Psi4 Research Update

Lori Burns, Sherrill Group

14 November 2014

User Access to BasisSet

Circa November 2014, basis set handling in PSI4 has been revamped. Fear not that the beloved libmints BasisSet object has changed- rather, the user specification of basis sets, programmer's API to BasisSet constructors, and the construction of BasisSet objects has been changed.

Advantages to New Scheme (aka *Why*)

- Defaults for fitting basis sets set on a per-atom basis (*e.g.*, DF-SCF on metal-organic with cc-pVDZ uses cc-pVDZ-JKFIT for the organic and Def2-tzvpp (or something) for the metal) so that the user shouldn't experience a failed job on account of incomplete fitting basis sets.
- All default info for auxiliary basis sets in one place. Programmer when calling for a new auxiliary BasisSet gives the fitting role if defaults need to be computed (e.g., JKFIT) and the orbital basis to compute defaults off of (e.g., get_option(BASIS)). This eliminates all the "corresponding_jkfit" boilerplate in `proc.py` and also means defaults can be assigned for non-uniform orbital basis sets.
- Assignment of basis sets to atoms proceeds through "all", "by_symbol" (e.g., "Co"), or "by_label" (e.g., H1 or Co_mine). There is *no* assignment to atoms by number (except a bit internally where it's safe) which can be ambiguous when the Molecule has been fragmented as for SAPT.
- Users don't need to "set basis basisname" after every *molecule {...}* definition or activation because basis sets are not attached to the molecule at time option is set but at time BasisSet is built. Similarly, once can define a *basis basisname {...}* block and use it for multiple molecules.

BasisSet gives the option name where any user intentions as to proper value may be found (D
which the new basis can be recalled (get_str('DF_BASIS_SCF')), the fitting role if defaults need to b

Disadvantage: qcdb.Molecule now participates in virtually every psi4 job. Let me know if there are problems w/basis set assignment to atoms.

```
set basis heavy-aug-cc-pvdz

molecule watdim {
0 1
O     -1.551007  -0.114520   0.000000
H     -1.934259   0.762503   0.000000
H_hb  -0.599677   0.040712   0.000000
--
0 1
O      1.350625   0.111469   0.000000
H      1.680398  -0.373741  -0.758561
H      1.680398  -0.373741   0.758561
}

energy('scf')  # water dimer in hadz

molecule ammdim {
0 1
N     -1.578718  -0.046611   0.000000
H     -2.158621   0.136396  -0.809565
H     -2.158621   0.136396   0.809565
H_hb  -0.849471   0.658193   0.000000
--
0 1
N      1.578718   0.046611   0.000000
H      2.158621  -0.136396  -0.809565
H      0.849471  -0.658193   0.000000
H      2.158621  -0.136396   0.809565
}

energy('scf')  # ammonia dimer in hadz
```

- Can set basis sets before molecule
- Can re-use basis sets for multiple molecules

```
basis minebas {
    assign aug-cc-pvdz
    assign H cc-pvtz
    assign H_hb aug-cc-pvtz
}

molecule watdim {
0 1
O     -1.551007  -0.114520   0.000000
H     -1.934259   0.762503   0.000000
H_hb  -0.599677   0.040712   0.000000
--
0 1
O      1.350625   0.111469   0.000000
H      1.680398  -0.373741  -0.758561
H      1.680398  -0.373741   0.758561
}

energy('scf')  # water dimer in minebas
set basis cc-pvdz
energy('scf')  # water dimer in dz
set basis minebas

molecule ammdim {
0 1
N     -1.578718  -0.046611   0.000000
H     -2.158621   0.136396  -0.809565
H     -2.158621   0.136396   0.809565
H_hb  -0.849471   0.658193   0.000000
--
0 1
N      1.578718   0.046611   0.000000
H      2.158621  -0.136396  -0.809565
H      0.849471  -0.658193   0.000000
H      2.158621  -0.136396   0.809565
}

energy('scf')  # ammonia dimer in minebas
```

- Custom basis sets can be set before molecule, too
- When named, custom basis sets can be recalled

```
molecule watdim {
0 1
O     -1.551007  -0.114520   0.000000
H     -1.934259   0.762503   0.000000
H_hb  -0.599677   0.040712   0.000000
--
0 1
O      1.350625   0.111469   0.000000
H      1.680398  -0.373741  -0.758561
H      1.680398  -0.373741   0.758561
}

set basis minebasfile
energy('scf')  # water dimer in minebas

set basis cc-pvdz
energy('scf')  # water dimer in dz


molecule ammdim {
0 1
N     -1.578718  -0.046611   0.000000
H     -2.158621   0.136396  -0.809565
H     -2.158621   0.136396   0.809565
H_hb  -0.849471   0.658193   0.000000
--
0 1
N      1.578718   0.046611   0.000000
H      2.158621  -0.136396  -0.809565
H      0.849471  -0.658193   0.000000
H      2.158621  -0.136396   0.809565
}

set basis minebasfile
energy('scf')  # ammonia dimer in minebas
```

Basis Search Path: here, then PSIPATH, then library. ok?

```
>>> cat minebasfile.gbs
spherical

! H_hb is aug-cc-pvtz
! H is cc-pvtz
! other are aug-cc-pvdz

****
H_hb     0
S   3   1.00
...
      0.2470000              1.0000000
****
H     0
S   3   1.00
     33.8700000              0.0060680
...
      1.0570000              1.0000000
****
N     0
S   8   1.00
   9046.0000000              0.0007000
...
      0.2300000              1.0000000
****
O     0
S   8   1.00
...
      0.3320000              1.0000000
****
```

- That same custom basis set can be stored in a file
- Files now allow label specs

```
=> Loading Basis Set <=

 Role: BASIS
 Keyword: BASIS
 Name: <function basisspec_psi4_yo__minebas at 0x10a51b488>
 atoms 1, 4   entry O          line   243 file /Users/loriab/linux/psihub/master/psi4/lib/basis/aug-cc-pvdz.gbs
 atoms 2, 5-6 entry H          line    20 file /Users/loriab/linux/psihub/master/psi4/lib/basis/cc-pvtz.gbs
 atoms 3      entry H          line    34 file /Users/loriab/linux/psihub/master/psi4/lib/basis/aug-cc-pvtz.gbs

=> Loading Basis Set <=

 Role: JKFIT
 Keyword: DF_BASIS_SCF
 Name:
 atoms 1, 4   entry O          line   269 file /Users/loriab/linux/psihub/master/psi4/lib/basis/aug-cc-pvdz-jkfit.gbs
 atoms 2, 5-6 entry H          line    50 file /Users/loriab/linux/psihub/master/psi4/lib/basis/cc-pvtz-jkfit.gbs
 atoms 3      entry H          line    69 file /Users/loriab/linux/psihub/master/psi4/lib/basis/aug-cc-pvtz-jkfit.gbs
```

- Above is basis {} block, below is custom gbs file
- See (with print > 1) exactly what gets loaded from where
- Results in identical orbital basis sets, but df_basis_scf differ b/c basis {...} block preserves info

```
=> Loading Basis Set <=

 Role: BASIS
 Keyword: BASIS
 Name: MINEBASFILE
 atoms 1, 4   entry O          line    82 file /Users/loriab/linux/psihub/master/psi4/tests/mints9/minebasfile.gbs
 atoms 2, 5-6 entry H          line    30 file /Users/loriab/linux/psihub/master/psi4/tests/mints9/minebasfile.gbs
 atoms 3      entry H_HB       line     8 file /Users/loriab/linux/psihub/master/psi4/tests/mints9/minebasfile.gbs

=> Loading Basis Set <=

 Role: JKFIT
 Keyword: DF_BASIS_SCF
 Name:
 atoms 1, 4     entry O          line   322 file /Users/loriab/linux/psihub/master/psi4/lib/basis/def2-qzvpp-jkfit.gbs
 atoms 2-3, 5-6 entry H          line    22 file /Users/loriab/linux/psihub/master/psi4/lib/basis/def2-qzvpp-jkfit.gbs
```

Programmer's Access to BasisSet Objects

To get a BasisSet object into your module, just call *pyconstruct* where formerly you called *construct*. There are two flavors, one for orbital basis sets and one for auxiliary basis sets. There's no difference in the BasisSet objects they return or even the code used to assemble them- the two flavors are just for sane argument naming and to establish different signatures for Boost binding.

