Electron Tunneling in Proteins Program (ETP) User Guide

Version 1.0

University of California, Davis

One Shields Ave, Davis, CA 95616

ETP Program License

ETP program relays on other third-party programs incorporated as modules, libraries or external programs,

A- Source codes embedded into ETP:

1- Qutemol module (http://qutemol.sourceforge.net/): open source, interactive, molecular visualization system licensed under GPLv2.0.

2- Solvate module (http://www.mpibpc.mpg.de/grubmueller/solvate): a program to construct an atomic solvent environment model for a given atomic macromolecule model (solute) for use in molecular dynamics simulations.

This software is distributed in the hope that it will be useful but *without any warranty*. The author does not accept responsibility to anyone for the consequences of using it or for whether it serves any particular purpose or works at all. No warranty is made about the software or its performance.

Use and copying of this software and the preparation of derivative works based on this software are permitted, so long as the following conditions are met:

- The copyright notice and this entire notice are included intact and prominently carried on all copies and supporting documentation.
- No fees or compensation are charged for use, copies, or access to this software. You may charge a nominal distribution fee for the physical act of transferring a copy, but you may not charge for the program itself.
- If you modify this software, you must cause the modified file(s) to carry prominent notices describing the changes, who made the changes, and the date of those changes. Please do inform the author about major modifications.
- Any work distributed or published that in whole or in part contains or is a derivative of this software or any part thereof is subject to the terms of this agreement. The aggregation of another unrelated program with this software or its derivative on a volume of storage or distribution medium does not bring the other program under the scope of these terms.

This software is made available *as is*, and is distributed without warranty of any kind, either expressed or implied. In no event will the author or his institutions be liable to you for damages, including lost profits, lost monies, or other special, incidental or consequential damages arising out of or in connection with the use or inability to use (including but not limited to loss of data or data being rendered inaccurate or losses sustained by third parties or a failure of the program to operate as documented) the program, even if you have been advised of the possibility of such damages, or for any claim by any other party, whether in an action of contract, negligence, or other tortuous action.

SOLVATE is written by Helmut Grubmüller, Theoretical Biophysics Group, Institut für Medizinische Optik, Ludwig-Maximilians-Universität München, München, Germany. Copyright © 1996-2013. All rights reserved.

B- Libraries distributed with ETP:

3- BALL library (http://www.ball-project.org): a library of algorithms and data structures for molecular modelling and computational structural bioinformatics licensed under LGPL.

4- Openbabel (http://openbabel.org/wiki/Main_Page): which is a chemical tool box used to calculate different chemical properties licensed under GPLv2.0.

5- VTK library (http://www.vtk.org): is an open-source, freely available software system for 3D computer graphics, image processing, and visualization licensed under BSD license.

6- LAPACK library (http://www.netlib.org/lapack):

7- CBLAS library (http://www.netlib.org/blas):

* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the bíogo project nor the names of its authors and contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

8- Python 2.5 (https://www.python.org): Python programming language and interpreter.

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

8. QT 4.8 library (http://www.qt.io): ETP GUI library licensed under LGPLv2.1.

9. Boost libraries (http://www.boost.org) :

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

C- External programs NOT distributed with ETP and should be obtained from their corresponding sites:

9- MSMS program (http://mgl.scripps.edu/people/sanner/html/msms_home.html): it is a molecular surface renderer program developed by Michael Sanner. It can be downloaded at (http://mgltools.scripps.edu/downloads).

10- Dowser program (http://danger.med.unc.edu/hermans/dowser/dowser.htm): a program which locates internal cavities and fills with energetically-favorable water molecules.

11- APBS program (http://www.poissonboltzmann.org): which is an Adaptive Poisson-Boltzmann Solver program designed to calculate the solvation properties.

12- RESP program (http://upjv.q4md-forcefieldtools.org): which is used to calculate the ESP charges based on different constrains.

ETP source code will be furnished upon request made to the sole ETP developer (Muhammad Hagras) at mahagras@ucdavis.edu. Please indicate in your email: (1) your name and title (2) your institute and (3) the purpose to obtain the ETP source code.

Contents

ETP GUI Anatomy

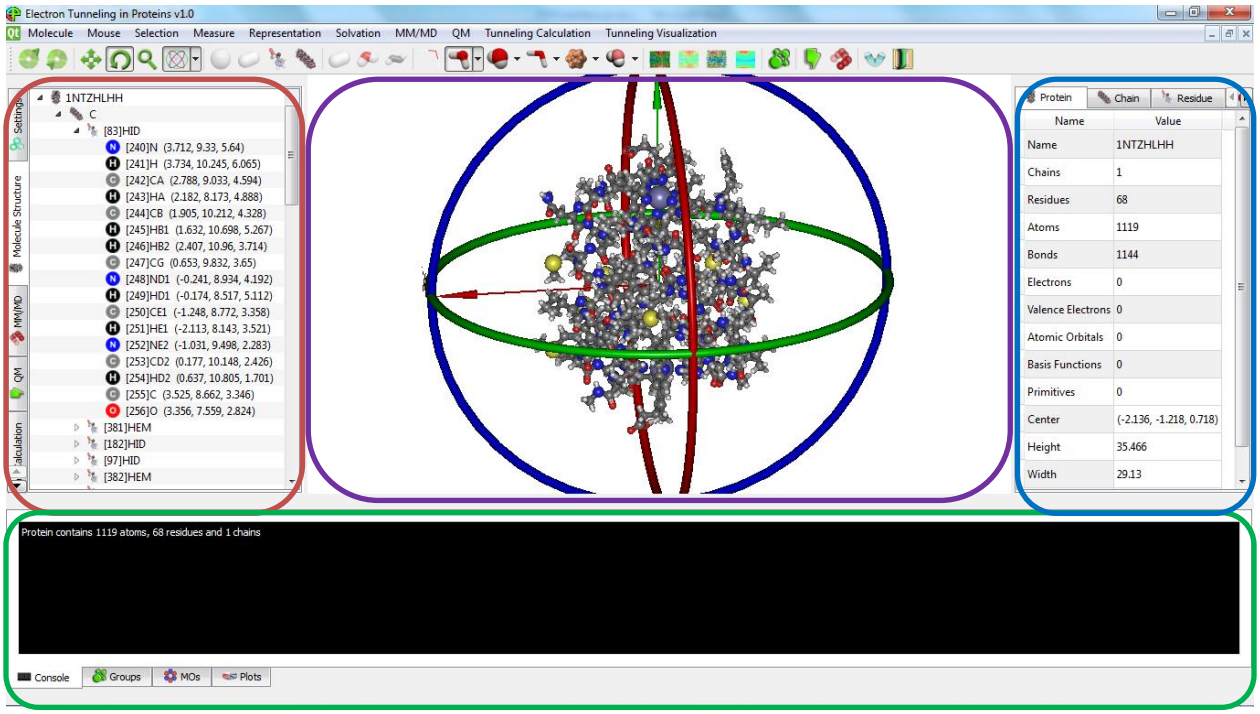
ETP is designed to be user-friendly and to provide versatile means to accomplish the desired calculation tasks. ETP GUI is composed broadly of four panes.

