Nexus User and Developer Meeting

Meeting 5: Quantum Package and QMCPACK

Jaron T. Krogel

26 April 2019

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Agenda for this Meeting

- Overview of Quantum Package + Nexus Interface

- Workflow demos:
  - Quantum Package: H_2O Hartree-Fock
  - Quantum Package: O_2 Selected CI
  - Quantum Package + QMCPACK: H_2O Hartree Fock ➞ VMC/DMC
  - Quantum Package + QMCPACK: O_2 Selected CI➞ VMC/DMC

- Q&A about Nexus features, workflow how to's, etc. + Updates

Files located at: `nexus_training/monthly_meetings/05_190426_qp_qmcpack/`

Overview of Quantum Package

Basic Quantum Package Usage: H_2O Hartree-Fock

```
# make quantum package executables available
source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc

# create basic ezfio input with a cc-pvtz basis
qp_create_ezfio -b cc-pvtz h2o.xyz

# force all QP variables to be created and initialized
qp_edit -c h2o.ezfio

# request one and two body integrals be saved
echo "Write" > h2o.ezfio/ao_one_e_ints/io_ao_one_e_integrals
echo "Write" > h2o.ezfio/mo_one_e_ints/io_mo_one_e_integrals
echo "Write" > h2o.ezfio/ao_two_e_ints/io_ao_two_e_integrals
echo "Write" > h2o.ezfio/mo_two_e_ints/io_mo_two_e_integrals

# run HF job
export OMP_NUM_THREADS=12
qp_run scf h2o.ezfio >h2o_scf.output 2>h2o_scf.error&
```

```
viribus>tail h2o_scf.output
.. >>>>> [ RES  MEM :       0.035549 GB ] [ VIRT MEM :       1.332722 GB ] <<<<< ..
.. >>>>> [ WALL TIME:       1.659105  s ] [ CPU  TIME:      17.032072  s ] <<<<< ..

* SCF energy                                        -76.03027837147562

.. >>>>> [ RES  MEM :       0.035549 GB ] [ VIRT MEM :       1.332722 GB ] <<<<< ..
.. >>>>> [ WALL TIME:       1.681538  s ] [ CPU  TIME:      17.243335  s ] <<<<< ..

Wall time : 3.1613965034484863s
```

For more information on running Quantum Package from the command line:
https://quantum-package.readthedocs.io/en/master/index.html

Quantum Package Input

- Quantum Package ezfio serves function of input file
- Really it is persistent program state on disk, so each run modifies and/or resets inputs
- Variable names are shown in bold below

See QP input docs: https://quantum-package.readthedocs.io/en/master/programmers_guide/index.html#index-of-modules

```
viribus>qp_create_ezfio -b cc-pvtz h2o.xyz
h2o.ezfio
viribus>qp_edit -c h2o.ezfio
viribus>tree h2o.ezfio/
h2o.ezfio/
├── ao_basis
│   ├── ao_basis
│   ├── ao_cartesian
│   ├── ao_coef.gz
│   ├── ao_expo.gz
│   ├── ao_md5
│   ├── ao_nucl.gz
│   ├── ao_num
│   ├── ao_power.gz
│   └── ao_prim_num.gz
├── ao_one_e_ints
│   ├── io_ao_integrals_e_n
│   ├── io_ao_integrals_kinetic
│   ├── io_ao_integrals_overlap
│   ├── io_ao_integrals_pseudo
│   └── io_ao_one_e_integrals
├── ao_two_e_erf_ints
│   ├── io_ao_two_e_integrals_erf
│   └── mu_erf
├── ao_two_e_ints
│   ├── direct
│   ├── io_ao_two_e_integrals
│   └── threshold_ao
├── becke_numerical_grid
│   └── grid_type_sgn
├── davidson
│   ├── davidson_sze_max
│   ├── disk_based_davidson
│   ├── distributed_davidson
│   ├── n_det_max_full
│   ├── n_states_diag
│   ├── only_expected_s2
│   ├── state_following
│   └── threshold_davidson
├── density_for_dft
│   ├── damping_for_rs_dft
│   ├── density_for_dft
│   └── no_core_density
├── determinants
│   ├── n_det
│   ├── n_det_max
│   ├── n_det_print_wf
│   ├── n_states
│   ├── read_wf
│   ├── s2_eig
│   ├── target_energy
│   ├── threshold_generators
│   └── used_weight
├── dft_keywords
│   ├── correlation_functional
│   ├── exchange_functional
│   └── hf_exchange
├── dressing
│   ├── dress_relative_error
│   ├── n_it_max_dressed_ci
│   └── thresh_dressed_ci
├── electrons
│   ├── elec_alpha_num
│   └── elec_beta_num
├── ezfio
│   ├── creation
│   ├── library
│   └── user
├── ijkl_ints_in_r3
│   ├── disk_access_ao_ijkl_r3
│   └── disk_access_mo_ijkl_r3
├── mo_one_e_ints
│   ├── io_mo_integrals_e_n
│   ├── io_mo_integrals_kinetic
│   ├── io_mo_integrals_pseudo
│   └── io_mo_one_e_integrals
├── mo_two_e_erf_ints
│   └── io_mo_two_e_integrals_erf
├── mo_two_e_ints
│   ├── io_mo_two_e_integrals
│   ├── no_ivvv_integrals
│   ├── no_vvv_integrals
│   ├── no_vvvv_integrals
│   └── threshold_mo
├── nuclei
│   ├── disk_access_nuclear_repulsion
│   ├── nucl_charge.gz
│   ├── nucl_coord.gz
│   ├── nucl_label.gz
│   └── nucl_num
├── perturbation
│   ├── correlation_energy_ratio_max
│   ├── do_pt2
│   ├── h0_type
│   ├── pt2_max
│   ├── pt2_relative_error
│   └── variance_max
├── pseudo
│   ├── do_pseudo
│   ├── pseudo_dz_k.gz
│   ├── pseudo_dz_kl.gz
│   ├── pseudo_grid_rmax
│   ├── pseudo_grid_size
│   ├── pseudo_klocmax
│   ├── pseudo_kmax
│   ├── pseudo_lmax
│   ├── pseudo_n_k.gz
│   ├── pseudo_n_kl.gz
│   ├── pseudo_v_k.gz
│   └── pseudo_v_kl.gz
├── qmcpack
│   └── ci_threshold
├── rsdft_ecmd
│   └── ecmd_functional
├── scf_utils
│   ├── frozen_orb_scf
│   ├── level_shift
│   ├── max_dim_diis
│   ├── mo_guess_type
│   ├── n_it_scf_max
│   ├── scf_algorithm
│   ├── threshold_diis
│   └── thresh_scf
├── two_body_dm
│   ├── ci_threshold
│   ├── mat_mul_svd_vectors
│   ├── ontop_approx
│   └── thr_ontop_approx
└── work
    └── empty
```

