FoldX Molecule Parametrization

Introduction

FoldX (http://foldxsuite.crg.eu/) is an application that provides a fast and quantitative estimation of the importance of the interactions contributing to the stability of proteins, protein–protein complexes, and protein–DNA complexes. The software recognize only a limited number of molecules in addition to the aminoacids and DNA nucleic acids. The list of recognized molecules can be found at http://foldxsuite.crg.eu/allowed-residues. The parameters of this molecules for stability calculations are included within the software, remaining not-modifiable.
Here we are introducing **ParamX**, a module for the parametrization of novel molecules to be recognized by FoldX software. This new molecules can be parametrized by the user in a single JSon file containing certain atomic and molecular information needed by the software to work and defined below. Also, the FoldX Yasara Plugin was expanded allowing to parametrize molecules both from scratch and taking existing molecules as templates.

Parameterization using JSon files

Json is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types. The { } characters are used to describe a new object consisting of field names and values. The [ ] characters are used to define a list of values assigned to one field.

An empty molecule file

In the website of FoldX an empty JSon file to parametrize molecules is provided, ***empty.json***. Also, a sample molecule parameterization, the ***uracil.json*** file, is available.
At the main level of the json file, there are three main attributes to be defined:
First, the *molName* attribute, which represents the molecule name, is a character string that can be defined freely by the user and is just descriptive. The *molCode* attribute is the three letter code of the molecule within the PDB files that will be provided to FoldX as input. The maximum length of this field is three letters, but can be one or two characters depending of the molecule (i.e. in the provided file with parameters of the uracil molecule, the code is "DU"). This field is case sensitive, which means that upper and lower case letters are considered different characters. Finally, within the *molParams* field, is the list of all the tables needed to parametrize the target molecule.

```
{
    "molName": "Uracyl",
    "molCode": "DU",
    "molParams": [
        {
            "dataType": "atomSolvation",
            "atomSolvation": []
        },{
            "dataType": "hydrogenBond",
            "hydrogenBond": []
        },{
            "dataType": "aminoAcidEntropy",
            "aminoAcidEntropy": []
        },{
            "dataType": "hydrogenPosition",
            "hydrogenPosition": []
        },{
            "dataType": "aminoAcidProperties",
            "aminoAcidProperties": []
        }
    ]
}
```


Figure 1: empty.json file, which contains all the fields that have to be defined to parametrize a new molecule to be recognized by FoldX.

The parameter tables

As is said above, within the ***molParams*** field the list of *tables* representing the different aspects of the molecule that have to be defined to parametrize the molecule can be found. For each table, the *dataType* field indicates FoldX which table is been defined, this field is fixed and the user does not have to modify them. The user only need to insert records for the different tables respecting the fields needed in each case as defined above. The information for each table has to be located within the brackets continuing the name of the table.

The **atomSolvation** table

This table contains all the information concerning solvation and Van der Waals interactions of each atom of the molecule that is being parametrized. One entry per atom is needed.

> **Important information:** The order of the atom parameters is important since other tables use the order of this table to recognize the atoms in the molecule. The first atom defined here will be then referenced as the atom 1 in other tables, continuing with the order consecutively.

This is the field description for this table:

Field	Allowed values	Description
atom	Three letter code of the atom (i.e. "C5", "OP2", etc.)	Atom code

volume	Floating point number	Atom volume
minOccupancy	Floating point number	Minimal volumetric occupancy (unfolded state)
maxOccupancy	Floating point number	Maximal volumetric occupancy (fully buried atom)
vdw	Floating point number	VdWaals unscaled energy
solvationEnergy	Floating point number	Atom solvation energy unscaled
level	-1: LEVEL_O, 0: LEVEL_N, 1: LEVEL_A, 2: LEVEL_B, 3: LEVEL_G, 4: LEVEL_D, 5: LEVEL_E, 6: LEVEL_Z, 7: LEVEL_H, 8: LEVEL_I, 9: LEVEL_K	This atom is O bb (-1), connected bb atoms in the residue (N and C, 0), CA(1), (CB,2), (CG,3), etc... to know when we are at i,i+3. **For non aminoacid-like molecules this value is fixed to 0.**
vdwInternalRadius	Floating point number	Radius used for internal vdWaals and to calculate phi.psi distribution
vdwClashesRadius	Floating point number	United atom radius used to detect VdWaals clashes
isHydrophobic	0: False, 1: True	Atom is hydrophobic
isBackbone	0: False, 1: True	Atom belongs to backbone
isPolar	0: False, 1: True	Atom is polar
cycleNumber	Integer number	This field has to have the same value for all atoms which are part of the same cycle (a numeric value). Value is 0 if doesn't belong to a cycle.
x, y, z	Floating point number	Coordinates of the atom in "standard" amino acid.
omega phi, psi, chi1, chi2, chi3, chi4, chi5, chi6, chi7	Integer number	The dihedral angle the definition is: -1: either angle doesn't exist or atom doesn't move when you change the angle 0:first atom for this residue or last of the previous one (N atom for psi) 1:first atom for this residue 2:second 3:third

