.pdb` for further refinement in BUSTER-TNT.

- To produce a Refmac-library file, please use:

```
% PDB2TNT -i TXL_tnt.pdb -n TXL -o taxotere -ccp4
```

- To add an entry to the O-library:

```
% PDB2TNT -i TXL_tnt.pdb -n TXL -Odict -ODB stereo_chem.odb
```

or

```
% PDB2TNT -i TXL_tnt.pdb -n TXL -Odict -ODB stereo_chem.odb  
-notor
```

if you do not wish to restrain flexible torsions

Please also check the messages provided by **PDB2TNT**.

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 02.02.07

MDL2TNT

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Contents

- General Information
 - Requirements and Installation
 - Command Line Switches and Arguments
 - Conversion Mechanism
 - Output
 - Examples
-

General Information

MDL2TNT is a Java-based command line tool to generate TNT-restraints from a 3D-MDL-file.

Requirements and Installation

MDL2TNT is the part of the MakeTNT-toolkit: please check the following links for information on Requirements and Installation.

Command Line Switches and Arguments

- Usage of MDL2TNT:

```
% MDL2TNT -i <input-file> [-n <tag>] \  
[-o <filename>] [-FLEXTOR] [-noB] \  
[-size <length of output>] \  
[-normBO][-HPOS][-ccp4][-ccp4_tnt][-Odict]\  
[-Odict][-ODB <O-database>][-OnoH][-notor][-TightRestrains]
```

Switch	Argument	Description	Possible Values	Compulsory	Note
-i	<mdl-file>	name of MDL-file submitted	no format restrictions, any extension	yes	
-n	<tag>	<3-Letter code to be used in TNT-dictionary file>	e.g.: ATP,GMP,ANY,THI,NGY,OUL,IKE	no	default = UNK Use your own!
-o	<filename>	unique filename for TNT-dictionaries	e.g.: kinase_inhibitor986B, ...	no	default = 3-letter code defined by -outname Choose something unique!
-noB		flag to suppress BCORREL restraints		no	
-FLEXTOR		output of default parameters for non-planar torsion angles		no	
-size	<length-of-output>	defines the overall size of the resulting dictionary	s (short) m (medium) l (long)	no	default : s (small) see also : Conversion Mechanism
-HPOS		Switch to ignore H-atom positions		no	Not recommended ! Use only if H-atoms are unreliable
-normBO		Assume normalised bond orders		no	Recommended if your file has normalised bond orders
-ccp4		flag to generate CCP4-restraints instead of TNT-restraints		no	
-ccp4_tnt		flag to generate CCP4-restraints AND TNT-restraints		no	
-Odict		flag to generate dictionaries for O instead of TNT-restraints		no	
-ODB	<O-database>	append non-redundant O-dictionaries to existing O-database		no	only relevant to the generation of O-dictionaries
-OnoH		exclude hydrogen atoms from O-dictionary entries		no	only relevant to the generation of O-dictionaries
-notor		flag to suppress torsion restraints for non-planar groups in O-dictionaries		no	only relevant in combination with -Odict
-TightRestrains		sigma for bond lengths: 0.020Å, sigma for bond angles: 2.0deg		no	default: 0.040Å, 5.0deg

Conversion Mechanism:

Restraint atoms are selected on the basis of a molecular graph generated from the bond-list of the MDL-file. Target values for restraints are taken from the 3D-coordinates. If present the positions of H-atoms are used to improve the quality of the restraints (this can be suppressed by the -HPOS option). As Buster-TNT presently does not support the refinement of hydrogen atoms, they are removed from the final restraint file.

If an atom has three neighbours bonded to it, an improper torsion is measured between those four atoms and if its value is close to 0 or 180 degrees a TRIGONAL restraint is generated, otherwise an IMPROPER torsion restraint is generated. IMPROPER torsion restraints are also generated for atoms surrounded by four (or more) neighbours.

Recognition of planar torsion restraints is mainly based on connectivity information, yet needs to be confirmed by an approximately planar constellation in the 3D-coordinates. If connectivity and coordinate information contradict each other (e.g. steric hindrances in large conjugated systems), restraints are not formed and a message to the user is written instead. Planarity recognition works best, if bond orders in your input file are normalised and you specify -normBO as an input flag.

The different options for the **-size** argument are:

- **s** (small, default):

Overlapping planes will result in one single TNT-PLANE restraint.

- **m** (medium):

Planar 5/6-membered rings result in one PLANE restraint each. Recommended for large planar systems and in structures of high resolution.

- **l** (long):

Each planar torsion angle will result in a separate PLANE restraint.

Output:

Using MDL2TNT in its default mode, the following output files are written into the working directory:

- TNT-dictionary file: <filename>.dic (e.g. ligand.dic)
- PDB-file without H-pos for refinement: <tag>.pdb (e.g. LIG_tnt.pdb)
- if -HPDB is set: PDB-file including original H-pos for refinement: <tag>.pdb (e.g. LIG_H_tnt.pdb)

With the -CCP4 flag, two files are generated:

- dictionary file for Refmac (e.g. ligand.GPhL.cif)

- PDB-file (with H-pos): LIG_tnt.pdb

If you choose the -Odict option, you will obtain:

- dictionary file: e.g.: ligand_O_refi_dict.txt
- dictionary header-file: ligand_O_refi_dict.hd (.bond_angles count line)
- PDB-file: LIG_tnt.pdb

If you choose -Odict, -ODB <O-database>

- the new dictionary will be appended to the existing O-database
 - if the database does not yet include an entry for a residue of the 3-letter code specified
-

Example:

Example input files (mdl824.mol, TXL.mol) can be found in: `samples/maketnt/MDL2TNT`.

Please note that TXL.mol has too many atoms/residue for O processing.

