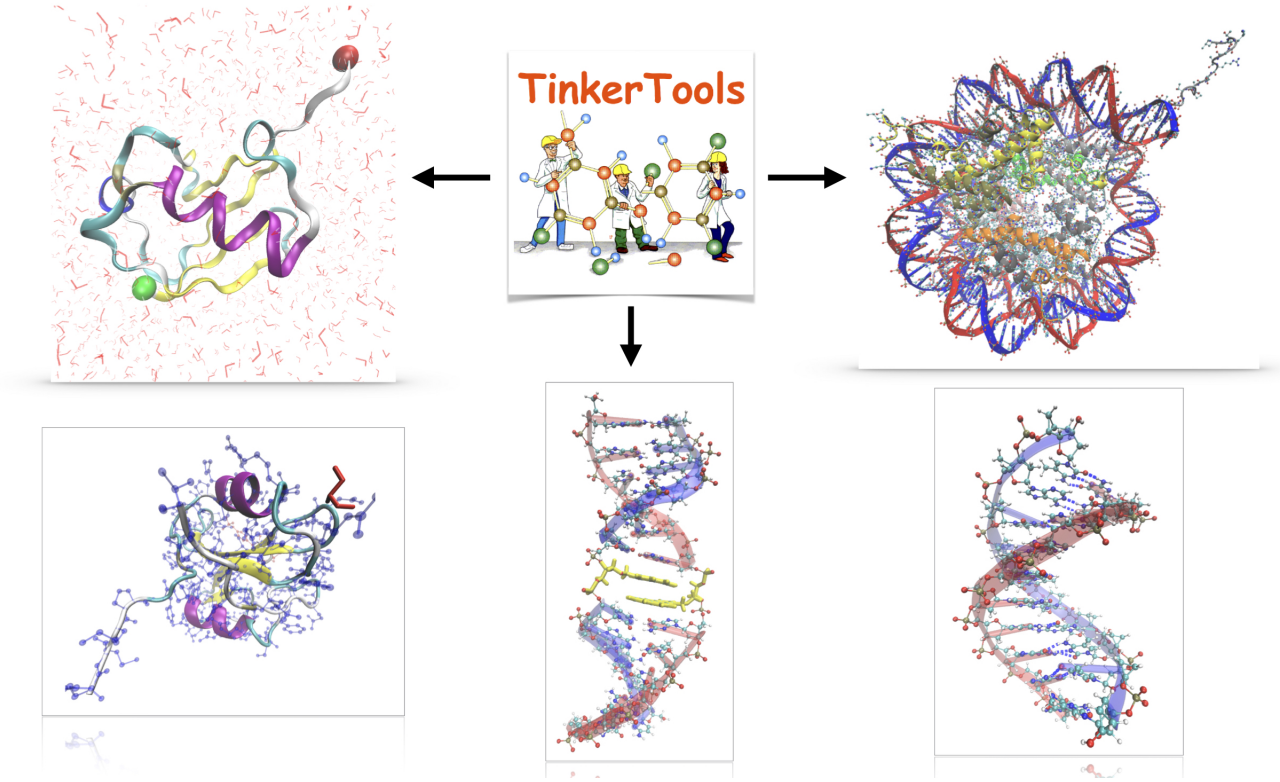
- Frédéric CELERSE
- Dina EL AHDAB
- Louis LAGARDERE
- Etienne DERAT
- Jean–Philip PIQUEMAL

How to set up a system for Tinker–HP ? TUTORIAL



⚠ Please visit the website: https://github.com/TinkerTools to obtain last updates of this tutorial and informations about other new tutorials.

Contents

1 Introduction **3**

2 Generating a Tinker XYZ file (TXYZ): several cases **4**
- 2.1 First case: the general case with ubiquitin 4
- 2.2 Second case: manage the protonation states still with ubiquitin 5
- 2.3 Third case: adding the crystal waters in the conversion 8
- 2.4 Fourth case: a first simple DNA case 10
- 2.5 Fifth case: a second DNA case mixing DNA and protein 11
- 2.6 Sixth case: a protein with a specific ligand 12
- 2.7 Seventh case: a DNA structure with non–usual nucleic acids 16

3 How to join two fragments to form a ligand ? **19**

4 Solvating and neutralizing a Tinker–HP system: the general procedure **21**
- 4.1 How to create a water box using Tinker modules ? 21
- 4.2 How to solvate a system using Tinker modules ? 22
- 4.3 How to neutralize a TXYZ system ? 23

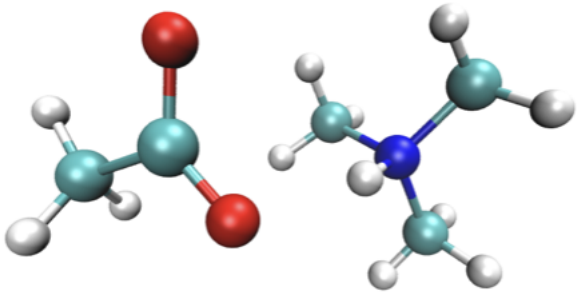
5 Other available Tinker tutorials **25**

1 Introduction

This tutorial is dedicated to few points relative to the Tinker XYZ file preparation protocol. Tinker XYZ file will be labeled as TXYZ file in the following of this tutorial. Indeed, a TXYZ file is more than a simple XYZ file. In addition to the number of atoms and the classical x/y/z cartesian coordinates, we can find additional informations (See Figure 1) such as:

- The atom type, which directly depends on the chosen force field related to the TXYZ file,
- The connectivities for each atoms, which could be partially attributed to the PSF informations in other simulation packages (CHARMM, NAMD, AMBER, ...)

Manually manage these parameters in the TXYZ preparation procedure could fastly become complicated. This is why we propose to sum a list of different cases you could often meet if you have to prepare a TXYZ file for a Tinker-HP simulation.



XYZ

```
21
   H      6.452575   -4.961420   -8.073297
   N      7.092541   -5.279817   -7.333865
   C      8.194742   -5.917805   -8.142721
   H      9.063044   -6.308931   -7.560229
   H      7.742956   -6.678958   -8.739743
   H      8.615509   -5.041600   -8.704687
   C      6.322927   -6.327683   -6.508201
   H      5.869650   -7.044235   -7.210021
   H      7.040435   -6.791576   -5.818699
   H      5.503991   -5.827558   -5.914070
   C      7.553615   -4.110065   -6.490444
   H      6.647051   -3.660315   -6.072634
   H      8.166915   -4.498185   -5.713955
   H      8.117199   -3.414381   -7.123686
   C     -4.788339   -4.550860   -2.612164
   H     -4.175040   -3.688858   -2.298099
   H     -5.088616   -5.210651   -1.807318
   H     -5.760796   -4.301374   -3.063026
   C     -4.034531   -5.321423   -3.710906
   O     -3.279409   -6.306235   -3.309398
   O     -4.261179   -5.158688   -4.964163
```