Nexus Quantum Package Input

```
>>> from nexus import QuantumPackageInput
>>> qpi = QuantumPackageInput('./h2o.ezfio')
>>> print qpi
  structure       = None
  ao_basis
    ao_basis        = cc-pvtz
    ao_cartesian    = False
    ao_md5          = b0e5878a56051339b81909a4b36ceeef
    ao_num          = 65
  end ao_basis
  ao_one_e_ints
    io_ao_integrals_e_n = None
    io_ao_integrals_kinetic = None
    io_ao_integrals_overlap = None
    io_ao_integrals_pseudo = None
    io_ao_one_e_integrals = None
  end ao_one_e_ints
  ao_two_e_erf_ints
    io_ao_two_e_integrals_erf = None
    mu_erf          = 0.5
  end ao_two_e_erf_ints
  ao_two_e_ints
    direct          = False
    io_ao_two_e_integrals = None
    threshold_ao    = 1e-15
  end ao_two_e_ints
  becke_numerical_grid
    grid_type_sgn   = 2
  end becke_numerical_grid
  davidson
    davidson_sze_max = 15
    disk_based_davidson = True
    distributed_davidson = True
    n_det_max_full  = 1000
    n_states_diag   = 4
    only_expected_s2 = True
    state_following = False
    threshold_davidson = 1e-10
  end davidson
  density_for_dft
    damping_for_rs_dft = 0.5
    density_for_dft = WFT
    no_core_density = full_density
  end density_for_dft
 determinants
    n_det           = 1
    n_det_max       = 1000000
    n_det_print_wf  = 10000
    n_states        = 1
    read_wf         = False
    s2_eig          = True
    target_energy   = 0.0
    threshold_generators = 0.99
    used_weight     = 1
  end determinants
  dft_keywords
    correlation_functional = short_range_LDA
    exchange_functional = short_range_LDA
    hf_exchange     = 0.0
  end dft_keywords
  dressing
    dress_relative_error = 0.001
    n_it_max_dressed_ci = 10
    thresh_dressed_ci = 1e-05
  end dressing
  electrons
    elec_alpha_num  = 5
    elec_beta_num   = 5
  end electrons
  ijkl_ints_in_r3
    disk_access_ao_ijkl_r3 = None
    disk_access_mo_ijkl_r3 = None
  end ijkl_ints_in_r3
  mo_one_e_ints
    io_mo_integrals_e_n = None
    io_mo_integrals_kinetic = None
    io_mo_integrals_pseudo = None
    io_mo_one_e_integrals = None
  end mo_one_e_ints
  mo_two_e_erf_ints
    io_mo_two_e_integrals_erf = None
  end mo_two_e_erf_ints
  mo_two_e_ints
    io_mo_two_e_integrals = None
    no_ivvv_integrals = False
    no_vvv_integrals = False
    no_vvvv_integrals = False
    threshold_mo    = 1e-15
  end mo_two_e_ints
  nuclei
    disk_access_nuclear_repulsion = None
    nucl_num        = 3
  end nuclei
  perturbation
    correlation_energy_ratio_max = 1.0
    do_pt2          = True
    h0_type         = EN
    pt2_max         = 0.0001
    pt2_relative_error = 0.002
    variance_max    = 0.0
  end perturbation
  pseudo
    do_pseudo       = False
    pseudo_grid_rmax = 10.0
    pseudo_grid_size = 1000
    pseudo_klocmax  = 0
    pseudo_kmax     = 0
    pseudo_lmax     = 0
  end pseudo
  qmcpack
    ci_threshold    = 1e-08
  end qmcpack
  rsdft_ecmd
    ecmd_functional = short_range_LDA
  end rsdft_ecmd
  run_control
  end run_control
  scf_utils
    frozen_orb_scf  = False
    level_shift     = 0.0
    max_dim_diis    = 15
    mo_guess_type   = Huckel
    n_it_scf_max    = 500
    scf_algorithm   = DIIS
    thresh_scf      = 1e-10
    threshold_diis  = 0.0
  end scf_utils
  two_body_dm
    ci_threshold    = 1e-05
    mat_mul_svd_vectors = True
    ontop_approx    = False
    thr_ontop_approx = 0.001
  end two_body_dm
```

All variables directly assignable by name in Nexus via generate_quantum_package

Quantum Package Running Modes

- Several sub-programs are run by Quantum Package to perform tasks such as Hartree-Fock, CIS, and CIPSI
- Modes that are touched on in Nexus demos here are bolded below

```
viribus>qp_run -h
qp_run - Quantum Package command

Usage:

  qp_run [-h] [-p <string>] [-s] [--] PROGRAM EZFIO_DIR

Arguments:

  PROGRAM                Name of the QP program to be run
  EZFIO_DIR              EZFIO directory

Options:

  -h  --help             Prints the help message.
  -p  --prefix=<string>  Prefix before running the program, like gdb
                         or valgrind.
  -s  --slave            Required to run slave tasks in distributed
                         environments.

Description:

  Executes a Quantum Package binary file among these:

  * cis
  * cisd
  * diagonalize_h
  * fci
  * fcidump
  * four_idx_transform
  * install
  * ks_scf
  * molden
```

```
  * print_ci_vectors
  * print_e_conv
  * print_ecmd_pbe_ontop
  * print_h0j
  * print_pgm
  * print_rsdft_variational_energy
  * print_wf
  * pt2
  * qmc_create_wf
  * qmc_e_curve
  * qp_ao_ijkl_r3_ints
  * qp_cipsi_rsh
  * reorder_dets
  * rs_ks_scf
  * save_for_qmcpack
  * save_natorb
  * save_one_e_dm
  * save_ortho_mos
  * scf
  * target_pt2_qmc
  * truncate_wf_spin
  * truncate_wf_spin_no_H
  * two_body_dm.main
  * uninstall
  * write_2_body_dm_fci_dump
  * write_effective_rsdft_hamiltonian
  * write_erf_and_regular_ints
  * write_integrals_erf
  * write_rsdft_h_read_ints
```

Setting QP Run Modes and Input Variables with Nexus

```
viribus>qp_run -h

  * cis
  * cisd
  * diagonalize_h
  * fci
  * fcidump
  * four_idx_transform
  * install
  * ks_scf
  * molden
  * print_ci_vectors
  * print_e_conv
  * print_ecmd_pbe_ontop
  * print_h0j
  * print_pgm
  * print_rsdft_variational_energy
  * print_wf
  * pt2
  * qmc_create_wf
  * qmc_e_curve
  * qp_ao_ijkl_r3_ints
  * qp_cipsi_rsh
  * reorder_dets
  * rs_ks_scf
  * save_for_qmcpack
  * save_natorb
  * save_one_e_dm
  * save_ortho_mos
  * scf
  * target_pt2_qmc
  * truncate_wf_spin
  * truncate_wf_spin_no_H
  * two_body_dm.main
  * uninstall
  * write_2_body_dm_fci_dump
  * write_effective_rsdft_hamiltonian
  * write_erf_and_regular_ints
  * write_integrals_erf
  * write_rsdft_h_read_ints
```

```python
#! /usr/bin/env python

...
from nexus import generate_quantum_package


scf = generate_quantum_package(
    # typical Nexus inputs, path/job/system, etc
    ...
    run_type     = 'scf',       # QP run mode
    ao_basis     = 'cc-pvtz',   # QP input variables
    )
```

