THC Analysis Scripts

This section contains descriptions of the scripts required to a) generate the THC data used in this work, and b) generate the plots and fits reported in this work based upon said data.

Generating THC Data

The results of the following steps are stored in the files "SI/PartialEnergyAnalysis" and "SI/PairsofGridPointsAnalysis" after extraction of the compressed SI file for the reader's convenience, and can be used in later steps to regenerate the figures and fits of this manuscript without rerunning CFOUR **or any of the scripts discussed below.**

The scripts required to reproduce the analysis in this manuscript are contained in the "THCpairpoints" sub-directory, as well as a public GitHub repository located at :
https://github.com/MatthewsResearchGroup/THCpairpoints . These scripts read various .dat files generated from the extract.py script after the successful completion of a CFOUR calculation. It should be noted that the THC-specific files can not be generated from the public version of CFOUR. Compressed .dat files used to generate the figures and data in this manuscript are provided in the GitHub repo. After extraction, .dat files for each given molecule/basis/wavefunction may be found in the following paths:
data/grid/$molecule/*.dat . The relevant file types are:

- fa.dat : diagonal of the virtual-virtual Fock Matrix
- fi.dat : diagonal of the occupied-occupied Fock Matrix
- t.dat : T2 amplitudes
- v.dat : abij electron repulsion integrals
- pvt.dat : pivots of the Cholesky grid pruning

- grid.dat : xyz coordinates and weight of a given grid point
- XI.dat : occupied-grid point collocation matrix
- XA.dat : virtual-grid point collocation matrix

Gathering Base Data via CFOUR

CFOUR calculations to generate data for these scripts must be run with Cartesian coordinates and in atomic units in order to be compatible with the Pair Point Analysis scripts below. Additionally, the scripts below only support RHF reference wavefunctions. The appropriate data files may be found on disk after the CFOUR calculation completes successfully and the "formatMolecule.sh" extraction script is run.

Sub-directory Layout

The partial energy analysis and pair point analysis scripts included in this repository expect a specific sub-directory layout for each molecule, where the name of each directory specifies the wavefunction and basis set for the given data, along with an additional sub-directory for THC grid data for any given basis set. The various options accepted by these scripts are given below:

- –molecule : any string
- –basis : any string
- –calc : { "MP2", "MP3", "CCSD" }

The path to the .dat files used in the following analysis then needs to adhere to the following pattern: $molecule/$basis/$calc for orbital data and $molecule/$basis/grid for grid data. For example, the orbital data for –molecule=thymine –basis=tz –calc=CCSD needs to be located in

```
thymine/tz/CCSD/*.dat
```

relative to the directory from which these scripts are called. The THC specific data for this same calculation would need to be located in

```
thymine/tz/grid/*.dat
```

If –calc=MP3 or –calc=MP2 then the calc directory should exist. For example, the orbital data for –molecule=thymine –basis=tz –calc=MP2 needs to be located in

```
thymine/tz/*.dat
```

and its THC data needs to be located in

```
thymine/tz/grid/*.dat
```

Likewise, if –basis=dz is used, the basis sub-directory should not exist. For example, the orbital data for –molecule=thymine –basis=dz –calc=CCSD needs to be located at

```
thymine/ccsd/*.dat
```

and the THC data at

```
thymine/grid/*.dat
```

These two rules compound, so if the options are –molecule=thymine –basis=dz –calc=MP2 the data needs to be located in

```
thymine/*.dat
```

and the THC data in

```
thymine/grid/*.dat
```

This description is also provided as a .pdf in the GitHub repository, and potential users are encouraged to defer to that description as these scripts may be modified or improved in future work.

Partial Energy Analysis

The partial energy analysis programs are located in “partialGridEnergy” sub-directory. This program takes in the input described above and generates a series of THC approximations at increasing values of χ for the given contributions of W_nT_2 to the Coulomb and exchange correlation energies. These values are stored in a .csv file in in the path

```
$molecule/partialenergyanalysis_$method_$basis.csv
```

This script must be called at the top level of the specific sub-directory structure described in the previous subsection.

It should be noted that the Partial Energy Analysis depends upon the docopt.cpp and randutils libraries.

Pair Point Analysis

The analysis of the pair point contributions to the THC approximations of W_nT_2 to the Coulomb and exchange correlation energies proceeds in several steps. The first step is to execute the “energyAnalysis” program, which accepts the –method input as described above and an additional parameter –npair that indicates the number of randomly selected pairs to be studied. This program must be called in the appropriate $molecule/$basis sub-directory for each method that will be analyzed. This program outputs energyAnalysis_$method.csv, which contains the WSS and correlation energy contributions for the generated sets of pairs.

It should be noted that energyAnalysis depends upon the docopt.cpp and randutils libraries.

Once the set of $molecule/$basis/$method/energyAnalysis_$method.csv for a set of calculations have been generated and accumulated into a file called $molecule/$basis/energyAnalysis.csv, the “make_ene_plot.py” script may be called to perform the bootstrapping and pair-point geometric analysis used to generate the correlation feature plots featured in the manuscript. This python script must be called from the top level directory of the pair-points analysis, and accepts only a fixed-format input:

```
python3 make_ene_plot.py $mol $basis $method $ftype $size
```

These options are summarized here:

- $mol : sub-directory name containing the data for a specific molecule.
- $basis : one of four strings, {adz, rtz, tz, dz }, which must also match the basis directory names in the sub-directory system.
- $method : a string that must match a method contained within $molecule/$basis/energyAnalysis.csv .
- $ftype : one of three strings, {all, no-cross, cross}. Analysis in this manuscript uses "all" exclusively.
- $size : an integer number of pairs of gridpoints, the same as those used –npair in energyAnalysis.