Orbital Basis

Give the function a Molecule object for which to build basis, a label for the basis (generally, BASIS), and a hint for finding the basis. This last argument gets used to find a python function by that name camoflaged (that's what `basis {...}` blocks in the input file get translated into) or failing that a string to find a gbs file defining the basis.

```
// simple
boost::shared_ptr<BasisSet> primary = BasisSet::pyconstruct_orbital(molecule,
    "BASIS", "CC-PVDZ");

// self-contained
boost::shared_ptr<BasisSet> primary = BasisSet::pyconstruct_orbital(Process::environment.molecule(),
    "BASIS", Process::environment.options.get_str("BASIS"));
```

Need outline of route through code? module/libmints/qcdb

Auxiliary Basis

Give the function a Molecule object for which to build basis, a label for the basis, a hint for finding the basis, a fitting role to apply if defaults need to be generated, and a hint for finding the orbital basis to build defaults against.

```
// simple
boost::shared_ptr<BasisSet> auxiliary = BasisSet::pyconstruct_auxiliary(molecule,
    "DF_BASIS_SCF", "",
    "JKFIT", "CC-PVDZ");

// self-contained and force Spherical
boost::shared_ptr<BasisSet> auxiliary = BasisSet::pyconstruct_auxiliary(Process::environment.molecule(),
    "DF_BASIS_SCF", Process::environment.options.get_str("DF_BASIS_SCF"),
    "JKFIT", Process:environment.options.get_str("BASIS"), 1);
```

5D/6D: user PUREAM trumps programmer pyconstruct arg trumps native basis setting

Adding Basis Option to Code

- Register new basis keyword with ⇗psi4/src/bin/psi4/read_options.cc (of course). The default should be the empty string.

```
options.add_str("DF_BASIS_ELST", "");
```

- Register new basis keyword with the input parser ⇗psi4/lib/python/inputparser.py. In the main function *process_input*, add it to the regex below. This ensures that users can define `basis_keyword basis_name {...}` blocks where the contents of the block get associated with basis_name and assigned to your basis_keyword.

```
basis_block = re.compile(r'^(\s*?)(basis|df_basis_scf|df_basis_mp2|df_basis_cc|df_basis_sapt)[=\s]*(\w*?)\s*\{(.*?)\}',
                         re.MULTILINE | re.DOTALL | re.IGNORECASE)
```

Approaching QC Tasks with the Right Tools

Python Crushing a Gnu

Antoine-Louis Barye, c. 1860

Interoperability: the Best of Two Codes

	Psi4	Cfour	Psi4 + Cfour
DFT Grad.	Yes	No	Yes
Anharm. Freq.	No	Yes	Yes
DFT Anharm. Freq.	No	No	Yes

- Restartable, parallel
- Any P4 or C4 grad. mtd.
- Options/basis either prog.
- Values avail. electronically, not just text output file

⟨CC|CC⟩

PSI4

+

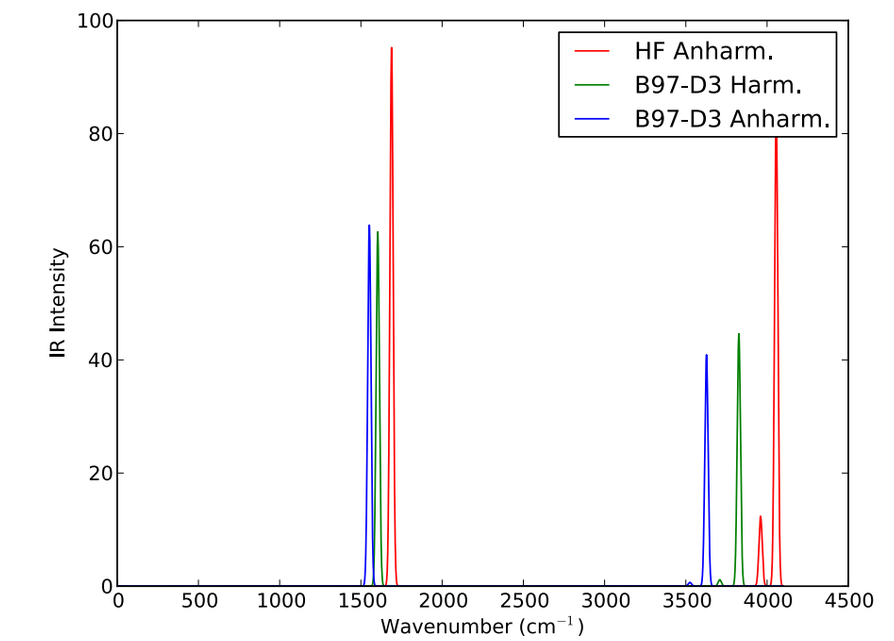
```
memory 4 gb

molecule h2o {
O  -1.51  -0.13   0.00
H  -1.95   0.73   0.00
H  -0.56   0.09   0.00
units angstrom
}

set basis aug-cc-pvdz

vpt2('b97-d3')
```

=

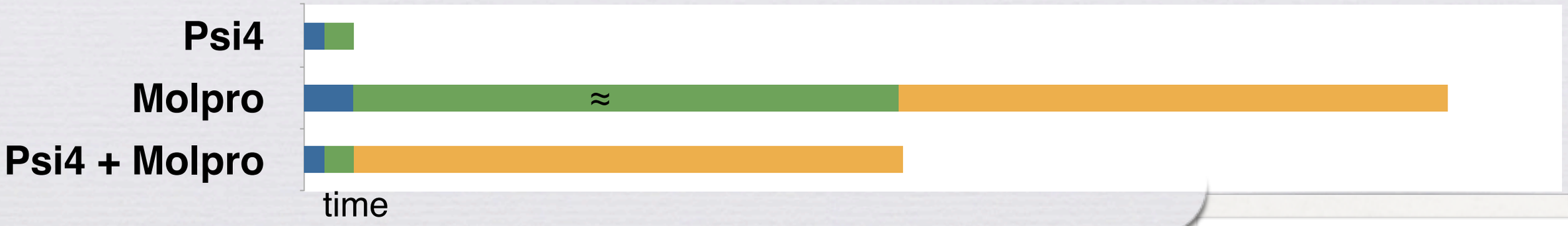


Best SCF-level vibrational spectrum from *Psi4*, from *Cfour*, and *together*.

thanks to Prof. John Stanton & Devin Matthews for help with interface

Interoperability: Pooling Capabilities

MP2 Interaction E. − SAPT0 Dispersion E. + TDDFT Dispersion E. = MP2C Interaction E.