- To produce a TNT-dictionary file for mdl824.mol, please run:

```
% MDL2TNT -i mdl824.mol -n LIG -o ligand
```

- For Refmac-restraints, run:

```
% MDL2TNT -i mdl824.mol -n LIG -o ligand -CCP4
```

- To add an entry to the O-library:

```
% MDL2TNT -i mdl824.mol -n LIG -Odict -ODB stereo_chem.odb
```

or

```
% MDL2TNT -i mdl824.mol -n LIG -Odict -ODB stereo_chem.odb  
-notor
```

if you do not wish to restrain flexible torsions

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 02.02.07

Comparative Test Runs for MDL2TNT and PDB2TNT

Evaluation : Generation of Restraints by MDL2TNT

Source of compounds	Number of compounds	Successful Generation of Restraints	Failures
NCI diversity data set	1990	1990	-

Evaluation : Accuracy of BOND recognition by PDB2TNT

The bond recognition algorithm of PDB2TNT has been tested using 3D-coordinates (but not the bond table !!) from the NCI diversity data set and bond-restraints generated this way have been compared to the ones obtained by MDL2TNT (with explicit knowledge of bond orders). Results are shown below:

Source of compounds	Number of compounds	Number of structures with fully correct bond restraints	Number of structures with missing bonds	Number of structures with too many bonds	Reasons for additional bonds
NCI diversity data set	1990	1970	-	20	<ul style="list-style-type: none"> ● bumps in input structure (12 compounds) ● too large tolerance for bond distances in metal complexes (8 compounds)

Evaluation : Accuracy of PLANE Restraints by PDB2TNT

Recognition of planar restraints by PDB2TNT has been tested by comparing planar restraints (format:s) obtained by its hybridisation-determining algorithm to the ones obtained from MDL2TNT. 2 test runs have been performed: one including the positions of H-atoms and one without knowledge of H-atom positions. Manual inspection of discrepancies between MDL-file and PDB-file derived restraints indicated that:

- planar restraints derived from MDL-files were always correct and thus preferable
- apart from 1 case, planar restraints derived from PDB-files had only minor errors and would still lead to accurate refinement
- inclusion of H-atoms reduces the errors in PDB-file derived restraints by more than 50%

Source of compounds	Number of compounds	Inclusion of H-positions	Differences between MDL- and PDB-derived restraints	Reason for differences:	
NCI diversity data set 3D-MDL-files	1990	yes	34	<ul style="list-style-type: none"> ● 24 cases: planar bond in non-planar 5-membered ring ● rest: double bonded sulfur, metal coordination etc. 	
NCI diversity data set 3D-MDL-files	1990	no	83		<ul style="list-style-type: none"> ● 41 cases: planar bond in non-planar 5-membered ring ● 29 cases: planar bond not recognised because of strains in bond angles ● rest: double bonded sulfur, metal coordination etc.

Evaluation : Small Molecule Refinements Using restraints Generated by MDL2TNT and PDB2TNT

Each cell in the table below represents one optimisation run, using the TNT-GEOM module without X-ray terms and a fast optimization routine kindly provided by Dr. Oliver Smart, Globalphasing.

Initial coordinates (with the correct stereochemistry around asymmetric centers) have been randomized to RMS deviations of 0.5 and 1.0 Ångströms and optimized with TNT-dictionaries, sizes m and l, derived either from a 3D-mdl file (M) or a PDB-file (P). For example, 0.5/m/M means, the compound has been distorted to an initial RMSD of 0.5 Ångströms, optimized with a TNT-dictionary file of size m derived from a 3D-MDL file.

Optimizations producing apparent errors in the geometry have been marked in red. The results indicate that:

- **Reliable restraints can be deduced from a given 3D-conformation.**
- **Restraints generated by PDB2TNT and MDL2TNT perform similar to MakeTNT-restraints**
- **The "m" dictionary format for PLANAR restraints seems to be the most robust one.**

compound number of atoms	structure	optimisation: jiggle[Å]/Size of -Restraint file/Source M:MDL2TNT; P:PDB2TNT no. refinement cycles/successful?							
		0.5/m/M	0.5/s/M	0.5/m/P	0.5/s/P	1.0/m/M	1.0/s/M	1.0/m/P	1.0/s/P
NCI124652/28	[IMAGE]	98 yes ^[a]	116 yes ^[a]	83 yes ^[a]	40 yes ^[a]	91 yes ^[a]	101 no	98 yes ^[a]	71 no
NIST04051410392D/10	[IMAGE]	36 yes	36 yes	45 yes	45 yes	50 yes	50 yes	106 yes	106 yes
NCI16498/25	[IMAGE]	44 yes	79 yes	56 yes	46 yes	80 yes	71 yes	79 yes	48 yes
NCI367798/48	[IMAGE]	120 yes	267 yes	115 yes	87 yes	115 yes	215 yes	91 yes	113 yes
NCI376722/29	[IMAGE]	96 yes ^[a]	98 yes ^[a]	75 yes ^[a]	58 yes ^[a]	101 yes ^[a]	156 yes ^[a]	102 yes ^[a]	91 yes ^[a]
NCI46/20	[IMAGE]	65 yes	107 yes	77 yes	92 yes	92 yes	148 no	103 yes	91 no
NCI56940/28	[IMAGE]	84 yes	98 yes	87 yes	63 yes	105 yes	116 yes	124 yes	104 no
NCI628503/58	[IMAGE]	104 yes	118 yes	81 yes	101 yes	109 yes	1091 no	115 yes	323 no
NCI631303/29	[IMAGE]	80 yes	154 yes	89 yes	78 yes	76 yes	91 yes	96 yes	113 yes
NCI640266/26	[IMAGE]	152 yes	78 yes	154 yes	57 yes	132 yes	73 yes	158 yes	79 yes

[a]: structure of good quality, except for positioning of metal ions(e.g. Cu, Co), where currently no parameters for non-bonded interactions are available in Buster/TNT

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 13.12.05

CIF2TNT

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Contents

- General Information
 - Requirements and Installation
 - Command Line Switches and Arguments
 - Conversion Mechanism
 - Output
 - Examples
 - Evaluation
-

General Information

CIF2TNT is a Java-based command line tool to generate TNT-restraints from a 3D-CIF-file. **Refmac-library files** can be converted by MakeTNT.

Requirements and Installation

CIF2TNT is part of the MakeTNT-toolkit: please check the following links for information on Requirements and Installation.

Command Line Switches and Arguments

- Usage of CIF2TNT:

```
% CIF2TNT -i <input-file> [-n <tag>]\
[-o <filename>] [-FLEXTOR] [-noB] \
[-size <length of output>]\
[-ccp4][ccp4_tnt][-Odict][-ODB <O-database>][-notor]
```

Switch	Argument	Description	Possible Values	Compulsory	Note
-i	<mdl-file>	name of CIF-file submitted	no format restrictions, any extension	yes	
-n	<tag>	<3-Letter code to be used in TNT-dictionary file>	e.g.: ATP,GMP,ANY,THI,NGY,OUL,IKE	no	default = UNK Use your own!
-o	<filename>	unique filename for TNT-dictionaries	e.g.: kinase_inhibitor986B, ...	no	default = 3-letter code defined by -outname Choose something unique!
-FLEXTOR		output of default parameters for non-planar torsion angles		no	
-noB		flag to suppress default BCORREL restraints		no	
-size	<length-of-output>	defines the overall size of the resulting dictionary	(short) m (medium) l (long)	no	default : s (small) see also : Conversion Mechanism
-ccp4		flag to generate CCP4-restraints instead of TNT-restraints		no	
-ccp4_tnt		flag to generate CCP4-restraints AND TNT-restraints		no	
-Odict		flag to generate dictionaries for O instead of TNT-restraints		no	
-ODB	<O-database>	append non-redundant O-dictionaries to existing O-database		no	only relevant to the generation of O-dictionaries
-notor		flag to suppress torsion restraints for non-planar groups in O-dictionaries		no	only relevant in combination with -Odict

Conversion Mechanism:

Restraint atoms are selected on the basis of a molecular graph generated from the bond-list of the CIF-file. Target values for restraints are taken from the 3D-coordinates. If present the positions of H-atoms are used to improve the quality of the restraints. As Buster-TNT presently does not support the refinement of hydrogen atoms, they are removed from the final restraint file.