		4:fourth 5:fourth atom for this residue or first for the next one (Ca for omega, C' for phi) 6:second atom for the next one 7: first for the next one 8: last of the previous one 9: third of the previous one
lastDihedral	Integer number	Last dihedral that exists for this amino acid
neighbourAtoms	List of atom names, example: ["OP1","OP2","O5'","C5'","O3P","999","999","999","999","999","999","999","999","999","999","999"]	A list of 16 elements, containing which atoms are considered neighbours to this atom and completing if necessary with "999" to fill the 16 positions needed. **Important:** this list must exactly contain 16 elements.

Table 1: List of fields for the atomSolvation table.

In the Figure 2, a dummy atomSolvation table containing only the parameters for the "P" ─phosphorus─ atom (all atoms have to be defined for this table) is shown. The x, y and z coordinates were defined in relation to the C1' atom which is the (0,0,0) coordinate (for further information see the providen file). Which atom is the origin in the parametrization is an election of the user and have no influence in the calculations. The neighbour atoms can be observed in the figure also

```
"molParams": [
    {
        "dataType": "atomSolvation",
        "atomSolvation": [
            {
                "atom": "P",
                "volume": 9.0,
                "minOccupancy": 165,
                "maxOccupancy": 375,
                "vdw": 0,
                "solvationEnergy": 0,
                "level": 0,
                "vdwInternalRadius": 1.37,
                "vdwClashesRadius": 1.49,
                "isHydrophobic": 0,
                "isBackbone": 1,
                "isPolar": 0,
                "cycleNumber": 0,
                "x": 3.486,
                "y": 3.377,
                "z": -2.03,
                "phi": 1,
                "psi": -1,
                "omega": 2,
                "chi1": -1,
                "chi2": 8,
                "chi3": 9,
                "chi4": -1,
                "chi5": -1,
                "chi6": -1,
                "chi7": -1,
                "lastDihedral": 9,
                "neighbourAtoms": [
                    "OP1","OP2","O5'","C5'","O3P",
                    "999","999","999","999","999",
                    "999","999","999","999","999",
                    "999"
                ]
            }
        ]
    },{
        "dataType": "waterRotamers"
```

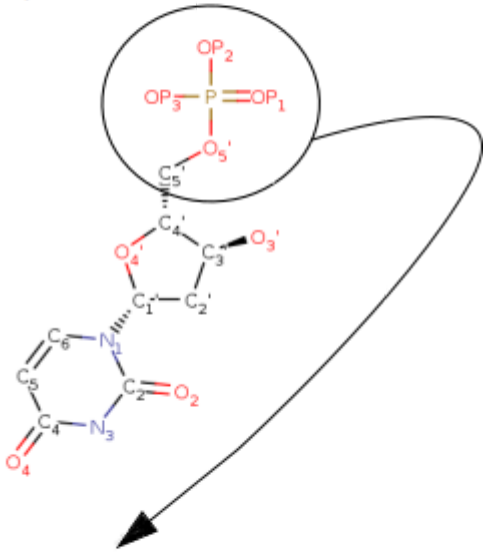



Figure 2: atomSolvation parameters for the P (phosphorus) atom of the uracil molecule. The neighbour atoms of this atom are highlighted and pointed to the json file *neighbourAtoms* parameter, indicating how it have to be defined..