Tinker XYZ

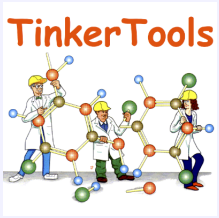
```
21
  1  H      6.452575   -4.961420   -8.073297   501    2
  2  N      7.092541   -5.279817   -7.333865   502    1     3     7    11
  3  C      8.194742   -5.917805   -8.142721   503    2     4     5     6
  4  H      9.063044   -6.308931   -7.560229   504    3
  5  H      7.742956   -6.678958   -8.739743   505    3
  6  H      8.615509   -5.041600   -8.704687   506    3
  7  C      6.322927   -6.327683   -6.508201   507    2     8     9    10
  8  H      5.869650   -7.044235   -7.210021   508    7
  9  H      7.040435   -6.791576   -5.818699   509    7
 10  H      5.503991   -5.827558   -5.914070   510    7
 11  C      7.553615   -4.110065   -6.490444   511    2    12    13    14
 12  H      6.647051   -3.660315   -6.072634   512   11
 13  H      8.166915   -4.498185   -5.713955   513   11
 14  H      8.117199   -3.414381   -7.123686   514   11
 15  C     -4.788339   -4.550860   -2.612164   401   16    17    18    19
 16  H     -4.175040   -3.688858   -2.298099   402   15
 17  H     -5.088616   -5.210651   -1.807318   403   15
 18  H     -5.760796   -4.301374   -3.063026   404   15
 19  C     -4.034531   -5.321423   -3.710906   405   15    20    21
 20  O     -3.279409   -6.306235   -3.309398   406   19
 21  O     -4.261179   -5.158688   -4.964163   407   19
```

Figure 1: Representation of a XYZ (on left) and its respective TXYZ (on right). Columns in red, green and yellow represents respectively: the atom ID, its atomic number in the force field and the connectivities of each atoms. These features are not present in a single XYZ file.

2 Generating a Tinker XYZ file (TXYZ): several cases

2.1 First case: the general case with ubiquitin

Input Files and Executables



- 1UBQ.pdb
- AMOEBABIO18.prm
- pdbxyz.x

To begin we will consider a simple and classical case, the ubiquitin protein. This system contains only usual amino–acids, which represents for us a standard reference for this tutorial. Let us see how to pass from a pdb structure (1UBQ.pdb) of this system to a TXYZ file for Tinker–HP.
In a terminal, you directly applied the tinker executable pdbxyz.x on the 1UBQ.pdb such as:

./pdbxyz.x

You will then enter the name of the pdb file (1UBQ.pdb) and the name of the desired force field (AMOEBABIO18.prm). Finally, 2 output files will be generated if the procedure well works:

1. an output file named **1UBQ.xyz** which corresponds to the TXYZ file
2. a sequence file named **1UBQ.seq**, which indicates the name of all amino–acids read and converted during the procedure. This file is useful when we want to check that all amino–acids have benn well taken into account during the procedure.

And when it does not work ? ...

⚠Note that this case (ubiquitin) is the universal case due to the simplicity of its amino–acids composition. In other cases, the procedure can failed, depicted by:

- No sequence file will be generated
- The atom types, number of connectivities and the respective connectivities will be set to 0

In this case, the problem you are encountering could be explained by several manner:

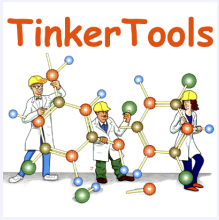
- The presence of non amino–acids (water, ions, ligand ...). In this case, a guideline to overcome this problem will be provide you in the 2.4 part of this tutorial.
- The presence of a modified amino–acids (Serine + adduct for example). In this case, a guideline to overcome this problem will be also provide you in the 2.5 part of this tutorial.

Output Files

- 1UBQP.xyz
- 1UBQP.seq

2.2 Second case: manage the protonation states still with ubiquitin

Input Files and Executables



- 1UBQ.xyz
- AMOEBABIO18.prm
- vmd

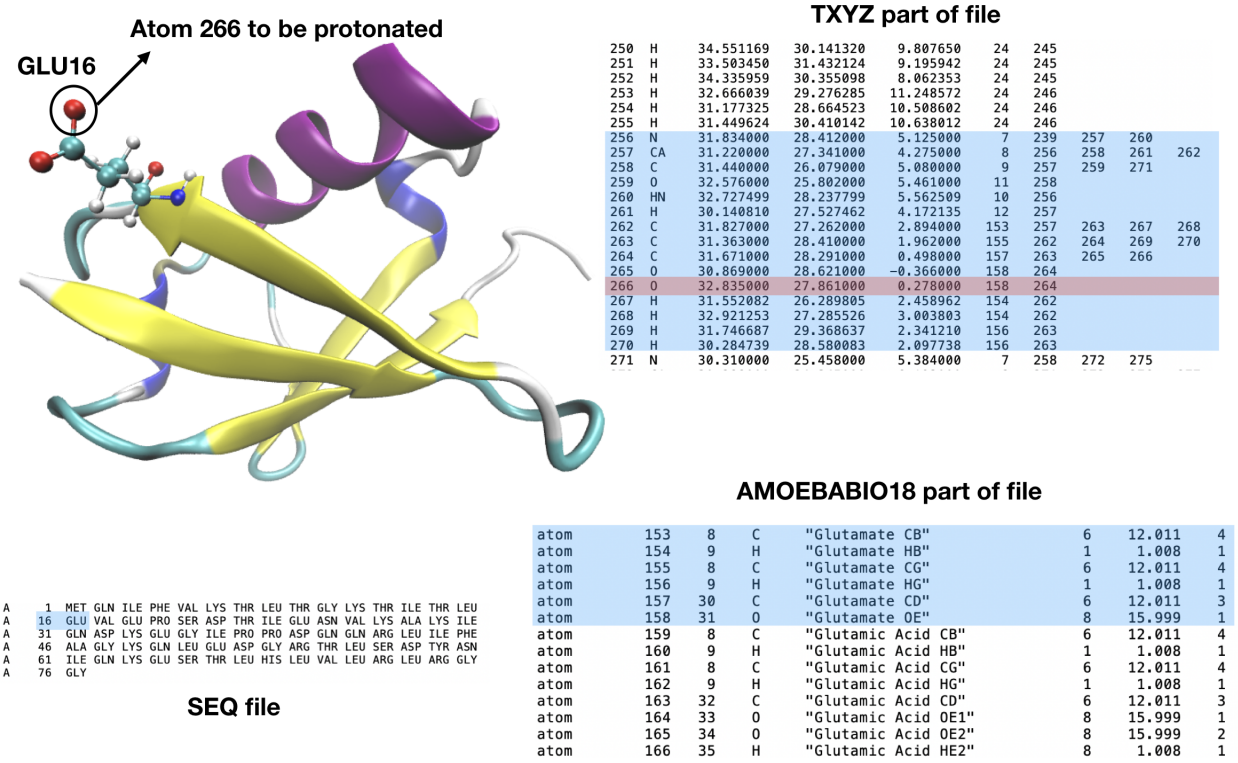


Figure 2: Representation of the GLU16 which would be protonated, and the specific lines of TXYZ, AMOEBABIO18 and SEQ files concerned.

