

Adaptive MD Tools

Adam Duster

June 13, 2019

Abstract

Adaptive MD Tools is a package that aims to help with getting partitions from completed MD simulations. The user supplies a topology file and trajectory file, and then selects the active zone center. The active zone center can be an individual atom, or one of several proton tracking algorithms including our indicator and the modified center of excess charge (mCEC). The program can then create a new trajectory where the active center is placed in the center of the box and the other atoms are wrapped around it based on supplied unit cell dimensions for an NVT simulation. The user can calculate the partitions from the permuted AP algorithm for a given permute order and buffer zone radius, and output them as xyz files to a folder.

1 Installation

This program is a python package with dependencies on MDAnalysis, Numba, and NumPY. I recommend that the user create a conda environment specifically for using this package. Then they can install the dependencies and use pip to install this program.

Prerequisites Python3 via Conda

Directions

1. Clone the package from github into the specified directory
`git clone https://github.com/adamduster/adaptive_md_tools`
2. Create a new conda environment with the name "qmmm_ap_tools"
`conda create --name qmmm_ap_tools python=3`
3. Enter the conda environment
`conda activate qmmm_ap_tools`
4. Change to the root directory of the downloaded package
`cd adaptive_md_tools`
5. Install the package using pip
`pip install -e .`
6. The installation should now be complete! You can call the file `AdaptiveMD.py` to run the program!

2 Usage

The program can be invoked by typing:

```
AdaptiveMD.py -i [your input file] > [your output path]
```

As stated in the abstract, this program has a couple functions with keywords that need to be specified together in the input file. Examples for each function are given in the **examples** folder found within the module directory. However, a brief walkthrough of the input file is given here. For the full list of keywords and more detail, see section 4.1.

All Input Files All input files must specify the coordinates, topology, and output coordinates with the following keywords:

1. `coordinates`
2. `topology`
3. `output_coords`

Wrapping simulations To wrap simulations, the dimensions of the period box must be specified using the `dimensions` keyword. The keyword `nowrap` can be used to prevent wrapping.

AP settings To accomplish this, the following variables can be specified:

1. `groups_file`
2. `active_radius`
3. `buffer_radius`
4. `pap_order`
5. `donor_index` to set active center as atom

Writing Partitions The following variables are relevant:

1. `write_partitions` (to flag for writing partitions)
2. `write_n_steps`
3. `elements_file` (if xyz files are to be written)
4. `write_prefix`

Indicator Variables The following variables are relevant:

1. `donor_index` to set donor for calculation
2. `rdh0`
3. `proton_types`
4. `ind_method`
5. `no_pre_topo_change`

Indicator Variables The following variables are relevant:

1. `donor_index` to set donor for calculation
2. `rdh0`
3. `proton_types`
4. `ind_method`
5. `no_pre_topo_change`

mCEC Variables The following variables are relevant:

1. `mcec`
2. `mcec_g`

3 Methodology

This program has several unpublished indicator methods enabled, as well as an implementation of the mCEC. The adaptive partition methodology is discussed elsewhere.

3.1 Indicators

This package contains a variety of proton indicators developed by our group, as well as the center of excess charge (mCEC) developed by König et. al. These indicators are developed to describe proton transfer in a variety of conditions. The location of these indicators can be appended to any arbitrary trajectory and used as the center of the active zone for the AP functionality of the package. Originally, these equations were only meant to describe proton transfer through water and molecules with one protonatable site (e.g. monoatomic anions). These sets of equations are described in section 3.1.1.

However it became necessary to describe proton transfer through molecules or residues with multiple protonation sites. We call these groups donor groups if they are in possession of the excess proton, or acceptor groups if the atoms could be protonated. There are many different attempts at solving this problem implemented within this package. Arguably only two are useful: the original indicator (Indicator0 here) and the indicator mentioned in our upcoming publication as "Indicator B" (Indicator4 here). In fact, as the Indicator4 reduces to Indicator0, there may even be no reason to use Indicator4. Indicator1 is well tested and may also be used, but based on the results of the publication this is not recommended.