Setting QP Run Modes and Input Variables with Nexus

```
viribus>qp_run -h

  * cis
  * cisd
  * diagonalize_h
  * fci
  * fcidump
  * four_idx_transform
  * install
  * ks_scf
  * molden
  * print_ci_vectors
  * print_e_conv
  * print_ecmd_pbe_ontop
  * print_h0j
  * print_pgm
  * print_rsdft_variational_energy
  * print_wf
  * pt2
  * qmc_create_wf
  * qmc_e_curve
  * qp_ao_ijkl_r3_ints
  * qp_cipsi_rsh
  * reorder_dets
  * rs_ks_scf
  * save_for_qmcpack
  * save_natorb
  * save_one_e_dm
  * save_ortho_mos
  * scf
  * target_pt2_qmc
  * truncate_wf_spin
  * truncate_wf_spin_no_H
  * two_body_dm.main
  * uninstall
  * write_2_body_dm_fci_dump
  * write_effective_rsdft_hamiltonian
  * write_erf_and_regular_ints
  * write_integrals_erf
  * write_rsdft_h_read_ints
```

```
scf = generate_quantum_package(
    run_type              = 'scf',
    ao_basis              = 'cc-pvtz',
    io_ao_two_e_integrals = 'Write’,
    )

fit = generate_quantum_package(
    run_type              = 'four_idx_transform',
    io_mo_two_e_integrals = 'Write',
    dependencies          = (scf,'other’) # generic dependency type
    )                                     # only wait for completion

cis = generate_quantum_package(
    run_type              = 'cis',
    cis_loop              = True,  # perform cis multiple times
    frozen_core           = True,  # don’t excite from core
    io_mo_two_e_integrals = 'Write’,
    )

fci0 = generate_quantum_package(
    run_type              = 'fci',
    n_det_max             = 5000,  # max determinant count
    dependencies          = (cis,'other’),
    )

sno = generate_quantum_package(
    run_type              = 'save_natorb',
    dependencies          = (fci0,'other’),
    )

fit0 = generate_quantum_package(
    run_type              = 'four_idx_transform',
    io_mo_two_e_integrals = 'Write',
    dependencies          = (sno,'other’),
    )

fci = generate_quantum_package(
    run_type      = 'fci',
    n_det_max     = 5000,
    dependencies  = (fit0,'other’),
    )
```

Workflow Demo:
Hartree Fock with Quantum Package

Quantum Package Example: H_2O Hartree-Fock

01_qp_h2o_hf/h2o_ae_hf.py

```
#! /usr/bin/env python

from nexus import settings,job,run_project
from nexus import generate_physical_system
from nexus import generate_quantum_package

settings(
    results       = '',
    status_only   = 0,
    generate_only = 0,
    sleep         = 3,
    machine       = 'ws12',
    qprc          = \
'/home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc',
    )

scf_job = job(cores=12,threads=12)

system = generate_physical_system(
    structure = 'H2O.xyz',
    )

scf = generate_quantum_package(
    identifier   = 'hf',        # log output goes to hf.out
    path         = 'h2o_ae_hf', # directory to run in
    job          = scf_job,
    system       = system,
    prefix       = 'h2o',       # create/use h2o.ezfio
    run_type     = 'scf',       # qprun scf h2o.ezfio
    ao_basis     = 'cc-pvtz',   # use cc-pvtz basis
    )

run_project()
```

```
viribus>source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc
viribus>./h2o_ae_hf.py
Applying user settings

Project starting
  checking for file collisions
  loading cascade images
    cascade 0 checking in
  checking cascade dependencies
    all simulation dependencies satisfied

  starting runs:
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  elapsed time 0.0 s  memory 101.68 MB
    Entering ./runs/h2o_ae_hf 0
      writing input files  0 hf   # qp_create_ezfio and qp_edit –c done here
    Entering ./runs/h2o_ae_hf 0
      sending required files  0 hf
      submitting job  0 hf
    Entering ./runs/h2o_ae_hf 0
      Executing:
        export OMP_NUM_THREADS=12

        source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc

        mpirun -np 1 qp_run scf h2o.ezfio

  elapsed time 3.2 s  memory 121.88 MB
  elapsed time 6.2 s  memory 101.73 MB
    Entering ./runs/h2o_ae_hf 0
      copying results  0 hf
    Entering ./runs/h2o_ae_hf 0
      analyzing  0 hf

Project finished

viribus>grep SCF runs/h2o_ae_hf/hf.out
* SCF energy                                               -76.03027837147553
```

Workflow Demo: Selected CI with Quantum Package

Quantum Package Example: O_2 Selected CI

02_qp_o2_selci/o2_selci.py

```
#! /usr/bin/env python

from nexus import settings,job,run_project
from nexus import generate_physical_system
from nexus import generate_quantum_package

settings(
    results       = '',
    status_only   = 0,
    generate_only = 0,
    sleep         = 3,
    machine       = 'ws12',
    qprc          = \
'/home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc',
    )

# define run details
qp_job = job(cores=12,threads=12)

# read in structure for oxygen dimer
dimer = generate_physical_system(
    structure = './O2.xyz',
    net_spin  = 2,
    )

# path, job, system details are shared across runs
qp_shared = dict(
    path   = 'O_dimer_selected_CI',
    job    = qp_job,
    system = dimer,
    prefix = 'fci', # single shared ezfio, rsync if different
    )

# run Hartree-Fock
scf = generate_quantum_package(
    identifier            = 'scf',
    run_type              = 'scf',
    ao_basis              = 'aug-cc-pvdz',
    io_ao_two_e_integrals = 'Write', # write 2e integrals
    four_idx_transform    = True,    # compute 2e integrals
    **qp_shared
    )
```

```
# initial selected CI run
fci0 = generate_quantum_package(
    identifier         = 'fci0',
    run_type           = 'fci',
    n_det_max          = 5000, # max determinant count
    save_natorb        = True, # write natural orbitals
    four_idx_transform = True, # compute 2e integrals
    dependencies       = (scf,'other'),
    **qp_shared
    )

# final selected CI based on natural orbitals
fci = generate_quantum_package(
    identifier    = 'fci',
    run_type      = 'fci',
    n_det_max     = 5000,
    dependencies  = (fci0,'other'),
    **qp_shared
    )

run_project()
```