+

PSI4

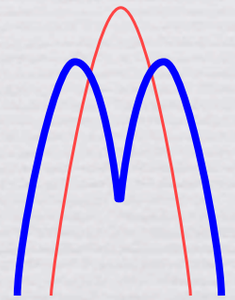
```
memory 4 gb

molecule {
0 1
O  -1.55  -0.11   0.00
H  -1.93   0.76   0.00
H  -0.60   0.04   0.00
--
0 1
O   1.35   0.11   0.00
H   1.68  -0.37  -0.76
H   1.68  -0.37   0.76
}

set basis aug-cc-pvdz

energy('mp2c')
```

=



+

```
***,default
memory,125,m
angstrom
geometry={
geometry={
O  -1.55  -0.11   0.00
H  -1.93   0.76   0.00
H  -0.60   0.04   0.00
O   1.35   0.11   0.00
H   1.68  -0.37  -0.76
H   1.68  -0.37   0.76
}

basis={
set,orbital; default,aug-cc-pvdz
set,mp2fit; default,aug-cc-pvdz/mp2fit
set,jkfit; default,aug-cc-pvdz/jkfit
}

gdirect
{df-hf,basis=jkfit}
{df-mp2,basis_mp2=mp2fit}
```

×3

```
***,set S22 index 2 HB-2 water_dimer cs
memory,125,m
nosym
angstrom
noorient
geometry={
O  -1.55  -0.11   0.00
H  -1.93   0.76   0.00
H  -0.60   0.04   0.00
O   1.35   0.11   0.00
H   1.68  -0.37  -0.76
H   1.68  -0.37   0.76
}

GTHRESH,ZERO=1.e-14,ONEINT=1.e-14,TWOINT=1.e-14,ENERG
Y=1.e-8,ORBITAL=1.e-8,GRID=1.e-8

basis={
set,orbital; default,aug-cc-pVDZ
set,jkfit;   default,aug-cc-pVDZ/jkfit
set,dflhf;   default,aug-cc-pVDZ/jkfit
set,mp2fit;  default,aug-cc-pVDZ/mp2fit
}

gdirect
ga=1101.2; gb=1102.2
ca=2101.2; cb=2102.2

dummy,4,5,6
{df-hf,basis=jkfit,locorb=0; start,atdens; save,$ga}
{df-ks,lhf; dftfac,1.0; start,$ga; save,$ca}

dummy,1,2,3
{df-hf,basis=jkfit,locorb=0; start,atdens; save,$gb}
{df-ks,lhf; dftfac,1.0; start,$gb; save,$cb}

{sapt,SAPT_LEVEL=3;intermol,ca=$ca,cb=
$cb,icpks=0,fitlevel=3,nlexfac=0.0,cfac=0.0
dfit,basis_coul=jkfit,basis_exch=jkfit,cfit_scf=3}
```

×2

```
cbs('mp2', corl_basis='cc-pV[TQ]Z', delta_wfn='c4-ccsd(t)', delta_basis='cc-pV[DT]Z')
```

This yields:

```
==> CBS <==

--------------------------------------------------------------------------------------------------------
   Stage                 Method / Basis                                 Energy [H]   Scheme
--------------------------------------------------------------------------------------------------------
     scf                   scf / cc-pvqz                              -1.10245974   highest_1
    corl                   mp2 / cc-pv[tq]z                           -0.03561890   corl_xtpl
   delta     c4-ccsd(t) - mp2 / cc-pv[dt]z                            0.03507767   corl_xtpl
   total                   CBS                                        -1.10300098
--------------------------------------------------------------------------------------------------------
```

All this is preparatory to separating qcdb and the non-Boost-module-calling portions of the PSI4 driver into common driver which calls psi4 among other progams. ok?

- Psi4-style input for all programs
- Molecule and BasisSet get translated into appropriate format
- options get translated into appropriate format but also selected psi4 options (e.g., scf convergence) get translated into driven program's options (with appropriate non-clobbering)
- drive external program and extract results all through python
- extract results into PsiVariables, PsiMatrices, transform results back to psi4 input orientation
- now results available to any driver wrapper or data writer
- segregate interface python code between those that call the Boost python module psi4 (lib/python/interface_cfour.py, lib/python/wrapper_cfour.py) and the remainder that do the translation/extraction/orientation (lib/python/qcdb)

Common Driver for Quantum Chemistry

GAMESS

NWCHEM

Q-CHEM

PSI4

QUANTUM CHEMISTRY COMMON DRIVER AND DATABASES

PetaChem

⟨CC|CC⟩

PSI Variables

```
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>//
//  Cfour c4-ccsd(t) Energy Results  //
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<//


Variable Map:
----------------------------------------------------------
"(T) CORRECTION ENERGY"       =>      -0.007263598030
"CCSD CORRELATION ENERGY"     =>      -0.275705492359
"CCSD TOTAL ENERGY"           =>     -76.338453952539
"CCSD(T) CORRELATION ENERGY"  =>      -0.007263598030
"CCSD(T) TOTAL ENERGY"        =>     -76.345717550569
"CFOUR ERROR CODE"            =>       0.000000000000
"CURRENT CORRELATION ENERGY"  =>      -0.007263598030
"CURRENT ENERGY"              =>     -76.345717550569
"CURRENT REFERENCE ENERGY"    =>     -76.062748460180
"MP2 CORRELATION ENERGY"      =>      -0.270191667755
"MP2 OPPOSITE-SPIN ENERGY"    =>      -0.204890356651
"MP2 SAME-SPIN ENERGY"        =>      -0.065301311104
"MP2 TOTAL ENERGY"            =>     -76.332940127935
"NUCLEAR REPULSION ENERGY"    =>       9.187331653300
"SCF TOTAL ENERGY"            =>     -76.062748460180
```

```
set basis 6-31g*
db('dfmp2','s22',subset='sm

from pprint import pprint

print_stdout('\nDB_RGT')
pprint(DB_RGT)

print_stdout('\nDB_RXN')
pprint(DB_RXN)

print_stdout('\ndf-mp2 interaction energy of water dimer (S22-2)')
print_stdout(DB_RXN['S22-2']['CURRENT ENERGY'])
```

The output to the screen is as follows.

```
DB_RGT
{'S22-16-dimer': {'CURRENT ENERGY': -155.37373581838636,
                  'DF-MP2 CORRELATION ENERGY': -0.523870772178089},
 'S22-16-monoA-unCP': {'CURRENT ENERGY': -78.29412053242164,
                       'DF-MP2 CORRELATION ENERGY': -0.2629759351596186},
 'S22-16-monoB-unCP': {'CURRENT ENERGY': -77.07606823017188,
                       'DF-MP2 CORRELATION ENERGY': -0.2594122526144091},
 'S22-2-dimer': {'CURRENT ENERGY': -152.40958884746667,
                 'DF-MP2 CORRELATION ENERGY': -0.3797598812113561},
 'S22-2-monoA-unCP': {'CURRENT ENERGY': -76.19905879745446,
                      'DF-MP2 CORRELATION ENERGY': -0.1887118848315123},
 'S22-2-monoB-unCP': {'CURRENT ENERGY': -76.19902978067739,
                      'DF-MP2 CORRELATION ENERGY': -0.1885738493735463 5},
 'S22-8-dimer': {'CURRENT ENERGY': -80.67416758080654,
                 'DF-MP2 CORRELATION ENERGY': -0.2844102558783027},
 'S22-8-monoA-unCP': {'CURRENT ENERGY': -40.336952636980364,
                      'DF-MP2 CORRELATION ENERGY': -0.14185962536715307},
 'S22-8-monoB-unCP': {'CURRENT ENERGY': -40.336952636980506,
                      'DF-MP2 CORRELATION ENERGY': -0.14185962536715097}}

DB_RXN
{'S22-16': {'CURRENT ENERGY': -0.0035470557928363178,
            'DF-MP2 CORRELATION ENERGY': -0.0014825844040612934},
 'S22-2': {'CURRENT ENERGY': -0.011500269334817403,
           'DF-MP2 CORRELATION ENERGY': -0.0024741470062974724},
 'S22-8': {'CURRENT ENERGY': -0.0002623068456699684,
           'DF-MP2 CORRELATION ENERGY': -0.0006910051439986686}}
```