1- The left pane (marked in red) is to display the molecular structure in a tree hierarchy format (where you can pick atoms or residue by clicking on the tree!) and to provide different I/O interfaces.

2- The lower pane (marked in green) is to display the different tasks warnings or outputs, display the structure of the different molecules in a tabular format, show the MOs data and to show the different pruning plots.

3- The right pane (marked in blue) is to display the different picked protein, chain , residue and atom information.

4- The middle pane (marked in magenta) is for 3D visualization.

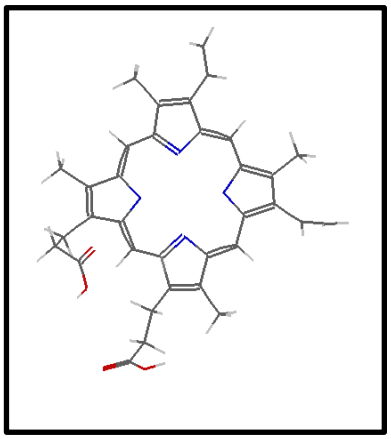


ETP Different Supported Molecular Representations

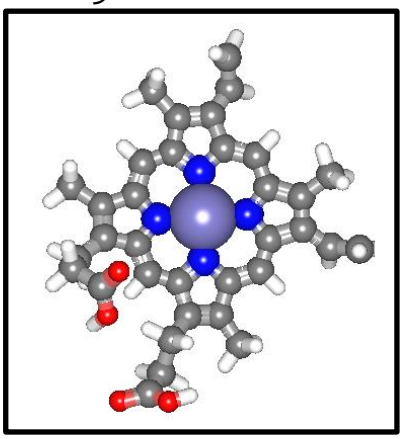
Supported Molecular Representations

ETP is built on two OpenGL engines (Qutemol imposter-based and VTK-based ones). Hence there is multitude of ways to obtain different visualization effects. Currently ETP supports lines, balls-and-sticks, vdw volumes and licorice representations besides the solvent-accessible surface. Currently the secondary structure representation is being developed. You can access the different representations through "Representation" menu or the corresponding toolbar buttons. 