When you directly applied the pdbxyz.x Tinker executable, protonation state are directly calculated according to their theoretical protonation state at the standard neutral pH (=7). A table summing the different protonation states of each concerned amino–acids is presented as following:

Amino acid	α–CO_2H pKa_1	αNH_3 pKa_2	side chain pKa_3
Arginine (ARG)	2.1	9.0	12.5
Aspartic Acid (ASP)	2.1	9.8	3.9
Cysteine (CYS)	1.7	10.4	8.3
Glutamic Acid (GLU)	2.2	9.7	4.3
Histidine (HIS)	1.8	9.2	6.0
Lysine (LYS)	2.2	9.0	10.5
Tyrosine (TYR)	2.2	9.1	10.1

Table 1: Standard pKa calculated at pH=7 for charged amino–acids. The main column of interest in this work is the fourth, but the other could be also usefull for other works.

Imagine for instance, according to Figure you want for any reason to protonate your GLU16, which is not protonated at pH=7 according to the side chain pKa value of Table 1 (4.3). In this way, how can you manually protonate it and in the same way adjust the atom types in the TXYZ file ? Let see simply how to do by following these instructions:

1. Firstly, you have to generate a set of x/y/z coordinates for the H atom wich you will link to the atom 266. To make this step, you can use several softwares such as:

   - VMD molfacture module
   - Gaussview (only with the GLU16 as input file)
   - Macmolplot
   - Avogadro
   - ...

   In our case, we used the molfacture module of VMD to generate a set of x/y/z coordinates for the H atom, which is:

   33.291695 / 27.695854 / 1.152159

2. Then, you add a line in the TXYZ between the atoms 266 and 267 corresponding to your new H atom. It has to be:

   10000 H 33.291695 27.695854 1.152159 166 266

   With:

   - 10000: Corresponds to a new number of atom. Has to be at least up to the last number of atom (here 1232).
   - H: Corresponds to a H atom
   - 33.291695 27.695854 1.152159: X/Y/Z coordinates of the new H atom
   - 166: Is the atom type in AMOEBABIO18 corresponding to the protonated H to the COOH group of GLU.
   - 266: H is linked to atom 266 as well as it corresponds to its protonated form.

3. The last things to do is to adjust the atom types of the CH_2–CH_2–COOH group according to the AMOEBABIO18 force field. In this way, the conversion to apply in the TXYZ is the following:

   - ATOM 262: Replace 153 by 159
   - ATOM 263: Replace 155 by 161
   - ATOM 264: Replace 157 by 163
   - ATOM 265: Replace 158 by 164 (is the unprotonated O)
   - ATOM 266: Replace 158 by 165 (is the protonated O)
   - ATOM 267: Replace 154 by 160
   - ATOM 268: Replace 154 by 160
   - ATOM 269: Replace 156 by 162
   - ATOM 270: Replace 156 by 162

Finally, a sum of what you have now is depicted here on Figure.

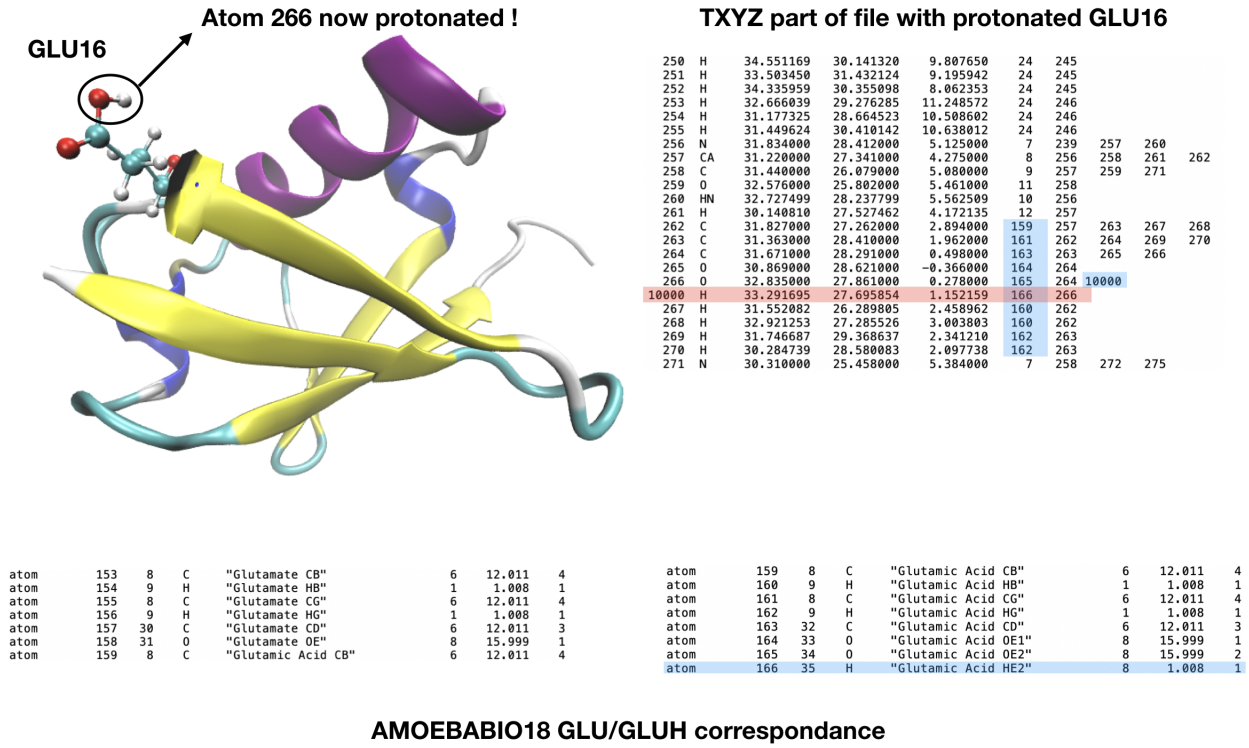


Figure 3: Representation of the protonated GLU16 with the atom types correspondance in the AMOEBABIO18 force field and the added line (red) and modified labels (blue).

Exercise 1: Try to protonate the ASP21.
Exercise 2: Try to deprotonate the LYS63.

2.3 Third case: adding the crystal waters in the conversion

Input Files and Executables

- 1UBQ.pdb
- AMOEBABIO18.prm
- pdbxyz.x
- xyzedit.x
- top_all22_prot.rtf (modified)
- CRY_generation.tcl
- WaterTranslator.f90

In many cases, crystal waters have to be considered more than simple water molecules. Indeed, they can be directly implicated in the stabilization of tertianary protein structures. In such a case, it becomes to be crucial to be able to include these water molecules in the initial txyz.
Unfortunately, the pdbxyz.x tinker module automatically delete all these crystal water molecules during the process. This is why we are suggesting here to present a methodology of how adding these water molecules in the txyz file.
Starting from the 1UBQ.pdb file, follow these instructions:

1. To begin, we have to divide the 1UBQ.pdb into 2 separate pdb files:

   - a file named **1UBQ_protein.pdb**, only containing the protein atoms,
   - a file named **1UBQ_cry.pdb**, only containing on its side the crystal water atoms.