Quantum Package Example: O_2 Selected CI

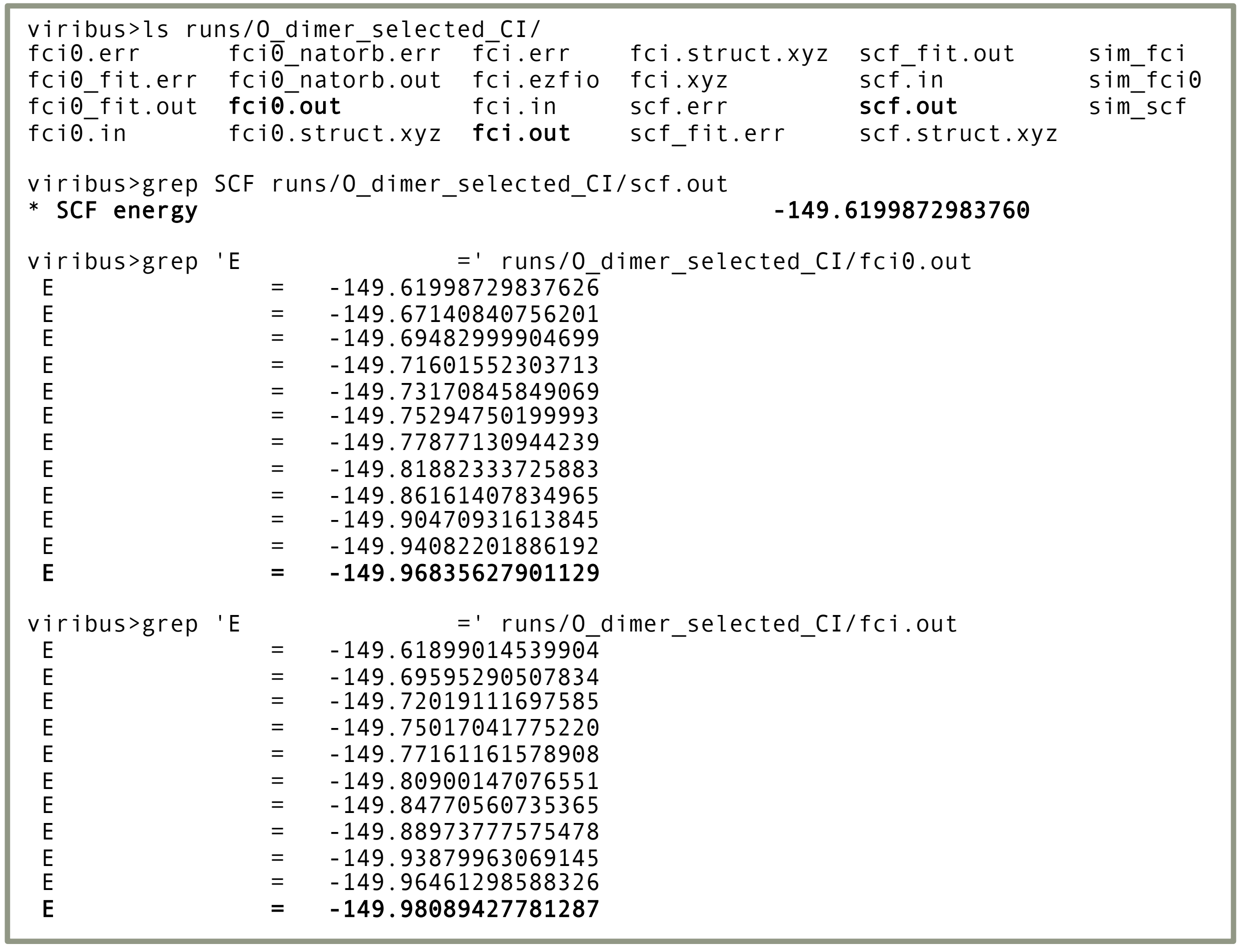
02_qp_o2_selci/o2_selci.py

```
viribus>source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc
viribus>./o2_selci.py

Applying user settings

Project starting
  checking for file collisions
  loading cascade images
    cascade 0 checking in
  checking cascade dependencies
    all simulation dependencies satisfied

  starting runs:
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  elapsed time 0.0 s  memory 101.95 MB
    Entering ./runs/O_dimer_selected_CI 0
      writing input files  0 scf
    Entering ./runs/O_dimer_selected_CI 0
      sending required files  0 scf
      submitting job  0 scf
    Entering ./runs/O_dimer_selected_CI 0
      Executing:
        export OMP_NUM_THREADS=12

        source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc

        mpirun -np 1 qp_run scf fci.ezfio >scf.out 2>scf.err

        echo "Write" > fci.ezfio/mo_two_e_ints/io_mo_two_e_integrals
        qp_run four_idx_transform fci.ezfio >scf_fit.out 2>scf_fit.err

  elapsed time 3.2 s  memory 122.12 MB
  elapsed time 9.2 s  memory 101.97 MB
    Entering ./runs/O_dimer_selected_CI 0
      copying results  0 scf
    Entering ./runs/O_dimer_selected_CI 0
      analyzing  0 scf

  elapsed time 12.2 s  memory 101.97 MB
    Entering ./runs/O_dimer_selected_CI 1
      writing input files  1 fci0
    Entering ./runs/O_dimer_selected_CI 1
      sending required files  1 fci0
      submitting job  1 fci0
```

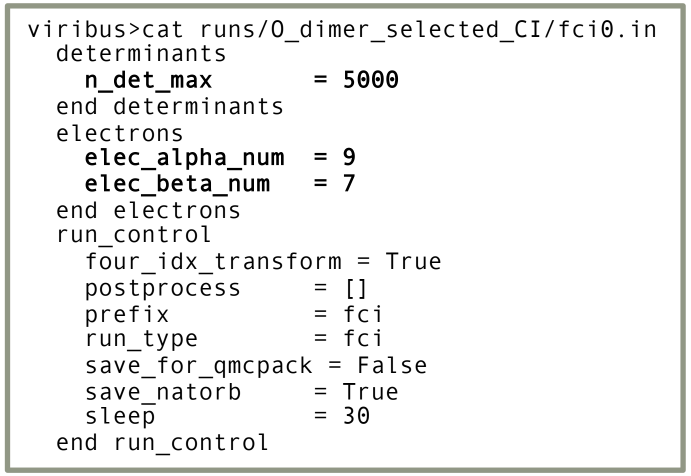
```
Entering ./runs/O_dimer_selected_CI 1
      Executing:
        export OMP_NUM_THREADS=12

        source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc

        mpirun -np 1 qp_run fci fci.ezfio >fci0.out 2>fci0.err

        qp_run save_natorb fci.ezfio >fci0_natorb.out 2>fci0_natorb.err

        echo "Write" > fci.ezfio/mo_two_e_ints/io_mo_two_e_integrals
        qp_run four_idx_transform fci.ezfio >fci0_fit.out 2>fci0_fit.err

  elapsed time 15.3 s  memory 344.29 MB
  ...
  elapsed time 30.4 s  memory 101.97 MB
    Entering ./runs/O_dimer_selected_CI 1
      copying results  1 fci0
    Entering ./runs/O_dimer_selected_CI 1
      analyzing  1 fci0

  elapsed time 33.5 s  memory 101.97 MB
    Entering ./runs/O_dimer_selected_CI 2
      writing input files  2 fci
    Entering ./runs/O_dimer_selected_CI 2
      sending required files  2 fci
      submitting job  2 fci
    Entering ./runs/O_dimer_selected_CI 2
      Executing:
        export OMP_NUM_THREADS=12

        source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc

        mpirun -np 1 qp_run fci fci.ezfio

  elapsed time 36.5 s  memory 358.10 MB
  ...
  elapsed time 45.6 s  memory 101.97 MB
    Entering ./runs/O_dimer_selected_CI 2
      copying results  2 fci
    Entering ./runs/O_dimer_selected_CI 2
      analyzing  2 fci

Project finished
```