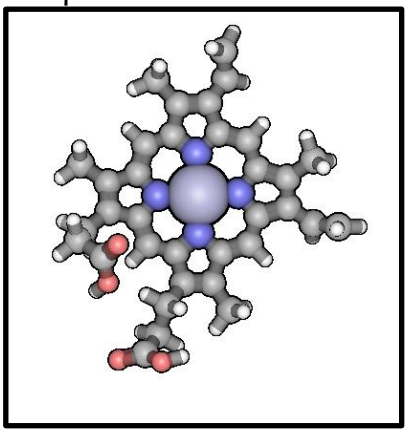
Lines



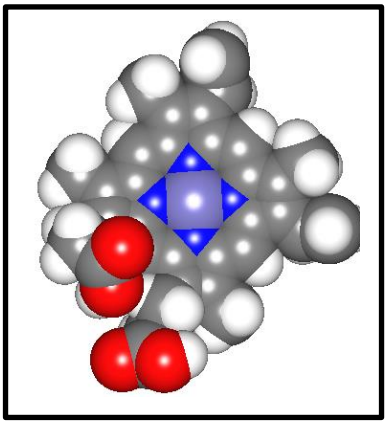
3D Balls-and-sticks



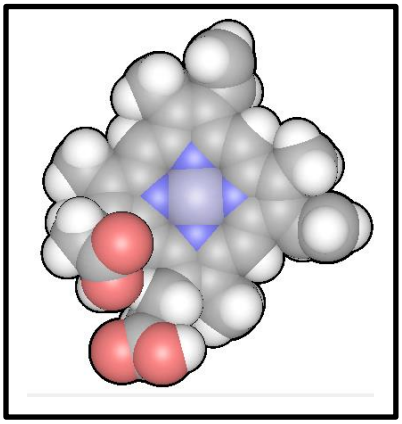
Imposter Balls-and-sticks



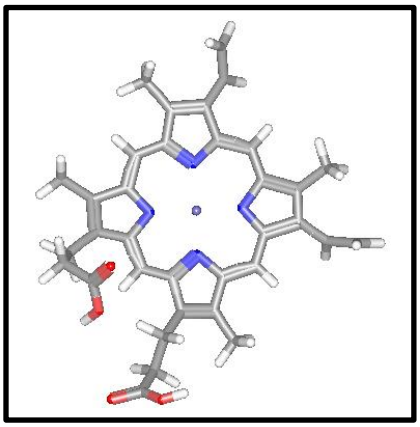
3D vdW Volumes



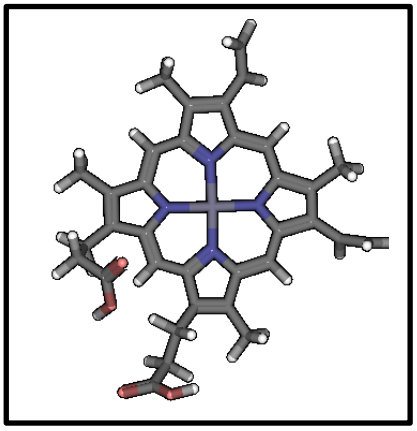
Imposter vdW Volumes



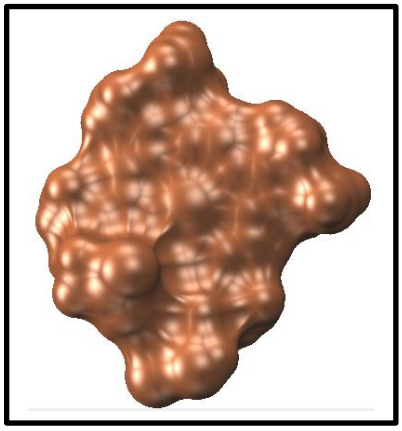
3D Licorice



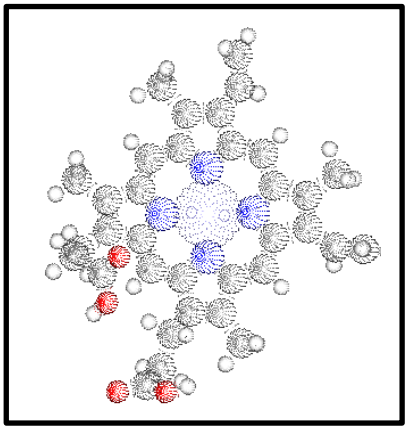
Imposter Licorice



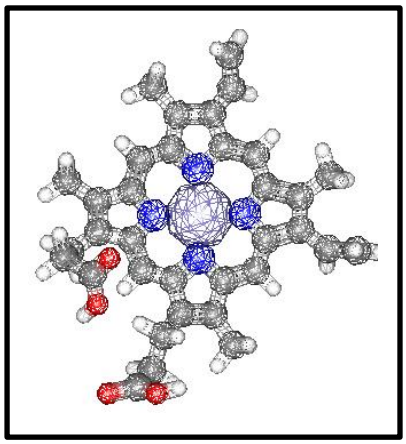
Solvent-Accessible Surface



3D Points

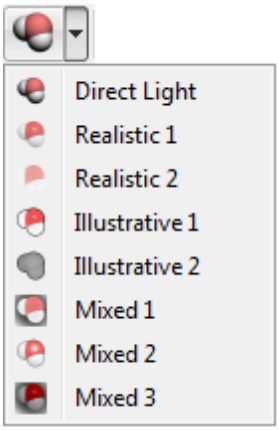


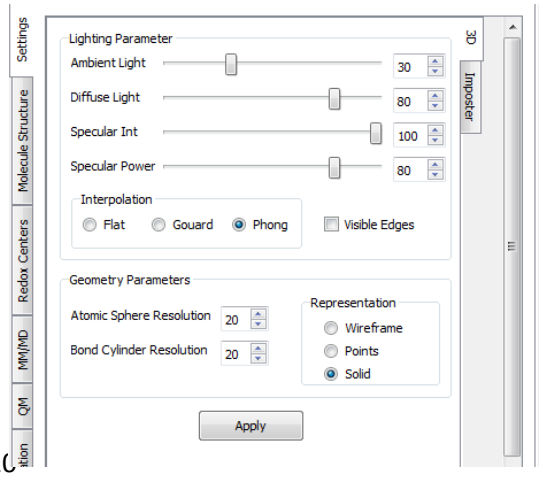
3D Wireframe



Molecular Representation Customization

Each representation can be customized through different "Settings" interfaces and preset imposter settings which you can access using the toolbar or through the menus "Representation=>Molecule=>Settings=>Create Custom" which will open the corresponding interfaces in the "Settings" tab. For SAS, you can access the Settings interface through "Representation=>Surface=>Settings".



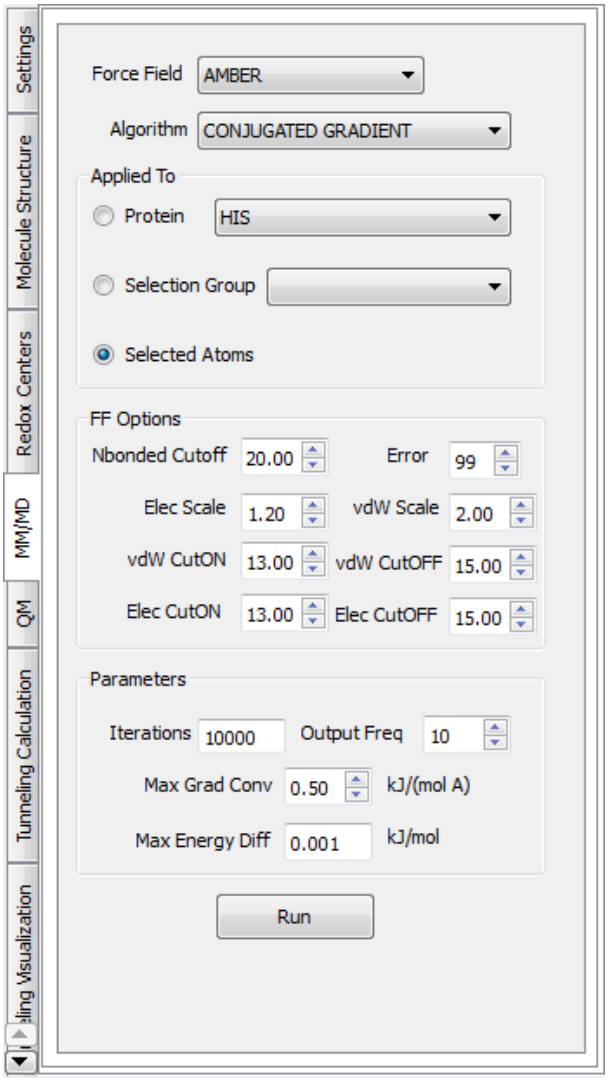


ETP Molecular Mechanics/Dynamics and Solvation Module

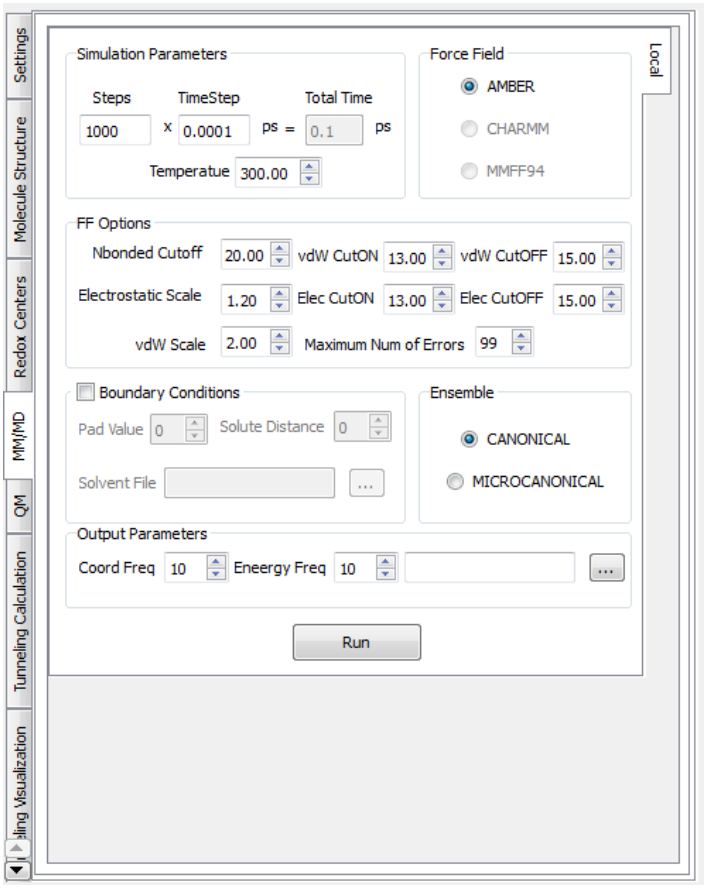
MM/MD Simulation

ETP is integrated with BALL library and hence supports MM/MD simulation through "MM/MD" menu which will open the corresponding interface in the MM/MD tab.