   To create these 2 files, you can type in your terminal:

   ```
   sed -n -e '321,922p' 1UBQ.pdb > 1UBQ_protein.pdb
   sed -n -e '924,981p' 1UBQ.pdb > 1UBQ_cry.pdb
   ```

   and add a the beginning of the both files the following line (which corresponds to the header of the pdb file):

   ```
   HEADER        CHROMOSOMAL PROTEIN
   ```

   and at the end of the both files the line only containing the word "END".

2. You can easily convert the 1UBQ_protein.pdb file into a txyz file by using the ./pdbxyz.x command seen below in the first part of this tutorial. A txyz named **1UBQ_protein.xyz** will be generated.

3. However, it appears to be more complicated for the 1UBQ_cry.pdb file. Indeed, if you try to convert it directly by the use of the pdbxyz.x command, it would not work as well as hydrogen atoms on each oxygen will be not determined. To overcome this problem, we can directly pass by the VMD software to generate a pdb owning the crystal water molecules with hydrogens. For that, open VMD and load the 1UBQ_cry.pdb file. You then go in the Tk console and type the following command:

   source CRY_generation.tcl

   This smooth tcl script working with VMD and the CHARMM topology file "top_all22_prot.rtf" will generate a pdb file, CRY.pdb, containing the oxygen and their corresponding hydrogen atoms.

4. You can then apply the pdbxyz.x module on the CRY.pdb file. Unfortunately, it still does not work to associate the right atom types for each water molecules, but the hydrogen atoms are here in the new CRY.xyz file.

5. We just have to convert the "0" in the CRY.pdb into the right atom types for the AMOEBA water in AMOEBABIO18.prm (349 for O and 350 for H). You can automatically make that by launching in the terminal the following code after compiling it:

   gfortran WaterTranslator.f90 -o WaterTranslator

   ./WaterTranslator

   A new file, CRY2.xyz is generated and owns the right atom types.

6. The last step consists of concatenating the both files, 1UBQ_protein.xyz and CRY2.xyz in one final txyz file. To make that, we can use the xyzedit.x module:

   (a) Type in the termina: ./xyzedit.x
   (b) Enter the txyz file name 1UBQ_protein.xyz
   (c) Enter the parameter file name AMOEBABIO18.prm
   (d) Choose the option 18
   (e) Enter the name file CRY2.xyz
   (f) Type on "enter" to quit the procedure

   A file named **1UBQ_protein.xyz_2** is created and corresponds to the final system ubiquitin + crystal waters for the AMOEBABIO18 force field. You can change the name by 1UBQ_final.xyz to finalize the procedure.
   You are now able to add crystal waters to a txyz file when it will be necessary, while this procedure can be generalized for all systems !

2.4 Fourth case: a first simple DNA case

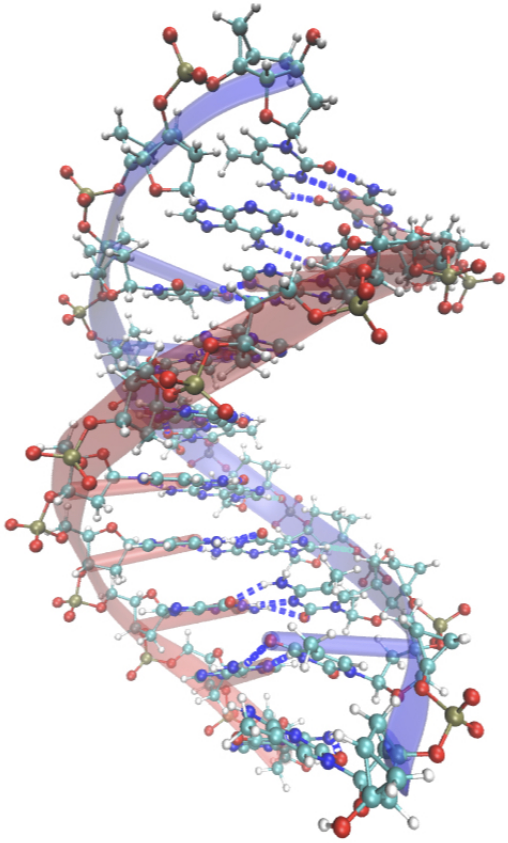


Figure 4: VMD representation of the 6GPI structure. The double brins (blue and red) follow the phorphorous atoms in brown and the H bonds between the both brins are depicted in blue dashed lines.

Input Files and Executables

- 6GPI.pdb
- AMOEBABIO18.prm
- pdbxyz.x

After having seen examples about protein systems, let see now few examples on the other well studied biological systems, DNA. Indeed, the AMOEBABIO18 polarizable force fields contains many polarizable parameters for DNA allowing us for studying of the DNA molecules. It is so of an important interest to be able to well prepared such a system for Tinker–HP. Let see how to do by taking as a first example a simple system is provided by the pdb file labeled 6GPI in the PDB data on internet. It corresponds to DNA duplex, characterized to be simple, classic and small. In order to convert this system from PDB to TXYZ, follow these instructions:

1. As seen previously, we just have to use the pdbxyz.x tinker module to convert the pdb in txyz file. So launch in a terminal this tinker module.
2. Enter the name of the pdb file: 6GPI.pdb
3. Enter the name of the force field: AMOEBABIO18.prm
4. Just type on enter to say that you consider all the chains in the conversion (all by default)

Finally, you well generated a txyz (6GPI.xyz) and seq (6GPI.seq) files. You could easily observed the 6GPI.xyz using VMD, and see that you have more than 1 frame ! This is due to the fact that the pdb file is not coming from a crystallography experiment but from a NMR study. You can so select the frame you desire. In practice, it is advised to consider the last one.

Output Files

- 6GPI.xyz
- 6GPI.seq

2.5 Fifth case: a second DNA case mixing DNA and protein

Input Files and Executables

- 1AOI.pdb
- AMOEBABIO18.prm
- pdbxyz.x

We will see now a more complex DNA system mixing DNA and amino acids residues. Such systems are not so complicated to be translated from pdb to txyz file. The procedure, applied here on the 1AOI.pdb, is similar to the previous one:

1. Launch in a terminal the pdbxyz.x tinker module.
2. Enter the name of the pdb file: 1AOI.pdb
3. Enter the name of the force field: AMOEBABIO18.prm
4. Just type on enter to say that you consider all the chains in the conversion (all by default)

You then generated a txyz (1AOI.xyz) and seq (1AOI.seq) files, meaning that the conversion well worked. This is now possible to prepare biological systems mixing DNA and amino acids for Tinker–HP simulations.

