A Tutorial for TheoDORE 3.0

Felix Plasser, Patrick Kimber

Loughborough, 2020

Department of Chemistry – Loughborough University

Contents

1 Before Starting

1.1 Introduction

This tutorial is intended to provide an overview over the functionalities of the TheoDORE program package. Various tasks of different complexity are discussed using interfaces to different quantum chemistry packages. It is advisable to go through the whole tutorial but it is of course possible to skip some of the later sections.

All input files are contained in the `EXAMPLES` directory in the TheoDORE distribution. They are the same files accessed by the `theo_test.bash` program.

1.2 Notation

The following notation is used:

```
This kind of font indiates what is seen on the screen
and the command lines that you should write <ENTER>  ! Comments come here
```

> *Important information related to TheoDORE but not necessarily connected to the current job comes in boxes like this.*

1.3 Installation

In case of using `bash` it should suffice to type

```
> source /mypath/TheoDORE_3.0/setpaths.bash  ! replace /mypath with your actual installation
```

to set up the required `THEODIR`, `PATH` and `PYTHONPATH` environment variables.

For more information, see

`https://theodore-qc.sourceforge.io/docs/installation.html`

2 Overview

To get an overview over all tools implemented in TheoDORE, simpy run.

```
> theodore
```

```
================================================================================
|                                TheoDORE 3.0                                  |
|         Theoretical Density, Orbital Relaxation and Exciton analysis         |
|                                                                              |
|                           Author: Felix Plasser                              |
|            Contributions by: L. Stojanovic, G. Hermann, S. Mai,              |
|                         M.F.S.J. Menger, P. Kimber                           |
--------------------------------------------------------------------------------
|                       References for the modules used                        |
|        (see also http://theodore-qc.sourceforge.net/literature.html)         |
|                                                                              |
|   Program citation:                                                          |
|     F. Plasser, J. Chem. Phys. (2020), 152, 084108.                          |
================================================================================
  optional arguments:
  ---------------------------------------------------
    -h/--help             | show this help message and exit

  Actions: options
  ---------------------------------------------------
    theoinp               Input generation for TheoDORE
    analyze_tden          Transition density matrix analysis
    analyze_tden_unr      Transition density matrix analysis (UHF/UKS)
    analyze_tden_es2es    Transition density matrix ana. (state-to-state)
    analyze_tden_soc      1TDM analysis for spin-orbit coupled states
    analyze_sden          State density matrix analysis
    analyze_nos           Analysis of natural orbital (NO) files
    parse_libwfa          Parse libwfa output from Q-Chem or OpenMolcas
    plot_vist             Read NICS values and prepare VIST plot
    plot_omfrag           Plot Omega matrices as pseudocolor matrix plot
    plot_om_bars          Plot Omega matrices as bar graphs
    plot_frag_decomp      Plot fragment decomposition of Omega matrix
    plot_graph            Graph plotting for potential curves etc.
    plot_graph_nx         Graph plotting (Newton-X)
    jmol_mos              Orbital/density plotting in Jmol
    jmol_vibs             Plotting of vibrations in Jmol
    vmd_plots             Automatic plotting of cube files in VMD
    draw_moments          Plotting of dipole and quadrupole moments
    babel                 Openbabel wrapper - conversion of coordinate files
    cc_opt                Analysis of geom. opt. or relaxed scan
    cc_check              Check if a logfile can be parsed with cclib
    extract_molden        Extract hole/particle parts from Molden file
    spectrum              Convoluted spectrum from analyze_tden output
    tden_ov               Transition density matrix overlap
    convert_table         Convert the output to latex/html table
    dgrid_prep            Prepare input for DGrid
    fcd                   Fragment charge difference analysis
```

- The first part of the header shows the main information about the program and version.
- The second part provides literature references (these are adjusted specifically to the modules called).
- In the third part all possible scripts are shown. The main ones are `theoinp` for input generation; `analyze_tden` and `analyze_sden` to drive the analyses.

To obtain information about a specific script and its input arguments, run for example

```
> theodore analyze_nos
```

```
================================================================================
|                                 TheoDORE 3.0                                 |
|         Theoretical Density, Orbital Relaxation and Exciton analysis         |
|                                                                              |
|                            Author: Felix Plasser                             |
|             Contributions by: L. Stojanovic, G. Hermann, S. Mai,             |
|                          M.F.S.J. Menger, P. Kimber                          |
--------------------------------------------------------------------------------
|                       References for the modules used                        |
|        (see also http://theodore-qc.sourceforge.net/literature.html)         |
|                                                                              |
|   Program citation:                                                          |
|     F. Plasser, J. Chem. Phys. (2020), 152, 084108.                          |
================================================================================

  positional arguments:
  ---------------------------------------------------
    no_files             | List of NO files in Molden format

  optional arguments:
  ---------------------------------------------------
    -h/--help            | show this help message and exit
    -f/--ifile           | Input file (optional)
    -r/--ref             | Reference MO file for computing AO overlap matrix
    -o/--occ_fac         | Multiply occupations with this factor
    -u/--unrestricted    | Use if unrestricted orbitals are present; Default=
                         | False, if set value=True
    -e/--rd_ene          | Interpret energies as occupations; Default=False,
                         | if set value=True

usage: /PhotoChem/programs/TheoDORE/Versions/TheoDORE_3.0/bin/theodore ... analyze_nos [-h] [-f ifile] [-r ref] [-o occ_fac

Analysis of natural orbital (NO) files
```

- The last line shows what the script does: "Analysis of natural orbital (NO) files"
- Arguments and keywords are shown above
- A summary of all scripts is given here:
  `https://theodore-qc.sourceforge.io/docs/usage.html`

3 Natural transition orbitals (Turbomole)

As a first step, we will plot the natural transition orbitals (NTOs) in the case of the formaldehyde dimer computed with RI-CC2 in Turbomole.

3.1 Input generation

The input files are taken from the EXAMPLES directory in the TheoDORE distribution

```
 > cp -r $THEODIR/EXAMPLES/STANDARD/fa2.ricc2/QC_FILES/ fa2.tutorial
```