Upon completion, "make_ene_plot.py" will generate a large .csv file:

```
$molecule/$basis/bootstrapping_$mol_$basis_$method_$ftype.csv
```

along with a large number of figures contained within a new sub-directory called "pic". This .csv file is then used in the fitting and final set of plot generation.

The results of the above steps are stored in the files "SI/PartialEnergyAnalysis" and "SI/PairsofGridPointsAnalysis" for the reader's convenience, and can be used in the following steps to regenerate the figures and fits of this manuscript without rerunning CFOUR **or any of the scripts discussed so far.**

Using Pre-Generated Data

At this point, it is expected that the results of the partial energy fitting for all molecules, methods, and basis sets have been accumulated into one .csv file called partialGridEnergy.csv.

Likewise, the results of the pair point analysis for all molecules, methods, basis sets, and numbers of pairs of points, have been accumulated into Energy_Analysis_Bootstrapping_Merge.csv . The versions of these files used in generating the plots and fits of this manuscript are located at SI/PartialEnergyAnalysis/partialGridEnergy.csv and SI/PairsofGridPointsAnalysis/Energy_Analysis_Bootstrapping_Merge.csv . The following sections describe how to use the scripts located in SI/PairsofGridPointsAnalysis and SI/PartialEnergyAnalysis to generate the figures and fits reported in this work.

These scripts have been confirmed to work with the following package versions:

- Conda : 24.4.0
- Python: 3.10.13
- Pandas: 1.5.3
- MatPlotLib: 3.7.1
- NumPy: 1.24.3
- SciPy: 1.11.1
- SciKit-Learn: 1.3.0

We recommend that you execute these scripts within a conda environment with specific package versions. This may be created via the following shell commands:

```
conda create -n "THC-plotting"
conda init
\\close shell and reopen
conda activate "THC-plotting"
conda install pandas=1.5.3
conda install matplotlib=3.7.1
conda install numpy=1.24.3

conda install scipy=1.11.1

conda install scikit-learn=1.3.0
```

Partial Energy Analysis Fits

“SI/PartialEnergyAnalysis/PartialEnergyFitting.py” generates fits of THC approximations of $W_n T_2$ contributions to the Coulomb and exchange correlation energies as a function of χ. The output of “partialGridEnergy” (described in the above section) for all species, wavefunctions, and basis sets needs to be contained within a single .csv file named partialGridEnergy.csv . “SI/PartialEnergyAnalysis/PartialEnergyFitting.py” then generates a .csv file of the fit parameters, PartialEnergyFittingParameters.csv, and a set of .png files that display the fits in a sub-directory called PartialEnergyFitting generated in the directory that the python script is called from.

Partial Energy Analysis Method Comparison

“SI/PartialEnergyAnalysis/PartialEnergyComparsionMethods.py” is a python script that will generate comparisons for all the unique methods contained within partialGridEnergy.csv (same file as in the above subsection) for the THC approximations of $W_n T_2$ contributions to the Coulomb and exchange correlation energies. The output .png files are stored in a sub-directory PartialEnergyMethodsComparison contained within the directory where “PartialEnergyComparsionMethods.py” is called from.

Pair Points Fitting

“SI/PairsofGridPointsAnalysis/PairsofGridPointsFitting.py” is a python script that generates the fit of the correlation feature used in the manuscript. It requires that Energy_Analysis_Bootstrapping_Merge.csv exists in the same directory from which it is called, and that this .csv file contains the accumulated results of the pair points analysis described above. Upon completion, this script will generate PairsofGridPointsFitting/Correlation_Feature_Gaussian_distribution_by_User.csv and PairsofGridPointsFitting/Correlation_Feature_t_coefficient_by_User.csv which contain the fits of the correlation feature to a Gaussian and Student's t-distribution, respectively.

Pair Points Plotting

"SI/PairsofGridPointsAnalysis/PairsofGridPointsPlotting.py" is a python script that generates the plots of the correlation features used in the manuscript. It requires that Energy_Analysis_Bootstrapping_Merge.csv exists in the same directory from which it is called, and that this .csv file contains the accumulated results of the pair points analysis described above. Upon completion, this script will generate a new subdirectory called "PairsofGridPointsFigures", which in turn contains a significant number of figures sub-directories nested as PairsofGridPointsFigures/$molecule/$basis/$method. In these directories there are a number of .png files that feature the plots listed below:

- EoverE : The ratio of the sum of exchange and Coulomb energy terms captured by pair grid points to the canonical correlation exchange and Coulomb energy terms, $\frac{\Delta E_x+\Delta E_c}{E_x+E_c}$
- WSS : the weighted subspace score introduced by the pair grid points
- ExoverEx : The ratio of exchange energy captured by pair grid points to the canonical correlation exchange energy, $\frac{\Delta E_x}{E_x}$
- EcoverEc : The ratio of Coulomb energy captured by pair grid points over the canonical correlation exchange energy, $\frac{\Delta E_c}{E_c}$
- ExoverE : The ratio of exchange energy captured by pair grid points to the total canonical correlation Coulomb energy, $\frac{\Delta E_x}{E_x+E_c}$
- EcoverE : The ratio of Coulomb energy captured by pair grid points to the total canonical correlation energy, $\frac{\Delta E_c}{E_x+E_c}$

WSS, EcoverE, and ExoverE are the types of plots used in the manuscript.