- QC language

Communicate to user in text form in output file

Communicate to user in data form through python dicts available after most wrappers

Communicate QC results to driver/wrapper. Once non-psi4 output translated to PsiVariables, MP2 correlation energy from Cfour looks like MP2 correlation energy from Psi4.

- New program acts on arrays of PsiVariable data (through pandas) and applies all the little QC equalities we all know to assemble all available model chemistries.

```
pv1['SCS-MP2 TOTAL ENERGY'] = {'func': sum, 'args': ['HF TOTAL ENERGY', 'SCS-MP2 CORRELATION ENERGY']}
pv1['SCS(N)-MP2 CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [0.0, 1.76, 'MP2 CORRELATION ENERGY', 'MP2 SAME-SPIN CORRELAT
pv1['SCS(N)-MP2 TOTAL ENERGY'] = {'func': sum, 'args': ['HF TOTAL ENERGY', 'SCS(N)-MP2 CORRELATION ENERGY']}
pv1['DW-MP2 CORRELATION ENERGY'] = {'func': dispersion_weighting, 'args': ['DW-MP2 OMEGA', 'MP2 CORRELATION ENERGY', 'SCS-MP2 CORRELATION EN
pv1['DW-MP2 TOTAL ENERGY'] = {'func': sum, 'args': ['HF TOTAL ENERGY', 'DW-MP2 CORRELATION ENERGY']}
pv1['SCS-MP2-F12 CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [1.2, 1.0/3.0, 'MP2-F12 CORRELATION ENERGY', 'MP2-F12 SAME-S
pv1['SCS-MP2-F12 TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'SCS-MP2-F12 CORRELATION ENERGY']}
pv1['SCS(N)-MP2-F12 CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [0.0, 1.76, 'MP2-F12 CORRELATION ENERGY', 'MP2-F12 SAME-S
pv1['SCS(N)-MP2-F12 TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'SCS(N)-MP2-F12 CORRELATION ENERGY']}
pv1['DW-MP2-F12 CORRELATION ENERGY'] = {'func': dispersion_weighting, 'args': ['DW-MP2-F12 OMEGA', 'MP2-F12 CORRELATION ENERGY', 'SCS-MP2-F12
pv1['DW-MP2-F12 TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'DW-MP2-F12 CORRELATION ENERGY']}
pv1['SCS-CCSD-F12A CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [1.27, 1.13, 'CCSD-F12A CORRELATION ENERGY', 'CCSD-F12A SA
pv1['SCS-CCSD-F12A TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'SCS-CCSD-F12A CORRELATION ENERGY']}
pv1['SCS(MI)-CCSD-F12A CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [1.11, 1.28, 'CCSD-F12A CORRELATION ENERGY', 'CCSD-F12
pv1['SCS(MI)-CCSD-F12A TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'SCS(MI)-CCSD-F12A CORRELATION ENERGY']}
pv1['SCS-CCSD-F12B CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [1.27, 1.13, 'CCSD-F12B CORRELATION ENERGY', 'CCSD-F12B SA
pv1['SCS-CCSD-F12B TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'SCS-CCSD-F12B CORRELATION ENERGY']}
pv1['SCS(MI)-CCSD-F12B CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [1.11, 1.28, 'CCSD-F12B CORRELATION ENERGY', 'CCSD-F12
pv1['SCS(MI)-CCSD-F12B TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'SCS(MI)-CCSD-F12B CORRELATION ENERGY']}
pv1['SCS-CCSD-F12C CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [1.27, 1.13, 'CCSD-F12C CORRELATION ENERGY', 'CCSD-F12C SA
pv1['SCS-CCSD-F12C TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'SCS-CCSD-F12C CORRELATION ENERGY']}
pv1['SCS(MI)-CCSD-F12C CORRELATION ENERGY'] = {'func': spin_component_scaling, 'args': [1.11, 1.28, 'CCSD-F12C CORRELATION ENERGY', 'CCSD-F12
pv1['DW-CCSD(T**)-F12 TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'DW-CCSD(T**)-F12 CORRELATION ENERGY']}
pv1['DW-CCSD(T**)-F12 CC CORRECTION ENERGY'] = {'func': difference, 'args': ['DW-CCSD(T**)-F12 CORRELATION ENERGY', 'MP2-F12 CORRELATION ENER
pv1['MP2C CC CORRECTION ENERGY'] = {'func': difference, 'args': ['MP2C DISP20 ENERGY', 'SAPT DISP20 ENERGY']}
pv1['MP2C CORRELATION ENERGY'] = {'func': sum, 'args': ['MP2C CC CORRECTION ENERGY', 'MP2 CORRELATION ENERGY']}
pv1['MP2C TOTAL ENERGY'] = {'func': sum, 'args': ['HF TOTAL ENERGY', 'MP2C CORRELATION ENERGY']}
pv1['MP2C-F12 CC CORRECTION ENERGY'] = {'func': sum, 'args': ['MP2C CC CORRECTION ENERGY']}
pv1['MP2C-F12 CORRELATION ENERGY'] = {'func': sum, 'args': ['MP2C CC CORRECTION ENERGY', 'MP2-
pv1['MP2C-F12 TOTAL ENERGY'] = {'func': sum, 'args': ['HF-CABS TOTAL ENERGY', 'MP2C-F12 CORREL
pv1['SAPT EXCHSCAL3'] = {'func': lambda x: x[0] ** 3, 'args': ['SAPT EXCHSCAL']}
pv1['SAPT HF(2) ALPHA=0.0 ENERGY'] = {'func': lambda x: x[0] - (x[1] + x[2] + x[3] + x[4]),
                                      'args': ['SAPT HF TOTAL ENERGY', 'SAPT ELST10,R ENERGY', 'SAPT EXC
                                               'SAPT IND20,R ENERGY', 'SAPT EXCH-IND20,R ENERGY']}
pv1['SAPT HF(2) ENERGY'] = {'func': lambda x: x[1] + (1.0 - x[0]) * x[2],
                                               'args': ['SAPT EXCHSCAL', 'SAPT HF(2) ALPHA=0.0 ENERGY',
pv1['SAPT HF(3) ENERGY'] = {'func': lambda x: x[1] - (x[2] + x[0] * x[3]),
                                      'args': ['SAPT EXCHSCAL', 'SAPT HF(2) ENERGY', 'SAPT IND30,R ENERGY', 'SAPT EXCH-IND30,R ENERGY']}
pv1['SAPT MP2(2) ENERGY'] = {'func': lambda x: x[1] - (x[2] + x[3] + x[4] + x[0] * (x[5] + x[6] + x[7] + x[8])),
                                       'args': ['SAPT EXCHSCAL', 'MP2 CORRELATION ENERGY', 'SAPT ELST12,R ENERGY',
                                                'SAPT IND22 ENERGY', 'SAPT DISP20 ENERGY', 'SAPT EXCH11(S^2) ENERGY',
                                                'SAPT EXCH12(S^2) ENERGY', 'SAPT EXCH-IND22 ENERGY', 'SAPT EXCH-DISP20 ENERGY']}
pv1['SAPT MP2(3) ENERGY'] = {'func': lambda x: x[1] - (x[2] + x[0] * x[3]),
                                       'args': ['SAPT EXCHSCAL', 'SAPT MP2(2) ENERGY', 'SAPT IND-DISP30 ENERGY', 'SAPT EXCH-IND-DISP30 ENERGY']}
```