Inside the fa2.tutorial directory, run the input program

```
 > theodore theoinp
```

```
Type of job (rtype):
  [ 1]      qcadc - Q-Chem ADC (libwfa output)
  [ 2]     libwfa - General libwfa output
  [ 3]    qctddft - Q-Chem TDDFT
  [ 4]       fchk - Q-Chem fchk file
  [ 5]   colmcscf - Columbus MCSCF
  [ 6]    colmrci - Columbus MR-CI (tden analysis)
  [ 7]      rassi - Molcas RASSI
  [ 8]        nos - Read natural orbitals (Molden format) for sden analysis: Columbus, Molcas, ...
  [ 9]      ricc2 - Turbomole ricc2
  [10]       escf - Turbomole escf
  [11]   terachem - Terachem (TDDFT)
  [12]      cclib - Use external cclib library: Gaussian, GAMESS, ...
  [13]       orca - ORCA TDDFT (using a Molden file and cclib)
  [14]        adf - ADF (TDDFT)
  [15]     tddftb - DFTB+ - TDDFTB
  [16]    dftmrci - DFT/MRCI
  [17]     onetep - ONETEP
Choice: [9]  9 <ENTER> ! theoinp tries to guess the program used according to the files present

Main file to read (rfile):
Choice (autocomplete enabled): [ricc2.out] <ENTER>

Read binary CCRE0 files? (read_binary):
Choice (y/n): [n] <ENTER> ! CCRE0 files are not available in this example.
! Otherwise binary files are recommended when available.

 *** Warning: in the case of ricc2 you have to delete the line
       implicit core=   x virt=    x
     from the control file before running tm2molden.
 ! Everything works here but in general one should remember this when using ricc2

MO file (Molden format)
 -> This file should ideally contain a square invertible coefficient matrix (mo_file):
Choice (autocomplete enabled): [molden.input] <ENTER>

Analysis of transition density matrices?
Choice (y/n): [y] <ENTER>

Perform CT number analysis?
Choice (y/n): [y] n <ENTER> ! Do not perform a charge transfer number analysis to keep things simple
```

```
Perform natural transition orbital (NTO) analysis? (comp_ntos):
Choice (y/n): [y] <ENTER>

Perform analysis of domain NTOs and conditional densities? (comp_dntos):
Choice (y/n): [n] <ENTER> ! See section 7

NTOs as Jmol script? (jmol_orbitals):
Choice (y/n): [y] y <ENTER> ! Type "y" if you have the JMOL program available

NTOs in Molden format (molden_orbitals):
Choice (y/n): [n] y <ENTER> ! Type "y" if you want files in MOLDEN format

Use alpha/beta rather then negative/positive to code for hole/particle orbitals? (alphabeta):
Choice (y/n): [n] n <ENTER> ! Only for special applications

NTOs in Cube file format (requires orbkit) (cube_orbitals):
Choice (y/n): [n] <ENTER> ! This option appears only when orbkit is installed

Calculation of Particle/Hole density (requires orbkit)? (comp_p_h_dens):
Choice (y/n): [n] <ENTER> ! Only with orbkit

Calculation of transition densities between ground state and excited states (requires orbkit) (comp_rho0n):
Choice (y/n): [n] <ENTER> ! Only with orbkit

Perform exciton analysis?
Choice (y/n): [y] n <ENTER>

Adjust detailed output options?
Choice (y/n): [n] <ENTER>

Name of input file
Choice: [dens_ana.in] <ENTER>
Finished: File dens_ana.in written.
```

After going through these steps, the file `dens_ana.in` with the following content is written:

```
rtype='ricc2'
rfile='ricc2.out'
read_binary=False
mo_file='molden.input'
comp_ntos=True
comp_dntos=False
jmol_orbitals=True
molden_orbitals=True
alphabeta=False
cube_orbitals=False
comp_p_h_dens=False
comp_rho0n=False
prop_list=['PRNTO', 'Z_HE']
```

> *To learn more about the available keywords check:*
> *https://sourceforge.net/p/theodore-qc/wiki/Keywords/*

3.2 Transition density matrix (1TDM) analysis

To run the 1TDM analysis to produce the NTOs, simply type:

```
> theodore analyze_tden
```

After some technical information, you will find the following output summary

```
state       dE(eV)     f  PRNTO   Z_HE ! label of the state / exc. energy / osc. strength / NTO participation ratio
--------------------------------------   / effective no. entangled states
1(1)a       4.174  0.000  1.943  2.000
2(1)a       4.192  0.000  1.952  1.999
3(1)a       7.944  0.000  1.849  1.937
4(1)a       8.021  0.164  1.882  1.958
5(1)a       8.755  0.000  1.991  2.012
6(1)a       8.763  0.052  1.998  2.015
```

3.3 Plotting of the orbitals

If you selected to export files in MOLDEN format, one file for each individual state will be present

```
nto_1-1-a.mld  nto_2-1-a.mld  nto_3-1-a.mld  nto_4-1-a.mld  nto_5-1-a.mld  nto_6-1-a.mld
```

You can visualize them with any program of your choice.

In the case of using JMOL a shortcut is available. To plot all the orbitals in one go, simply type

```
> jmol -n nto_jmol.spt
```

Then JMOL will create `.png` files for all the orbitals. To look at all these orbitals at once open the file `nto.html` in a browser, as shown in Fig. 1.

> *In both cases two NTO pairs are required to describe the transition, in agreement with* $\mathrm{PR}_{\mathrm{NTO}} \approx 2$. *This is a general observation for excitonic states in interacting chromophores, see Ref. [1] for further discussion and references.*

Technical note: To get the precise pictures shown here, modify the `nto_jmol.spt` script

```
load molden.input FILTER "nosort"
mo titleformat ""
rotate x 90 ! add
rotate y 10 ! add
...
```