If an atom has three neighbours bond to it, an improper torsion is measured between those four atoms and if its value is close to 0 or 180 degrees a TRIGONAL restraint is generated, otherwise an IMPROPER torsion restraint is generated. IMPROPER torsion restraints are also generated for atoms surrounded by four (or more) neighbours.

Recognition of planar torsion restraints is mainly based on connectivity information, yet needs to be confirmed by an approximately planar constellation in the 3D-coordinates. If connectivity and coordinate information contradict each other (e.g. steric hindrances in large conjugated systems), restraints are not formed and a message to the user is written instead.

The different options for the **-size** argument are:

- **s** (small, default):

Overlapping planes will result in one single TNT-PLANE restraint.

- **m** (medium):

Planar 5/6-membered rings result in one PLANE restraint each. Recommended for large planar systems and in structures of high resolution.

- **l** (long):

Each planar torsion angle will result in a separate PLANE restraint.

Output:

Using CIF2TNT in its default mode, following output files are written into the working directory:

- TNT-dictionary file: <filename>.dic (e.g. taxotere.dic)
- PDB-file without H-pos for refinement: <tag>_tnt.pdb (e.g. TXL_tnt.pdb)

With the **-CCP4** flag, two files are generated:

- dictionary file for Refmac (e.g: taxotere.GPhL.cif)
- PDB-file: <filename>.fromPDB.mdl (e.g. TXL_tnt.pdb)

With the **-Odict** option, you will obtain:

- dictionary file: e.g.: TXL_O_refi_dict.txt
- dictionary header-file: TXL_O_refi_dict.hd (.bond_angles count line)
- PDB-file: TXL_tnt.pdb

If you choose **-Odict**, **-ODB** <O-database>:

- the new dictionary will be appended to the existing O-database
 - if the database does not yet include an entry for a residue of the 3-letter code specified
-

Example:

An example input file (TXL.cif) can be found in: `samples/maketnt/CIF2TNT`.

- To generate a TNT-dictionary, use the command:

```
% CIF2TNT -i TXL.cif -n TXL -o taxotere
```

- To produce a Refmac-library file, please use:

```
% CIF2TNT -i TXL.cif -n TXL -o taxotere -ccp4
```

- To add an entry to the O-library:

```
% CIF2TNT -i TXL.cif -n TXL -Odict -ODB stereo_chem.odb
```

or

```
% CIF2TNT -i TXL.cif -n TXL -Odict -ODB stereo_chem.odb -notor
```

if you do not wish to restrain flexible torsions

Evaluation:

10 example 3D-CIF-files with idealised (except: BCD) coordinates (3-letter codes: 638, 858, BCD, DAQ, KAB, MTC, PMY, RAP, RR6, STU) have been downloaded from the MSDchem-database.

CIF2TNT has been used to obtain geometry restraints for these molecules.

The restraints have been tested to re-optimize (Gelly, O.Smart, GlobalPhasing) experimental coordinates from the Hicup-database after an additional distortion of 0.25Å. Manual inspection showed consistent optimization for all 10 test-cases (data available on request).

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modified: Fri May 15 12:07:21 BST 2009

MOL22TNT

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Contents

- General Information
 - Requirements and Installation
 - Command Line Switches and Arguments
 - Conversion Mechanism
 - Output
 - Examples
 - Evaluation
-

General Information

MOL22TNT is a Java-based command line tool to generate TNT-restraints from a Tripos *.mol2 file.

Requirements and Installation

MOL22TNT is part of the MakeTNT-toolkit: please check the following links for information on Requirements and Installation.

Command Line Switches and Arguments

- Usage of MOL22TNT:

```
% MOL22TNT -i <input-file> [-n <tag>]\
[-o <filename>] [-FLEXTOR] [-noB] \
[-size <length of output>][[-TightRestrains]\
[-ccp4][[-ccp4_tnt]][-Odict <O-database>][[-notor][[-OnoH][[-HPDB]
```

Switch	Argument	Description	Possible Values	Compulsory	Note
-i	<mol2-file>	name of mol2-file submitted	no format restrictions, any extension	yes	
-n	<tag>	<3-Letter code to be used in TNT-dictionary file>	e.g.: ATP,GMP,ANY,THI,NGY,OUL,IKE	no	default = UNK Use your own!
-o	<filename>	unique filename for TNT-dictionaries	e.g.: kinase_inhibitor986B, ...	no	default = 3-letter code defined by -outname Choose something unique!
-FLEXTOR		output of default parameters for non-planar torsion angles		no	
-noB		flag to suppress the output of default BCORREL restraints		no	
-size	<length-of-output>	defines the overall size of the resulting dictionary	s (short) m (medium) l (long)	no	default : m (medium) see also : Conversion Mechanism
-ccp4		flag to generate CCP4-restraints instead of TNT-restraints		no	
-ccp4_tnt		flag to generate CCP4-restraints AND TNT-restraints		no	
-Odict		flag to generate dictionaries for O instead of TNT-restraints		no	
-ODB	<O-database>	append non-redundant O-dictionaries to existing O-database		no	only relevant to the generation of O-dictionaries
-OnoH		exclude hydrogen atoms from O-dictionary entries		no	only relevant to the generation of O-dictionaries
-notor		flag to suppress torsion restraints for non-planar groups in O-dictionaries		no	only relevant in combination with -Odict
-HPDB		request of PDB-file including hydrogen positions of original MOL2-file		no	
-TightRestrains		sigma for bond lengths: 0.020Å, sigma for bond angles: 2.0deg		no	default: 0.040Å, 5.0deg

Conversion Mechanism:

Restraint atoms are selected on the basis of a molecular graph generated from the bond-list of the mol2-file. Target values for restraints are taken from the 3D-coordinates. If present the positions of H-atoms are used to improve the quality of the restraints. As Buster-TNT presently does not support the refinement of hydrogen atoms, they are removed from the final restraint file.