Energy Minimization



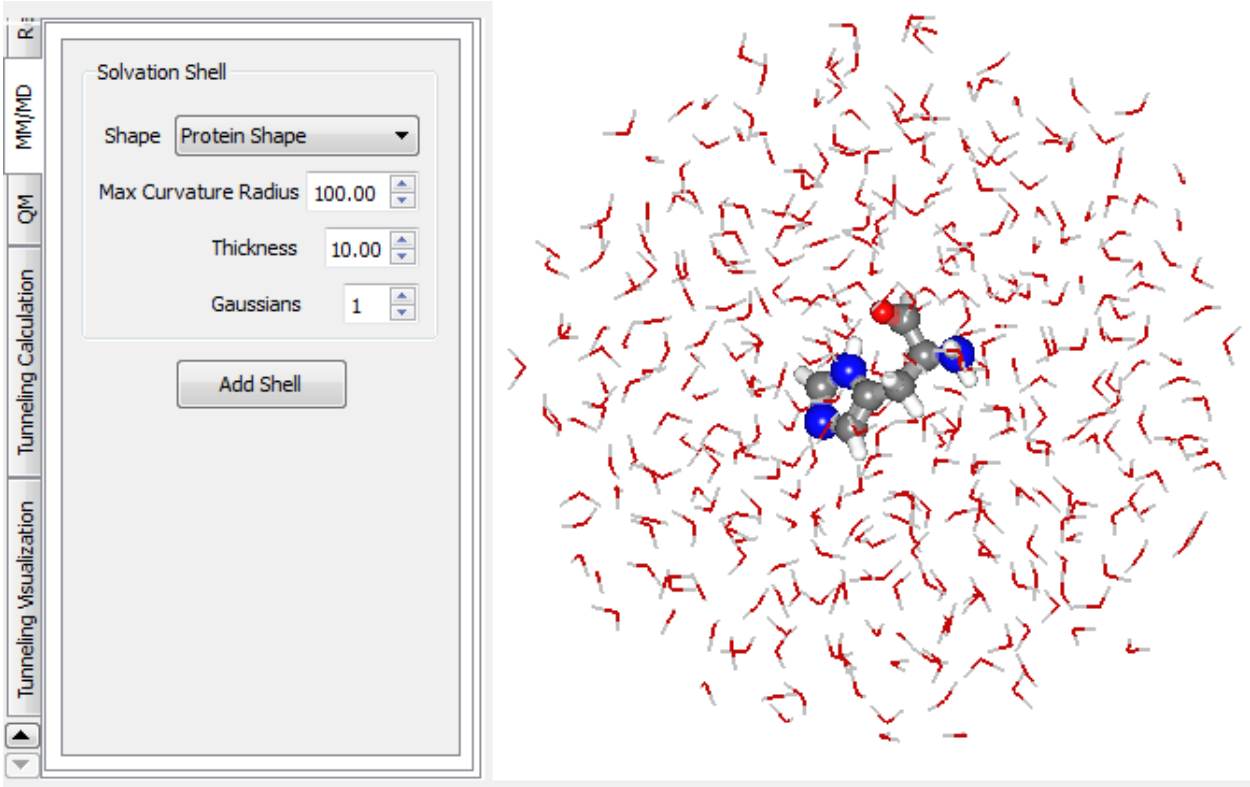
MD Simulation



Note: currently only AMBER force field is supported.

Solvation Module

Solvate program is embedded in ETP program and can be accessed through "Solvation=>Solvate…" which will open its interface.

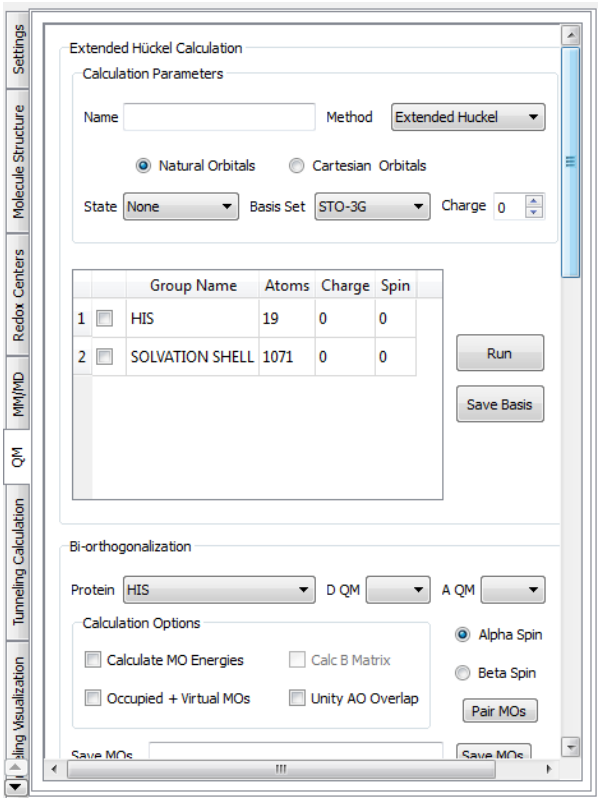


NOTE: Dowser module is still under development.

ETP Quantum Mechanical Calculation Module

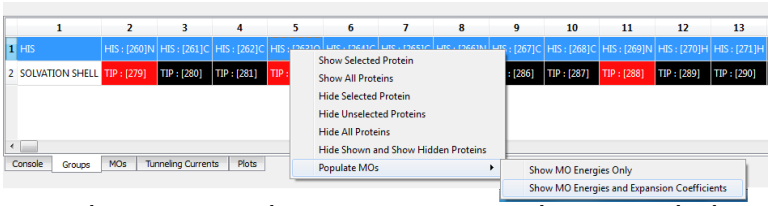
Extended H*ü*ckel Calculation Module

ETP has a full operational extended hückel module that you can access through "QM=>Electronic Structure Calculation…".