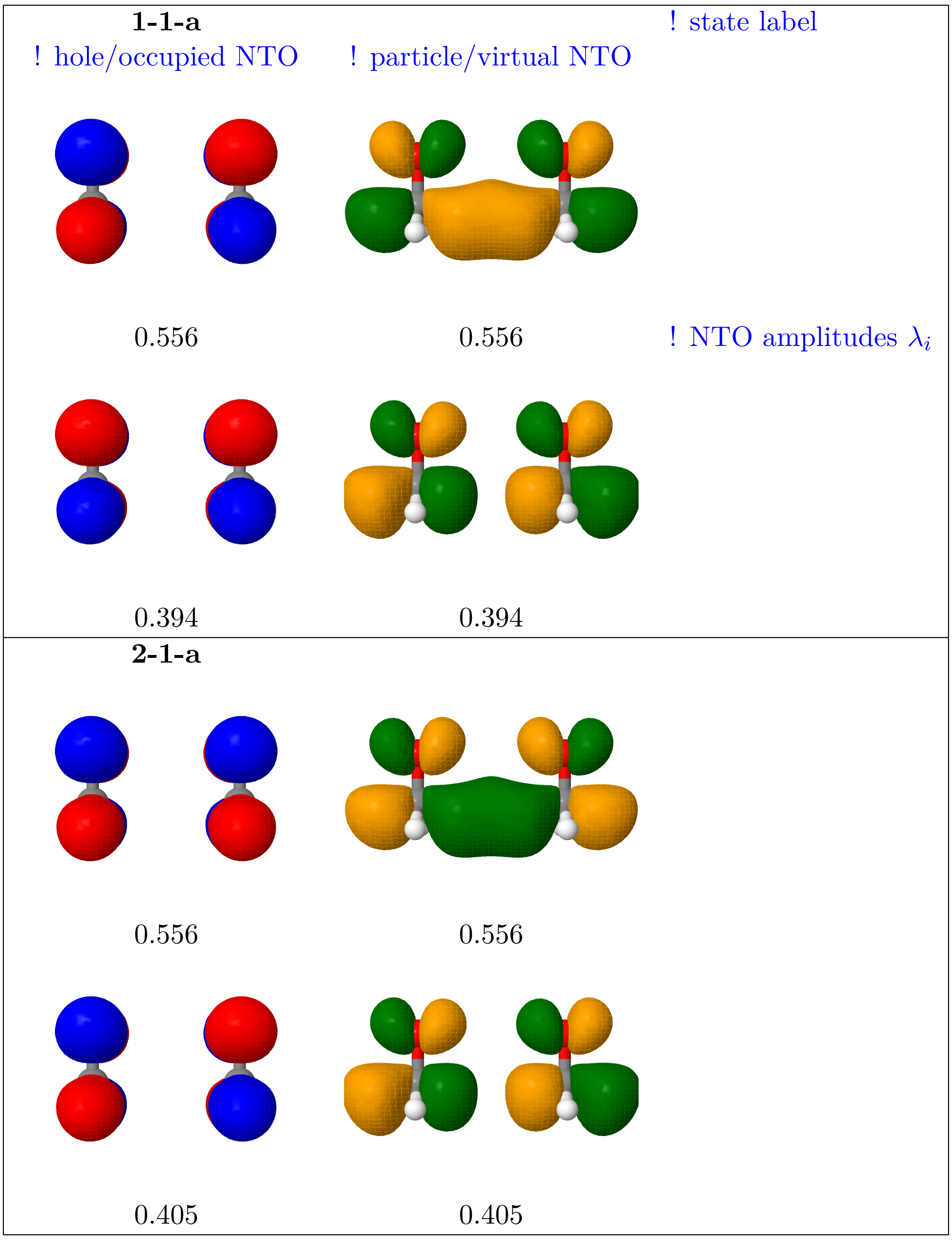


Figure 1: NTOs for the formaldehyde dimer.

4 Charge transfer number and exciton analysis (Turbomole)

In this part an analysis of the charge transfer numbers is carried out. This requires dividing the system into fragments, which are analyzed together. Choosing which atoms are grouped into individual fragments and how these fragments are arranged is the first critical step in the charge transfer number analysis. Special care has to be taken when choosing these fragments, and it is often necessary to try different settings.

4.1 Input generation

First, it is helpful to look at the molecule using a molecular structure editor (e.g. Avogadro) and display the atom lables.

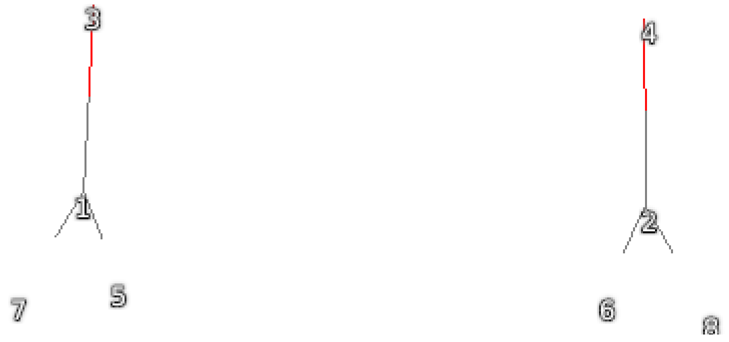


Figure 2: Atom numbering in the formaldehyde dimer.

In the present simple case we want to divide our molecule into two fragments, one for each molecule. One fragment will contain the atom indices 1,3,5,7 the other one 2,4,6,8.

After deciding on the fragment definition run

```
> theodore theoinp
```

And start out the same as in Section 3.1. Then continue:

```
Perform CT number analysis?
Choice (y/n): [y] y <ENTER> ! This time we want to do the CT number analysis

Mode for specifying molecular fragments (at_lists):
  [ 1] Manual input
  [ 2] Automatic generation by fragment (using python-openbabel)
  [ 3] Automatic generation for transition metal complexes (using python-openbabel)
  [ 4] Mixed manual/automatic generation (using python-openbabel)
  [ 5] Automatic generation by element (using python-openbabel)
  [ 6] Leave empty and fill out later
Choice: 1 <ENTER> ! We use "Manual input" here, for other options see Section 6

Input the indices of the atoms belonging to fragment 1:
(separated by spaces)
Choice: 1 3 5 7 <ENTER> ! Atom indices according to Figure 2
```

```
Input the indices of the atoms belonging to fragment 2:
(separated by spaces)
Choice: 2 4 6 8 <ENTER>

Input the indices of the atoms belonging to fragment 3:
(separated by spaces)
Choice:<ENTER> ! Leave empty to quit

Checking whether the at_lists definition is valid ...
at_lists= [[1, 3, 5, 7], [2, 4, 6, 8]]
  2 lists with individual numbers of entries:
[4, 4] ! Two fragments with four atoms each
  8 total entries, with maximal value 8

Formula for Omega matrix computation
   0 - simple, 1 - Mulliken, 2 - Lowdin (Om_formula):
Choice: [2] <ENTER>

Omega descriptors to be computed:
  [ 1] Standard set
  [ 2] Transition metal complex
  [ 3] None
Choice: [1]<ENTER>

Print-out of electron/hole populations
  [ 1] None
  [ 2] For fragments
  [ 3] For fragments and individual atoms
Choice: [1]1 <ENTER> ! for the symmetric case this analysis does not really help

Perform natural transition orbital (NTO) analysis?
Choice (y/n): [y] n <ENTER> ! already did that before ...

Perform analysis of domain NTOs and conditional densities? (comp_dntos):
Choice (y/n): [n] <ENTER>

Calculation of transition densities between ground state and excited states (requires orbkit) (comp_rho0n):
Choice (y/n): [n] <ENTER>

Perform exciton analysis?
Choice (y/n): [y] <ENTER> ! Let's do the exciton analysis here, as well

Compute approximate exciton size?
Choice (y/n): [y] <ENTER>

Adjust detailed output options?
Choice (y/n): [n] <ENTER>

Name of input file
Choice: [dens_ana.in] <ENTER>
```

4.2 Transition density matrix (1TDM) analysis

Again run:

```
> theodore analyze_tden
```

Now, a more extended print-out is available:

```
state       dE(eV)    f      Om     POS     PR      CT     COH    CTnt  RMSeh
-------------------------------------------------------------------------
1(1)a       4.174  0.000  0.950  1.500  2.000  0.027  1.056  0.000  1.237
2(1)a       4.192  0.000  0.961  1.500  2.000  0.032  1.067 -0.000  1.248
3(1)a       7.944  0.000  0.971  1.500  2.000  0.167  1.385  0.000  2.107
4(1)a       8.021  0.164  0.968  1.500  2.000  0.198  1.467 -0.000  2.186
5(1)a       8.755  0.000  0.973  1.500  2.000  0.851  1.341 -0.000  3.433
6(1)a       8.763  0.052  0.973  1.500  2.000  0.816  1.429  0.000  3.378
```

The meaning of these values is discussed in Refs [2, 3] and at the documentation wiki. Only a brief explanation shall be given here:

- In the case of using `ricc2` the first value `Om` or Ω is just a normalization factor. In cases, where an exact 1TDM is available, this is the one-electron excitation character.

- The values POS=1.500 and PR=2.000 in all cases mean that the excitation is distributed evenly between fragment 1 and fragment 2 (for symmetry reasons)

- The crucial information lies in the CT value. CT$\approx$ 0 for the first four excited states, meaning that these are mostly coupled local excitations (Frenkel excitons). For the last two states CT is greater than 0.8 indicating that these are charge resonance states.

- The trend in the CT values is also reflected by the (approximated) root-mean square electron-hole separation (RMSeh, also denoted $\tilde{d}_{exc}$) given in Å [5]. This value is is about equal to the intermolecular separation of 3.5 Å in the case of the charge resonance states while it is significantly smaller for the locally excited states.