Quantum Package Example: O_2 Selected CI

`02_qp_o2_selci/o2_selci.py`



```
viribus>ls runs/O_dimer_selected_CI/
fci0.err       fci0_natorb.err   fci.err    fci.struct.xyz  scf_fit.out     sim_fci
fci0_fit.err   fci0_natorb.out   fci.ezfio  fci.xyz         scf.in          sim_fci0
fci0_fit.out   fci0.out          fci.in     scf.err         scf.out         sim_scf
fci0.in        fci0.struct.xyz   fci.out    scf_fit.err     scf.struct.xyz

viribus>grep SCF runs/O_dimer_selected_CI/scf.out
* SCF energy                                       -149.6199872983760

viribus>grep 'E               =' runs/O_dimer_selected_CI/fci0.out
 E               =   -149.61998729837626
 E               =   -149.67140840756201
 E               =   -149.69482999904699
 E               =   -149.71601552303713
 E               =   -149.73170845849069
 E               =   -149.75294750199993
 E               =   -149.77877130944239
 E               =   -149.81882333725883
 E               =   -149.86161407834965
 E               =   -149.90470931613845
 E               =   -149.94082201886192
 E               =   -149.96835627901129

viribus>grep 'E               =' runs/O_dimer_selected_CI/fci.out
 E               =   -149.61899014539904
 E               =   -149.69595290507834
 E               =   -149.72019111697585
 E               =   -149.75017041775220
 E               =   -149.77161161578908
 E               =   -149.80900147076551
 E               =   -149.84770560735365
 E               =   -149.88973777575478
 E               =   -149.93879963069145
 E               =   -149.96461298588326
 E               =   -149.98089427781287
```



```
viribus>cat runs/O_dimer_selected_CI/fci0.in
  determinants
    n_det_max        = 5000
  end determinants
  electrons
    elec_alpha_num   = 9
    elec_beta_num    = 7
  end electrons
  run_control
    four_idx_transform = True
    postprocess      = []
    prefix           = fci
    run_type         = fci
    save_for_qmcpack = False
    save_natorb      = True
    sleep            = 30
  end run_control
```

Workflow Demo:
Hartree Fock with Quantum Package
+ VMC/DMC with QMCPACK

QP + QMCPACK Example: H_2O Hartree-Fock to DMC

03_qp_h2o_hf_qmcpack/h2o_ae_hf_qmc.py

```
#! /usr/bin/env python

from nexus import settings,job,run_project
from nexus import generate_physical_system
from nexus import generate_quantum_package
from nexus import generate_convert4qmc
from nexus import generate_cusp_correction
from nexus import generate_qmcpack

settings(
    results       = '',
    sleep         = 3,
    machine       = 'ws12',
    qprc          = \
'/home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc',
    )

scf_job = job(cores=12,threads=12)
c4q_job = job(cores=1)
qmc_job = job(cores=12,threads=12)

system = generate_physical_system(
    structure = 'H2O.xyz',
    )

# perform Hartree-Fock
scf = generate_quantum_package(
    identifier       = 'hf',      # log output goes to hf.out
    path             = 'H2O/hf',  # directory to run in
    job              = scf_job,
    system           = system,
    prefix           = 'h2o',     # create/use h2o.ezfio
    run_type         = 'scf',     # qprun scf h2o.ezfio
    ao_basis         = 'cc-pvtz', # use cc-pvtz basis
    save_for_qmcpack = True,      # write h5 file for qmcpack
    )

# convert orbitals to QMCPACK format
c4q = generate_convert4qmc(
    identifier   = 'c4q',
    path         = 'H2O/hf',
    job          = c4q_job,
    hdf5         = True,          # use hdf5 format
    dependencies = (scf,'orbitals'),
    )
```

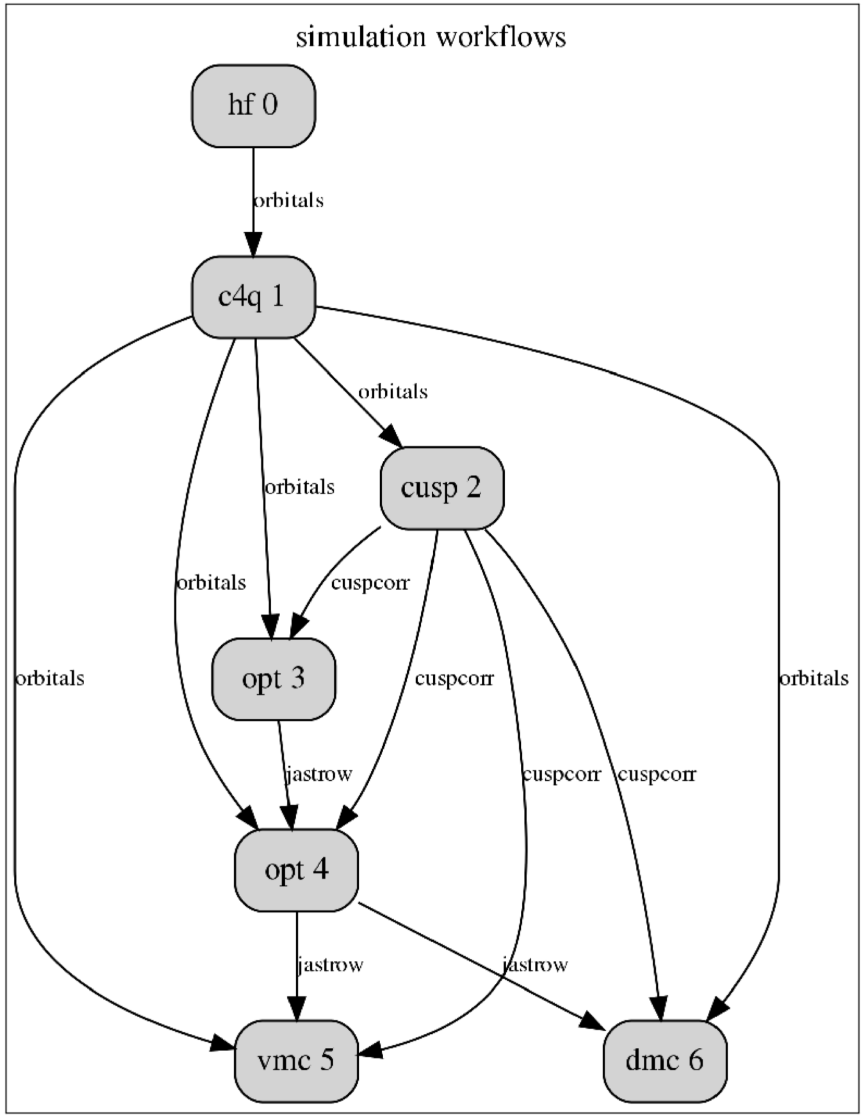
```
# calculate cusp correction
cc = generate_cusp_correction(
    identifier   = 'cusp',
    path         = 'H2O/cuspcorr',
    job          = qmc_job,
    system       = system,
    dependencies = (c4q,'orbitals'),
    )

# optimize 2-body Jastrow
optJ2 = generate_qmcpack(
    identifier      = 'opt',
    path            = 'H2O/optJ2',
    job             = qmc_job,
    system          = system,
    J2              = True, # jastrow defaults
    J2_rcut         = 8.0,  # shorter cutoff
    qmc             = 'opt', # use opt defaults
    minmethod       = 'oneshiftonly',
    init_cycles     = 3,
    init_minwalkers = 0.1,
    cycles          = 3,
    samples         = 25600,
    dependencies    = [(c4q,'orbitals'),
                       (cc,'cuspcorr')],
    )

# optimize 3-body Jastrow
optJ3 = generate_qmcpack(
    identifier      = 'opt',
    path            = 'H2O/optJ3',
    job             = qmc_job,
    system          = system,
    J3              = True, # jastrow defaults
    qmc             = 'opt',
    minmethod       = 'oneshiftonly',
    init_cycles     = 3,
    init_minwalkers = 0.1,
    cycles          = 3,
    samples         = 51200,
    dependencies    = [(c4q,'orbitals'),
                       (cc,'cuspcorr'),
                       (optJ2,'jastrow')],
    )
```

```
# run VMC with QMCPACK
qmc = generate_qmcpack(
    identifier   = 'vmc',
    path         = 'H2O/vmc',
    job          = qmc_job,
    system       = system,
    jastrows     = [],
    qmc          = 'vmc', # use vmc defaults
    dependencies = [(c4q,'orbitals'),
                    (cc,'cuspcorr'),
                    (optJ3,'jastrow')],
    )

# run DMC with QMCPACK
qmc = generate_qmcpack(
    identifier   = 'dmc',
    path         = 'H2O/dmc',
    job          = qmc_job,
    system       = system,
    jastrows     = [],
    qmc          = 'dmc', # use dmc defaults
    eq_dmc       = True,  # add equil run
    dependencies = [(c4q,'orbitals'),
                    (cc,'cuspcorr'),
                    (optJ3,'jastrow')],
    )
```