Implement this in psi4 so that programmers only need define minimum psi variables and remainder are derived/validated. ok?

```
pv2['a56z'] = {'func': xtpl_power, 'args': [3.0, 6, 'a6z', 'a5z']}
pv2['dtzf12'] = {'func': xtpl_power, 'args': [3.0, 3, 'tzf12', 'dzf12']}
pv2['tqzf12'] = {'func': xtpl_power, 'args': [3.0, 4, 'qzf12', 'tzf12']}
# Hill xtpl for CCSD-F12b from Table X of JCP 131 194105 (2009)
    # TODO should only be applied to CCF12, as per definition (literally, only CCF12B)
pv2['hillcc_adtz'] = {'func': xtpl_power, 'args': [2.483070, 3, 'atz', 'adz']}
pv2['hillcc_atqz'] = {'func': xtpl_power, 'args': [4.255221, 4, 'aqz', 'atz']}
```

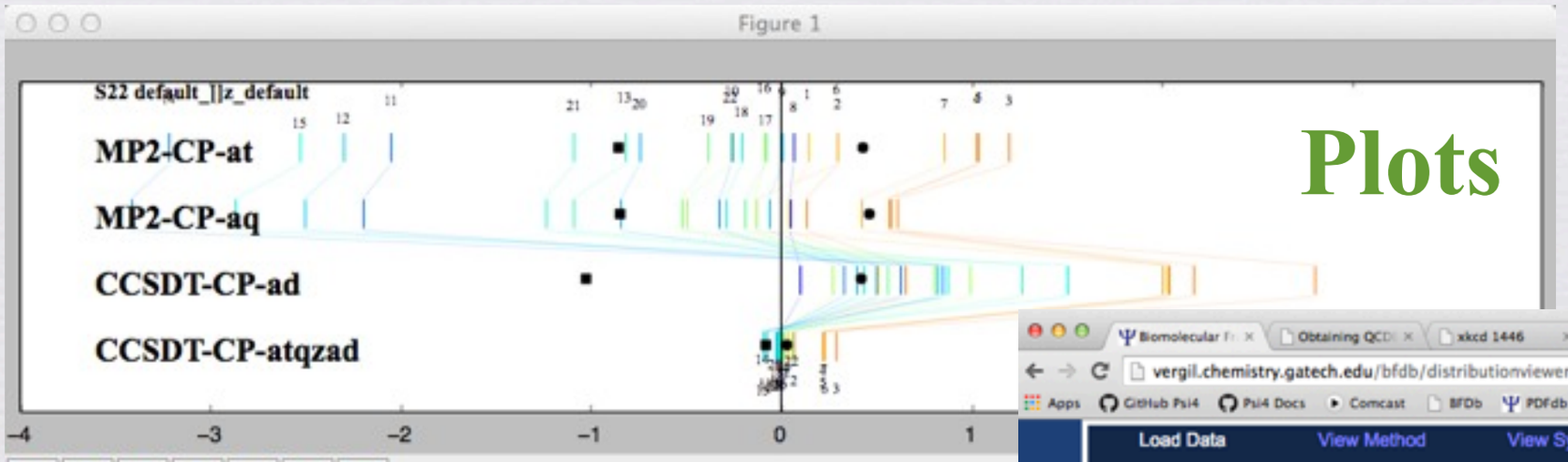
Extend PsiVariables to PsiArrays for gradients, etc.

QDDB Frills

```
import sys
sys.path.append('/Users/loriab/linux/qcdb')
import qcdb

asdf = qcdb.Database(['s22'])
asdf.load_qcdata_byproject('pt2')

print asdf
print asdf.mcs.keys()
asdf.plot_modelchems(['MP2-CP-atz', 'MP2-CP-aqz', 'CCSDT-CP-adz', 'CCSDT-CP-atqzadz'])
```