If an atom has three neighbours bonded to it, an improper torsion is measured between those four atoms and if its value is close to 0 or 180 degrees (tolerance: 10 degrees), a TRIGONAL restraint is generated, otherwise an IMPROPER torsion restraint is generated. IMPROPER torsion restraints are also generated for atoms surrounded by four (or more) neighbours.

A PLANE restraint is formed around bonds:

- of Tripos-bond types:
1, 2, ar, am
- displaying planar torsion angles (tolerance: 10 degrees).

The different options for the **-size** argument are:

- **s** (small):
Overlapping planes will result in one single TNT-PLANE restraint.
 - **m** (medium, default):
Planar 5/6-membered rings result in one PLANE restraint each. Recommended for large planar systems and in structures of high resolution.
 - **l** (long):
Each planar torsion angle will result in a separate PLANE restraint.
-

Output:

Using MDL2TNT in its default mode, the following output files are written into the working directory:

- TNT-dictionary file: <filename>.dic (e.g. ligand.dic)
- PDB-file without H-pos for refinement: <tag>.pdb (e.g. LIG_tnt.pdb)

With the **-CCP4** flag, two files are generated:

- dictionary file for Refmac (e.g: ligand.GPhL.cif)
- PDB-file (with H-pos): LIG_tnt.pdb

If you choose the **-Odict** option, you will obtain:

- dictionary file: e.g.: ligand_O_refi_dict.txt

- dictionary header-file: ligand_O_refi_dict.hd (.bond_angles count line)
- PDB-file: LIG_tnt.pdb

If you choose -Odict, -ODB <O-database>:

- the new dictionary will be appended to the existing O-database
 - if the database does not yet include an entry for a residue of the 3-letter code specified
-

Example:

Example input files (mdl189978.mol2, TXL.mol2) can be found in: samples/maketnt/MOL22TNT.

Please note that TXL.mol2 has too many atoms/residue for O processing.

- To produce a TNT-dictionary file for mdl824.mol, please run:

```
% MOL22TNT -i mdl189978.mol2 -n LIG -o ligand
```

- For Refmac-restraints, run:

```
% MOL22TNT -i mdl189978.mol2 -n LIG -o ligand -CCP4
```

- To add an entry to the O-library:

```
% MOL22TNT -i mdl189978.mol2 -n LIG -Odict -ODB stereo_chem.odb
```

or

```
% MOL22TNT -i mdl189978.mol2 -n LIG -Odict -ODB stereo_chem.odb
-notor
```

if you do not wish to restrain flexible torsions

Evaluation:

Derivation of TNT-dictionary restraints from *.mol2 files has been tested on a data set of 1000 compounds from the NCI-database Dictionaries from 20 *.mol2 files (NSC-Numbers: 325, 333, 334, 335, 396, 400, 408, 463, 479, 533, 551, 556, 558, 604, 741, 790, 811, 870, 871, 872) have been checked for their performance in re-optimizing (Gelly, O.Smart, Globalphasing) a corresponding PDB-file (generated by Corina) after an initial distortion to an RMSD-value of 0.25Å. Manual inspection showed consistent optimization for all 20 test-cases (data available on request).

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 02.02.07

EditTNT

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Contents

- General
- Requirements and Installation
- How to start EditTNT
- How to use EditTNT
- Output
- Tutorial
 1. Setup
 2. Restraints from Corina
 3. Restraints from xplo2d (XPLOR): adding/removing restraints

General Information

EditTNT is a Java-based interactive tool for editing and visualization of TNT-restraints. All types of restraints are supported. You can modify existing restraints or delete them and add new ones. Visualization is carried out by JMOL. Provided with a PDB-file (see below: How to start EditTNT), EditTNT can call Gelly (O.Smart) to optimize the PDB-file with the parameters chosen (including the inversion of chiralities by sign change of the corresponding improper torsion angles).

Requirements and Installation

EditTNT is part of the MakeTNT-toolkit: please check the following links for information on Requirements and Installation

How to start EditTNT

EditTNT can be used to visualize restraints in PDB, MDL and Tripos-MOL2-files

```
% EditTNT [TNT-dictionary-file] [structure-file]
```

where:

Argument	Description	Note
TNT-dictionary-file	TNT-style-restraints-file	
structure-file	name of PDB, MDL, or Tripos Mol2-file	matches up with the TNT-dictionary file

See also

```
% EditTNT -h
```

for a list of command line arguments. For examples, please refer to the Tutorial section of this documentation.

How to Use EditTNT

- The main table
 - Click on the tab BOND, ANGLE, TORSION, etc. to view a table of restraints of the given kind
 - Use the mouse to select the first restraint you want to visualize in the table
 - Scroll up and down the table with the arrow keys to search for restraints
 - NEW in MakeTNT2.1.1: Click on an atom in the Jmol-window to highlight the restraints it is involved in
image
 - Double click on a cell in order to edit restraints
 - The display will update as soon as you press ENTER or move to another cell
 - For planarity restraints atom counts will automatically update upon the modification of atom records
 - Check the display before you save the table
- NEW in MakeTNT2.1.1: Interactive Addition/Duplication of Restraints
 - Click on the "Add Restraint" or "Duplicate Restraint" - button in the main window
 - Click on atoms in the Jmol-display to select atoms for the new restraint, they will show up in the extra table
 - Click on a selected atom (NO DOUBLE CLICK) to remove it again
 - Insert the new restraint into the main table by clicking on "Add Restraint To Table"

image

- **Gelly optimization:**

- Click on "Run Gelly" to optimize your structure to the restraints chosen (only possible if pdb-file was submitted)
 - Once the optimization is complete (< 10s), switching to the optimized conformation (and back) will become available
 - Optimization will always start from the original pdb-file
 - If you would like an optimization to start from a previous gelly run, please restart EditTNT with the gelly-output PDB-file and the modified dictionary (*.gelly)
 - Please also notice the messages provided by EditTNT
 - To use gelly for changing chiralities, check the example below
-

Output:

If you click on Save Table on the menu, the table will be saved into a file with the extension ".user". If you have chosen to optimize a structure with Gelly, the latest update of the table will be saved into a file with the extension ".gelly", the optimized structure will be saved into a file with the extension ".opt.pdb" and a file "gelly_refine.log" will show you the course of the gelly - optimization and its final agreement with the restraints chosen by you.

Tutorial:

1. Setup

Please copy the folder `samples/maketnt` of your MakeTNT-toolkit distribution over:

```
% cp -pri $BDG_home/MakeTNT/samples/maketnt .
```

This will create a local copy of the examples that we provide. In this folder you will find restraint files generated by Corina, the Libcheck program (part of Refmac and CCP4) and xplo2d (XPLOR option), as well as the corresponding TNT-dictionary files converted by MakeTNT. For details of MakeTNT usage, please refer to the MakeTNT-documentation.