QP + QMCPACK Example: H_2O Hartree-Fock to DMC

03_qp_h2o_hf_qmcpack/h2o_ae_hf_qmc.py

```
viribus>./h2o_ae_hf_qmc.py --graph_sims
```

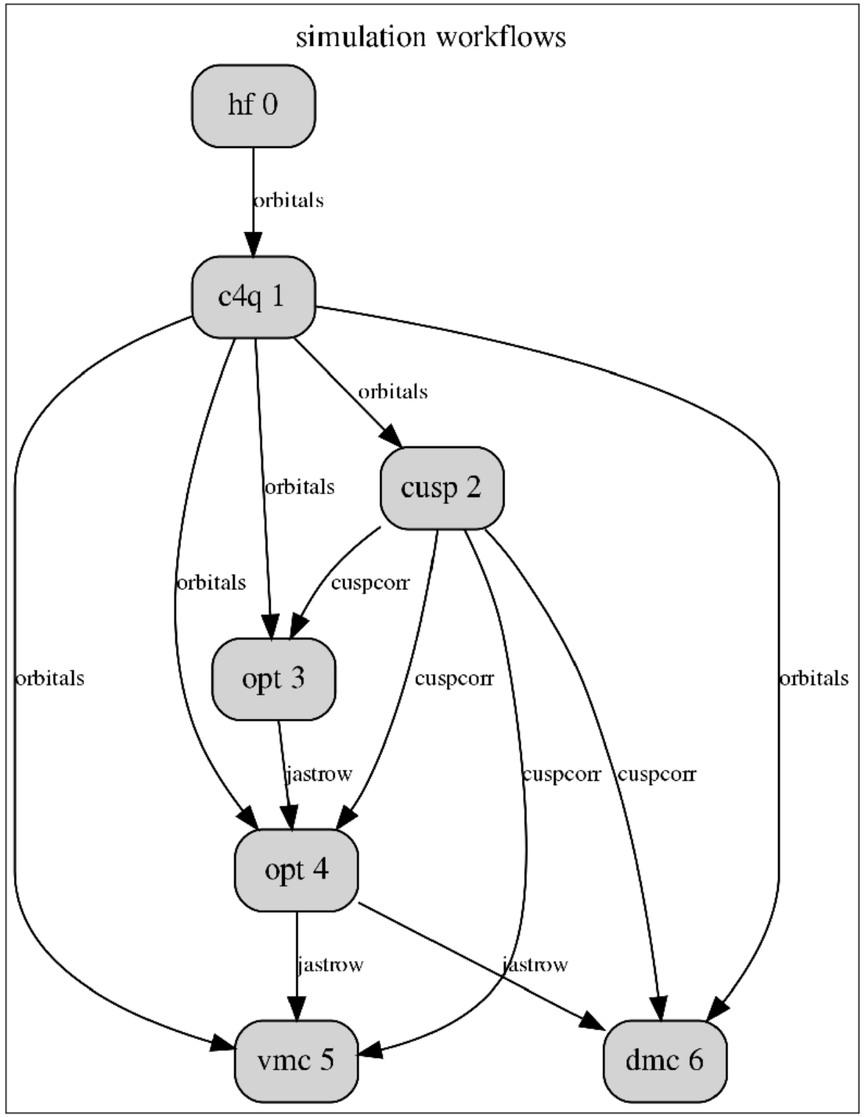


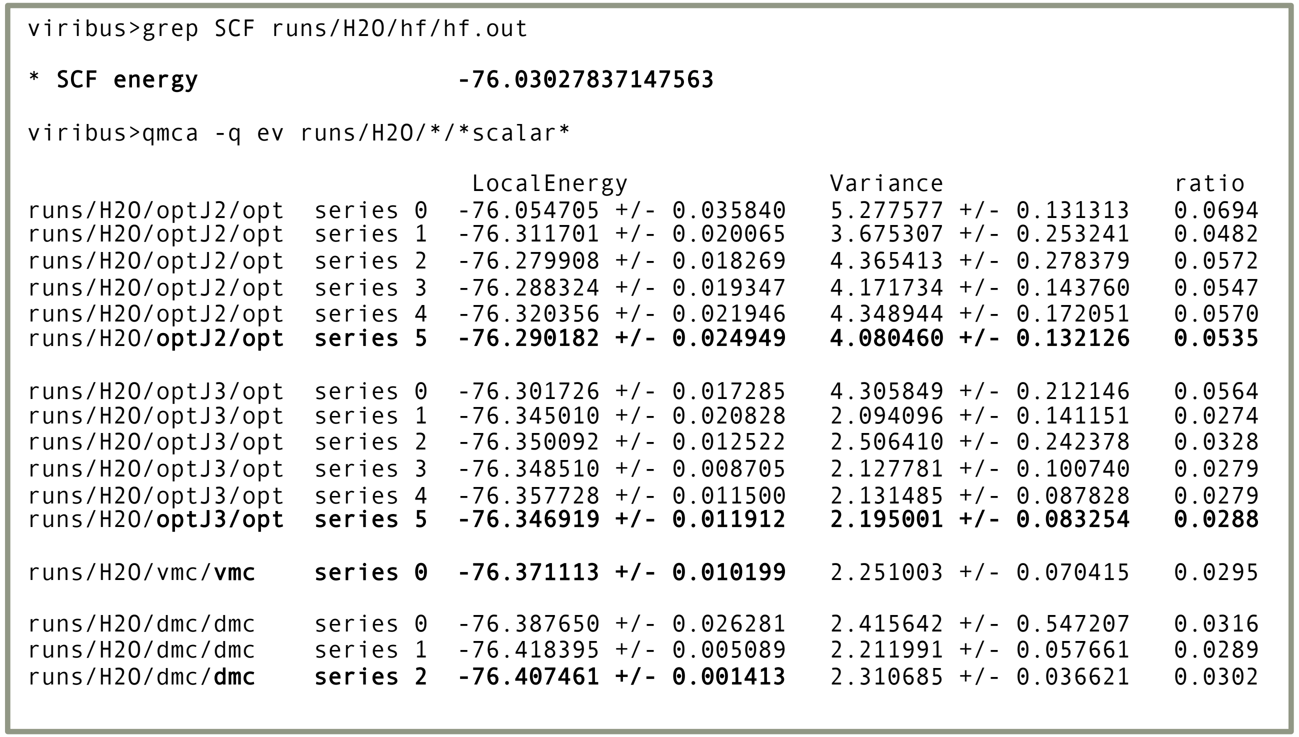
```
viribus>source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc
viribus>./h2o_ae_hf_qmc.py

Project starting
  ...
    Entering ./runs/H2O/hf 0
      Executing:
        export OMP_NUM_THREADS=12

        source /home/j1k/apps/quantum_package/qp2-2.0.0-beta/quantum_package.rc

        mpirun -np 1 qp_run scf h2o.ezfio >hf.out 2>hf.err

        qp_run save_for_qmcpack h2o.ezfio >hf_savewf.out 2>hf_savewf.err
    ...
    Entering ./runs/H2O/hf 1
      Executing:
        export OMP_NUM_THREADS=1
        mpirun -np 1 convert4qmc -QP hf_savewf.out -prefix c4q -hdf5
    ...
    Entering ./runs/H2O/cuspcorr 2
      Executing:
        export OMP_NUM_THREADS=12
        mpirun -np 1 qmcpack cusp.in.xml # add cuspCorrection="yes"
    ...                                  # produces up/down *.cuspInfo.xml
    Entering ./runs/H2O/optJ2 3
      Executing:
        export OMP_NUM_THREADS=12
        mpirun -np 1 qmcpack opt.in.xml
    ...
    Entering ./runs/H2O/optJ3 4
      Executing:
        export OMP_NUM_THREADS=12
        mpirun -np 1 qmcpack opt.in.xml
    ...
    Entering ./runs/H2O/vmc 5
      Executing:
        export OMP_NUM_THREADS=12
        mpirun -np 1 qmcpack vmc.in.xml
    ...
    Entering ./runs/H2O/dmc 6
      Executing:
        export OMP_NUM_THREADS=12
        mpirun -np 1 qmcpack dmc.in.xml
```