Important information: For drug-like molecules the angles have to be set to -1. The angle flexibility relative to other atoms is defined relatively to aminoacid-like and nucleotide-like molecules and could have unexpected results for other type of molecules.

The **aminoAcidProperties** table

This table defines general properties for the target molecule, which means that only has to have one entry per parameter file.

Field	Allowed values	Description
isNatural	0: False, 1: True	Molecule is natural
molecularWeight	Floating point number	Molecular weight
extinctionCoefficient	Floating point number	Extinction coefficient
maxDeltaG	Floating point number	Maximum value of ΔG

Table 2: List of fields for the aminoAcidProperties table.

The **aminoAcidEntropy** table

The data defined in this table is general for the molecule, it contains the information concerning to the side chain entropy.
This is the field description for this table:

Field	Allowed values	Description
centreAtom	Integer number	Center atom, based on the **atomSolvation** table entry number, zero based.
secondAtom	Integer number	Second atom, based on the **atomSolvation** table entry number, zero based.
radius	Floating point number	Radius of the molecule
moleculeEntropy	Floating point number	Molecule entropy

Table 3: List of fields for the aminoAcidEntropy table.

The **hydrogenPosition** table

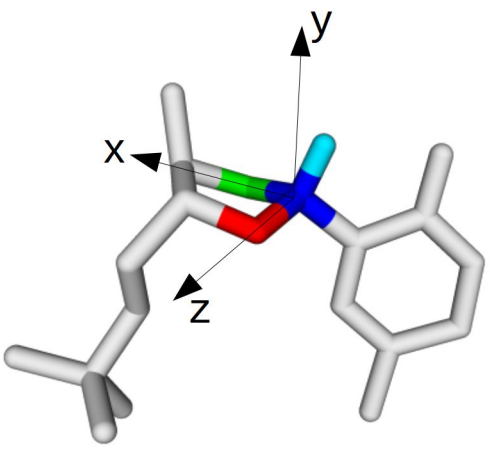
This table contains the information of hydrogen coordinates on backbone and side chain positions relevant for H-bonding.
This is the field description for this table:

Field	Allowed values	Description
atom	Three letter code of the atom (i.e. "C5", "OP2", etc.)	Atom code within the molecule where the hydrogen coordinates are been defined.
atomPartner1	Three letter code of the atom (i.e. "C5", "OP2", etc.)	First atom marked as partner
atomPartner2	Three letter code of the atom (i.e. "C5", "OP2", etc.)	Second atom marked as partner

hydrogenName	Three letter code of the atom (i.e. "H22", "H4", etc.)	Syntax of the H atom name, used to name hydrogens in the output when apply.
isExplicit	0: False, 1: True	Explicit hydrogen
isProtonated	0: False, 1: True	Atom is protonated
isVirtual	0: False, 1: True	Virtual hydrogen
isCarbonyl	0: False, 1: True	Carbonyl hydrogen, is used to locate B pseudoatoms mimicking the free orbital of the O
x	Floating point number	Coordinate x relative to the atom and its partners
y	Floating point number	Coordinate y relative to the atom and its partners
z	Floating point number	Coordinate z relative to the atom and its partners

Table 4: List of fields for the hydrogenPosition table.

This is an example of the hydrogen positions in the C1' atom of the uracil molecule:




```
{
    "atom": "C1'",
    "atomPartner1": "C2'",
    "atomPartner2": "O4'",
    "isProtonated": 1,
    "isExplicit": 1,
    "isVirtual": 0,
    "isCarbonyl": 0,
    "x": -0.487,
    "y": 0.887,
    "z": -0.27,
    "hydrogenName": "H1"
},
```


Figure 3: The atom C1 (blue) which have one explicit hydrogen to define (light blue) builds the coordinate system based in its two partners. The partner1 is C2' (green) and defines the X axis together with the atom C1; the partner 2 is O4' (red) and is used to define the XZ plane as third point. In this constructed space C1 is the origin which is then used as reference to define the three coordinates of the hydrogen.