Output Files

- 1AOI.xyz
- 1AOI.seq

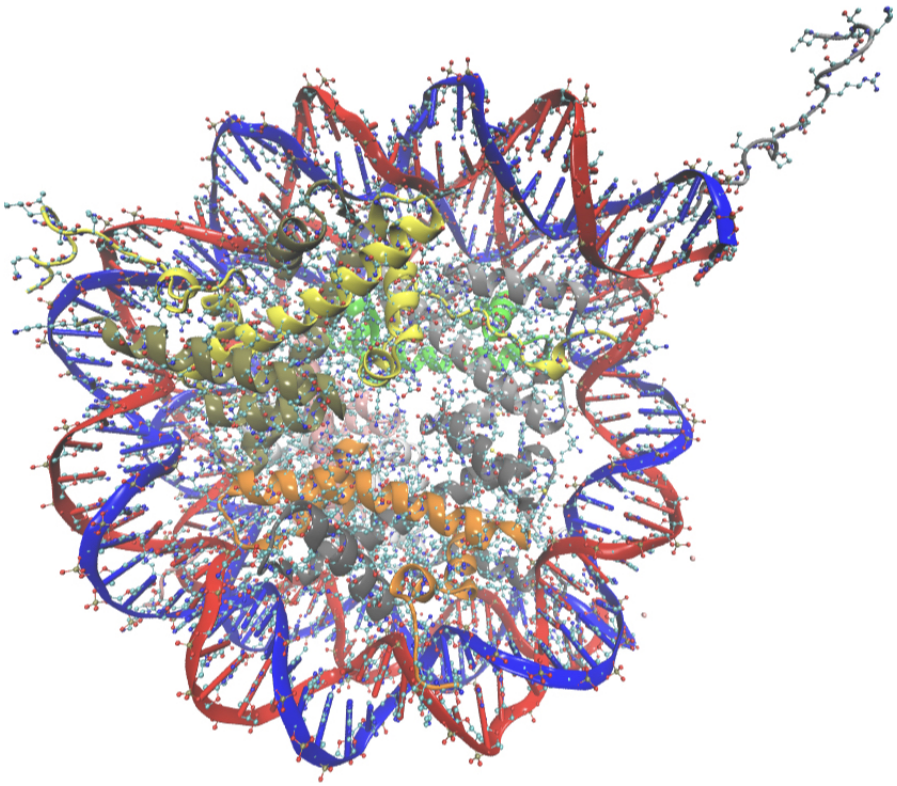


Figure 5: VMD representation of the 1aoi structure. The double brins (blue and red) follow the phorphorous atoms in brown and form a cavity where protein are localized as it corresponds to the depicted alpha–helix in grey, yellow and orange.

2.6 Sixth case: a protein with a specific ligand

Input Files and Executables

- 3S8L.pdb
- AMOEBABIO18.prm
- pdbxyz.x
- xyzedit.x

The next case of this tutorial will concern a classical case in the field of drug design. Indeed, adding non–biological ligands in order to try to inhibit protein's or enzyme's activity (for instance) is one the most studied field in the frame of the MD. Therefore, be able to

combine a protein with a non–usual ligand in a TXYZ become to be interesting, especially when it has been seen before that the pdbxyz.x tinker module does not work to convert non–biological residues. Let us see in this subsection how we can prepare a good TXYZ containing both the protein main system and its ligand we want to add.
The system chosen here is a smooth protein–ligand interaction one named 3S8L.pdb in the PDB data. This system is made off:

- 1 main protein chain A
- a second little chain B named pYAc4cN
- a glycerol (labeled GOL)
- 2 peptides, non–polymere and ions

As you could see, the composition of this system is quite complex, so to limit the complexity of this study we will limit us only to a system containing the main protein chain A and the GOL. In this way, we will see how to prepare the TXYZ corresponding to the interaction between the GOL and the chain A.

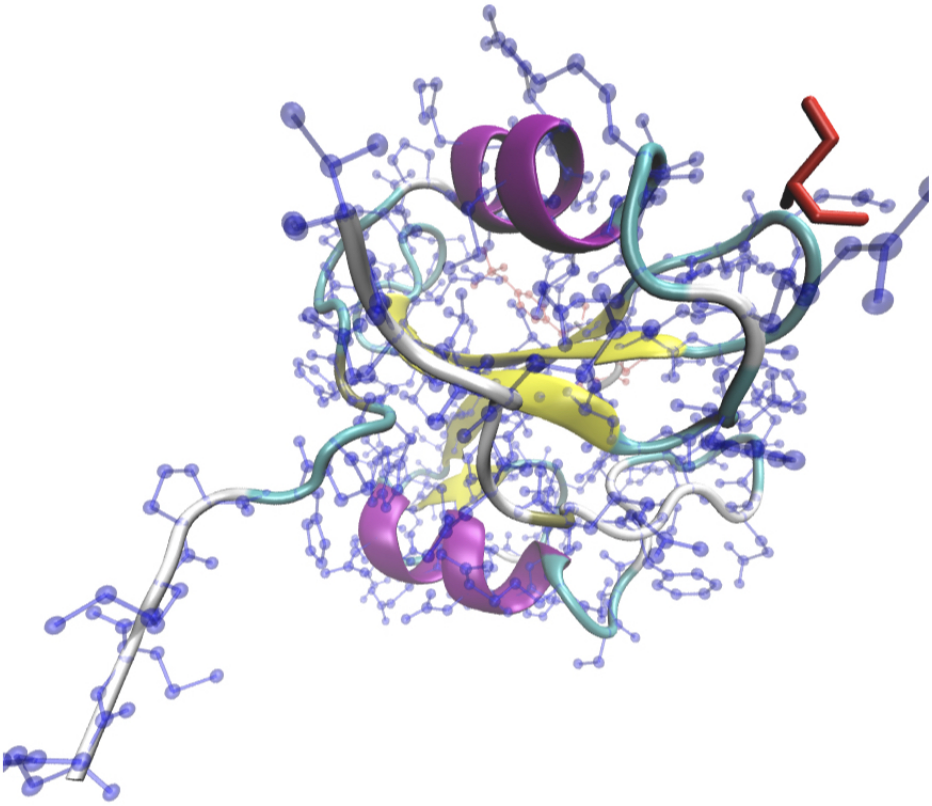


Figure 6: VMD representation of the 3s8l structure. Secondary structure of the protein is depcited with the newcartoon representation, while the non–hydrogen atoms of the chain A of the protein are depicted in blue. The GOL ligand is depicted in red at the right up of the figure.

The first part consists of converting the main protein chain A from PDB to TXYZ, which

is not a hard task as well as it is similar to all the other procedure using pdbxyz.x. Follow theses instructions:

1. Launch the pdbxyz.x tinker module in a terminal
2. Enter the name of the system: 3S8L.pdb
3. Type A as well as you only want here the main protein chain A
4. Ignore the "Set of Alternate Atom Locations" by just typing on enter
5. Enter the force field name: AMOEBABIO18.prm

You well successed to generate the 3S8L.xyz and 3S8L.seq files, containing the chain A of your protein. Now, the hardest task is staying to overcome.