QP + QMCPACK Example: H_2O Hartree-Fock to DMC

`03_qp_h2o_hf_qmcpack/h2o_ae_hf_qmc.py`

```
viribus>./h2o_ae_hf_qmc.py --graph_sims
```





```
viribus>grep SCF runs/H2O/hf/hf.out

* SCF energy                   -76.03027837147563

viribus>qmca -q ev runs/H2O/*/*scalar*

                                LocalEnergy               Variance                 ratio
runs/H2O/optJ2/opt  series 0  -76.054705 +/- 0.035840   5.277577 +/- 0.131313   0.0694
runs/H2O/optJ2/opt  series 1  -76.311701 +/- 0.020065   3.675307 +/- 0.253241   0.0482
runs/H2O/optJ2/opt  series 2  -76.279908 +/- 0.018269   4.365413 +/- 0.278379   0.0572
runs/H2O/optJ2/opt  series 3  -76.288324 +/- 0.019347   4.171734 +/- 0.143760   0.0547
runs/H2O/optJ2/opt  series 4  -76.320356 +/- 0.021946   4.348944 +/- 0.172051   0.0570
runs/H2O/optJ2/opt  series 5  -76.290182 +/- 0.024949   4.080460 +/- 0.132126   0.0535

runs/H2O/optJ3/opt  series 0  -76.301726 +/- 0.017285   4.305849 +/- 0.212146   0.0564
runs/H2O/optJ3/opt  series 1  -76.345010 +/- 0.020828   2.094096 +/- 0.141151   0.0274
runs/H2O/optJ3/opt  series 2  -76.350092 +/- 0.012522   2.506410 +/- 0.242378   0.0328
runs/H2O/optJ3/opt  series 3  -76.348510 +/- 0.008705   2.127781 +/- 0.100740   0.0279
runs/H2O/optJ3/opt  series 4  -76.357728 +/- 0.011500   2.131485 +/- 0.087828   0.0279
runs/H2O/optJ3/opt  series 5  -76.346919 +/- 0.011912   2.195001 +/- 0.083254   0.0288

runs/H2O/vmc/vmc    series 0  -76.371113 +/- 0.010199   2.251003 +/- 0.070415   0.0295

runs/H2O/dmc/dmc    series 0  -76.387650 +/- 0.026281   2.415642 +/- 0.547207   0.0316
runs/H2O/dmc/dmc    series 1  -76.418395 +/- 0.005089   2.211991 +/- 0.057661   0.0289
runs/H2O/dmc/dmc    series 2  -76.407461 +/- 0.001413   2.310685 +/- 0.036621   0.0302
```

Workflow Demo:
Selected CI with Quantum Package
+ VMC/DMC with QMCPACK

QP + QMCPACK Example: O_2 Selected CI to DMC

`04_qp_o2_selci_qmcpack/o2_selci_vmc_dmc.py`

```
#! /usr/bin/env python

from nexus import settings,job,run_project
from nexus import generate_physical_system
from nexus import generate_quantum_package
from nexus import generate_convert4qmc
from nexus import generate_cusp_correction
from nexus import generate_qmcpack

settings(
    results       = '',
    status_only   = 0,
    generate_only = 0,
    sleep         = 3,
    machine       = 'ws12',
    qprc          = ...
    )

# define run details
qp_job  = job(cores=12,threads=12)
c4q_job = job(cores=1)
qmc_job = job(cores=12,threads=12)

# read in structure for oxygen dimer
dimer = generate_physical_system(
    structure = './O2.xyz',
    net_spin  = 2,
    )

# path,job,system shared across runs
qp_shared = dict(
    path   = 'O_dimer/selci',
    job    = qp_job,
    system = dimer,
    prefix = 'fci', # single shared ezfio
    )

# run Hartree-Fock
scf = generate_quantum_package(
    identifier            = 'scf',
    run_type              = 'scf',
    ao_basis              = 'aug-cc-pvdz',
    io_ao_two_e_integrals = 'Write',
    four_idx_transform    = True,
    **qp_shared
    )
```

```
# initial selected CI run
fci0 = generate_quantum_package(
    identifier         = 'fci0',
    run_type           = 'fci',
    n_det_max          = 5000,
    save_natorb        = True,
    four_idx_transform = True,
    dependencies       = (scf,'other'),
    **qp_shared
    )

# selected CI based on natural orbitals
fci = generate_quantum_package(
    identifier       = 'fci',
    run_type         = 'fci',
    n_det_max        = 5000,
    save_for_qmcpack = True,
    dependencies     = (fci0,'other'),
    **qp_shared
    )

# convert orbitals/multidet
c4q = generate_convert4qmc(
    identifier   = 'c4q',
    path         = 'O_dimer/selci',
    job          = c4q_job,
    hdf5         = True,
    dependencies = (fci,'orbitals'),
    )

# calculate cusp correction
cc = generate_cusp_correction(
    identifier   = 'cusp',
    path         = 'O_dimer/cuspcorr',
    job          = qmc_job,
    system       = dimer,
    dependencies = (c4q,'orbitals'),
    )
```