2. Restraints from Corina

To look at TNT-restraints generated for the compound taxotere by Corina and converted into TNT-format by MakeTNT, please go to the `maketnt/CORINA` directory and type:

```
% EditTNT taxotere.dic TXL.mol
```

This will start the interface as shown above. Please select PLANE in the tab-header section. Click on an atom to highlight the planar restraints it is involved in. A count of restraints found for this atom will appear in the text window towards the bottom of EditTNT. You can also use the mouse keys to scroll up and down the list of planar restraints.

Looking at the table you will find, that for this molecule all PLANE restraints are computed in a very satisfactory way, as is usually the case when using Corina-derived parameters. In the next sections, we will learn how to edit and - if necessary - modify restraints generated by other sources.

3. Restraints from Libcheck/Refmac

To view TNT-restraints derived from a Libcheck/Refmac descriptions, please go to the `maketnt/Refmac` directory and type:

```
% EditTNT adenosinetriphosphate.dic ATP.pdb
```

Select the PLANE-table again and look at the only entry, which describes the planarity of the ATP-purine ring. If you wish to add the N6 atom to this list, double click on the cell with the Header Atoms. It will change colour and you can scroll to either end of the list using the arrow keys on your keyboard. You can delete text in the cell using the backspace key and add text by just typing it into the cell. To add the "N6" atom to the list, go to the end of it and type ", N6".

Alternatively, click on "Duplicate Restraint" to start a new table with the same atom atoms. To change the selection of atoms, either

- click on the atom in the JMOL display window, this will
 - selected atoms
 - deselect selected atoms
- or
- double click in the atoms cell to edit, a single click outside this cell, but inside the table will re-enable selection from the molecular viewer

Once you are happy with your new restraint, insert it into the main table by pressing the respective button and remove the original restraint from there.

A similar mechanism can be used to add new restraints using the "Add Restraint" button.

To save the new restraint list, select Save Table. The TNT-dictionary file will be saved using the original name with the extension ".user". Before saving the list, please press "Enter" to make sure that all the latest changes are included.

4. Restraints from xplo2d (XPLOR): adding/removing restraints

For a demonstration of TNT-restraints derived from xplo2d (XPLOR-format), please go to the `maketnt/XPLOR` directory and type:

```
% EditTNT rosiglitazone.dic BRL.pdb
```

Please select the PLANE-table again. You will see four different PLANE-restraints generated for rosiglitazone. The first restraint describes the planar thiazolidinedione group and is correct, as are restraints 2 and 4 (for the benzyl and pyridinyl group, respectively). However, the third restraint (4 atoms) is derived from a particular conformation rather than from a chemical rule. If you are interested

in refining rosiglitazone in conformations different to the one in the structure shown, you probably want to remove that restraint. To do so, simply select it and click on the [Remove Restraint](#) - button. This removes the restraint in the table, but not in the underlying file. To make the change permanent, please save the new restraints by clicking on [Save Table](#).

5. Restraints from PDB2TNT: using Gelly to invert the chirality around an atom

To find out how to use gelly interactively within EditTNT, please enter the `maketnt/PDB2TNT` directory and type:

```
% EditTNT TXL.dic TXL_tnt.pdb
```

Now, let's assume you would like to invert the chirality around the C34 atom. To do that, you enter the [IMPROPER](#) card in EditTNT, select the 8th restraint from the top, change its value from -30.7 to 30.7, press the ENTER key on the keyboard and then press the "Run Gelly" button on the interface. You will get a message that gelly has started running. Once optimization is complete (<10s), you will be notified by an EditTNT-message. Simultaneously, a button for loading the new structure into EditTNT/JMOL will become active. Another button will take you back to the original structure. The "Run-Gelly" button could be modified by an option to specify parts of the molecule to be fixed during the optimization. Please let us know whether such an option would be useful to you.

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 10.07.08

EditREFMAC

Copyright © 2004-2008 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2008) M. Brandl

Contents

- General
 - Requirements and Installation
 - How to start EditREFMAC
 - How to use EditREFMAC
 - Output
 - Example
-

General Information

EditREFMAC is a Java-based interactive tool to edit and visualize REFMAC-restraints. You can modify existing restraints or delete them and add new ones. Visualization by rasmol is generated at runtime.

Requirements and Installation

EditREFMAC is part of the MakeTNT-toolkit: please check the following links for information on Requirements, Installation and

How to start EditREFMAC

Please start **EditREFMAC** using the command:

```
% EditREFMAC [REFMAC-dictionary-file] [PDB-file] [ligand-3-letter  
code]
```

where:

Argument	Description	Note
Refmac-library-file	Refmac-library-file(including monomer description of ligand)	please check MakeTNT's current format requirements for REFMAC restraint dictionaries
PDB-file	name of PDB-file	matches up with Refmac library-file
3-letter code	3-letter code of the residue you wish to edit	only restraints for this residue will be edited

See also

```
% EditREFMAC -h
```

for a list of command line arguments. For an example, please refer to the Example section of this documentation.

How to Use EditREFMAC

EditREFMACJMOL.png

- Click on the tab atom, tree, bond, etc. to view a table of restraints of the given kind
- Use the mouse to select the first restraint you want to visualize in the table
- Scroll up and down the table with the arrow keys to visualize/search for restraints
- Check the update on the display before you save the table

Example:

- Please enter samples/maketnt/EditREFMAC and execute: EditREFMAC 1A29_22_lib.cif TFP.pdb TFP

Output:

- If you click on Save Table on the menu, the **complete** table will be saved into a file with the extension ".user". This allows you to visualize and change data for a particular monomer without loss of the other information generated by Refmac. If you submit e.g. a library file generated by Refmac, it should still work in Refmac-refinement.
- The Save and Convert to TNT option saves the complete table with the extension ".user", subsequently converts the restraints for the monomer specified to a TNT-dictionary file (name: 3-letter-residue-code ".fromRefmac.dic") and starts EditTNT on the TNT-dictionary file for further inspection and editing. Please note, that this tool only converts the monomer specified and not the complete library.

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modified: Fri Jul 17 16:03:42 BST 2009

MakeLINK

Copyright © 2004-2007 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2007) M. Brandl, (2008) T. Womack

Contents

- General Information
 - Requirements and Installation
 - Command Line Switches and Arguments
 - Mechanism
 - Output
 - Examples
-

General Information

MakeLINK scans the 3D-structure of a model (PDB-file) for linkages between residues and assigns TNT-dictionaries to them. It creates information in the format of a TNT-sequence file for use in BUSTER and autoBUSTER.