Plots

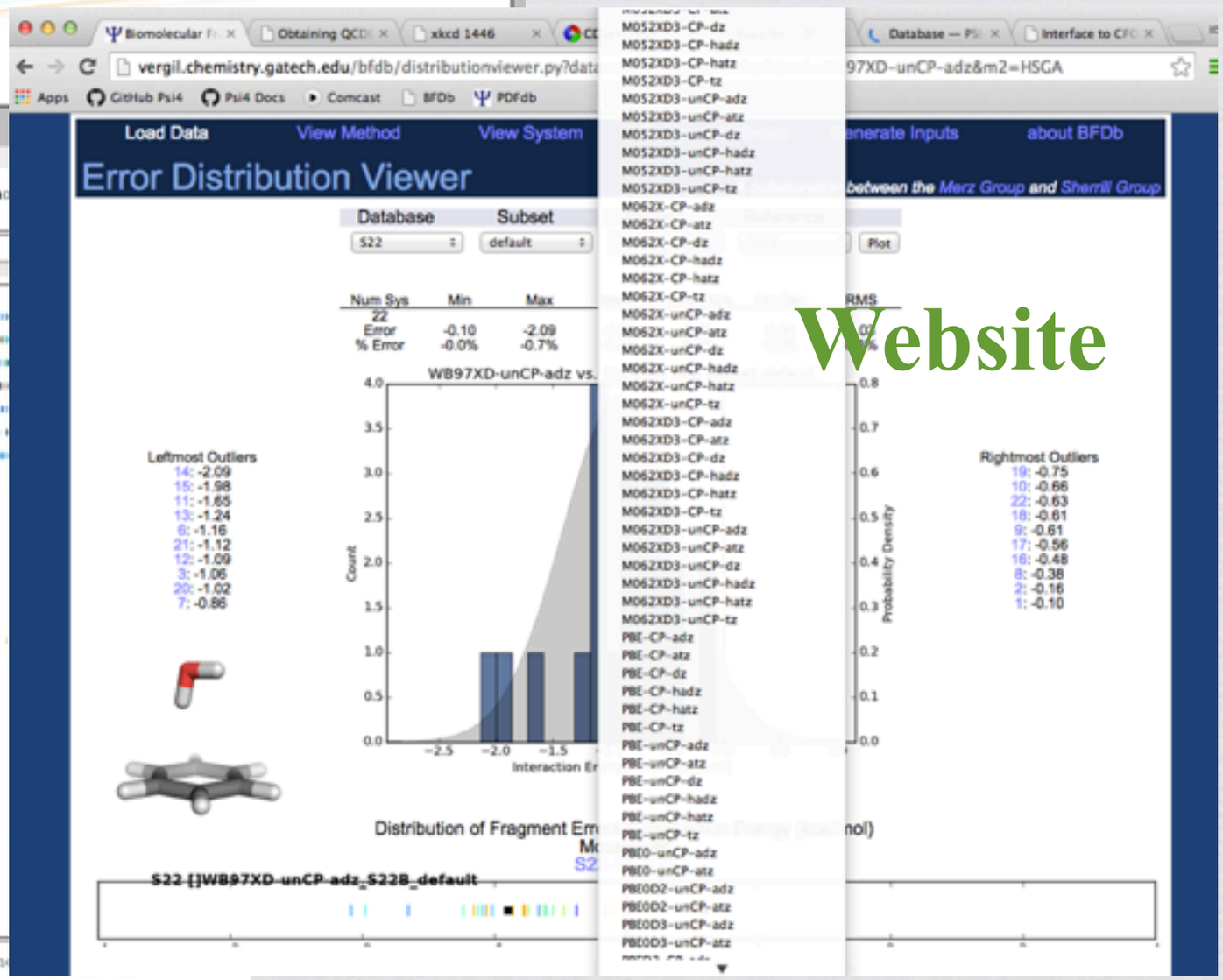
LaTeX Tables

TABLE 4. Mean absolute error (kcal/mol) of the interaction energy for databases and their subsets with DFT metho
aug-cc-pVDZ.

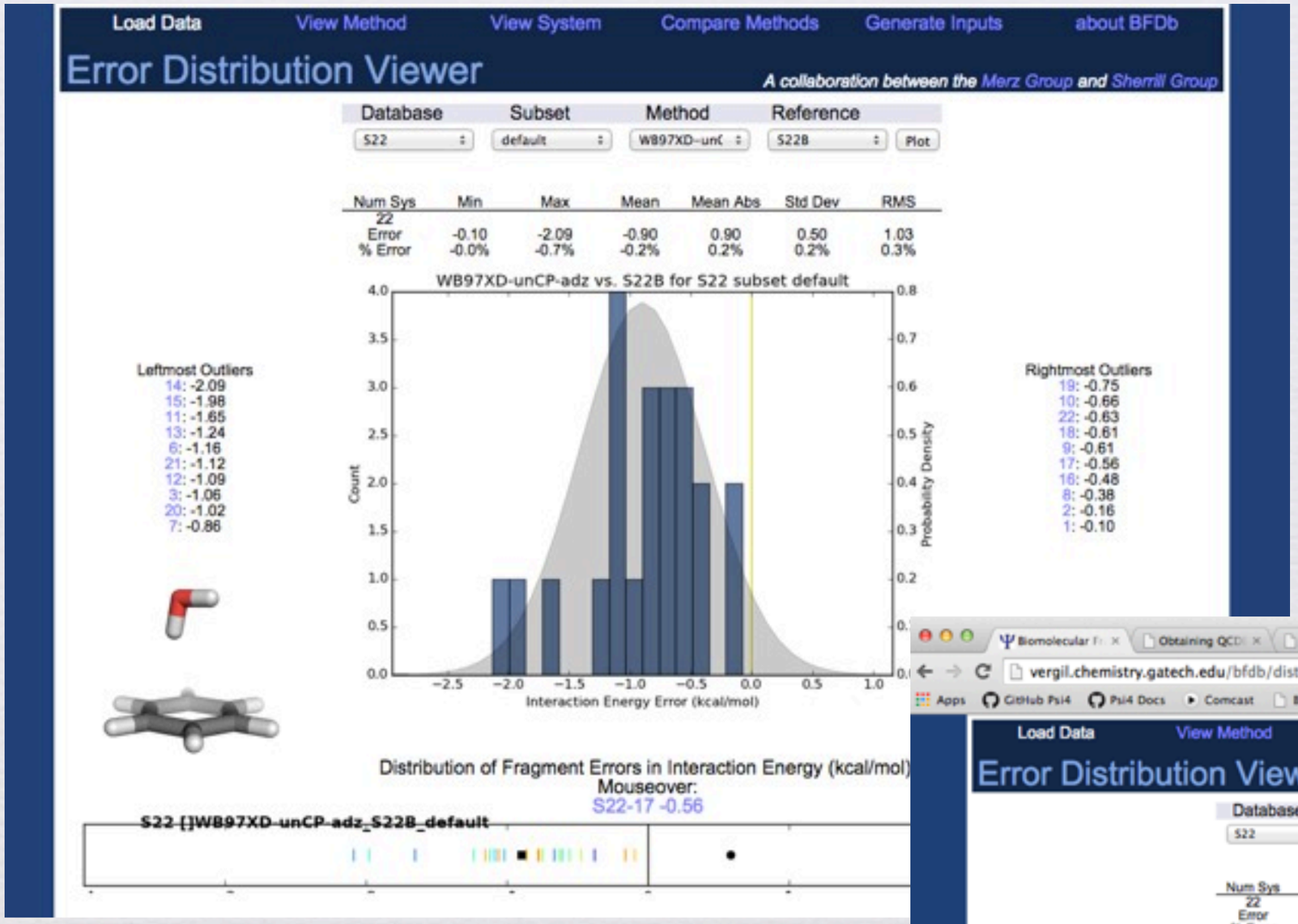
Method & Basis Set	S22				NBC10			HBC6	HSG				Avg	Error Distribution[a]
	HB	MX	DD	TT	MX	DD	TT	HB/TT	HB	MX	DD	TT	TT	
aug-cc-pVDZ														
B97	3.73	5.65	4.59	4.70	4.08	3.16	3.44	2.26	4.57	2.42	2.74	2.94	3.33	
BP86	1.76	4.48	3.83	3.40	3.77	2.88	3.00	0.93	2.66	2.32	2.37	2.40	2.43	
B3LYP	1.[illegible]	4.13	3.70	[illegible]	3.52	2.82	3.00	0.74	2.27	1.61	2.00	1.97	2.22	
B970	[illegible]	4.43	3.82	[illegible]	2.75	2.09	2.28	0.83	2.30	1.25	1.50	1.57	1.85	
PBE	0.60	2.82	2.35	1.96	2.05	1.71	1.82	0.39	1.10	1.00	1.06	1.05	1.31	
PBE0	0.49	2.42	2.20	1.74	1.92	1.66	1.74	0.47	0.95	0.99	1.20	1.13	1.27	
B2PLYP	0.[illegible]	1.[illegible]	1.[illegible]	1.[illegible]	1.[illegible]	0.95	1.02	0.30	0.85	0.49	0.58	0.60	0.74	
XYG3	0.[illegible]	1.[illegible]	1.[illegible]	1.[illegible]	0.[illegible]	0.42	0.43	0.62	0.84	0.46	0.47	0.52	0.52	
M05-2X	0.45	0.40	0.24	0.37	0.15	0.25	0.22	0.46	0.29	0.15	0.17	0.20	0.31	
PBE0-D2	1.32	0.61	0.33	0.75	0.32	0.21	0.24	1.36	0.80	0.27	0.53	0.52	0.72	
PBE-D2	1.46	0.97	0.78	1.07	0.73	0.50	0.57	1.32	1.09	0.58	1.10	1.00	0.99	
B970-D2	0.31	0.47	0.23	0.37	0.15	0.19	0.18	0.46	0.54	0.33	0.66	0.58	0.40	
BP86-D2	1.08	0.82	0.58	0.82	0.83	0.38	0.52	1.00	0.41	0.24	0.67	0.55	0.72	
B2PLYP-D2	0.90	1.48	1.09	1.17	0.86	0.67	0.73	0.63	0.76	0.67	1.00	0.90	0.86	
B97-D2	0.51	0.74	0.64	0.63	0.58	0.52	0.54	0.49	1.11	0.21	0.86	0.77	0.61	
B3LYP-D2	1.37	1.17	0.63	1.06	0.54	0.28	0.36	1.03	0.80	0.57	1.02	0.91	0.84	
M06-2X	0.38	0.83	0.82	0.69	0.55	0.47	0.50	0.45	0.68	0.15	0.37	0.37	0.50	
ωB97X-D	0.70	0.92	1.07	0.90	0.78	0.82	0.81	0.72	0.43	0.43	0.88	0.73	0.79	
B3LYP-XDM	0.48	0.30	0.49	0.41	0.40	0.33	0.35	0.61	0.47	0.38	0.51	0.48	0.46	
M05-2X-D3	0.37	1.09	0.80	0.80	0.65	0.67	0.67	0.70	0.95	0.90	0.84	0.87	0.76	
PBE0-D3	1.38	0.84	0.65	0.95	0.62	0.56	0.58	1.43	0.96	0.73	0.81	0.82	0.94	
PBE-D3	1.10	0.51	0.55	0.71	0.51	0.56	0.55	1.12	0.80	0.75	0.99	0.92	0.82	
BP86-D3	1.25	1.27	1.20	1.24	1.17	0.77	0.89	1.13	0.62	0.39	0.90	0.77	1.01	
B2PLYP-D3	0.77	1.23	0.97	1.00	0.76	0.65	0.68	0.58	0.64	0.76	0.89	0.83	0.77	
B97-D3	0.28	0.53	0.60	0.47	0.62	0.43	0.49	0.41	0.73	0.29	0.67	0.60	0.49	
B3LYP-D3	1.11	0.79	0.50	0.80	0.39	0.35	0.36	0.93	0.59	0.74	0.84	0.79	0.72	
M06-2X-D3	0.26	1.22	1.17	0.90	0.83	0.78	0.80	0.56	0.97	0.42	0.73	0.71	0.74	