4.3 Electron-hole correlation plots

Electron-hole correlation plots are pseudocolor matrix plots representing the charge-transfer numbers Ω_{AB}. To create electron-hole correlation plots, run

```
> theodore plot_omfrag
```

Simply use all default values and then look at `OmFrag.html` in a browser (Fig. 3).

Here, the results are rather trivial since there are only two fragments in the calculation, which are equivalent for symmetry reasons. The locally excited (Frenkel) states are represented by black boxes on the main diagonal (going from lower left to upper right) while the charge resonance states are distinguished by off-diagonal contributions. Note, however, that distinguishing between Frenkel and charge resonance states without these tools is quite challenging.

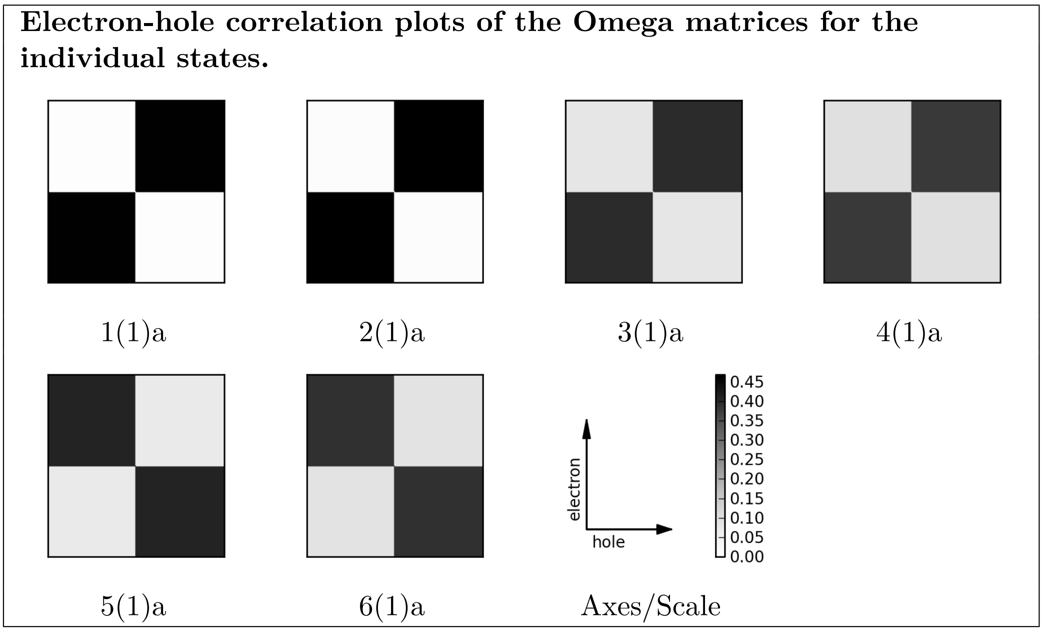


Figure 3: Electron-hole correlation plots for the formaldehyde dimer.

5 Interface to the external cclib library (Gaussian 09)

Gaussian, GAMESS, Orca, and some other programs can be parsed through the cclib library [4]. cclib is included in the default installation. The present example also uses the `python-openbabel` package for automatic input generation (but you can also use the manual input mode from the previous example).

Start by copying the relevant files

```
> cp -r $THEODIR/EXAMPLES/CCLIB/fa2.cclib/QC_FILES/ fa2.cclib.tutorial
```

5.1 Check the log file

When using cclib, one should start by checking whether the file can be parsed correctly

```
> theodore cc_check gaussian.log
```

```
...
Essential attributes:
       mocoeffs ... True
      atombasis ... True
          natom ... True
          homos ... True
     moenergies ... True
     etenergies ... True
         etsyms ... True
         etsecs ... True
```

```
Optional attributes:
         etoscs ... True
     aooverlaps ... False
         mosyms ... True

Attributes for structure parsing and creation of Molden file:
         gbasis ... False ! gbasis is missing - no Molden files
          natom ... True
     atomcoords ... True
        atomnos ... True


 gaussian.log can be parsed by using rtype=’cclib’ in dens_ana.in.! this is the important part
 But conversion to Molden format is not possible
```

5.2 Input generation

As usual:

```
> theodore theoinp
```

```
Type of job (rtype):
  [ 1]      qcadc - Q-Chem ADC (libwfa output)
  [ 2]     libwfa - General libwfa output
  [ 3]    qctddft - Q-Chem TDDFT
  [ 4]       fchk - Q-Chem fchk file
  [ 5]   colmcscf - Columbus MCSCF
  [ 6]    colmrci - Columbus MR-CI (tden analysis)
  [ 7]      rassi - Molcas RASSI
  [ 8]        nos - Read natural orbitals (Molden format) for sden analysis: Columbus, Molcas, ...
  [ 9]      ricc2 - Turbomole ricc2
  [10]       escf - Turbomole escf
  [11]   terachem - Terachem (TDDFT)
  [12]      cclib - Use external cclib library: Gaussian, GAMESS, ...
  [13]       orca - ORCA TDDFT (using a Molden file and cclib)
  [14]        adf - ADF (TDDFT)
  [15]     tddftb - DFTB+ - TDDFTB
  [16]    dftmrci - DFT/MRCI
  [17]     onetep - ONETEP
Choice: 12 <ENTER>

Main file to read (rfile):
Choice (autocomplete enabled): gaussian.log <ENTER>

Note: If used in connection with ORBKIT it is preferable to have an externally generated Molden file.
Do you have an externally generated Molden file?
Choice (y/n): [n] <ENTER>
! If available a Molden file generated by Gaussian would be preferable but we do not need it here

Analysis of transition density matrices?
Choice (y/n): [y] <ENTER>

Perform CT number analysis?
Choice (y/n): [y] <ENTER>
Fragment definition for CT nubmer analysis

Mode for specifying molecular fragments (at_lists):
  [ 1] Manual input
  [ 2] Automatic generation by fragment (using python-openbabel)
  [ 3] Automatic generation for transition metal complexes (using python-openbabel)
  [ 4] Mixed manual/automatic generation (using python-openbabel)
  [ 5] Automatic generation by element (using python-openbabel)
  [ 6] Leave empty and fill out later
Choice:2 <ENTER> ! since there are two well-separated molecules, we can use the automatic mode

Automatic generation of at_lists partitioning ...

Coordinate file (coor_file):
Choice (autocomplete enabled): gaussian.log <ENTER> ! simply take the log file

Format of coordinate file (coor_format):
Choice: g09 ! format, as recongized by openbabel
```