It is particularly useful:

- for structures with **carbohydrate components**, since using it will guarantee that the linkages between individual sugar moieties conform to the geometry of well curated standard libraries (in the default mode, autoBUSTER will restrain those bonds to the parameters found in the starting model).
 - if you repeatedly need to define links involving **confidential compounds**. MakeLINK's libraries can be extended by the user. This enables MakeLINK to automatically recognise a link, e.g. to a covalently bound inhibitor and to create the reference to a user-curated dictionary embodying all your knowledge about its geometry. By default, autoBUSTER will just restrain the geometry of such bonds to the input model.
 - if you would like to use a TNT-sequence file including **parts of the protein missing in your current model**. MakeLINK allows you to amend a TNT-sequence file generated from e.g. a protein sequence database, but does not remove links not present in the PDB-structure submitted.
-

Requirements and Installation

MakeLINK is installed with the MakeTNT-toolkit: please check the following links for information on Requirements and Installation.

Command Line Switches and Arguments

- Usage of MakeLINK:

```
% MakeLINK -p <pdb-file> [-s <input-sequence-file>]
[-o <output-sequence-file>] [-d <link-parameter-file>] [-a]
[-l] [-debug]
```

Argument	Description	Possible Values	Compulsory	Note
-p <pdb-file>	name of PDB-file submitted	no format restrictions, any extension	yes	
-s <TNT-sequence file>	name of TNT-sequence file including standard entries	no format restrictions, any extension	no; -a is assumed if neither -s nor -o option is given	generated by pdb2seq or manually
-o <TNT-sequence file>	name to give to output sequence file	no format restrictions, any extension	no; is filled in as <i>pdb-file.seq</i> or <i>seq-file.1</i> if not given	no
-d <link-parameter file>	Name of library of user defined residue-residue links	no format restrictions, any extension	No; but you can use more than one -d option	user generated, example

Switch	Description
-a	Generate the sequence file automatically from the PDB input
-l	Generate linkages for all contacts, not just those described in LINK or SSBOND records in the input PDB file
-ss	Generate linkages for all SSBOND records in the input PDB file, whether the atoms are in contact or not

Mechanism

- All inter-residue contacts
 - involving B,C,N,O,F,Si,P,S,Cl,As,Se,Br,I,Hg atoms
 - shorter than a distance d :

$$d = r\text{-atom1} + r\text{-atom2} + 0.30\text{\AA};$$

with $r\text{-atom1}$ and $r\text{-atom2}$ representing the element specific covalent radii (loaded by default from `$BDG_home/database/lists/atom.dat`)

are identified. The contacts are searched for in the following three files:

1. main connectivity library `$BDG_home/tnt/data/connect.dat`
 2. user defined connectivity files (e.g. `$BDG_home/samples/maketnt/MAKELINK/1EAU/SBDK.connect`)
 3. library of atom-specific connectivity data (loaded by default from `$BDG_home/database/files/MakeTNT/connectAt.dat`)
- If no link is found in any of the lookup-tables, MakeLINK creates a PDB-file of the linked residues, calls PDB2TNT (-link option) to generate a link-dictionary file, and writes the contents of the link-dictionary file to the output sequence file
 - If the `-ss` option is used, the SSBOND cards rather than the input geometry are used to generate the sequence file. This is useful if a refinement of your structure gives electron density indicating the presence of a disulphide bond: you can add an SSBOND card to the PDB file and rerefine to pull the SG atoms into place.

MakeLINK trusts the input positions to have the right atom contacts, if not necessarily correct geometry for linkages that it recognises; so if you have for example added a sugar residue to your molecule, and MakeLINK does not produce the linkage you expect, please move the sugar so that the contact it makes with the residue it should link to is shorter. It will be re-refined to a curated geometry for the appropriate linkage.

Output:

Various temporary files are written into the working directory during the run, but should be deleted automatically.

Using the `-o` option, you can choose your own name for the TNT-sequence file produced by MakeLINK.

If you do not enter a name for the output sequence file, the name of the output sequence file is composed of:

- the parameter to `-s` with a ".1" at the end, if a sequence file was submitted;
- `input_pdbname.seq`, if no input sequence was submitted;

Examples:

Example input files (pdb1uvq.ent, pdb1eau.ent) can be found in \$BDG_home/samples/maketnt/MAKELINK/1EAU and \$BDG_home/samples/maketnt/MAKELINK/1UVQ.

- The normal usage is simply

```
MakeLINK -p pdb1uvq.ent -a
```

which will make a sequence file pdb1uvq.ent.seq; if you look at this file, you will see lines like

```
RESIDUE B|19 ASN 20 XGPEPTIDE 1193 S001
```

which indicates that the ND1->C1 linkage between ASN residue 20 and (NAG) residue 1193 has been recognised.

- If your input file does not have chain identifiers for all atoms, the program will complain; in that case, do

```
pdbchk -p pdb1eau.ent -o pdb1eau.chk.pdb then
```

```
MakeLINK -p pdb1eau.chk.pdb -a
```

- As another example, consider

```
MakeLINK -p pdb1eau.chk.pdb -a
```

This will complain

```
A link of type U01 has been found between OG of A|195 and C2 of B|1
There is no linkage between those atoms described in the LINK
cards
Please add a LINK record or re-run with -l
```

that there is no LINK card describing the SER->BDK linkage in pdb1eau.chk.pdb, so re-run with the -l option to have it generate a sequence file anyway.

That sequence file will call the link 'U01'. If you happen to have a description of this kind of linkage handy - say in

```
$BDG_home/samples/maketnt/MAKELINK/1EAU/SBDK.connect - you can use
MakeLINK -d SBDK.connect -p pdb1eau.chk.pdb -a -l
```

and get a sequence file in which the link has a more sensible name. Note that you have to use the -l SBDK.dic option to autoBUSTER so that it knows the geometry for the residue.

Advanced configuration

The locations of the element-properties file, main connectivity library and atom-specific connectivity library can be specified using the environment variables

MakeTNT_connect	The main connectivity library
MakeTNT_elements	The element-properties file
MakeTNT_connectat	The atom-specific connectivity library

or on the command line using entries of the form
MakeTNT_connect=/home/user/my_connect.dat.

Tom Womack & Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modified: Thu Jul 2 14:55:48 BST 2009

Validated TNT-restraints

Copyright © 2004-2006 Global Phasing Limited

All rights reserved.

This software is proprietary to and embodies the confidential technology of Global Phasing Limited (GPhL). Possession, use, duplication or dissemination of the software is authorised only pursuant to a valid written licence from GPhL.