The **hydrogenBond** table

This table contains the parameters which describe the information for Hbond formation. One or more records have to be defined for all the atoms that can establish hydrogen bonds. This is the field description for this table:

Field	Allowed values	Description
atom	Three letter code of the atom (i.e. "C5", "OP2", etc.)	Atom code
donor	Integer number	Number of donor possibilities
acceptor	Integer number	Number of acceptor possibilities
hydrogens	Integer number	Number of possible hydrogens or waters
dummyHydrogens	Integer number	Number of hydrogens placed by FoldX
hydrogensMovility	1: The H can rotate 2: Coordination of the H is restricted	
doubleBond	0: False, 1: True	Atom makes a double bond
charge	Floating point number	Charge
hydrogenName	Integer number	Atom on which the donor atom is bound, based on the **atomSolvation** table entry. "X" is hydrogen is virtual.
pKa	Floating point number	pKa
isDipoled	0: Partial charge, 1: Not partial charge	
isCharged	0: Polar, 1: Charged	
minBondDistance	Floating point number	Minimal distance for Hbonding
partialCovalentContribution	Floating point number	Default partial covalent contribution
explicitSolvation	Floating point number	Common values: -0.0 when atom field is a hydrogen, -0.4 when atom field a charged atom, -0.2 when atom field is neutral atom, -1.0 when atom field is an ion
bondTolerance	Floating point number	Tolerance for h-bonding. Typical values: 0.0 when

		atom field is a hydrogen, 0.5 when **atom** field is a backbone atom, 0.22 when **atom** field is a sidechain atom, 0.2 when **atom** field is an ion.
hybridization	0: SP2_N_ORB1, 1: SP2_N_H1, 2: SP2_N_H2, 3: SP2_O_ORB2, 4: SP3_O_H1ORB2, 5: SP3_N_H3, 6: NO_HYBRID	Distance deduced from sum of radii to take into account H-bonds

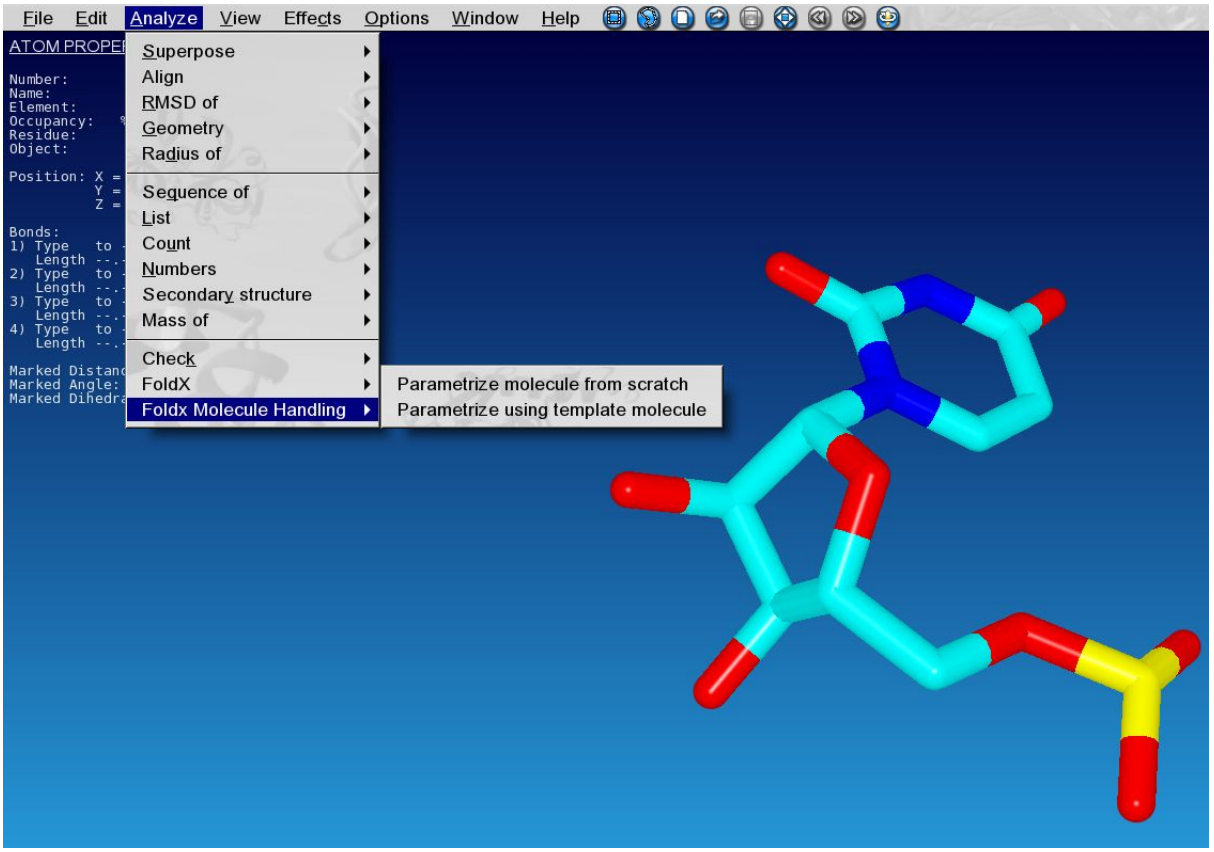
Table 5: List of fields for the hydrogenBond table.