The relevant formats are:
GAUSSIAN - g03, g09
GAMESS - gamout
Q-CHEM - qcout

```
*** Fragment composition *** ! Check that everything worked
  Fragment 1: C H2 O ! This looks reasonable: two formaldehyde molecules
  Fragment 2: C H2 O

Checking whether the at_lists definition is valid ...
at_lists= [[1, 3, 5, 7], [2, 4, 6, 8]] ! correct indices
  2 lists with individual numbers of entries:
[4, 4]
  8 total entries, with maximal value 8

Formula for Omega matrix computation
   0 - simple, 1 - Mulliken, 2 - Lowdin (Om_formula):
Choice: [2]  <ENTER>

Omega descriptors to be computed:
  [ 1] Standard set
  [ 2] Transition metal complex
  [ 3] None
Choice: [1]  <ENTER>

Print-out of electron/hole populations
  [ 1] None
  [ 2] For fragments
  [ 3] For fragments and individual atoms
Choice: [1] 2 <ENTER>

Perform natural transition orbital (NTO) analysis?
Choice (y/n): [y] n <ENTER> ! no possibility to visualize them if MOLDEN export does not work

Perform analysis of domain NTOs and conditional densities? (comp_dntos):
Choice (y/n): [n]  <ENTER>

Calculation of transition densities between ground state and excited states (requires orbkit) (comp_rho0n):
Choice (y/n): [n]  <ENTER>

Perform exciton analysis?
Choice (y/n): [y]  <ENTER>

Compute approximate exciton size?
Choice (y/n): [y]  <ENTER>

Adjust detailed output options?
Choice (y/n): [n]  <ENTER>

Name of input file
Choice: [dens_ana.in]  <ENTER>
```

```
Finished: File dens_ana.in written.
```

The following file `dens_ana.in` was created:

```
rtype='cclib'
rfile='gaussian.log'
coor_file='gaussian.log'
coor_format='log'
at_lists=[[1, 3, 5, 7], [2, 4, 6, 8]]
Om_formula=2
eh_pop=1
comp_ntos=False
comp_dntos=False
comp_ntos=False
comp_rho0n=False
prop_list=['Om', 'POS', 'PR', 'CT', 'COH', 'CTnt', 'RMSeh']
```

5.3 Transition density matrix (1TDM) analysis

Again run:

```
> theodore analyze_tden
```

The output looks similar as it did before only that at the TDDFT/PBE level the CT states are lower in energy and the separation between local and CT states is not as clear cut.

```
state       dE(eV)    f     Om    POS     PR     CT    COH   CTnt  RMSeh
--------------------------------------------------------------------------
1SingA2     3.583  0.000  1.000  1.500  2.000  0.697  1.730  0.000  3.111
2SingB1     3.635  0.000  1.000  1.500  2.000  0.736  1.635  0.000  3.187
3SingB1     4.242  0.000  1.001  1.500  2.000  0.263  1.634  0.000  2.090
4SingA2     4.284  0.000  1.001  1.500  2.000  0.302  1.729 -0.000  2.202
5SingB2     7.793  0.011  1.001  1.500  2.000  0.956  1.091  0.000  3.531
6SingA1     7.851  0.001  1.000  1.500  2.000  0.991  1.019  0.000  3.589
...
```

6 Advanced fragment input and double excitations (Columbus)

The secure way for fragment definition is always the manual mode described in Section 3.1. In some cases, i.e. when the fragments of interest are separate molecules, one can use the option “Automatic generation from coordinate file” in `theoinp`. A more sophisticated method for automatic fragment definition is described in the next section. This method relies on the Avogadro molecular structure editor and the availability of the `python-openbabel` package.

This method is described in the next two sections. If you just wish to run the job without the input generation, copy the `dens_ana.in file` given at the bottom of Section 6.2.

Take the input files from the `EXAMPLES` directory in the TheoDORE distribution

```
> cp -r $THEODIR/EXAMPLES/STANDARD/hexatriene.colmrci/QC_FILES/ hexatriene.tutorial
```

6.1 Fragment preparation using Avogadro

First the `geom` file in COLUMBUS format has to be converted to the more common xyz format.

```
> theodore babel geom geom.xyz
```

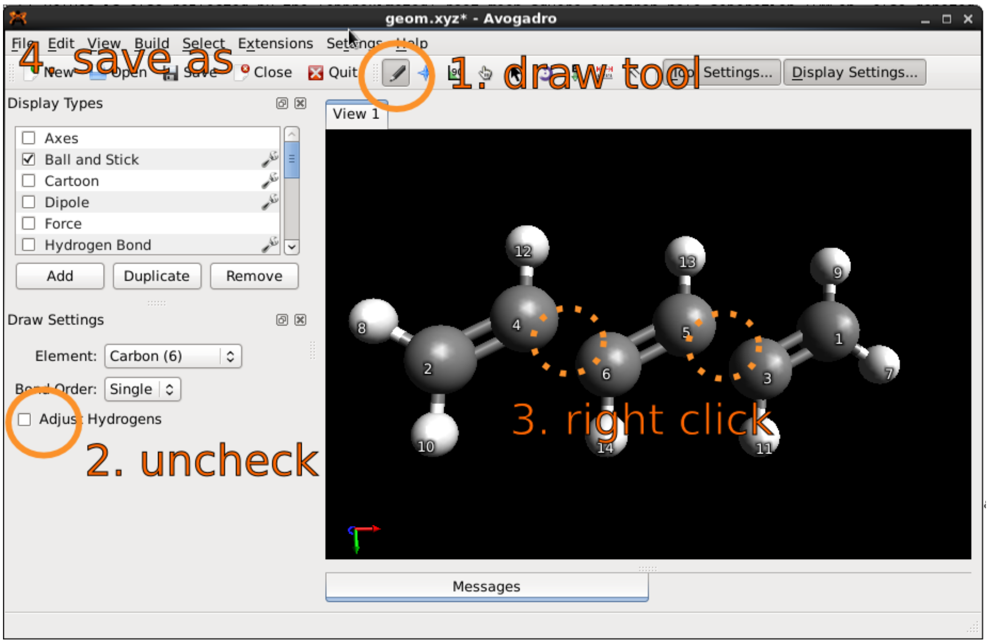
> *This step is specific for COLUMBUS. In many other cases AVOGADRO can directly read the structure file or logfile.*

This file can be opened with AVOGADRO

```
> avogadro geom.xyz
```

In AVOGADRO the following steps have to be performed (Figure 4)

1. Click the pen (draw tool)
2. Uncheck "Adjust Hydrogens"
3. Right-click the bonds that you wish to delete to divide the molecule into fragments
4. Save the file as `geom.mol` ! use a structure format with explicit bonds



4. save as

1. draw tool

☐ Axes
☑ Ball and Stick
☐ Cartoon
☐ Dipole
☐ Force
☐ Hydrogen Bond

Element: Carbon (6)

Bond Order: Single

☐ Adjust Hydrogens

2. uncheck

3. right click

Figure 4: Fragment definition using AVOGADRO.

