RESP2 Documentation

resp2

Aug 23, 2019

CONTENTS

CHAPTER

ONE

INSTALLATION

To install all necessary packages we recommend to follow these steps.

1.) Download the RESP2 package:

```
git clone git@github.com:MSchauperl/RESP2.git
```

2.) Install and activate the conda environment 'RESP2'

```
cd RESP2
conda env create -f devtools/conda-envs/RESP2_environment.yaml
conda activate RESP2
```

3.) Install the RESP2 package

```
python setup.py develop
```

4.) Download the respyte package

```
cd ..
git clone https://github.com/lpwgroup/respyte.git
```

5.) Install respyte package

```
cd respyte
python setup.py develop
cd ..
```

CHAPTER

TWO

EXAMPLE

See Jupyter notebook: https://github.com/MSchauperl/RESP2/blob/master/example/Ethanol_Example.ipynb

CHAPTER

THREE

FUNCTIONS

3.1 RESP2 charge generation and ForceBalance Input generation

resp2.py includes the basic functions to parameterize a molecule with RESP2 charges. It allows to create a molecule from a Smiles string and obtain 3D structures. It uses openeye's omega to generate conformations. Respyte is used to select ESP grid points Psi4 is used for the QM calculations.

It can generate RESP2 charges with any given Delta value.

It can also be used to scale RESP1 charges (only neutral molecules).

resp2.resp2.**create_fb_input**(*name=''*, *targets=[]*, *forcefield='smirnoff99Frosst.offxml'*, *port='3333'*, *type='single'*, *mol2_files=[]*, *convergence='tight'*)

Module to create a ForceBalance input file for training or testing purposes.

Parameters

- **name** – Name of the ForceBalance input file you want to create.
- **targets** – The targets you want to inlcude in the run.
- **forcefield** – Name of the forcefield file.
- **port** – Port to use for work_queue.
- **type** – Single point or an optimization.
- **mol2_files** – List of mol2 files necessary for the calculation. Usually this should include all RESP2 charge files.
- **convergence** – <tight> or <loose> convergence criteria.

Returns 0 if succesful.

resp2.resp2.**create_fb_input_header**(*output=None*, *port='3333'*, *type='single'*, *forcefield='smirnoff99Frosst.offxml'*, *mol2_files=[]*, *convergence='tight'*)

This function is used to generate the header of the ForceBalance input file ($options section). The targets are not included in this function. Is used by create_fb_input.

Parameters

- **output** – File to write to.
- **forcefield** – Name of the forcefield file.
- **port** – Port to use for work_queue.
- **type** – Single point or an optimization.

- **mol2_files** – List of mol2 files necessary for the calculation. Usually this should include all RESP2 charge files.
- **convergence** – <tight> or <loose> convergence criteria.

Returns

resp2.resp2.**create_std_target_file**(*name=''*, *density=None*, *folder=None*, *hov=None*, *dielectric=None*)

This function creates the target data.csv files required by ForceBalance. Up to now only 3 properties are supported. Is used by create_target.

Parameters

- **name** – Name of the molecule. Folders are named accordingly.
- **density** – Density of the molecule in kg / m3
- **hov** – Heats of Vaporization in kJ /kcal / mol
- **dielectric** – Dielectric constant

Returns 0 if successful

resp2.resp2.**create_target**(*smiles=''*, *name=''*, *folder=None*, *density=None*, *hov=None*, *dielectric=None*, *resname='MOL'*, *nmol=700*, *tries=2000*)

This functions creates a target including folder structure mol2 files and the data.csv file. Charges are done separate.

Parameters

- **smiles** – SMILES Code of the molecule.
- **name** – Name of the molecule. Folders are named accordingly.
- **density** – Density of the molecule in kg / m3
- **hov** – Heats of Vaporization in kJ /kcal / mol
- **dielectric** – Dielectric constant
- **folder** – Name of the folder for the target. If not specified. {name}-liquid is used.
- **resname** – Abbreviation of the Residue. Specified in the mol2
- **nmol** – Number of molecules in the liquid simulation box.
- **tries** – Number of tries to create the liquid simulation box. For bulky molecules higher values are necessary.

Returns

resp2.resp2.**create_smifile_from_string**(*smiles=''*, *filename=''*)

Writes a SMILES string to a file.

Parameters

- **smiles** – SMILES Code of the molecule.
- **filename** – Filename (.smi)

Returns

resp2.resp2.**create_conformers**(*infile=None*, *outfile=None*, *resname=None*, *folder=None*, *name=None*)

This function takes a mol1 file and runs Openeye's omega to create conformers for the molecules The conformers are stored in separated files, adding the number of the conformer at the end of the filename

Parameters

- **infile** – Path to input file
- **outfile** – Path to output file return
- **folder** – Name of the folder for the target. If not specified. {name}-liquid is used.
- **resname** – Abbreviation of the Residue. Specified in the mol2

Returns Number of conformers for this molecule

resp2.resp2.**optimize_conformers**(*opt=True*, *name=''*, *resname='MOL'*, *number_of_conformers=1*, *folder=None*)

Optimize all conformers using psi4. This is done in a 3 step approach were the level of theory is increased stepwise. The resulting structures ares saved as xyz files. If opt = False the optimization is omitted and only the files are copied.