3.1.1 Original Indicator

This is implemented as Indicator0. The position of the proton indicator is a linear combination of the coordinates of the donor D and all possible acceptors A_j within a predefined enlisting radius r_{LIST} from D :

$$\mathbf{X}_I = \frac{1}{g_I} \left(\mathbf{X}_D + \sum_{j=1}^J \sum_{m=1}^M g(\rho_{mj}) \mathbf{X}_{A_j} \right) \quad (1)$$

Here, \mathbf{X}_D and \mathbf{X}_{A_j} are the Cartesian coordinates of the donor oxygen and the j -th possible acceptor oxygen respectively, $g(\rho_{mj})$ is the weight function associated with ρ_{mj} (the ratio of the projected donor-acceptor vector), M is the total number of H_m (hydrogen atoms bonded to D), J is the total number of possible acceptors within a radius of size r_{LIST} and g_I is a normalization factor. The ratio ρ_{mj} is a metric for how close a given atom H_m is to the donor D versus a possible acceptor A_j and is calculated according to the following equation:

$$\rho_{mj} = \frac{\mathbf{r}_{\text{DH}_m} \cdot \mathbf{r}_{\text{DA}_j}}{|\mathbf{r}_{\text{DA}_j}|^2} \quad (2)$$

where $\mathbf{r}_{\text{DH}_m} = \mathbf{X}_{H_m} - \mathbf{X}_D$ and $\mathbf{r}_{\text{DA}_j} = \mathbf{X}_{A_j} - \mathbf{X}_D$. The linear combination of coordinates is normalized by g_I , which is calculated according to equation 3 using the weight function described in equation 4:

$$g_I = 1 + \sum_{j=1}^J \sum_{m=1}^M g(\rho_{mj}) \quad (3)$$

$$g(x) = \begin{cases} 0 & \text{if } 1 \leq x \\ -6x^5 + 15x^4 - 10x^3 + 1 & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x < 0 \end{cases} \quad (4)$$

The weight function $g(\rho_{mj})$ depends on the reduced variable x , which is now determined by:

$$x = x(\rho_{mj}) = 1 - \frac{\rho_{mj} - \rho_{mj}^0}{\rho_{\text{max}} - \rho_{mj}^0} \quad (5)$$

$$\rho_{\text{max}} = \frac{r_{\text{DH}}^0}{r_{\text{DH}}^0 + r_{\text{AH}}^0} \quad (6)$$

where $\rho_{mj}^0 = \left(\frac{r_{DH}^0}{r_{\text{LIST}}}\right)$, r_{DH}^0 is a parameter that is set to slightly larger than the equilibrium $D-H$ bond distance (e.g. 1.00 Å in a hydronium ion) to reduce the sensitivity to the $D-H$ vibrations near equilibrium, r_{AH}^0 is similarly a parameter for the $A-H$ bond distance, r_{LIST} is the threshold distance for enlisting possible acceptors, and ρ_{max} represents the percentage of the distance that a transferring proton needs to travel from the donor towards an acceptor before the donor and acceptor switch their status. In this work, the parameters r_{DH}^0 or r_{AH}^0 are based on gas-phase geometries optimized at the B3LYP/6-31G* level for the protonated species and rounded up to the nearest 0.1 Å.

Switching To reiterate, Donor switching occurs theoretically at $\rho = \rho_{\text{max}}$. Theoretically, switching is time-reversible in this case. It should be noted however, that switching is non time-reversible in most practical situations. However, in the case of finite timesteps, switching will occur when $\rho \geq \rho_{\text{max}}$. In cases of proton transfer between two species, this is typically not an issue as the location of the indicator will be very similar in both cases. This is because the normalized weight of the donor will be equal to one, and the weight of the acceptor is approximately equal to 1 around that time. Unfortunately there is a nightmare scenario which occurs in situations with concerted proton transfer. Here a proton can travel part of the way down a water wire belong to the original donor.

3.2 Heuristic for Molecular Topology Update for Intramolecular Reactions

Updating the molecular topology on-the-fly is important for adaptive QM/MM simulations as the H^+ is transported along the path, because if molecules in the active zone later travel into the environmental region, the simulation will contain serious artifacts. While the original indicator could deal with intermolecular topology changes, it was necessary to add additional rules for dealing with intramolecular topology changes. One example is the tautomerization of GLU-, which sees H^+ transfer between the two atoms. The heuristic is as follows. For each donor $k \in K$, and for each other member of the donor group $j \in K, j \neq k$ and for each proton m bonded to k , we calculate the following ratio:

$$F = \frac{r_{km}}{r_{km} + r_{jm}} \quad (7)$$

Here r_{km} and r_{jm} are the distances between the proton and the k -th or j -th atom respectively. If $F > 0.5$, then the m -th proton is closer to the j -th member of the donor group than the atom it is currently bonded to. The topology is then revised by deleting the bond between k and m , and adding another bond between j and m . The angles and dihedrals will also be adjusted accordingly.