6.2 Input generation

Now run `theoinp` using the newly created `geom.mol` file as a template.

```
 > theodore theoinp
```

```
Type of job (rtype):
  [ 1]      qcadc - Q-Chem ADC (libwfa output)
  [ 2]     libwfa - General libwfa output
  [ 3]    qctddft - Q-Chem TDDFT
  [ 4]       fchk - Q-Chem fchk file
  [ 5]   colmcscf - Columbus MCSCF
  [ 6]    colmrci - Columbus MR-CI (tden analysis)
  [ 7]      rassi - Molcas RASSI
...
Choice: [6]  <ENTER>

MO file (Molden format)
 -> This file should ideally contain a square invertible coefficient matrix (mo_file):
Choice (autocomplete enabled): [MOLDEN/molden_mo_mc.sp]  <ENTER>

Analysis of transition density matrices?
Choice (y/n): [y]  <ENTER>

Perform CT number analysis?
Choice (y/n): [y] y <ENTER>
Fragment definition for CT nubmer analysis

Mode for specifying molecular fragments (at_lists):
  [ 1] Manual input
  [ 2] Automatic generation by fragment (using python-openbabel)
  [ 3] Automatic generation for transition metal complexes (using python-openbabel)
  [ 4] Mixed manual/automatic generation (using python-openbabel)
  [ 5] Automatic generation by element (using python-openbabel)
  [ 6] Leave empty and fill out later
Choice: 2 <ENTER> ! use automatic generation if python-openbabel is available
Automatic generation of at_lists partitioning ...

Coordinate file (coor_file):
Choice (autocomplete enabled): [geom] geom.mol <ENTER> ! specify the newly created file

Format of coordinate file (coor_format):
Choice: [mol]  <ENTER>

*** Fragment composition ***
  Fragment 1: C2 H3
  Fragment 2: C2 H3
  Fragment 3: C2 H2 ! the central $C_2H_2$ fragment is at the end ...
  ! ... his has to be changed (see below)
Checking whether the at_lists definition is valid ...
at_lists= [[1, 3, 7, 9, 11], [2, 4, 8, 10, 12], [5, 6, 13, 14]]
  3 lists with individual numbers of entries:
[5, 5, 4]
  14 total entries, with maximal value 14

Formula for Omega matrix computation
   0 - simple, 1 - Mulliken, 2 - Lowdin (Om_formula):
Choice: [2]  <ENTER>

Omega descriptors to be computed:
  [ 1] Standard set
  [ 2] Transition metal complex
  [ 3] None
Choice: [1]  <ENTER>
```

```
Print-out of electron/hole populations
  [ 1] None
  [ 2] For fragments
  [ 3] For fragments and individual atoms
Choice: [1] 2 <ENTER>

Perform natural transition orbital (NTO) analysis? (comp_ntos):
Choice (y/n): [y] n <ENTER>

Perform analysis of domain NTOs and conditional densities? (comp_dntos):
Choice (y/n): [n]  <ENTER>

Calculation of transition densities between ground state and excited states (requires orbkit) (comp_rho0n):
Choice (y/n): [n]  <ENTER>

Perform exciton analysis?
Choice (y/n): [y] n <ENTER>

Were there frozen core orbitals in the calculation?
Choice (y/n): [y] n <ENTER> ! for general COLUMBUS jobs frozen core orbitals would have to be specified here

Adjust detailed output options?
Choice (y/n): [n]  <ENTER>

Name of input file
Choice: [dens_ana.in]  <ENTER>
Finished: File dens_ana.in written.
```

In the `dens_ana.in` file, it is necessary to adjust the fragment definitions in `at_lists` to make sure that the central C_2H_2 fragment is really in the middle. The file should look like this:

```
rtype="colmrci"
mo_file="MOLDEN/molden_mo_mc.sp"
coor_file="geom.mol"
coor_format="mol"
at_lists=[[1, 3, 7, 9, 11], [5, 6, 13, 14], [2, 4, 8, 10, 12]]
Om_formula=2
eh_pop=1
comp_ntos=False
comp_dntos=False
comp_ntos=False
comp_rho0n=False
prop_list=["Om", "POS", "PR", "CT", "COH", "CTnt"]
```

When using the automatic fragment definition, it is generally advisable to check the results using a graphical representation of the molecule (c.f. Figure 4) and to adjust things if necessary.

6.3 Transition density matrix (1TDM) analysis

As always:

```
> theodore analyze_tden
```

```
I2.1-2
-------------------------------------------------------
       Fragment        h+        e-       sum      diff
-------------------------------------------------------
         C2 H3    0.12852   0.11292   0.24144   0.01561
         C2 H2    0.17536   0.20658   0.38194  -0.03122
```

```
        C2 H3    0.12852   0.11292   0.24144   0.01561
-------------------------------------------------------
                 0.43241   0.43241   0.86482   0.00000
-------------------------------------------------------

File ehFrag.txt with information about e/h populations written.

state       dE(eV)    f      Om    POS     PR     CT    COH   CTnt
------------------------------------------------------------------
I1.1-2      5.565  0.000  0.375  2.000  2.966  0.812  2.715  0.000
I2.1-1      6.530  1.254  0.865  2.000  2.897  0.617  2.861  0.000
I2.1-2      6.772  0.006  0.432  2.000  2.837  0.897  1.833 -0.000
```

The Ω value serves as a method-independent measure of double excitation character [3]. values close to one indicate single excitation character whereas $\Omega < 0.8$ shows (partial) doubly excited character. In the present case, the first and third states show predominant double excitation character ($\Omega < 0.5$). Note, that for low Ω values the 1TDM analysis does not provide a complete description and one might resort to the difference density matrix instead.

7 Fragment decomposition for a transition metal complex

A more compact representation for showing the different local and charge transfer contributions to an excited state has been worked out in Ref. [7]. For this example, we are going to use a small Ir complex with three bidentate ligands. First get the files from the EXAMPLES directory:
> cp -r $THEODIR/EXAMPLES/EXTRA/ir_c3n3.qctddft/QC_FILES/ ir_c3n3.qctddft.tutorial