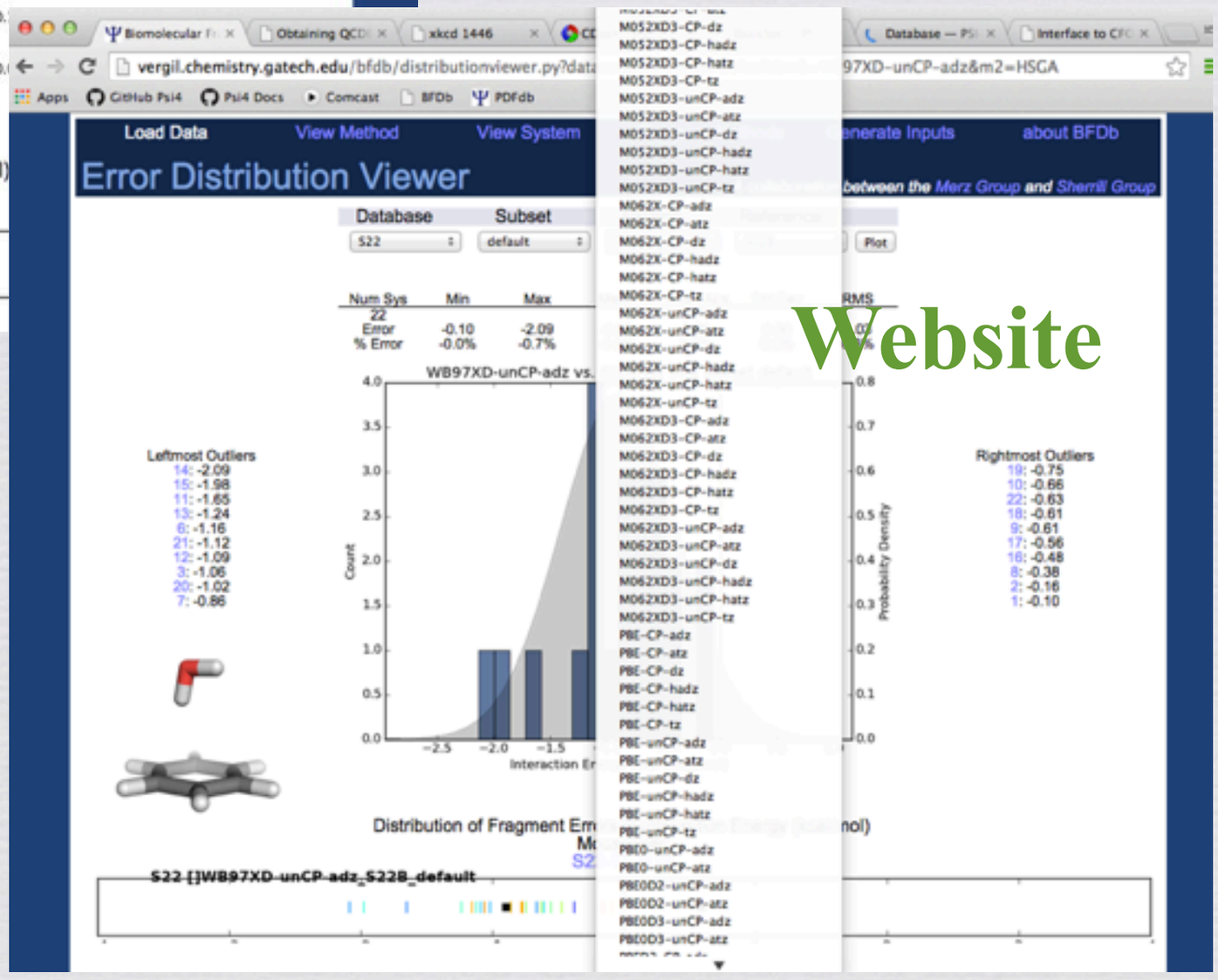
[a] Errors with respect to Gold Standard (see Sec. II D for plot details). Guide lines are at 0, 0.3, and 1.0 kcal/mol overbound (−) an
[b] Only equilibrium and near-equilibrium systems included. (All S22 and HSG, 50/194 NBC10, 28/118 HBC6.)

Website



QDDB Frills


Load Data
View Method
View System
Compare Methods
Generate Inputs
about BFDb
Error Distribution Viewer
A collaboration between the Merz Group and Sherrill Group
Database
Subset
Method
Reference
S22
default
WB97XD-unC
S22B
Plot
Num Sys
Min
Max
Mean
Mean Abs
Std Dev
RMS
22
Error
-0.10
-2.09
-0.90
0.90
0.50
1.03
% Error
-0.0%
-0.7%
-0.2%
0.2%
0.2%
0.3%
WB97XD-unCP-adz vs. S22B for S22 subset default
Count
Probability Density
Interaction Energy Error (kcal/mol)
Leftmost Outliers
14: -2.09
15: -1.98
11: -1.65
13: -1.24
6: -1.16
21: -1.12
12: -1.09
3: -1.06
20: -1.02
7: -0.86
Rightmost Outliers
19: -0.75
10: -0.66
22: -0.63
18: -0.61
9: -0.61
17: -0.56
16: -0.48
8: -0.38
2: -0.16
1: -0.10
Distribution of Fragment Errors in Interaction Energy (kcal/mol)
Mouseover:
S22-17 -0.56
S22 []WB97XD-unCP-adz_S22B_default