Author: (2004-2006) M. Brandl

TNT-restraints for the following compounds and their linkages have been added to the TNT-standard library:

- modified amino acids and amino acid modifying groups
 - with PDB-counts of 15 or higher and/or
 - requested by users (please do not hesitate to ask for additional ones)
- Cofactors: ATP, ADP, ANP, cyclic-AMP, FAD, FADH, NAD, NADH
- Carbohydrates: pyranoses, NAG, SIA

Restraints are based on standard PDB atom and residue naming documented below. For each compound, please find links to:

- chemical formula (column2)
- example PDB-structure (last but one column)
- superposition (last column) of example PDB-structure (MODEL 1) and structure optimized by Gelly(O.Smart, MODEL 2)

Non-standard Amino Acids

Residue restraints were:

- derived(PDB2TNT) from Corina-generated structures deposited in the MSDchem database[1]
- adjusted for charges and conjugation in phosphate groups

Link restraints have either been taken from the standard library or high-resolution PDB-structures

- [1] ● MSDsite: behind the scene: The technology used in database searching and retrieval for the analysis and viewing of bound ligands and active sites. Golovin, A., Dimitropoulos, D., Oldfield, T. and Henrick, K. (2004) The eCheminfo 2004 Conference "Applications of Cheminformatics and Modelling to Drug Discovery 8-19 November.
- MSD database and MSD database services A. Golovin, T. J. Oldfield, J. G. Tate, S. Velankar, G. J. Barton, H. Boutselakis, D. Dimitropoulos, J. Fillon, A. Hussain, J. M. C. Ionides, M. John, P. A. Keller, E. Krissinel, P. McNeil, A. Naim, R. Newman, A. Pajon, J. Pineda, A. Rachedi, J. Copeland, A. Sitnov, S. Sobhany, A. Suarez-Uruena, J. Swaminathan, M. Tagari, S. Tromm, W. Vranken and K. Henrick (2004) E-MSD: an integrated data Nucleic Acids Research, 32 (Database issue), D211-D216. 2004

modification/modified amino acid	3-letter code/link to formula	parent amino acid	CATEGORY in connect.dat	example coordinates from PDB	superposition of PDB/optimised coordinates
3-SULFINOALANINE	CSD	CYS	AAREST	CSD-1acd.pdb	CSD-1acd-opt.pdb
4-HYDROXYPROLINE	HYP	PRO	AAREST	HYP-1ag7.pdb	HYP-1ag7-opt.pdb
4-METHYL-4-[(E)-2-BUTENYL]-4,N-METHYL-THREONINE	BMT	THR	AANME	BMT-1bck.pdb	BMT-1bck-opt.pdb
5-HYDROXYPROLINE	5HP	GLU	AAREST	5HP-1a39.pdb	5HP-1a39-opt.pdb
ACETYL_GROUP	ACE	-	ACE	ACE-155c.pdb	ACE-155c-opt.pdb
ALPHA-AMINOBUTYRIC_ACID	ABA	ALA	AAREST	ABA-1a3p.pdb	ABA-1a3p-opt.pdb
ALPHA-AMINOISOBUTYRIC_ACID	AIB	ALA	AAREST	AIB-1ai1.pdb	AIB-1ai1-opt.pdb
AMINO_GROUP	NH2	-	AAREST	NH2-1a0m.pdb	NH2-1a0m-opt.pdb
CARBOXY_GROUP	CBX	-	CBX	CBX-1an5.pdb	CBX-1an5-opt.pdb
CYSTEINE-S-DIOXIDE	CSW	CYS	AAREST	CSW-1c0t.pdb	CSW-1c0t-opt.pdb
CYSTEINESULFONIC_ACID	OCS	CYS	AAREST	OCS-1cs8.pdb	OCS-1cs8-opt.pdb
D-ALANINE	DAL	ALA	AAREST	DAL-1cwj.pdb	DAL-1cwj-opt.pdb
D-ARGININE	DAR	ARG	AAREST	DAR-1cvq.pdb	DAR-1cvq-opt.pdb
D-ASPARAGINE	DSG	ASN	AAREST	DSG-1t5m.pdb	DSG-1t5m-opt.pdb
D-ASPARTATE	DSP	ASP	AAREST	DSP-1bfw.pdb	DSP-1bfw-opt.pdb
D-CYSTEINE	DCY	CYS	AAREST	DCY-1czq.pdb	DCY-1czq-opt.pdb
DECARBOXY(PARAHYDROXYBENZYLIDENE-IMIDAZOLIDINONE)THREONINE	CRO	CRO	CRO	CRO-1c4f.pdb	CRO-1c4f-opt.pdb
D-GLUTAMATE	DGL	GLU	AAREST	DGL-1czq.pdb	DGL-1czq-opt.pdb
D-GLUTAMINE	DGN	GLN	AAREST	DGN-1yj1.pdb	DGN-1yj1-opt.pdb
D-HISTIDINE	DHI	HIS	AAREST	DHI-1czq.pdb	DHI-1czq-opt.pdb
D-ISOLEUCINE	DIL	ILE	AAREST	DIL-1cvq.pdb	DIL-1cvq-opt.pdb
D-ISOVALINE	DIV	VAL	AAREST	DIV-1ee7.pdb	DIV-1ee7-opt.pdb
D-LEUCINE	DLE	LEU	AAREST	DLE-1bdw.pdb	DLE-1bdw-opt.pdb
D-LYSINE	DLY	LYS	AAREST	DLY-1c4b.pdb	DLY-1c4b-opt.pdb