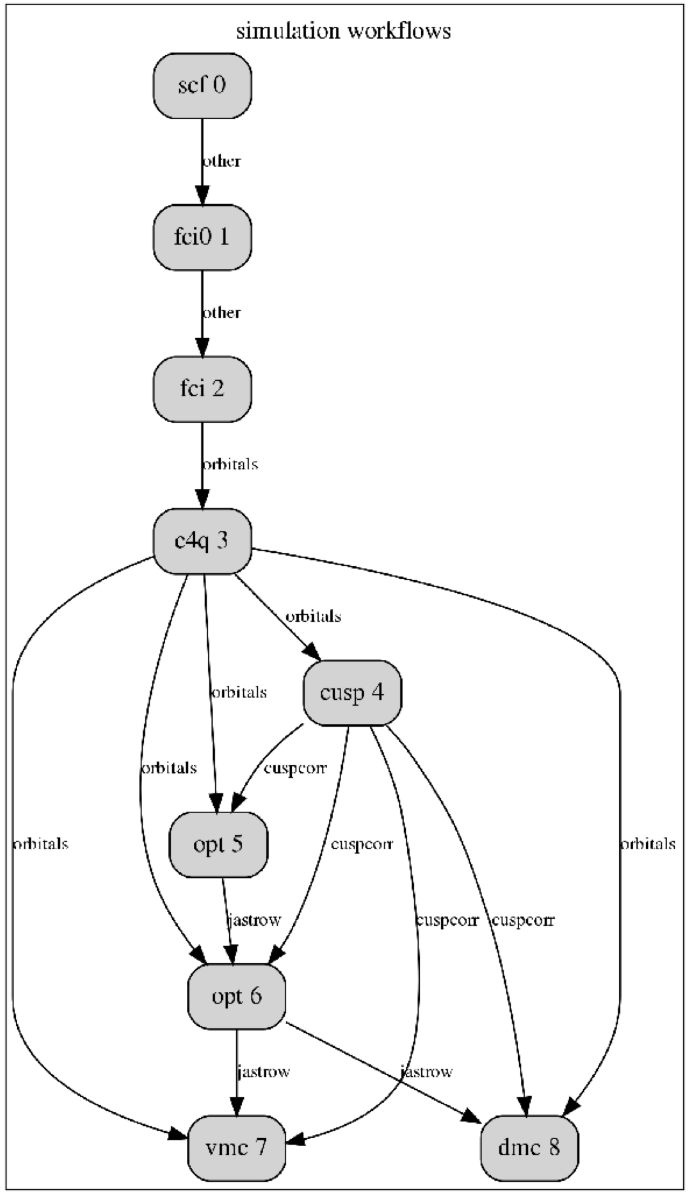
```
# optimize 2-body Jastrow
optJ2 = generate_qmcpack(
    identifier      = 'opt',
    path            = 'O_dimer/optJ2',
    job             = qmc_job,
    system          = dimer,
    J2              = True,
    qmc             = 'opt',
    minmethod       = 'oneshiftonly',
    init_cycles     = 3,
    init_minwalkers = 0.1,
    cycles          = 3,
    samples         = 25600,
    dependencies    = [(c4q,'orbitals'),
                       (cc,'cuspcorr')],
    )

# optimize 3-body Jastrow
optJ3 = generate_qmcpack(
    identifier      = 'opt',
    path            = 'O_dimer/optJ3',
    job             = qmc_job,
    system          = dimer,
    J3              = True,
    qmc             = 'opt',
    minmethod       = 'oneshiftonly',
    init_cycles     = 3,
    init_minwalkers = 0.1,
    cycles          = 3,
    samples         = 51200,
    dependencies    = [(c4q,'orbitals'),
                       (cc,'cuspcorr'),
                       (optJ2,'jastrow')],
    )
```

```
# run VMC with QMCPACK
qmc = generate_qmcpack(
    identifier   = 'vmc',
    path         = 'O_dimer/vmc',
    job          = qmc_job,
    system       = dimer,
    jastrows     = [],
    qmc          = 'vmc',
    dependencies = [(c4q,'orbitals'),
                    (cc,'cuspcorr'),
                    (optJ3,'jastrow')],
    )

# run DMC with QMCPACK
qmc = generate_qmcpack(
    identifier   = 'dmc',
    path         = 'O_dimer/dmc',
    job          = qmc_job,
    system       = dimer,
    jastrows     = [],
    qmc          = 'dmc',
    vmc_samples  = 1024,
    eq_dmc       = True,
    dependencies = [(c4q,'orbitals'),
                    (cc,'cuspcorr'),
                    (optJ3,'jastrow')],
    )
```

QP + QMCPACK Example: O_2 Selected CI to DMC

04_qp_o2_selci_qmcpack/o2_selci_vmc_dmc.py

```
viribus>./o2_selci_vmc_dmc.py --graph_sims
```



```
viribus>grep SCF runs/O_dimer/selci/scf.out
* SCF energy         -149.6199872983761

viribus>grep 'E              =' runs/O_dimer/selci/fci0.out
 E               =   -149.96839697045442

viribus>grep 'E              =' runs/O_dimer/selci/fci.out
 E               =   -149.99187556469764

viribus>qmca -q ev runs/O_dimer/*/*scalar*

                                   LocalEnergy                Variance                ratio
runs/O_dimer/optJ2/opt  series 0  -150.018615 +/- 0.040825  14.594573 +/- 2.613162   0.0973
runs/O_dimer/optJ2/opt  series 1  -150.280100 +/- 0.039179   9.031973 +/- 0.485841   0.0601
runs/O_dimer/optJ2/opt  series 2  -150.211582 +/- 0.046759   9.256717 +/- 0.621347   0.0616
runs/O_dimer/optJ2/opt  series 3  -150.210220 +/- 0.034927   8.957077 +/- 0.476612   0.0596
runs/O_dimer/optJ2/opt  series 4  -150.181311 +/- 0.036444  10.004254 +/- 0.614452   0.0666
runs/O_dimer/optJ2/opt  series 5  -150.171909 +/- 0.034095   8.589145 +/- 0.323540   0.0572

runs/O_dimer/optJ3/opt  series 0  -150.203770 +/- 0.031103   9.899471 +/- 0.678129   0.0659
runs/O_dimer/optJ3/opt  series 1  -150.276269 +/- 0.026324   7.612160 +/- 0.495191   0.0507
runs/O_dimer/optJ3/opt  series 2  -150.239839 +/- 0.024805   7.290338 +/- 0.340443   0.0485
runs/O_dimer/optJ3/opt  series 3  -150.233542 +/- 0.022657   7.343151 +/- 0.428958   0.0489
runs/O_dimer/optJ3/opt  series 4  -150.253519 +/- 0.018772   7.380013 +/- 0.366188   0.0491
runs/O_dimer/optJ3/opt  series 5  -150.268180 +/- 0.019884   7.065689 +/- 0.266124   0.0470

runs/O_dimer/vmc/vmc    series 0  -150.211488 +/- 0.023634   6.881802 +/- 0.195692   0.0458

runs/O_dimer/dmc/dmc    series 0  -150.315266 +/- 0.115869   6.840526 +/- 0.966921   0.0455
runs/O_dimer/dmc/dmc    series 1  -150.410008 +/- 0.015410   9.247801 +/- 0.930152   0.0615
runs/O_dimer/dmc/dmc    series 2  -150.332674 +/- 0.003840   7.730539 +/- 0.119554   0.0514
```

Question & Answer + Updates

Question & Answer + Updates

- Q&A for Nexus features, workflow how to’s, code details, etc

- Updates on current development

Next Meeting

- Date/Time: Friday June 21st 1-2pm EST
- BlueJeans link: https://bluejeans.com/190169126
- Tentative topic: PySCF
- Feature demo and/or topic requests welcome!