1- After selecting the desired input parameters (basis set, charges, and the molecule) hit the Run button.

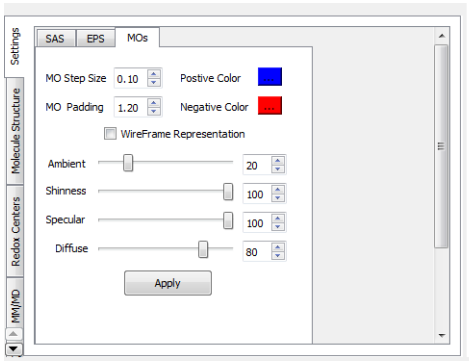
2- Go to the output pane (at the bottom) and right click and choose Populate MOs.

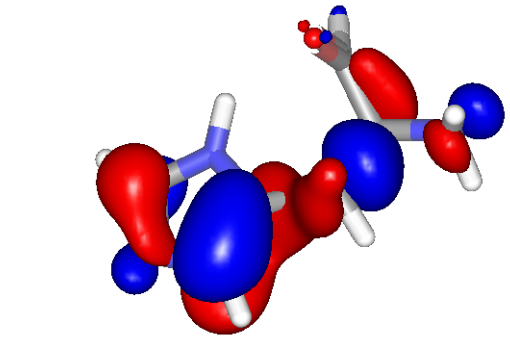


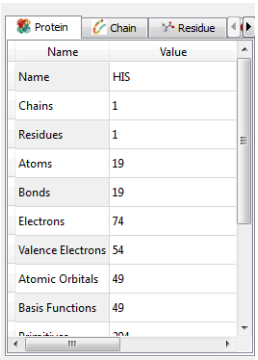
3- Then go to the MOs output tab to read the different MO energies (and expansion coefficients).

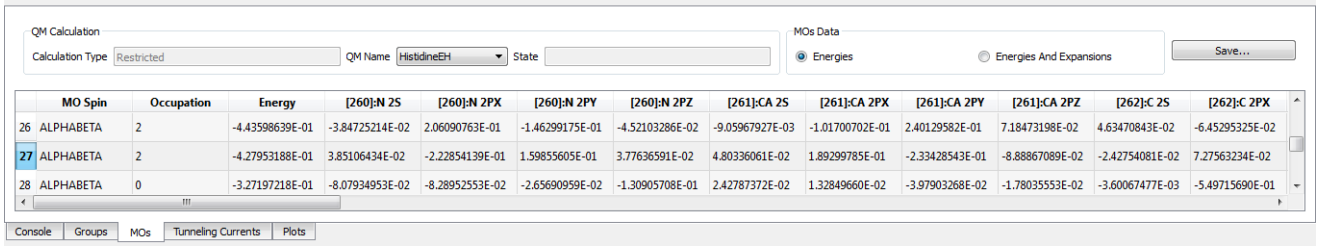
4- By clicking on any MO row, the corresponding MO will be rendered. Clicking again will hide it.

5- MOs rendering can be customized through "Representation=>Surface=>Settings=>MOs tab"



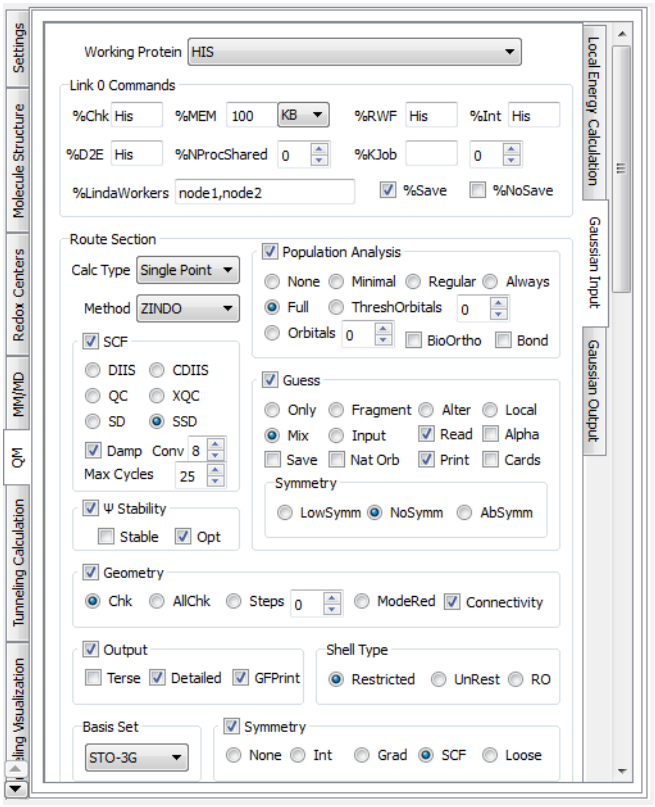


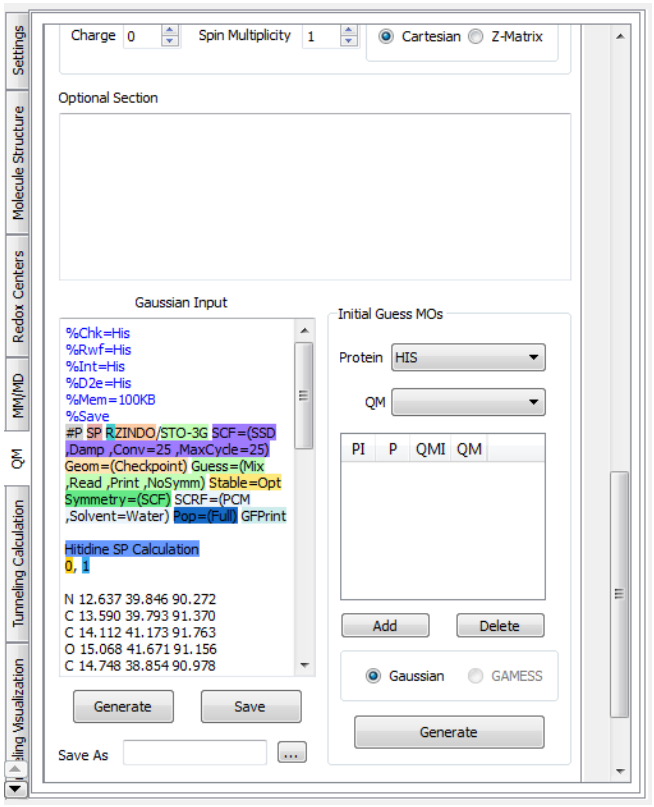




Gaussian Input Generation Module

ETP is equipped with a QM interface that automatically generates Gaussian input file by selecting the appropriate options.