This can cause discontinuities if used with the original indicator and any indicator based on Indicator4, because in those methods, the specific bonds between a donor in a donor group and its hydrogens are used to calculate the projection vectors. In Indicator1 however, this is no issue as all protons are considered bonded to the center of geometry of the donor group.

3.3 Indicator A (1)

This extension to the original indicator equation considers each donor or acceptor group as a singular unit with its location determed by its center of geometry. For a donor group, all protons bonded to the individual protonation sites are considered to be bonded to the COG of the group instead. The projection vectors between the donor and acceptors are calculated from the COG of the donor group to the COG of an acceptor group. Therefore the location of the donor group becomes:

$$\mathbf{X}_D = \frac{1}{K} \sum_k^K \mathbf{X}_{D_k} \quad (8)$$

Likewise the location of each acceptor is expressed as:

$$\mathbf{X}_A = \frac{1}{K} \sum_k^K \mathbf{X}_{A_k} \quad (9)$$

Table 1: Suggested Parameters for Indicator Algorithms

	Original Algorithm	Algorithm A	Algorithm B
Donor Location	Site ^α	Center ^β	Site ^α
Reference point for Acceptor Search and Projections	Site ^α	Center ^β r_{DH}^0	Site ^α
Lysine	1.0	1.0	1.0
Aspartate Glutamate	1.0	1.9	1.0
Histidine	1.0	2.1	1.0
Arginine	1.0	2.1	1.0

where k is the k -th protonatable site of the acceptor group.

This algorithm requires very few changes to the definition for the H+ indicator. The geometric or mass center is the reference point for the searching of possible acceptors (when the side chain acts as the donor) and for the calculations of $g(\rho_{mj})$. The parameter r_{DH}^0 are again determined based on B3LYP-optimized geometries of gas-phase models, but from the geometric center of the residue to the proton. Due to the large size of the side chain functional groups, r_{DH}^0 is elongated in order to make sure that all relevant acceptors are included in the equation. The parameters suggested for various amino acids are shown in column 3 of Table 3.3.

3.4 Indicator B (4)

Indicator4, or IndicatorB in the publication, considers the side chain as a whole, but it recognizes that the multiple protonation sites may be situated in different local environments and thus participate differently. Therefore, when all protonation sites of the side chain act collectively as donors, they are treated independently in the search of possible acceptors and the subsequent calculations of weights for these acceptors (Fig. 2c). More specifically, a local search of possible acceptors is carried out for each protonation site D_k , and the indicator position \mathbf{X}_I is computed by:

$$\mathbf{X}_I = \frac{1}{g_I} \left(\mathbf{X}_D + \sum_{k=1}^K \sum_{j=1}^J \sum_{m=1}^{M_k} g(\rho_{mjk}) \mathbf{X}_{A_j} \right) \quad (10)$$

Here, k is the index for the protonation sites of the side chain, \mathbf{X}_{D_k} is the position of the k -th protonation site D_k , M_k is the number of covalently bonded H atoms to the atom at D_k , and ρ_{mjk} and g_{mjk} are the generalized projection and normalized weights for D_k respectively. Here, \mathbf{X}_D is calculated using Equation 8. However, the location of the individual protonation site D_k is used as the reference point for enlisting possible acceptors and calculating the projection ratio. The r_{LIST} and r_{DH}^0 parameters should use the same values as they would with the original algorithm (see column 4 of Table 3.3). The entire sidechain is represented by the geometric center so that when there are no acceptors, the center of geometry is the location of the proton indicator.

The normalization constant is adjusted to accomodate the extra projection vectors and is calculated by:

$$g_i = 1 + \sum_k^K \sum_j^J \sum_m^{M_k} g_{kjm}(\rho_{kjm}) \quad (11)$$

Note that the geometric center is given a weight of 1.

3.5 Indicator B Variations

3.5.1 Increase Weighting of the Likely Acceptors (Indicator 6)

The main origin of discontinuities in the indicator’s position occurs when multiple projection vectors are pointed towards acceptors in the opposite directions. These projections cancel each other out and the location of the indicator ends up on the donor instead of along the D-H bond length. This correction tries to weight the most likely projection even further using a logit to reduce the degree of this cancellation.