Parametrization using the Yasara FoldX plugin

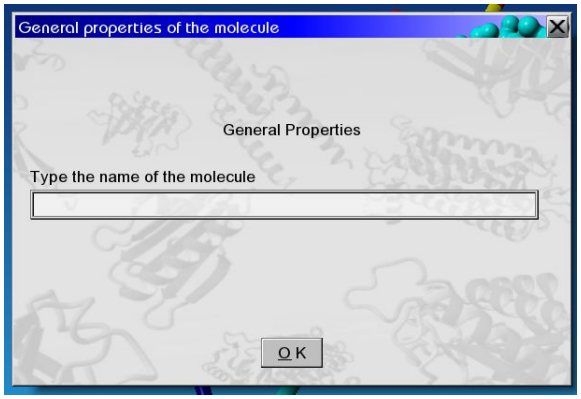
The Foldx Molecule Handling plugin was developed to aid the user in the molecule parameterization in a more friendly way. The only requisite needed to parametrize a desired molecule is to have defined a PDB file containing the molecule: the name of each atom in the molecule has to be consistent with the name of that atom when the target molecule is within a full PDB used as input in a FoldX run.

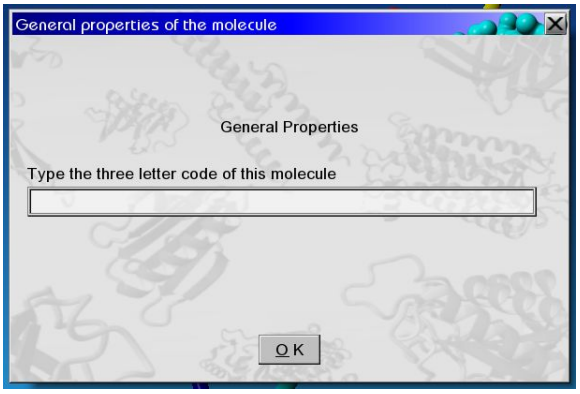
Start a molecule parametrization

After your molecule PDB file is loaded in the Yasara software, the user have to select one option within the menu Analyze -> FoldX Molecule Handling. A molecule can be parametrized from the scratch of taking another molecule already recognized by FoldX as a template. The criteria to copy the parameters if this second option is chosen, will be the to use the same atom name: all the parameters for the atom named X in the existing molecule will be copied to the atom called X in the target molecule.



The only two fields needed to start a molecule parametrization is the molecule name (which is only informative) and the three letter code of the molecule, which has to be consistent with the code used for the molecule in the input PDB files of the further FoldX runs. The number of characters that can be used for the three letter code is one to three.