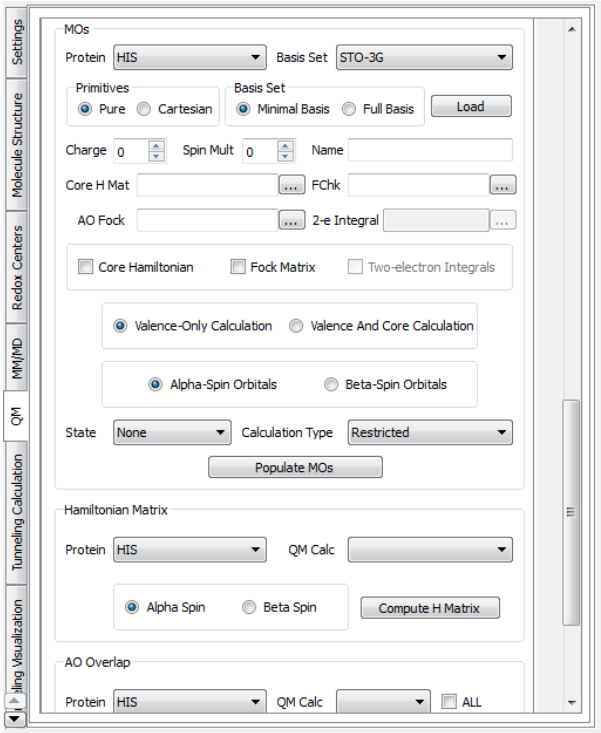




The generated Gaussian input can be saved into a file that needs to be directed into Gaussian program.

Gaussian Output Reader Module

In addition, ETP reads different Gaussian output files including the geometry optimization output, formatted checkpoint files, core Hamiltonian and AO Fock matrix output files.

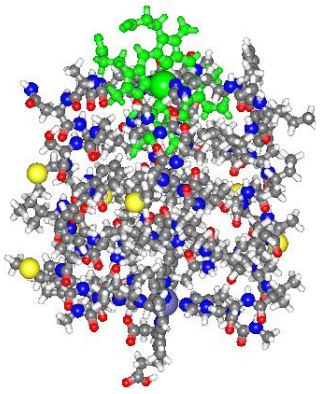


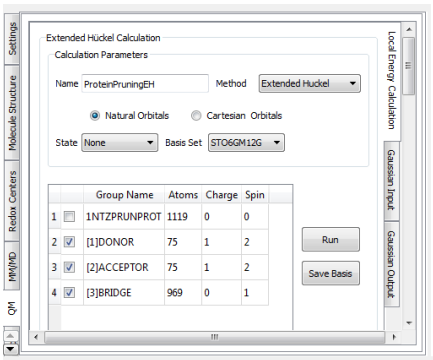
ETP Tunneling Calculation Module

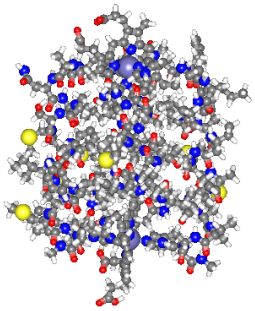
Tunneling calculation is a lengthy multi-tier procedure that basically falls into two categories:
1- Protein pruning, and **2- Electron tunneling pathway calculation.**

Protein Pruning

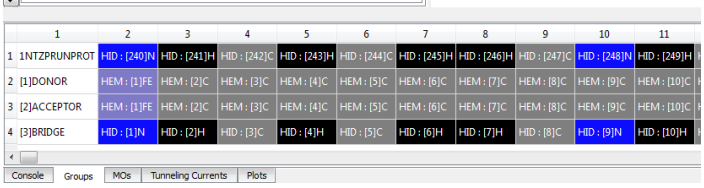
1- Pruning starts by selecting the donor, acceptor, and bridge parts into new groups through "**Selection=>Groups**" or the "**Groups**" toolbar button.

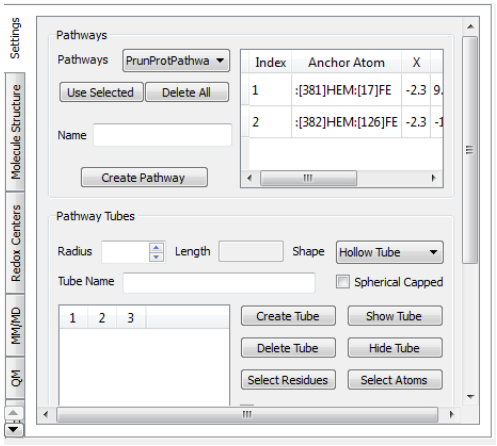






2- Then run EH calculation on all three new groups (Donor/Acceptor/Bridge).





3- Next we need to analyze the different calculation outputs using the MOs output pane as discussed before and select the appropriate tunneling MOs.

4- Then you need to create Donor=>Acceptor Pathway through "**Tunneling Calculation=> Pruning=>Tunneling Tube**" or "**Tunneling Tube**" toolbar button.

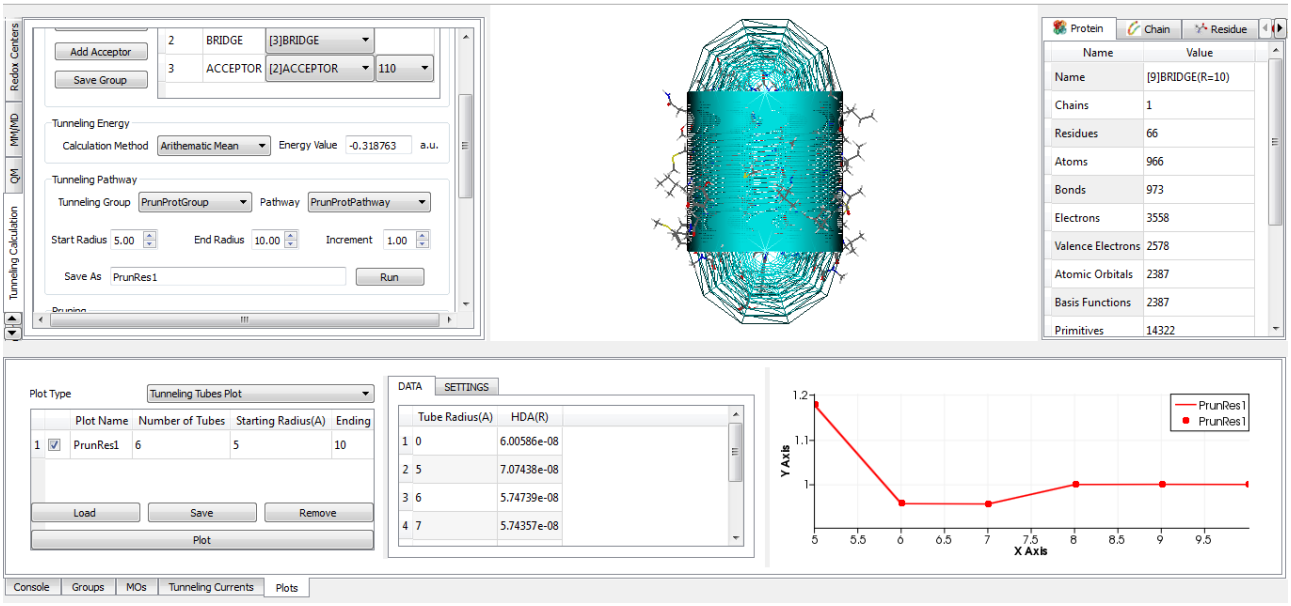
5- Afterwards, we need to create tunneling group through "**Tunneling Calculation=> J Calculation**" or through "**J Calculation**" toolbar button.