1. To begin, we have to isolate the GOL residue in one pdb file. So create a file named GOL.pdb and place in the coordinates of the GOL residue with:
   - As first line: HEADER GOL RESIDUE
   - Last line: END
2. The next step is now to determine the position of missed H atoms. To perform this step, you can use software such like VMD or even Gaussview as the molecule is small. Note that this is in the same spirit as the third case of this part (cf adding crystal water molecules). When you obtained the desired H atoms position, place them in the GOL.pdb file like:

   ATOM 904 H1 GOL A 2 33.291695 27.695854 1.152159 1.00 0.00 H

   In total, 8 hydrogen atoms have to be added: 1 on each O atom (so 3 in total) and 5 on the C atoms (2/1/2). An example is provided on Figure 7 Finally, by changing "HETATM" by "ATOM" and by adjusting the number of each atoms, your final GOL.pdb should look like:

```
HEADER GOL RESIDUE
ATOM  898  C1  GOL A   2     -21.057 -10.540   0.997  1.00 36.18           C
ATOM  899  O1  GOL A   2     -21.075 -10.310   2.338  1.00 40.30           O
ATOM  900  C2  GOL A   2     -21.199  -9.244   0.244  1.00 40.21           C
ATOM  901  O2  GOL A   2     -19.935  -8.625   0.004  1.00 39.10           O
ATOM  902  C3  GOL A   2     -21.753  -9.676  -1.050  1.00 36.30           C
ATOM  903  O3  GOL A   2     -20.900 -10.567  -1.594  1.00 38.15           O
ATOM  904  H1  GOL A   2     -21.928 -11.120   0.774  1.00  0.00           H
ATOM  905  H2  GOL A   2     -20.085 -10.923   0.765  1.00  0.00           H
ATOM  906  H3  GOL A   2     -20.797 -11.103   2.802  1.00  0.00           H
ATOM  907  H4  GOL A   2     -21.793  -8.535   0.782  1.00  0.00           H
ATOM  908  H5  GOL A   2     -19.236  -9.175   0.365  1.00  0.00           H
ATOM  909  H6  GOL A   2     -22.663 -10.199  -0.843  1.00  0.00           H
ATOM  910  H7  GOL A   2     -21.762  -8.818  -1.690  1.00  0.00           H
ATOM  911  H8  GOL A   2     -20.113 -10.631  -1.048  1.00  0.00           H
END
```

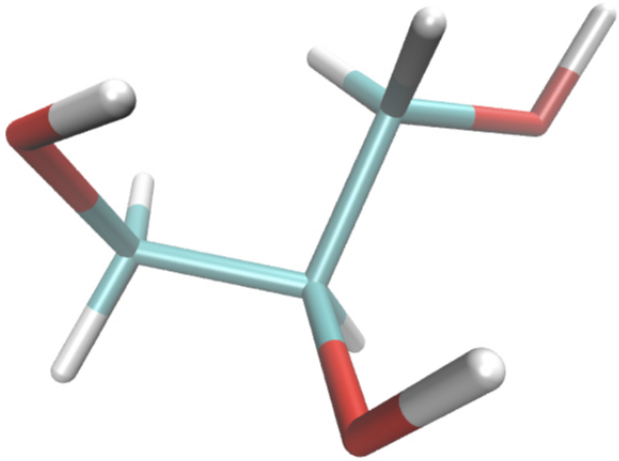


Figure 7: VMD representation of the GLO system with its 8 hydrogen atoms.

3. Finally, we just have to concatenate the both TXYZ file 3S8L.xyz and GOL.xyz. To do that, let use the xyzedit.x tinker module:

   (a) Launch the xyzedit.x tinker module in a terminal

   (b) Enter the name of the protein: 3S8L.xyz

   (c) Enter the name of the force field: AMOEBABIO18.prm

   (d) Select option 18

   (e) Enter the name of the GOL ligand: GOL.xyz

   (f) Type on enter and quit the procedure

   You finally generated the 3S8L.xyz_2, which you can change as 3S8L-GOL.xyz, which corresponds to your final TXYZ system containing the chain A of your initial protein and the GOL ligand.

The non–usual GOL residue

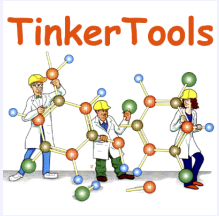
⚠As you surely already guessed, the GOL is not a common ligand, so all its atom type set to be 0 in the TXYZ file. To perform a Tinker–HP study with this file, do not forget to parametrize the ligand according to the AMOEBA protocol of Ren et al., free available on internet. Once done, change each 0 of the atom types of your ligand with the new atom types defined in your parametrization, and you will be ready to perform a Tinker–HP study with such a system.

Output Files

- 3S8L-GOL.xyz

2.7 Seventh case: a DNA structure with non–usual nucleic acids

Input Files and Executables



- 1S88.pdb (frame 19)
- AMOEBABIO18.prm
- pdbxyz.x
- xyzedit.x

The last example of this part of tutorial concerns the procedure for translating a PDB to a TXYZ file when one or several modified amino or nucleic acids are localized in the main chain(s). This procedure is really useful especially when it aims at understanding experimental mutations by use of MD. However, this case is really dependant about what is clearly desired, in this line we will here only be focused on one DNA case where the two main chains contain one non–usual nucleic acid (2DM) (cf Figure 8). Starting from the 19^th structure of

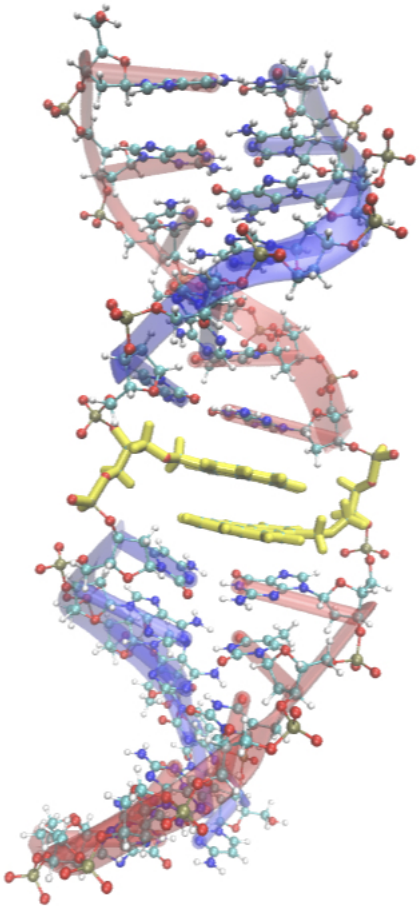


Figure 8: VMD representation of the 1s88 structure. The double brins (blue and red) follow the phorphorous atoms in brown and the 2DM residues for A and B are depicted in yellow.

the 1S88.pdb file (copy this structure file in a new file called **1S88_frame19.pdb**), we start by using the pdbxyz.x tinker module on it with the AMOEBABIO18.prm parameter file (see case 1: the ubiquitin protein for example). The procedure is completely the same as all the other cases of this part. At the end, you must obtain two files:

1. 1s88_frame19.seq
2. 1s88_frame19.xyz

From a first side, it seems to work. However, if you take care with the number of residues, your pdb file indicates that you have 13 nucleic acids per chains while the 1s88_frame19.seq only indicate that 12 nucleic acids per chains have been converted ..
It thus misses one nucleic acid per chain, which corresponds to .. the 2DM non–usual nucleic acid. To add it for both nucleic chains in the TXYZ file, follow these instructions:

1. Copy in a first file labeled **2DM_chainA.pdb** the coordinates of the 2DM of the chain A
2. Copy in a second file labeled **2DM_chainB.pdb** the coordinates of the 2DM of the chain B
3. Apply the pdbxyz.x tinker module on the both files with the force field AMOEBABIO18.prm. You then obtain two new files labeled **2DM_chainA.xyz** and **2DM_chainB.xyz**.
4. Apply the xyzedit.x as follow:
   - Enter the name of the 1S88_frame19.xyz file
   - Enter the name of the force field AMOEBABIO18.prm
   - Select option 18 to concatenate several xyz files
   - Enter the name of the first 2DM file 2DM_chainA.xyz
   - Select again option 18
   - Enter the name of the second 2DM file 2DM_chainB.xyz
   - Just type on enter to quit the executable and then generate the new file 1S88_frame19.xyz_2.
5. Make a copy of the new file and label it as **1S88_final.xyz**.
6. As you could observe, several atom names are missed and x/y/z coordinates are not right. Just add a "H" label in the blanks (as it corresponds to H atoms) and modify their respective x/y/z coordinates according to the initial 1s88_frame19.pdb file.
7. The final thing to do is to adjust the connectivity of each "H" atoms in the 1S88_final.xyz file and so also the respective connectivity for each other related atoms. Here is depicted all lines you must modified (with modifications in bold) for the 2DM owning to the chain A:
   - *For the "H" atoms and their respective C' carbons:*
   - H1'1 ATOM 829: 829 **H 5.173000 -9.034000 21.257000** 0 **808**
   - H1'2 ATOM 830: 830 **H 4.007000 -7.737000 21.533000** 0 **808**

- C1' ATOM 808: 808 C1' 4.145000 -8.714000 21.084000 0 806 809 **829 830**
- H3'1 ATOM 832: 832 **H 1.997000 -9.437000 19.005000** 0 **810**
- H3'2 ATOM 833: 833 **H 1.921000 -8.224000 20.206000** 0 **810**
- C3' ATOM 810: 810 C3' 2.399000 -8.483000 19.282000 0 809 811 **832 833**
- H4'1 ATOM 834: 834 **H 0.413000 -7.052000 19.114000** 0 **812**
- H4'2 ATOM 835: 835 **H 0.386000 -7.307000 17.365000** 0 **812**
- C4' ATOM 812: 812 C4' 0.965000 -6.891000 18.188000 0 811 813 **834 835**
- *For the P atom of 2DM, we need to adjust the connectivity of the 2DM nucleic acid with the previous one:*
- P ATOM 803: 803 P 3.058000 -9.811000 23.265000 0 804 805 806 **165**
- O ATOM 165: 165 O -5.320000 4.149000 17.964000 338 163 **803** (DELETE 185 !)
- *For the O atom of 2DM, we need to adjust the connectivity of the 2DM nucleic acid with the next one:*
- O3' ATOM 807: 807 O3' 4.541000 -9.626000 18.925000 0 809 **185**
- P ATOM 185: 185 P -4.785000 5.953000 23.817000 343 **807** 186 187 188 (DELETE 165 !)

Boring procedure ...

⚠ This procedure appears to be boring, but is mandatory when at least one non–usual amino or nucleic acid is present in a initial pdb file. The VMD software can help you to identify the connectiviy between the 2DM and nucleic residues. This is the same for 2DM in the chain B, so the result is directly done in a file named **1S88_final2.xyz**. However, you can try to find it by yourself as it represents a good exercise to learn how to manage non–usual residues with Tinker and is completely similar in terms of methodology to the previous one.

8. Copy 1S88_final2.xyz in **1S88_final3.xyz**. Finally, delete the " ' " for 2DM atoms which have it. You have finally generate your final TXYZ file. Note that while 2DM is not an usual nucleic acid, it has to be parametrized and thus once it should be done, each "0" will be adjust according to the each respective new atomic number.

Output Files

- 1S88_final3.xyz

3 How to join two fragments to form a ligand ?

Input Files and Executables

- scaffold1.pdb
- fragment1.pdb
- VMD

If you wish to build a new small molecule, generally to serve as a ligand, from two pre-existing fragments, you may face some problems while trying to auto generating a psf file on VMD. Here is a way you can proceed alternatively.

- First, you should load the two molecules that you which to combine.
- Rotate the visualization state so you can be able to see the two concerned atoms in each of the two fragments.
- In order to move manually a fragment to the appropriate position, choose from the VMD main window the following options "Mouse" : "Move" : "Rep". Or click directly the number 9 on your keyboard. This will allow you the rotate freely the two fragments simultaneously and by pressing the "shift" key, you can move and rotate the selected fragment only.

Polar effects

⚠While building your molecule keep in mind to take into consideration the effects of some functional groups having polar effects on their neighbors. However, the minimization and the dynamic that will proceed this step should confer the appropriate geometry for your molecule.

- Once the two fragments are in the desired position to form a single molecule, make sure that you do not have any extra atoms.

If not, you can

- Label these extra atoms
- Open a copy of the PDB file
- Search for these atoms
- Delete them.

In the present example, we merged manually two carbon atoms in order to form one ligand. We chose only one of them (it does not matter which one) to delete in its appropriate PDB file to avoid having exactly two atoms on the same coordinates.

You are ready now to save the new coordinates of each of your fragments in a new file.

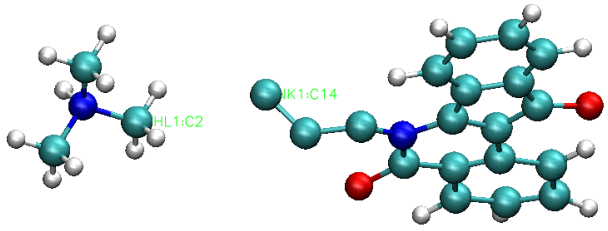


Figure 9: VMD representation of two fragments before forming the final ligands. The carbon atoms to merge are labeled in green. Three extra hydrogen atoms are still bonded to the carbon

- Just press the "File" button in the VMD Main window : "Save Coordinates".
- To proceed safely you can save each of the new coordinates in two different new PDB files and then merge them in a final new file that corresponds to the full new ligand coordinates.