D-PHENYLALANINE	DPN	PHE	AAREST	DPN-1b0q.pdb	DPN-1b0q-opt.pdb
D-PROLINE	DPR	PRO	AAREST	DPR-1ic9.pdb	DPR-1ic9-opt.pdb
D-SERINE	DSN	SER	AAREST	DSN-1t5n.pdb	DSN-1t5n-opt.pdb
D-THREONINE	DTH	THR	AAREST	DTH-1a7z.pdb	DTH-1a7z-opt.pdb
D-TRYPTOPHANE	DTR	DTR	AAREST	DTR-1czq.pdb	DTR-1czq-opt.pdb
D-TYROSINE	DTY	TYR	AAREST	DTY-1uno.pdb	DTY-1uno-opt.pdb
D-VALINE	DVA	VAL	AAREST	DVA-1mic.pdb	DVA-1mic-opt.pdb
FORMYL_GROUP	FOR	-	FOR	FOR-1a4.pdb	FOR-1a4-opt.pdb
GAMMA-CARBOXY-GLUTAMIC_ACID	CGU	GLU	AAREST	CGU-1awy.pdb	CGU-1awy-opt.pdb
ISOVALERIC_ACID	IVA	-	AAREST	IVA-1apt.pdb	IVA-1apt-opt.pdb
LYSINE_NZ-CARBOXYLIC_ACID	KCX	LYS	AAREST	KCX-1aa1.pdb	KCX-1aa1-opt.pdb
LYSINE-PYRIDOXAL-5'-PHOSPHATE	LLP	LYS	AAREST	LLP-1a8i.pdb	LLP-1a8i-opt.pdb
N-CARBOXYMETHIONINE	CXM	MET	AAREST	CXM-1aiq.pdb	CXM-1aiq-opt.pdb
N-FORMYLMETHIONINE	FME	MET	AAREST	FME-1bq9.pdb	FME-1bq9-opt.pdb
N-METHYLLEUCINE	MLE	LEU	AANME	MLE-1bck.pdb	MLE-1bck-opt.pdb
N-METHYLVALINE	MVA	VAL	AANME	MVA-1c5f.pdb	MVA-1c5f-opt.pdb
NORLEUCINE	NLE	LEU	AAREST	NLE-1a8k.pdb	NLE-1a8k-opt.pdb
O-PHOSPHOTYROSINE	PTR	TYR	AAREST	PTR-1a1b.pdb	PTR-1a1b-opt.pdb
ORNITHINE	ORN	ALA	AAREST	ORN-1b2h.pdb	ORN-1b2h-opt.pdb
PHOSPHOSERINE	SEP	SER	AAREST	SEP-1b4g.pdb	SEP-1b4g-opt.pdb
PHOSPHOTHREONINE	TPO	THR	AAREST	TPO-1bkx.pdb	TPO-1bkx-opt.pdb
PYROGLUTAMIC_ACID	PCA	GLU	AAREST	PCA-1a8j.pdb	PCA-1a8j-opt.pdb
PYRUVOYL_GROUP	PVL	-	PVL	PVL-1hq6.pdb	PVL-1hq6-opt.pdb
SARCOSINE	SAR	GLY	AANME	SAR-1bck.pdb	SAR-1bck-opt.pdb
S-HYDROXY-CYSTEINE	CEA	CYS	AAREST	CEA-1cxp.pdb	CEA-1cxp-opt.pdb
S-HYDROXYCYSTEINE	CSO	CYS	AAREST	CSO-1dmp.pdb	CSO-1dmp-opt.pdb
S-MERCAPTOCYSTEINE	CSS	CYS	AAREST	CSS-1bi0.pdb	CSS-1bi0-opt.pdb
S-OXY_CYSTEINE	CSX	CYS	AAREST	CSX-1fw8.pdb	CSX-1fw8-opt.pdb
S,S-(2-HYDROXYETHYL)THIOCYSTEINE	CME	CYS	AAREST	CME-1a1v.pdb	CME-1a1v-opt.pdb
SULFONATED_TYROSINE	TYS	TYR	AAREST	TYS-1a2c.pdb	TYS-1a2c-opt.pdb
TERT-BUTYLOXYCARBONYL_GROUP	BOC	-	BOC	BOC-1btw.pdb	BOC-1btw-opt.pdb
TOPO-QUINONE	TPQ	PHE	AAREST	TPQ-1a2v.pdb	TPQ-1a2v-opt.pdb
TYROSINE-O-SULPHONIC_ACID	STY	TYR	AAREST	STY-1c5l.pdb	STY-1c5l-opt.pdb

Cofactors

Residue restraints were generated in 4 steps:

- a high-resolution structure from the hiccup database was chosen as a starting point
- the structure was corrected for atom-typing and charged groups and its geometry optimized using the MMFF force field and a constant dielectric constant of 80(water)
- PDB2TNT was run on the resulting optimized structure
- P=O bonds were normalized manually to correct for electron delocalization

cofactor	3-letter code/link to formula	example coordinates from the Hiccup database	superposition of Hiccup/optimised coordinates
ADP	ADP	adp_exp.pdb	adp-opt.pdb
AMP	AMP	amp_exp.pdb	amp-opt.pdb
ATP	ATP	atp_exp.pdb	atp-opt.pdb
cAMP	CMP	cmp_exp.pdb	cmp-opt.pdb
COENZYME A[1]	COA	coa_exp.pdb	coa-opt.pdb
FAD	FAD	fad_exp.pdb	fad-opt.pdb
FLAVIN_MONONUCLEOTIDE	FMN	fmn_exp.pdb	fmn-opt.pdb
NADP	NAP	nap_exp.pdb	nap-opt.pdb
NADPH	NDP	ndp_exp.pdb	ndp-opt.pdb

[1] MMMFFs optimized Ac-CoA has been used as a template structure

Carbohydrates

TNT-restraints for:

- all sugars listed in the table below
- all O-glycosidic linkages between those monomers (please use MakeLINK to add the appropriate link to your TNT-sequence file)
- N-glycosidic linkage between ASN and any pyranose (including NAG), O-glycosidic linkages between SER/THR and any pyranose (including MAN; please use MakeLINK to add to your sequence file)

cofactor	3-letter code/link to formula	example monomer coordinates from the Hiccup database	superposition of PDB/optimised coordinates for monomer	example environment	optimized example environment
D-GLUCOSE	BGC GLC	bgc_msd.pdb glc_msd.pdb	bgc_msd-opt.pdb glc_msd-opt.pdb	1od3.pdb 3chb.pdb	1od3-opt.pdb 3chb-opt.pdb
D-MANNOSE	MAN BMA	bma_msd.pdb man_msd.pdb	bma_msd-opt.pdb man_msd-opt.pdb	1ib4.pdb 1oh4.pdb	1ib4-opt.pdb 1oh4-opt.pdb
FUCOSE	FUC	fuc_msd.pdb	fuc_msd-opt.pdb	1e4m.pdb	1e4m-opt.pdb
D-GALACTOSE	GAL GLA	gal_msd.pdb gla_msd.pdb	gal_msd-opt.pdb gla_msd-opt.pdb	3chb.pdb 1oh4.pdb	3chb-opt.pdb 1oh4-opt.pdb
N-ACETYL-D-GLUCOSAMINE	NAG	nag_msd.pdb	nag_msd-opt.pdb	1e4m.pdb	1e4m-opt.pdb
N-ACETYL-D-GALACTOSAMINE	NGA	nga_msd.pdb	nga_msd-opt.pdb	3chb.pdb	3chb-opt.pdb
O-SIALIC_ACID	SIA	sia_msd.pdb	sia_msd-opt.pdb	3chb.pdb	3chb-opt.pdb
D-XYLOPYRANOSE	XYS	xys_msd.pdb	xys_msd-opt.pdb	1e4m.pdb	1e4m-opt.pdb

Maria Brandl, <buster-develop@GlobalPhasing.com>

Last modification: 13.12.05