6- Select the calculation method to obtain the tunneling energy (Currently it is a mere average of the donor/acceptor MO energies).

7- Select the starting, ending radii and the increment value, name for the calculation then hit Run button.

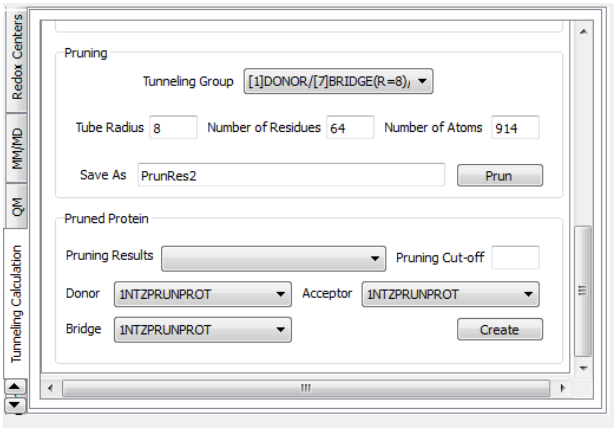
8- After the calculation ends (when the program begins responding again!), the desired number of tubes with different radii will appear and different protein groups will be created in the groups output pane.

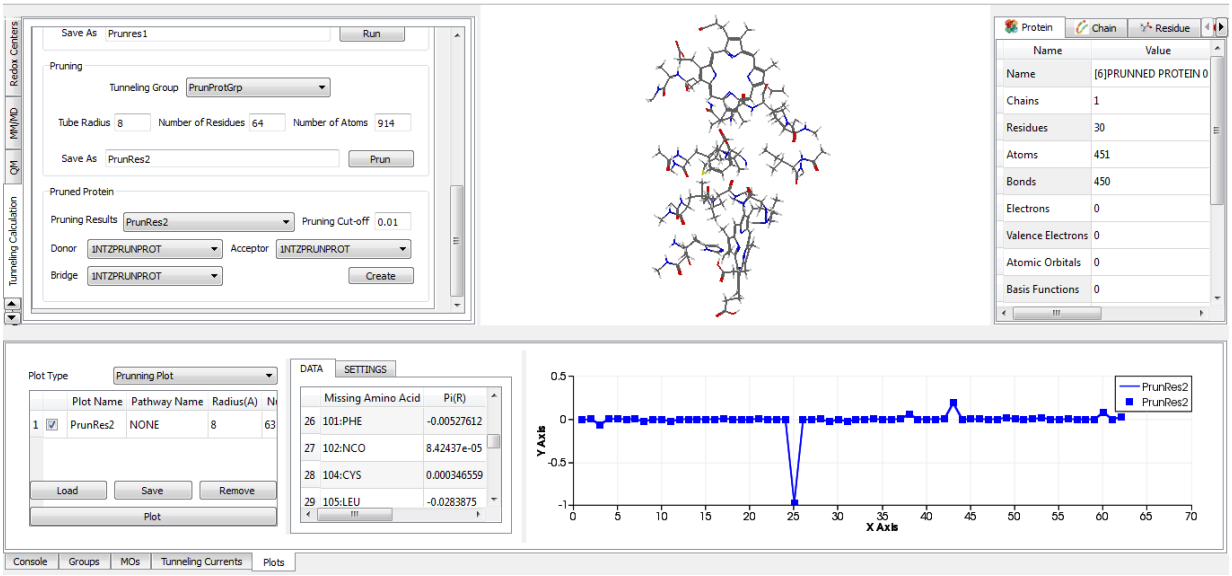
9- You can analyze the results in the plot output pane.



10- You can hide those tubes in the "Tunneling Tubes" dialog. You need to refresh the "Pathways" combo and then select all the tubes and hit "Hide Tube"

11- By analyzing the results, tube with radius 8A seems reasonable. Choose the corresponding group and hit Prun button.





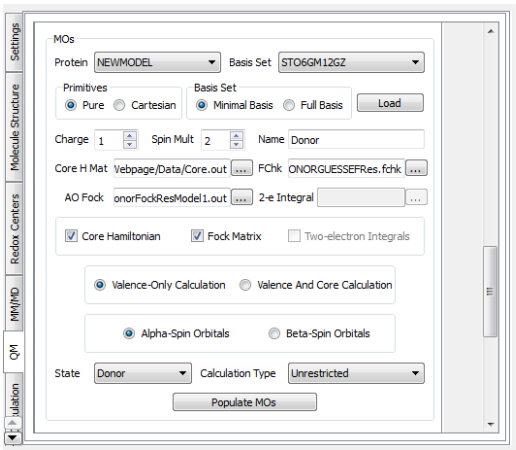
13- The results can be plotted as shown above.

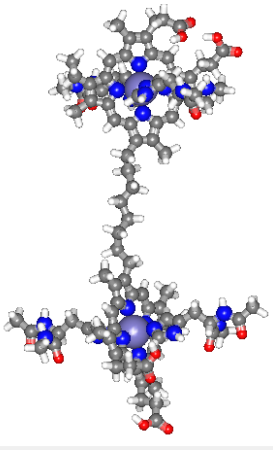
14- To create the final pruned protein, you need to specify the donor/acceptor and the bridge molecules along with the Pi cutoff value then hit "Create" button. The pruned protein is shown above. Observe that number of atoms is halved in the pruning process.