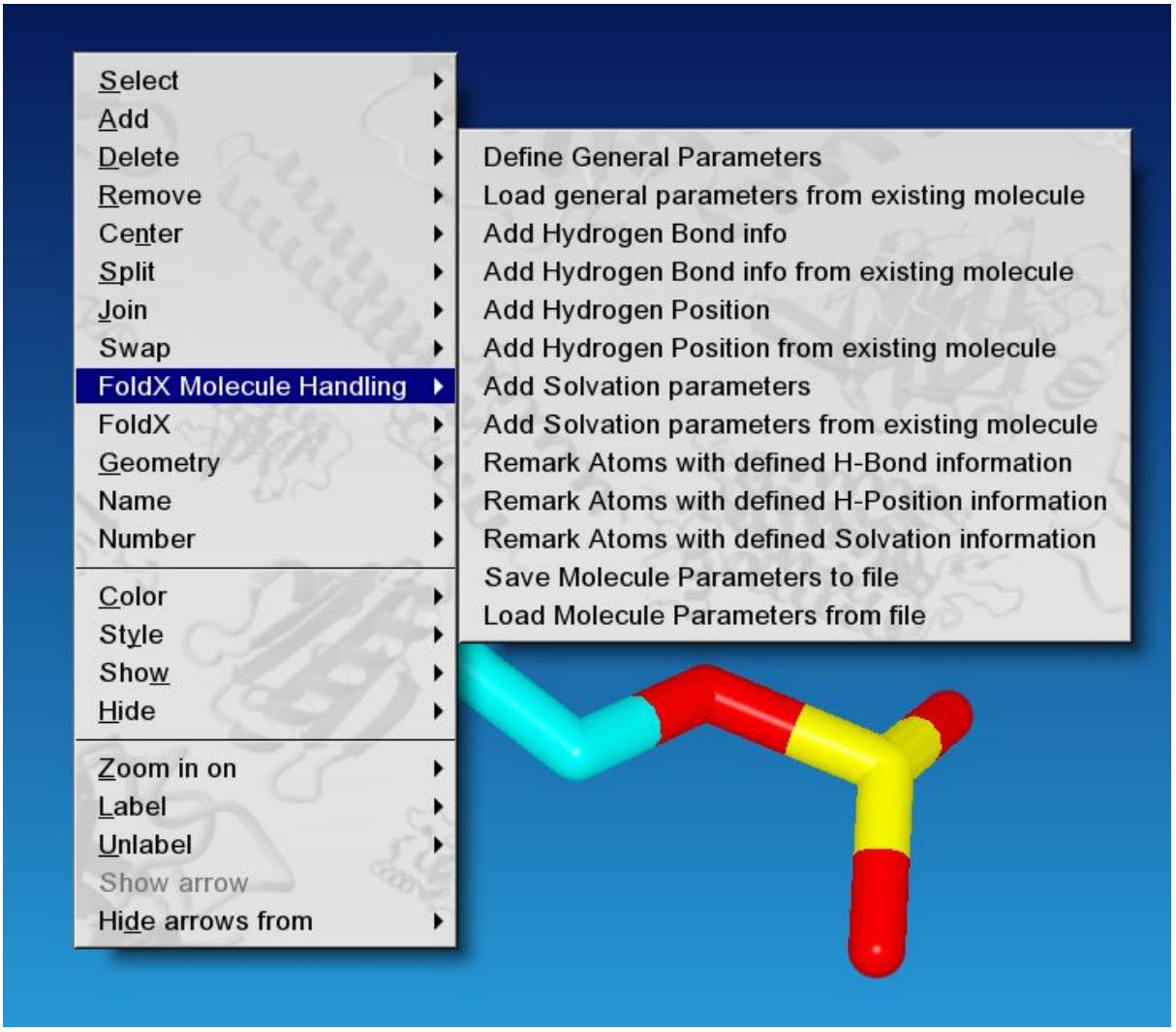


There are two main ways to define the molecule parameters with the Yasara plugin. One is to input them manually field by field, and the other is to copy the target parameters from an already parametrized molecule.

Important information: The rotabase file for the ParamX version has to be actualized. If you have a previous version of this file, please download it from the FoldX website.