Here the equation 10 is modified by replacing:

$$\begin{aligned} & g_{kjm}(\rho_{kjm}) \text{ with } \exp[g_{kjm}(\rho_{kjm})] \\ & \text{and} \\ & \text{equation 11 with } g_i = e + \sum_{k=1}^K \sum_{j=1}^J \sum_{m=1}^M g(\rho_{mjk}) \end{aligned}$$

Brief Description of Performance It almost universally improved the performance of Indicator B or had no real deleterious performance.

3.5.2 Change \mathbf{X}_D to a Weighted Coordinate

This method also tried to solve the same problem as the first but with a different approach. Here the coordinate \mathbf{X}_D in equation 10 was changed from the geometric center of the protonatable sites to the following:

$$\mathbf{X}_D = \sum_k^K w_k \mathbf{X}_{D_k} \quad (12)$$

$$w_k = \frac{\sum_{m \in M_k} r_{km} - r_{DH}^0}{\sum_k^K r_{km} - r_{DH}^0} \quad (13)$$

The idea was that the location of the indicator should be weighted further towards donor groups that had bonds that were further from equilibrium. In practice, it seems theoretically very unsatisfactory as it just makes everything more complicated and invalidates the correction in the original equation. This should not be used.

3.5.3 Intramolecular ρ_{kjm} Included

This extension aimed to solve the problem of proton tautomerization by including the other atoms in a donor group as possible acceptors. Thus, the projection weight would be calculated between the m -th proton bonded to the k -th other protonatable site, and the normalized projection weight is multiplied by the k - j vectors and added just like if they were a normal acceptor.

This method performed the same as Indicator B when these projection vectors were negative and significantly worse when they weren’t.

3.5.4 Correction Term For Proton Transfer

In this method, a correction term $\mathbf{X}_D - \mathbf{X}_{D_k}$ is added into equation 10 to remove a discontinuity from proton transfer in the limit of the eigen state when a donor is donating a proton to a group with multiple protonatable sites. The equation becomes:

$$\mathbf{X}_I = \frac{1}{g_I} \left(\mathbf{X}_D + \sum_{k=1}^K \sum_{j=1}^J \sum_{m=1}^M g(\rho_{mjk}) (\mathbf{X}_{A_j} + \mathbf{X}_{D_k} - \mathbf{X}_D) \right) \quad (14)$$

Consider the situation of a proton being transferred from a group with one protonation site (i.e. a water) to a group with multiple protonation sites (i.e. a histidine) Also assume the group with multiple protonation sites is not poised to donate a proton to any other group. In this case, the indicator will be superimposed on the proton between the donor (water) and acceptor (N atom of histidine) as $\rho_{mkj} = 0.5$.

However, after transfer and if there was no correction, the proton would not be in the same location even though the ρ_{mkj} of the donor group would still equal 0.5. Because when ρ_{mkj} , $g_{mkj} = 1$ and expanding the expression for the donor COG, Equation 10 would reduce to:

$$\frac{\mathbf{X}_{A_j} + \mathbf{X}_D}{2} \quad (15)$$

This would place the proton halfway between the COG of the donor group and the acceptor. However, when we transferred the proton, we were halfway between the protonatable site of the group and the other molecule! Thus adding this correction term adds the difference from the COG to the protonatable site of the group and prevents this jump.

Although this is a good idea, it did not perform well on a test suite with concerted-like transfers and was subsequently scrapped.

3.6 mCEC

The mCEC is implemented as such:

$$\sum_{h \in H} \mathbf{r}_h - \sum_{j \in J} w_j \mathbf{r}_j + \sum_{h \in H} \sum_{j \in J} f_{sw}(r_{hj}) + \zeta' \quad (16)$$

Here, ζ' is a correction term to account for groups of atoms with multiple protonatable sites. It can be expressed as:

$$\sum_{g \in G} \sum_{k \in g} \sum_{j \in g} m_k (\mathbf{r}_j - \mathbf{r}_k) \quad (17)$$

where g is a residue with multiple protonation sites in the collection of all residues G , and j and k are acceptor molecules within the same protonation site. The multiplier m is a differentiable maximum function times a constant related to the number of protons in the residue:

$$m = \frac{P}{N} * \sum_h f_{sw}(r_{hj})^{16} / \sum_h f_{sw}(r_{hj})^{15} \quad (18)$$

In the implementation within this code, the mCEC cannot be used alone and must be used with the one of the other indicator based methods. Those track the topology.