Electron Tunneling Pathway Calculation

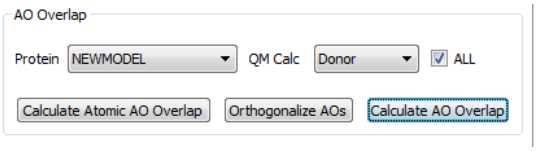
14- Afterwards, you need to run Gaussian job on the pruned protein. You can use the Gaussian interface to generate that input file as discussed above.

15- Then you need to input the generated Gaussian donor and acceptor output files into the Gaussian output reader module as described above and shown below.





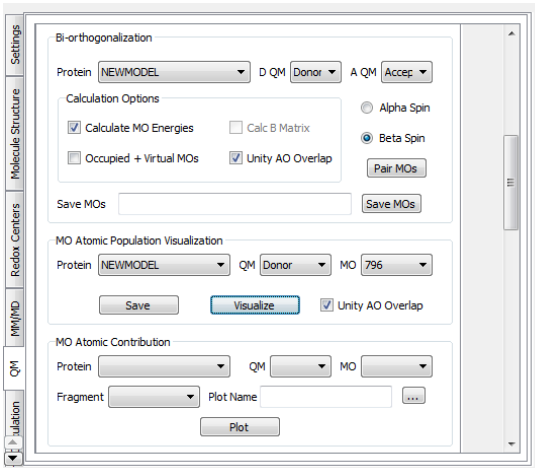
16- Then, you need to calculate AO overlap matrix and then the orthogonalization matrix.

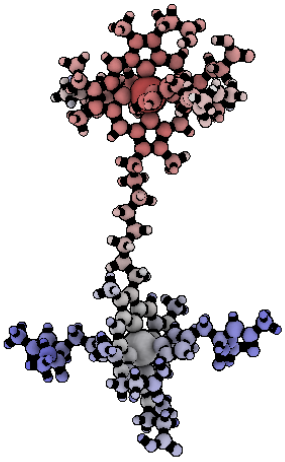


17- Then, you need to apply the bi-orthogonalization scheme.

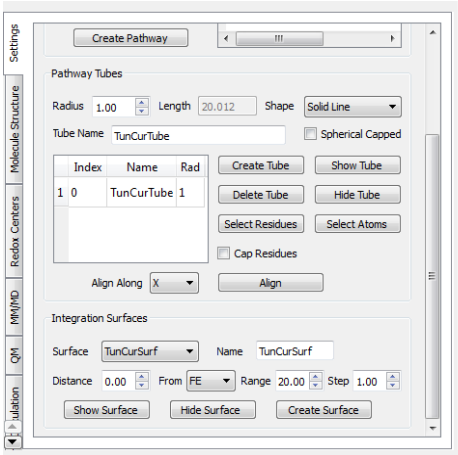
18- Afterwards, you will obtain in the Temp folder a csv file for the overlap values and the MO energies.

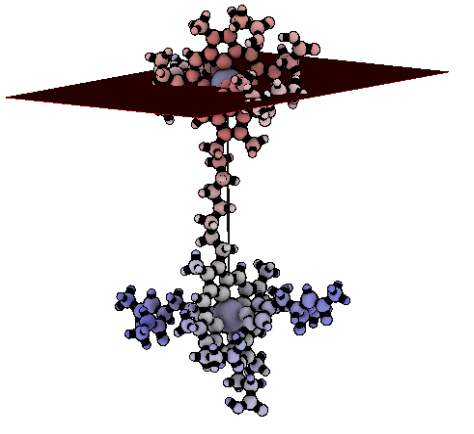
19- You can visualize the atomic population of the different MOs.





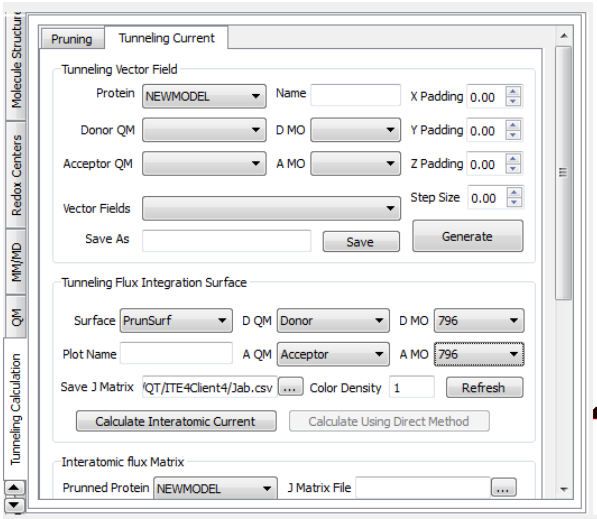
20- Then you need to create pathway, solid tube and integration surface.

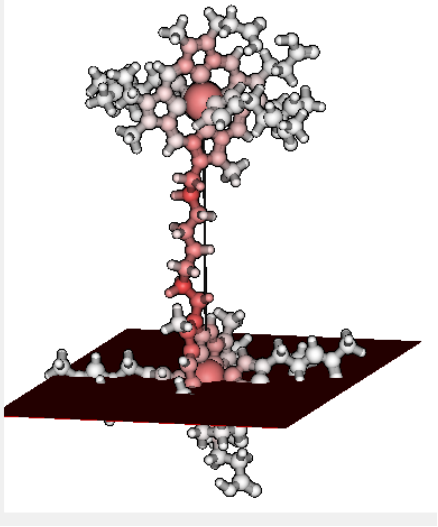




21- Then you need to go to "Tunneling Calculation=>J Calculation" or press "J Calculation" toolbar button.

22- Choose the integration surface, donor/acceptor QM calculation and the tunneling MO indices. Set the color intensity and hit the "calculate interatomic current" button.





23- You will see the surface moving along the tube and the pruned protein colored.

24- An integrals.csv data file will be generated at Temp folder with the tunneling flux results along the pathway.

Notes: 1- Coloring schemes for tunneling current and atomic population work best in the imposter mode. An issue which is currently under investigation.

2- Starting from step 14, we switched to another pruned system for the sake of simplicity but obviously you can continue with the one generated at step 13.

3- The generated file integrals.csv in the Temp folder contains the final desired product which is the tunneling flux element.

ETP Tunneling Currents Visualization Module

ETP is equipped with 4 visualization schemes for the tunneling currents (3D vectors, hedgehog, streamlines and 3D volumes).

1- You need first to calculate the tunneling flux density at certain grid points as shown to the right.

2- Afterwards, you have to switch to the desired visualization scheme through "**Tunneling Visualization=> J**" or through the corresponding toolbar button as shown below.