Manually define the parameter tables entries

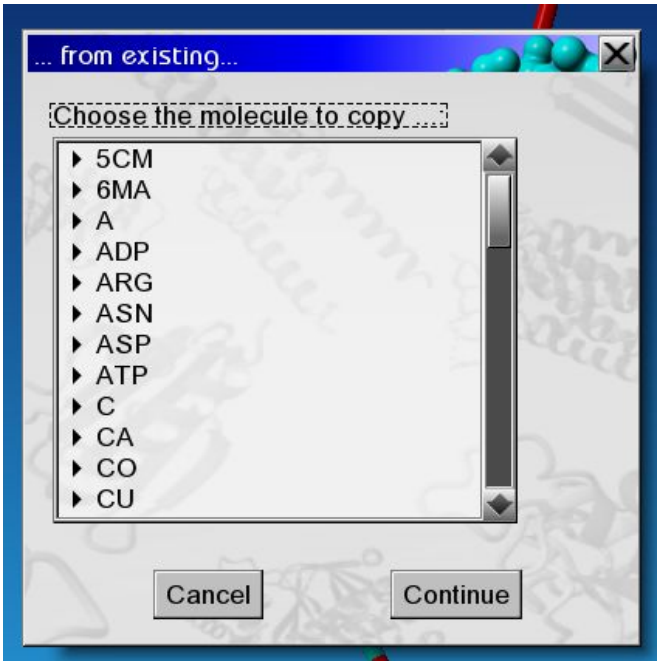
Once the molecule have been named the real parametrization starts. To access the parametrization menu, an atom of the molecule has to be selected, and then right clicking it the "FoldX Molecule Handling" option will display all the actions that can be done.



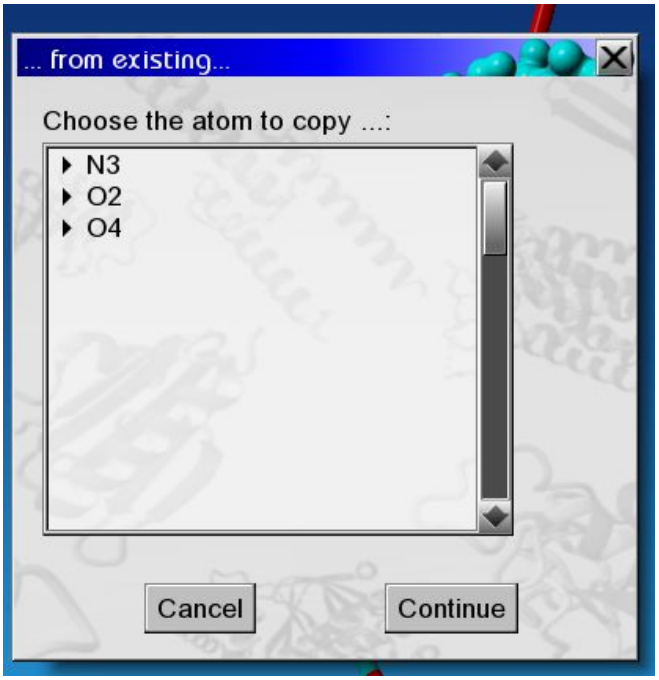
As can be seen, for all the tables described in the first part of this tutorial the option of defining a new entry manually and from existing molecule is defined.
When a manual option is selected, all the fields defined for the selected table will be asked and the parameter will be added to the molecule.

Define parameters based on parametrized molecules

When the user choose and "Add" option from existing molecule, first, all the molecules which have defined at least one parameter for the selected table will be displayed.



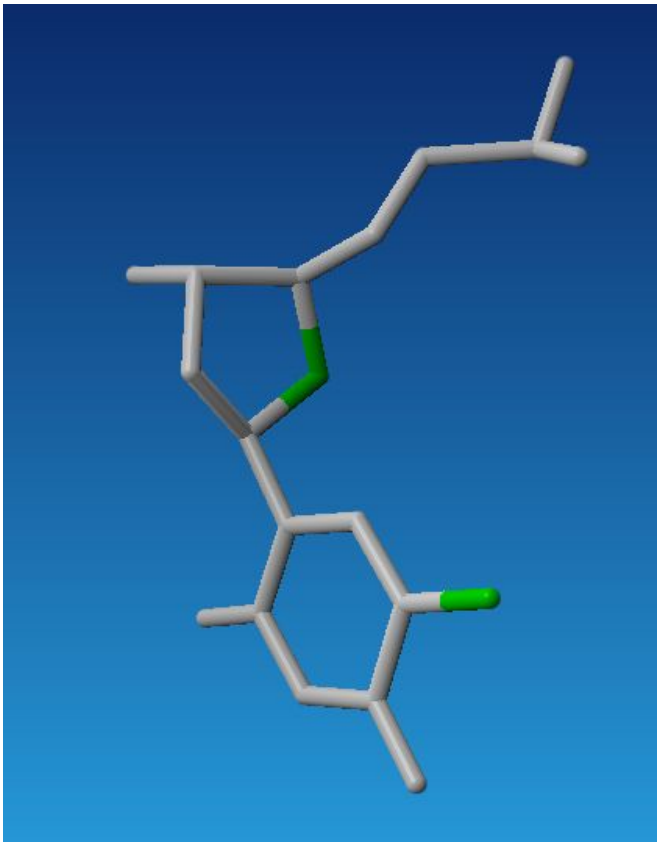
Most of the parameters have to be defined on reference of one atom of the molecule. The selected atom will be the target of the copied parameter, but the source atom could not have the same name in the parametrized molecule, then the atoms with entries in the target table for the selected molecule are displayed.