4 Input File

4.1 Keywords

active_radius (float)

The active radius for adaptive partitioning calculations.

allow_hop (int to bool)

If the indicator is calculated, indicator hops will only be allowed if this is set to 1.

buffer_radius (float)

The buffer radius for adaptive partitioning calculations.

coordinates (path to coordinate file)

This keyword allows the user to supply the coordinates file for the program. It is required. The suffix of the file will indicate to MDAnalysis what kind of coordinate file it is.

dcd_pbc 1

This keyword doesn't do anything. Some of the files used in previous research projects have it, so this is kept to ensure those files are compatible with this version of the code.

dimensions (float) (float) (float)

The x, y, and z dimensions of the cell for wrapping the trajectory. The cell is assumed to have constant volume.

pap_order (int)

The truncation order for the PAP method.

donor_index (1-based integer of donor index)

This keyword is used to specify the donor at step 1.

elements_file (file type) (path to elements file)

This keyword is used to specify the file with the element symbol for each of the atoms in the system. It is needed to write the .xyz files if the partitions are to be output.

groups_file (path to group file)

This keyword is used to specify the AP groups file. It is needed for calculating the partitions during an AP simulation. It is updated as proton transfer happens.

ind_method (int)

The indicator method to use for the calculation. It can be set to:

0 — Original indicator. See section 3.1.1

1 — Indicator A. See section 3.3.

2 — Defunct, do not use.

3 — Defunct, do not use.

4 — Indicator B. See section 3.4.

5 — Defunct, do not use.

6 — Indicator method described in 3.5.1

7 — Indicator method described in 3.5.3. Should not be used.

9 — Indicator method described in 3.5.2

11 — Indicator method described in 3.5.4

ind_output_freq (integer)

The frequency in steps to output the indicator location into the xyz file.

indicator_verbose

Print out tons of indicator info.

mcec (atom type 1) (float reference state 1) (atom type 2) (reference state 2) ..

This keyword turns on the mCEC. Each atom type that is to be included as an acceptor should be specified here followed by its reference state. All atom types must also have an associated

mcec_g (1-based index),(1-based index),(...),(int reference state) ...

This keyword turn on the mCEC correction for groups with multiple protonationable sites. Each group is represented by a string with atom indices separated by commas followed finally by the reference protonation state for the group. For example, the entry:

```
mcec_g 5,8,7,4 18,19,0 35,36,1
```

specifies three groups with multiple protonatable sites. The first group is contains atom indices 5,8, and 7. It has 4 protons in its reference state. The second group contains atom indices 18 and 19 and has no protons in its reference state.

no_pre_topo_change

Disable the topology changing heuristic for intramolecular proton transfer found in 3.2

nowrap

Don't wrap the simulation around the center of the periodic box or active zone.

proton_types (type 1) (type 2) (type 3)

The proton types to be considered for mCEC and indicator calculations. Protons will only be considered bonded to the donor atoms if they have the specified type.

rdh0 (atom type 1) (float) (atom type 2) (float) ...

This keyword sets the rdh0 parameters for each atom **type**. An example of using this code is:

```
rdh0 N 1.0 OT 1.0 CLA 1.4
```

rlist (float)

This sets the r_{LIST} parameter.

structure (path to structure file) (structure type)

This keyword allows the user to specify the topology of the system. When doing indicator calculations with proton transfers, only .mol2 topology files can be used. Otherwise, any filetype MDAAnalysis accepts is fine.

verbose 1

Turn on the verbose flag. Currently it must be set to 1.

write_partitions (path to folder to write partitions) (file type)

Write the AP partitions that would be in the trajectory to the specified folder. The files will have the extension given by the file type argument to the keyword. Note that xyz is the most tested format, but any format supported by MDAAnalysis should work if the required info is there.

write_n_steps (int)

Write the partitions every n steps.

write_prefix (desired prefix)

Prefix files created by the program with some output prefix to tidy up the mess that this program creates.

5 Developers Notes

See the pdoc folder in the root directory for a complete list of all submodules and subroutines.