vergil.chemistry.gatech.edu/bfdb/distributionviewer.py?data
Load Data
View Method
View System
Error Distribution Viewer
Database
Subset
S22
default
Plot
M052XD3-CP-dz
M052XD3-CP-hadz
M052XD3-CP-hatz
M052XD3-CP-tz
M052XD3-unCP-adz
M052XD3-unCP-atz
M052XD3-unCP-dz
M052XD3-unCP-hadz
M052XD3-unCP-hatz
M052XD3-unCP-tz
M062X-CP-adz
M062X-CP-atz
M062X-CP-dz
M062X-CP-hadz
M062X-CP-hatz
M062X-CP-tz
M062X-unCP-adz
M062X-unCP-atz
M062X-unCP-dz
M062X-unCP-hadz
M062X-unCP-hatz
M062X-unCP-tz
M062XD3-CP-adz
M062XD3-CP-atz
M062XD3-CP-dz
M062XD3-CP-hadz
M062XD3-CP-hatz
M062XD3-CP-tz
M062XD3-unCP-adz
M062XD3-unCP-atz
M062XD3-unCP-dz
M062XD3-unCP-hadz
M062XD3-unCP-hatz
M062XD3-unCP-tz
PBE-CP-adz
PBE-CP-atz
PBE-CP-dz
PBE-CP-hadz
PBE-CP-hatz
PBE-CP-tz
PBE-unCP-adz
PBE-unCP-atz
PBE-unCP-dz
PBE-unCP-hadz
PBE-unCP-hatz
PBE-unCP-tz
PBE0-unCP-adz
PBE0-unCP-atz
PBE0D2-unCP-adz
PBE0D2-unCP-atz
PBE0D3-unCP-adz
PBE0D3-unCP-atz
Website

Public Service Announcements

- New docs target “make sphinxmini” that bypasses most of the autodocing steps to build quickly.

Interest in a “state variable” for options (beyond the “energy(‘scf’)” runtime Options object) that incorporates options settings from user (check), driver, read_option (check), and module complex-defaulting?

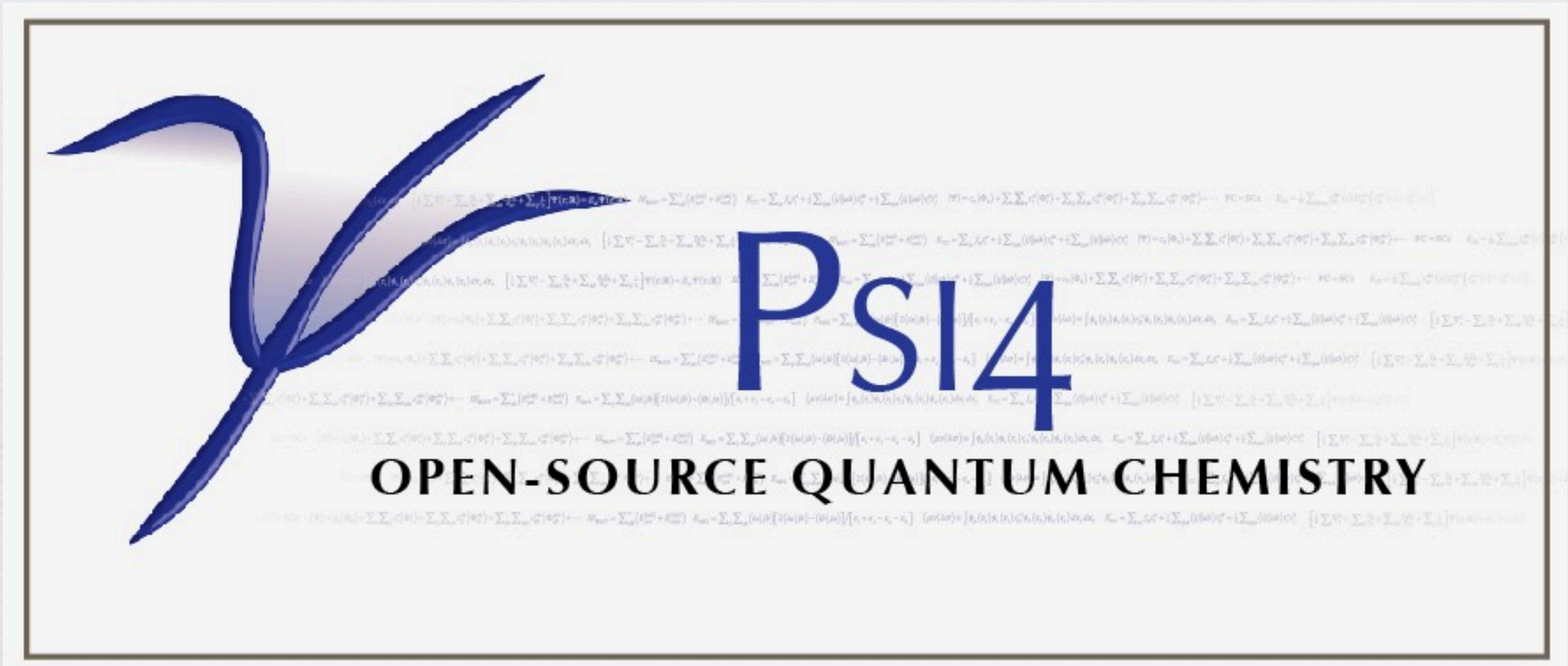
Psi4 Logos: SMALLCAPS and Implications

Do we want "Psi" in small caps? If so, what happens with the "4"?

Psi4 Logos: SMALLCAPS and Implications

Current

Psi4 Logos: SMALLCAPS and Implications



Database Tools

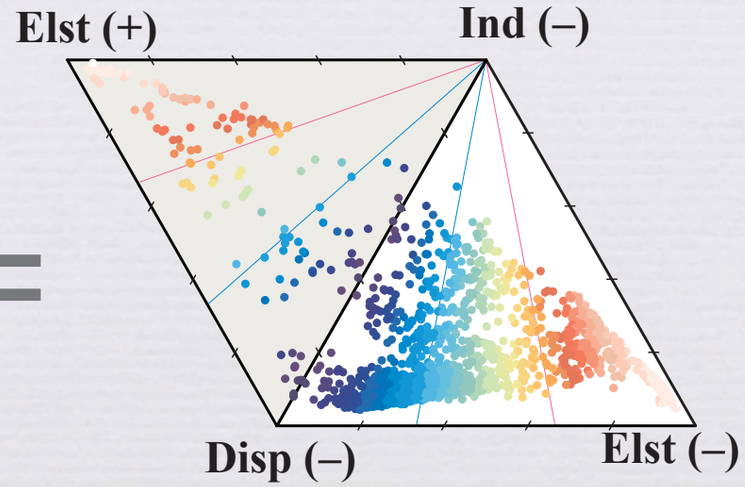
PSI4 +

```
memory 2.5 gb

set basis jun-cc-pvdz

database('sapt0', 'ssi')
```

=



Databases built into Psi4 simplify job execution and result collection

- Database is simple python (text) file
- Build by script from xyz files
- Restartable, parallel
- Tabulate QC values
- Access any model chemistry
- Couple with other wrappers (*e.g.*, basis extrapolation, geometry opt.)

Each colored dot characterizes a noncovalent interaction in the database *SSI*, as computed by the method *SAPT0*.

Faver, Burns, Marshall, Zheng, Sherrill, & Merz, *in preparation*.