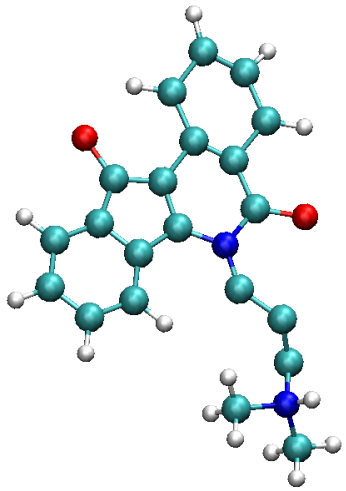


Figure 10: VMD representation of the final ligand structure

Output Files

- ligand1.xyz

4 Solvating and neutralizing a Tinker–HP system: the general procedure

Once the system is translated in the TXYZ format, we have to solvate and neutralize the system. Let see how to perform that with use of tinker modules.

4.1 How to create a water box using Tinker modules ?

Input Files and Executables

- WAT.xyz
- AMOEBABIO18.prm
- xyzedit.x

The first step to be able to solvate a system is to create a box and to fill it. In this case, we will create a cubic box of dimension 60 × 60 × 60 Å, and fill it with water molecules. Before beginning the procedure using the xyzedit.x tinker module, we have to know how many water molecules we must add in the box. Indeed, this choice is crucial for the validity of a dynamic, as well as it has to strictly respect the water's chemical condition which stipulate that it density is equal to **1000 Kg/m^3**.

Calculation tools !

1. Starting from the volumetric mass definition, we can easily determine what is the mass of water in a box of 60 × 60 × 60 Å $= 2.16 \times 10^{-25}$ m^3. We thus obtain:

$$m = \rho \times V = 1000 \times 2.16 \times 10^{-25} = 2.16 \times 10^{-22} Kg \tag{1}$$

2. We can then know how many molecules account for the water box displayed by a volume of 1 m^3. To do that, we just use the easy formula:

$$n = \frac{m}{M} = \frac{N}{N_a}$$

So we obtain:

$$N = \frac{m \times N_a}{M} = \frac{2.16 \times 10^{-22} \times 10^3 \times 6.02 \times 10^{23}}{18} = 7224$$

3. A total number of 7224 water molecules have to be add to the water box to ensure that the volumetric mass is right.

Automatic calculation !

⚠Note that this calculation can be automatically done by using a python code named *vol2mol*, free available on http://chembytes.wikidot.com/yasc#vol2mol

Knowing now the number of water molecules to add in the water box, we can use the xyzedit.x tinker module to create and fill the water box. Please launch this executable in a terminal:

./xyzedit.x

And enter the name of your txyz water molecule (WAT.xyz) and the corresponding force field parameter file (AMOEBABIO18.prm). Select then option 19 labeled "Create and Fill a Periodic Boundary Box" and enter in this order:

- The number of replicas: 7224
- The X/Y/Z box size: 60 60 60
- Just type on enter to allow refinning of periodic box configuration
- The associated force field: AMOEBABIO18.prm

A water box is then created (WAT.xyz_2) and then minimized (WAT.xyz_3). The final water box is thus the WAT.xyz_3 file, which can be renamed as waterbox.xyz.

Output Files

- waterbox.xyz

4.2 How to solvate a system using Tinker modules ?

Input Files and Executables

- ubiquitin.xyz
- waterbox.xyz
- AMOEBABIO18.prm
- xyzedit.x

Once the water box is created, we have to include our system within it, and so delete water molecules. This procedure is automatically done by still performing the xyzedit.x executable. Launch the executable in a terminal and follow these instructions:

1. Enter the name of the txyz file of the system: ubiquitin.xyz
2. Enter the corresponding parameter file: AMOEBABIO18.prm
3. Select the option "Soak Current Molecule in Box of Solvent": 20

4. Enter the name of the water box file: waterbox.xyz

You have created now a new file, ubiquitin.xyz_2, which contains your system included in the water box. Your system is now fully solvated.

Sursolvation !

⚠When the system is big and contain one or many hydrophobic cores, we have to be careful that no water molecules have been placed in these regions. This is so important to check that using VMD and if necessary, translated these molecules outside these regions or directly deleted them.

Size of the water box !

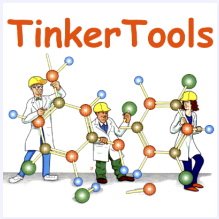
⚠The size of the water box is also a crucial step. Indeed, this is important to ensure that the main system does not interact with its fictive replicas in the PBC conditions. Therefore in all x/y/z direction we need a distance of 10 Å, which is consistent with the Tinker–HP PME set up (an ewald cutoff of 7 Å and a VDW cutoff of 9 Å). To have access to these values, a TCL code named "minimalDIST.tcl" is free available.

Output Files

- ubiquitin.xyz_2

4.3 How to neutralize a TXYZ system ?

Input Files and Executables



- ubiquitin_2.xyz
- 1UAO.xyz
- AMOEBABIO18.prm
- analyze.x
- xyzedit.x (Tinker 8.7)

The last step to do is to neutralize the entire system if its total electrostatic charge is not equal to 0. This step is mandatory since PME has to be use with neutral systems to avoid mathematical divergence.
Thus, by taking of the ubiquitin.xyz_2 file obtained before, the first step is to know the total charge of it in order to see if we have to neutralize the system with ions or not. To have a direct access to this information, please follow these instructions:

1. Open a terminal and launch analyze.x and load you ubiquitin.xyz_2 txyz file and the force field (AMOEBABIO18.prm)

2. Enter the option M (Electrostatic Moments and Principle Axes)

Finally, you have to obtain a total charge of 0.00, which means that your system is totally neutral, and so it is not usefull to add specific ions to neutralize the system.
However, we would like to see here a system which is not initially neutral, so take another txyz file labeled 1UAO.xyz.
You can then perform the same protocol in VMD than the ubiquitin.xyz_2 file. Finally, you could appreciate that your total charge on the screen is equal to -4.00. Therefore, 4 Na^+ sodium ions can be added to neutralize the system. Here is the way to proceed:

1. Launch xyzedit.x and load your 1UAO.xyz file and the force field (AMOEBABIO18.prm)
2. Enter the option 21 (Place Monoatomic Ions around a Solute)
3. Enter the first and last number of the solute (1 138 in this current case).
4. Enter the atom type of the desired ion you want to add (here 352 as it corresponds to the Na^+ atom type in the AMOEBABIO18.prm file) and the number of copies (here 2)

Finally, a file named 1uao.xyz_2 is generated, containing the added ions at the end of the file. You can check if your system is now neutralized by proceeding of the analyze instructions described below.

5 Other available Tinker tutorials

This tutorial does not encompass all the possible systems they could prepare for Tinker–HP but just provide you an additional tool to give help. To complete it, we depict links and name of other already existing tutorials speaking on Tinker–HP.

1. **Chembytes: Build molecular structures for TINKER** http://chembytes.wikidot.com/tnk-tut00
2. **Chembytes: Build a solvent box in TINKER** http://chembytes.wikidot.com/solventbox
3. **AMOEBA advanced potential energies workshop** https://sites.google.com/site/amoebaworkshop/exercise-1
4. **Tinkergpu:Tinker-tut** https://biomol.bme.utexas.edu/tinkergpu/index.php?title=Tinkergpu:Tinker-tut
5. **TINKER** http://www.its.caltech.edu/~rphil/aph150/manuel_tinker.pdf