EHPath 2.0 instructions

The EHPath program uses python libraries to analyze charge transport pathways. The charge (electron or hole) donor, bridge(s), and acceptor(s) need to be defined respectively in *csv* files. For more information on the EHPath program, please refer to [1] SI Appendix S4 Description of EHPath.

The revamped EHPath 2.0 (or EHPath multirun) package (`EHPath_multirun.py`) is an extension of the EHPath program which allows a user to analyze such pathways with a large number of input files.

A run of EHPath 2.0 requires a command like the following:

`python path/to/EHPath_multirun.py` {*pdb_list*} {*cutoff_num*} {*total_paths*} {hole or electron} {*alpha_reorg*} {*path/to/main_dir*}

An example run would be:

```
python EHPath_multirun.py pdb_list 4 1 hole 1 1rrq6D273/HT
```

The *pdb_list* is located in the *main_dir* and each row contains a protein pathway to be analyzed. An example *pbd_list* looks like this:

```
1rrq31
1rrq32
1rrq33
1rrq34
1rrq35
1rrq36
```

For each pdb in the *pdb_list*, `EHPath_multirun.py` will call either `EHPath_single_electron.py` or `EHPath_single_hole.py` depending on user input to run the original EHPath.

The input files should be located in an *input* folder within the *main_dir*. The donor node *csv* file should be named 'Donor_Nodes_' + pdb + '.csv', the bridge nodes file should be labelled 'Bridging_Nodes_' + pdb + '.csv' and the acceptor node file 'Acceptor_Nodes_' + pdb + '.csv'. The detailed specifications of these *csv* files can be found in [1] SI Appendix S4 .

After a successful EHPath multirun, the output from each pdb will be individually saved in the *output* folder within the *main_dir*.

We used the following folder organization layout:

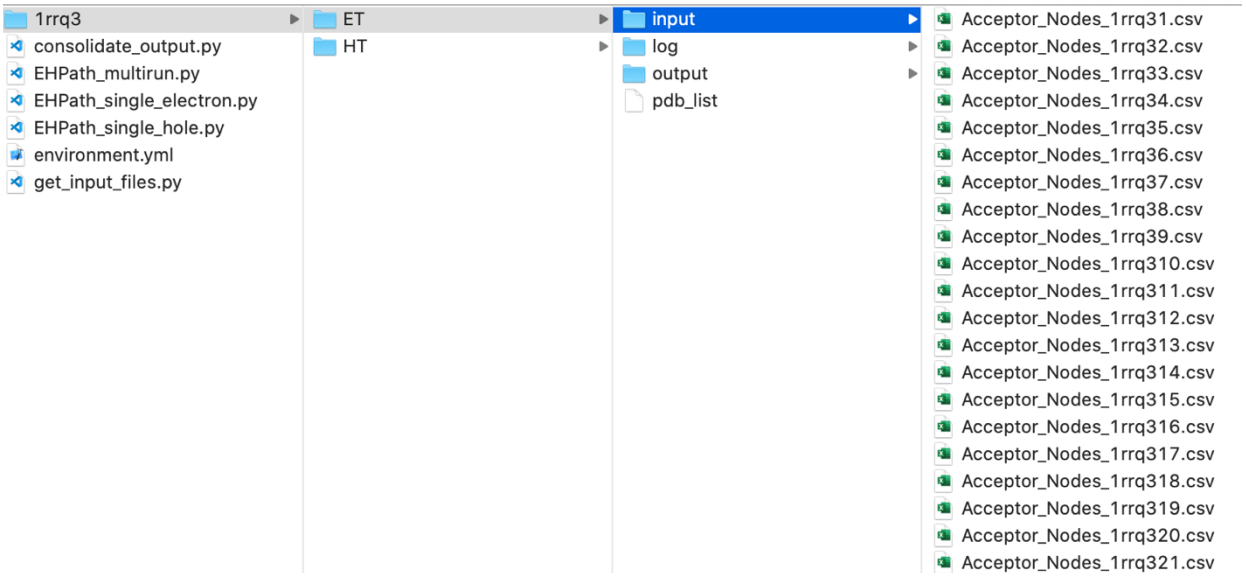

Fig. 1 Sample folder layout of an EHPath 2.0 production run. All the Python files are located in the top-most directory. Electron transport (ET) and Hole transport (HT) paths of the same protein

are put into separate folders and labelled as such. Each of the ET and HT folders contains a *pdb_list* of all the pdb to be analyzed and an *input* folder with the corresponding donor, bridge, and acceptor node *csv* files.

To make it easier for users, a `get_input_files.py` script can be used to create the *pdb_list*. In the folder containing the `EHPath_multirun.py` script, use the bash command:

`python get_input_files.py` {*protein*} {*pathway* (HT or ET)}

An example is:

```
python get_input_files.py 1rrq6D273 ET
```

To consolidate all the output files, we can use the `consolidate_output.py` script:

`python consolidate_output.py` {*protein*} {*pathway* (HT or ET)}

An example is:

```
python consolidate_output.py 1rrq6D273 HT
```

EHPath multirun is also parallelized to enable speed-up when run on an HPC cluster. A Conda environment file (*environment.yml*) is also supplied for compatibility. Please see (https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-from-an-environment-yml-file) for installation details.

1. Teo, R.D., et al., *Mapping hole hopping escape routes in proteins.* Proceedings of the National Academy of Sciences, 2019. **116**(32): p. 15811-15816.