Parameters

- **opt** – True if optimization should be performed.
- **name** – Name of the molecule. Folders are named accordingly.
- **resname** – Abbreviation of the Residue. Specified in the mol2
- **number_of_conformers** – Number of conformers for this molecule
- **folder** – Name of the folder for the target. If not specified. {name}-liquid is used.

Returns

resp2.resp2.**create_respyte**(*type='RESP1'*, *name=''*, *resname='MOL'*, *number_of_conformers=1*, *opt_folder=None*)

This function creates the respyte input files to generate the selection of ESP grid points by calling the function create_respyte_input_files. Additionally, it generates the input for the psi4 QM calculation at the requested level of theory.

Theory level is determined by the type of the calculation. RESP1 uses HF/6-31G*; RESP2GAS uses PW6BP94/aug-cc-pV(D+d)Z; RESP2LIQUID uses PW6BP94/aug-cc-pV(D+d)Z with PCM (water).

Parameters

- **type** – Defines what type of QM calculation to perform
- **name** – Name of the compound
- **resname** – 3 letter abbreviation of the compound
- **number_of_conformers** – Number of conformers used for this compound
- **opt_folder** – Name of the folder used for optimize_conformers. If not specified. {name}-liquid is used.

Returns 0 if successful

resp2.resp2.**calculate_respyte**(*type='RESP1'*, *name=''*, *resname='MOL'*, *number_of_conformers=1*)

This function performs the psi4 calculation and the respyte calculation and checks if the calculation was successful.

Parameters

- **type** – defines what type of QM calculation to perform
- **name** – name of the compound
- **resname** – 3 letter abbreviation of the compound
- **number_of_conformers** – Number of conformers used for this compound

Returns 0 if successful

resp2.resp2.**create_respyte_input_files**(*type='RESP1'*, *name=''*, *resname='MOL'*, *number_of_conformers=1*)

This function performs the psi4 calculation and the respyte calculations and checks if the calculation was successful.

Parameters

- **type** – Defines what type of QM calculation to perform
- **name** – Name of the compound
- **number_of_conformers** – Number of conformers used for this compound

Returns 0 if successful

resp2.resp2.**create_charge_file**(*name=''*, *resname='MOL'*, *delta=0.0*, *type='RESP1'*)

This function creates a MOL2 file with either RESP1 scaled charges or RESP2 charges with a certain mixing parameter.

Parameters

- **name** – Name of the compound.
- **resname** – 3 letter abbreviation of the compound.
- **delta** – Mixing parameter given as absolute value (not percent)
- **type** – RESP1 or RESP2 type charges

Returns

resp2.resp2.**create_RESP2**(*smi=None*, *folder=''*, *opt=True*, *name=''*, *resname='MOL'*, *delta=1.0*, *density=None*, *hov=None*, *dielectric=None*)

Creates a mol2 file with RESP2 charges from a mol2 file (resname.mol2) or from a smiles string.

Parameters

- **folder** – folder to write the output files.
- **opt** – True when generated conformers should be locally optimized.
- **name** – Name of the compound
- **density** – Density of the molecule in kg / m3
- **hov** – Heats of Vaporization in kJ /kcal / mol
- **dielectric** – Dielectric constant
- **folder** – Name of the folder for the target. If not specified. {name}-liquid is used.
- **resname** – Abbreviation of the Residue. Specified in the mol2
- **delta** – Fraction (in percent) of liquid charges. default=1.0

Returns

CHAPTER

FOUR

OTHER MODULES

The intial version of this code is part of the openforcefield package and was written by Lee-Ping Wang. This version was adapted for the use in the RESP2 package.

resp2.create_mol2_pdb.**CalculateMolecularWeight**(*mol*)
Calculate the molecular weight for an OpenEye molecule.

Parameters

mmol [OEGraphMol]

Returns

float Molecular weight in g/mol

resp2.create_mol2_pdb.**CalculateBoxSize**(*nmol*, *molwt*, *density*)
Calculate the size of a solvent box.

Parameters

nmol [int] Number of molecules desired for the box

molwt [float] Molecular weight in g/mol

density [float] Estimated density in kg/m3 (this should be about 40-50% lower than the real liquid density)

Returns

float Length of a cubic solvent box in nm.

resp2.create_mol2_pdb.**GenerateBox**(*pdbin*, *pdbout*, *box*, *nmol*, *tries*)
Call genbox. (Confirmed working with Gromacs version 4.6.7 and 5.1.4). Mainly checks whether genbox ran correctly.

Parameters

pdbin [str] Name of input PDB file containing a single molecule.

pdbout [str] Name of output PDB file containing solvent box.

box [float] Solvent box size, should be determined previously.

nmol [int] Number of molecules to go into the solvent box

tries [int] Parameter for genbox to try inserting each molecule (tries) times

Returns

None If successful, produces "pdbout" containing solvent box.

resp2.create_mol2_pdb.**main**()

Provide a text file containing a single SMILES string and three-letter residue name. Receive (res).pdb and (res).mol2 files containing a single molecule with conformation. Receive (res)-box.pdb containing a box with specified number

Dependencies: OpenEye tools (for creating molecule from SMILES) openmoltools (for calling OpenEye to generate conformer) Gromacs 4.6.7 or 5.1.4 (for calling genbox to create solvent box) ForceBalance 1.5.x (for putting information back that was thrown away by genbox)

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

r

INDEX