All the entries in the target table for the selected atom of the previously selected molecule are copied to the target atom of the molecule that is been parametrized.

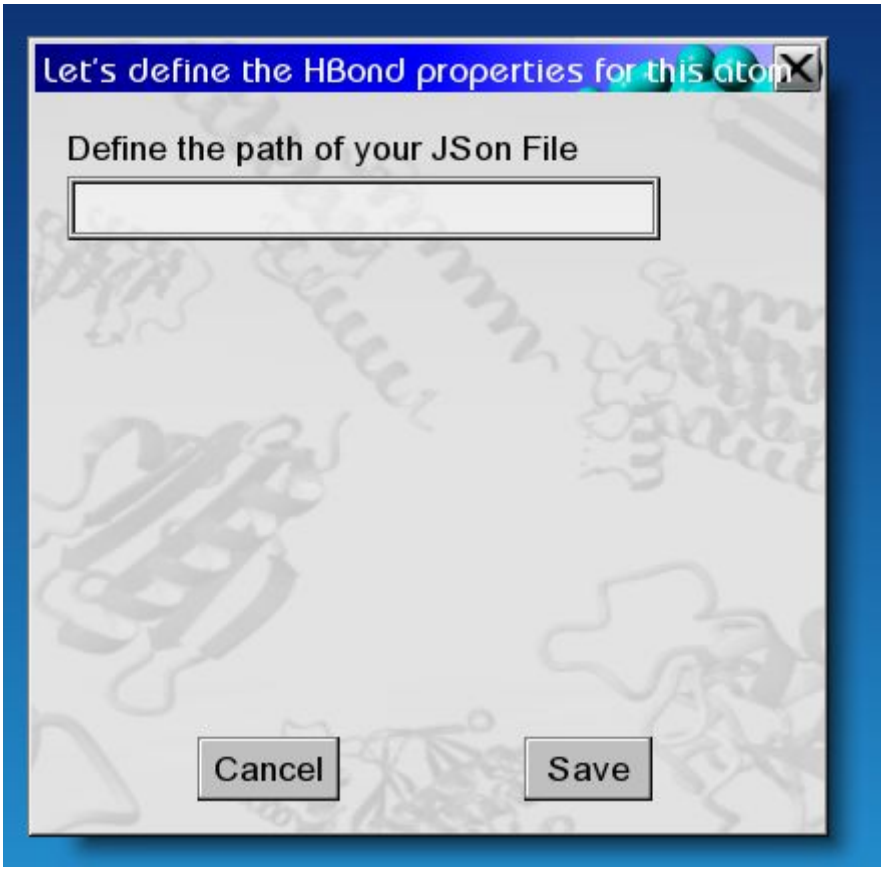
Important information: atomic parameters are copied "blindly", only by replacing the molecule and atom name of the template molecule and atom to the target ones and keeping all the other fields equal. Using this option is under user's responsibility, which have to check and modify any field that is needed to the proper parametrization of the molecule.

The parametrized atoms for the selected table will be colored in green and the rest of the molecule will be white colored.



Saving and loading the molecule parameters

The parametrized molecules has to be stored in a .json file to be processed by foldx software. Select the "Save molecule parameters to file" will display an input box to define the path of this .json file.



There is also the option of load the molecule parameters from a file in order to allow the parametrization of a molecule in more than one session. This option is also useful when two similar molecules are being parametrized, and the molecule parameters of one of them is a good starting point to parametrize the next one.