7.1 Input generation

Call > theodore theoinp

```
Type of job (rtype):
  [ 1]      qcadc - Q-Chem ADC (libwfa output)
  [ 2]     libwfa - General libwfa output
  [ 3]    qctddft - Q-Chem TDDFT
  [ 4]       fchk - Q-Chem fchk file
  [ 5]   colmcscf - Columbus MCSCF
...
Choice: [1] 3 <ENTER>

Main file to read (rfile):
Choice (autocomplete enabled): [qchem.out]  <ENTER>

Did you run "state_analysis=True"? (read_libwfa):
Choice (y/n): [n] y <ENTER>
! We started with a state analysis (libwfa) in Q-Chem and are only post-processing results now.

Read TDA rather than full TDDFT results? (TDA):
Choice (y/n): [n]  <ENTER>

Analysis of transition density matrices?
Choice (y/n): [y]  <ENTER>

Perform CT number analysis?
Choice (y/n): [y]  <ENTER>
Fragment definition for CT nubmer analysis

Mode for specifying molecular fragments (at_lists):
  [ 1] Manual input
  [ 2] Automatic generation by fragment (using python-openbabel)
  [ 3] Automatic generation for transition metal complexes (using python-openbabel)
  [ 4] Mixed manual/automatic generation (using python-openbabel)
  [ 5] Automatic generation by element (using python-openbabel)
  [ 6] Leave empty and fill out later
Choice: 3 <ENTER> ! For transition metal complexes, openbabel can automatically distinguish between the metal centre
                    and ligands provided you give the index of the metal centre.
! If openbabel is not activated, use option [6] and set up at_lists manually in dens_ana.in:
at_lists=[[1], [2, 6, 23, 7, 16, 15, 5, 14], [3, 12, 25, 13, 22, 20, 9, 21], [4, 19, 11, 18, 10, 17, 8, 24]]
Automatic generation of at_lists partitioning ...

Coordinate file (coor_file):
Choice (autocomplete enabled): [qchem.out] qchem.mol <ENTER>
Detected file type: mol

Format of coordinate file (coor_format):
Choice: [mol]  <ENTER>
```

```
Input the index of the transition metal atom (or indices of the corresponding fragment)
Choice: 1 <ENTER> ! In this case Ir is atom 1 (you can check this in avogadro)

*** Fragment composition ***
  Fragment 1: Ir
  Fragment 2: C3 H4 N
  Fragment 3: C3 H4 N
  Fragment 4: C3 H4 N

Checking whether the at_lists definition is valid ...
at_lists= [[1], [2, 23, 6, 16, 7, 5, 15, 14], [3, 25, 12, 22, 13, 9, 20, 21], [4, 11, 19, 10, 18, 17, 8, 24]]
  4 lists with individual numbers of entries:
[1, 8, 8, 8]
  25 total entries, with maximal value 25

Omega descriptors to be computed:
  [ 1] Standard set
  [ 2] Transition metal complex
  [ 3] None
Choice: [1] 2 <ENTER> ! Use special descriptors for TM complexes

Print-out of electron/hole populations
  [ 1] None
  [ 2] For fragments
  [ 3] For fragments and individual atoms
Choice: [1]  <ENTER>

Perform exciton analysis?
Choice (y/n): [y]  <ENTER>

Compute approximate exciton size?
Choice (y/n): [y]  <ENTER>

Molecular coordinates for exciton analysis:

Coordinate file (coor_file):
Choice (autocomplete enabled): [qchem.xyz]  <ENTER>

Format of coordinate file (coor_format):
Choice: [xyz]  <ENTER>

Parse exciton information from libwfa analysis?
Choice (y/n): [n]  <ENTER>

Parse 1DDM exciton information from libwfa analysis?
Choice (y/n): [n]  <ENTER>

Adjust detailed output options?
Choice (y/n): [n]  <ENTER>

Name of input file
Choice: [dens_ana.in]  <ENTER>
Finished: File dens_ana.in written.
```

7.2 Transition density matrix analysis and decomposition

Run > theodore analyze_tden

This gives the results for the first six excited states:

```
state       dE(eV)    f     Om    POSi   POSf    PR     CT     MC     LC    MLCT   LMCT   LLCT  RMSeh
----------------------------------------------------------------------------------------------------------
S_1          3.948  0.031  1.012  1.889  2.852  3.058  0.646  0.043  0.311  0.513  0.034  0.098  2.397
S_2          3.999  0.035  1.013  1.948  2.910  2.653  0.586  0.098  0.316  0.465  0.052  0.069  2.249
S_3          4.000  0.035  1.012  1.797  2.482  2.633  0.586  0.098  0.316  0.465  0.052  0.069  2.250
S_4          4.408  0.040  1.004  1.650  2.436  2.498  0.788  0.075  0.136  0.569  0.032  0.187  2.685
S_5          4.410  0.041  1.004  1.772  3.154  2.474  0.789  0.075  0.136  0.569  0.032  0.187  2.685
S_6          4.426  0.008  1.004  1.705  2.651  3.011  0.753  0.112  0.134  0.535  0.053  0.166  2.621
```

The analysis shows that all states have around 50% of MLCT character and varying amounts of LC and LLCT admixture.

The results for each state can now be decomposed into contributions from local excitations, MLCT and LLCT.

Call > theodore plot_om_bars

```
Name of the file with the Omega matrix entries (OmFfile):
Choice (autocomplete enabled): [OmFrag.txt]  <ENTER>

Name of the file with the tden information (tdenfile):
Choice (autocomplete enabled): [tden_summ.txt]  <ENTER>

Width of the plot (cm) (width):
Choice: [7.000000]  <ENTER>
Please enter the different excitation components to be plotted
    - leave empty to finish

Name of component 1 (e.g. MLCT or AB)
Choice: MLCT <ENTER>

Color for plotting
Choice: blue <ENTER> ! blue, green, yellow, red are available

 *** Fragment pairs belonging to MLCT ***
  Enter two indices between 1 and 4, separated by spaces
  Leave empty to finish

Hole/electron indices for pair 1
Choice: 1 2 <ENTER> ! CT from metal to ligand 1

Hole/electron indices for pair 2
Choice: 1 3 <ENTER> ! CT from metal to ligand 2

Hole/electron indices for pair 3
Choice: 1 4 <ENTER> ! CT from metal to ligand 3

Hole/electron indices for pair 4
Choice:  <ENTER> ! Leave empty to switch to the next component
 ... switching to next component.

Name of component 2 (e.g. MLCT or A-B)
Choice: LMCT <ENTER>
```

```
Color for plotting
Choice: green <ENTER>

 *** Fragment pairs belonging to LMCT ***
  Enter two indices between 1 and 4, separated by spaces
  Leave empty to finish

Hole/electron indices for pair 1
Choice: 2 1 <ENTER> ! CT from ligand 1 to metal

Hole/electron indices for pair 2
Choice: 3 1 <ENTER>

Hole/electron indices for pair 3
Choice: 4 1 <ENTER>

Hole/electron indices for pair 4
Choice:  <ENTER>
 ... switching to next component.

Name of component 3 (e.g. MLCT or A-B)
Choice: LC ! ligand-centred local excitations

Color for plotting
Choice: red <ENTER>

 *** Fragment pairs belonging to LC ***
  Enter two indices between 1 and 4, separated by spaces
  Leave empty to finish

Hole/electron indices for pair 1
Choice: 2 2 <ENTER> ! Local excitation on ligand 1

Hole/electron indices for pair 2
Choice: 3 3 <ENTER>

Hole/electron indices for pair 3
Choice: 4 4 <ENTER>

Hole/electron indices for pair 4
Choice:  <ENTER>
 ... switching to next component.

Name of component 4 (e.g. MLCT or A-B)
Choice: LLCT <ENTER> ! Ligand to ligand charge transfer

Color for plotting
Choice: yellow

 *** Fragment pairs belonging to LLCT ***
  Enter two indices between 1 and 4, separated by spaces
  Leave empty to finish

Hole/electron indices for pair 1
Choice: 2 3 <ENTER>

Hole/electron indices for pair 2
Choice: 2 4 <ENTER> ! We have contributions from each ligand to the two other ligands

Hole/electron indices for pair 3
Choice: 3 2 <ENTER>

Hole/electron indices for pair 4
Choice: 3 4 <ENTER>

Hole/electron indices for pair 5
```

```
Choice: 4 2 <ENTER>

Hole/electron indices for pair 6
Choice: 4 3 <ENTER>

Hole/electron indices for pair 7
Choice:  <ENTER>
 ... switching to next component.

Name of component 5 (e.g. MLCT or A-B)
Choice:  <ENTER> ! Leave empty to finish

 ... component input finished.
  File Om_bars.tex written.
  -> Create plots using: pdflatex Om_bars.tex
```

As the end of the interactive script suggests, run: > `pdflatex Om_bars.tex`

Once this is finished, open the resulting pdf file using a suitable program e.g.:

> `okular Om_bars.pdf`

The results of the decomposition are as shown:

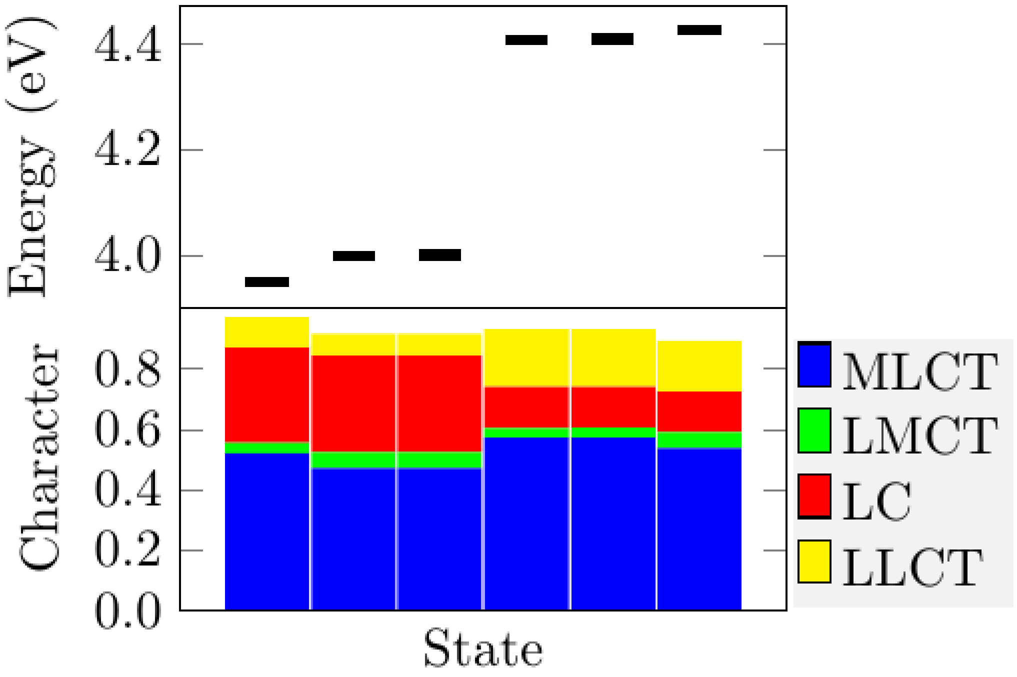


Figure 5: The first six excited states for the Ir complex decomposed into contributions from MLCT, LMCT and local excitations on ligands

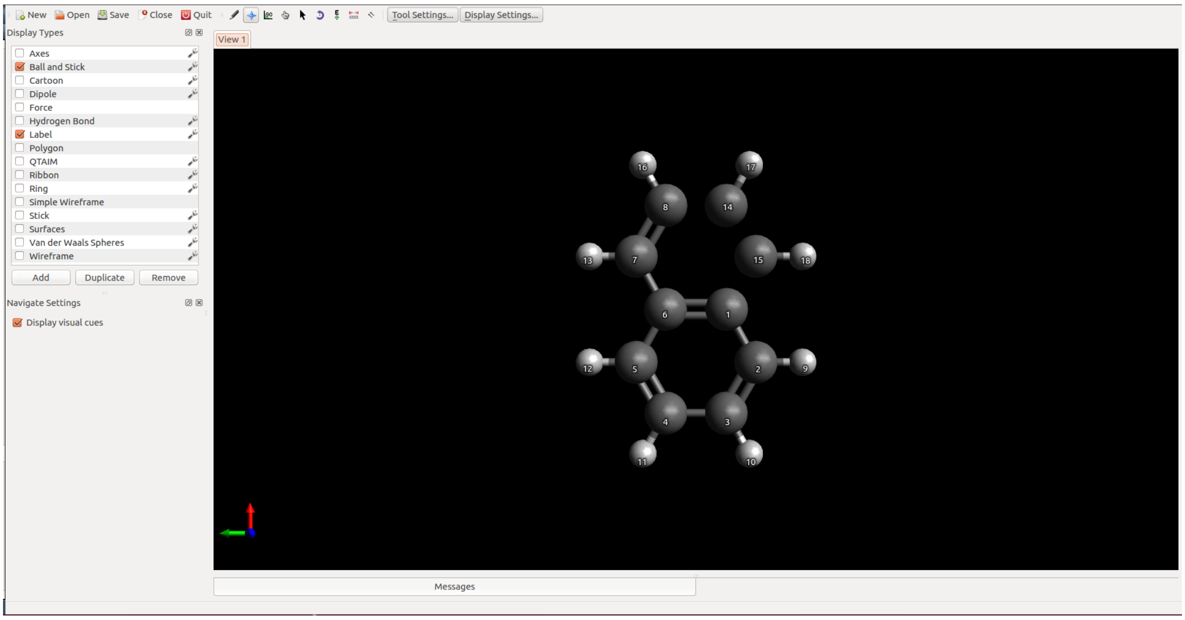
8 Domain NTO and conditional density analysis

It is possible to visualise excited state correlation using TheoDORE. This is done by plotting domain NTOs and conditional densities. The idea is the consider the excited state using an 'electron-hole' picture. A hole is fixed on a fragment of the molecule and the resulting conditional electron density is observed. Further explanation can be found in Ref [8]. This procedure generally works if it is possible to run `analyze_tden.py` and there is a Molden file with orbital information. Exporting the densities as cube files requires `ORBKIT` which is not installed by default but is available from `https://github.com/felixplasser/orbkit`.

Get the input files from the `EXAMPLES` directory

```
> cp -r $THEODIR/EXAMPLES/STANDARD/naphth.fchk/QC_FILES/ naphth.fchk.tutorial
```

From the tutorial folder, open the .xyz file with Avogadro, create fragments for the napthalene molecule and save it as a .mol file as done in Section 6.



Display Types

- ☐ Axes
- ☑ Ball and Stick
- ☐ Cartoon
- ☐ Dipole
- ☐ Force
- ☐ Hydrogen Bond
- ☑ Label
- ☐ Polygon
- ☐ QTAIM
- ☐ Ribbon
- ☐ Ring
- ☐ Simple Wireframe
- ☐ Stick
- ☐ Surfaces
- ☐ Van der Waals Spheres
- ☐ Wireframe

Navigate Settings

- ☑ Display visual cues

In the present case, we want to separate two symmetry-unique CH groups from the remaining molecule. This allows us to view correlation effects between the individual atoms.

8.1 Input generation

Call > theodore theoinp

```
Type of job (rtype):
  [ 1]      qcadc - Q-Chem ADC (libwfa output)
  [ 2]     libwfa - General libwfa output
  [ 3]    qctddft - Q-Chem TDDFT
  [ 4]       fchk - Q-Chem fchk file
  [ 5]   colmcscf - Columbus MCSCF
  [ 6]    colmrci - Columbus MR-CI (tden analysis)
...
Choice: [4]  <ENTER>

Main file to read (rfile):
Choice (autocomplete enabled): [qchem.fchk]  <ENTER>

Analysis of transition density matrices?
Choice (y/n): [y]  <ENTER>

Perform CT number analysis?
Choice (y/n): [y]  <ENTER>
Fragment definition for CT nubmer analysis

Mode for specifying molecular fragments (at_lists):
  [ 1] Manual input
  [ 2] Automatic generation by fragment (using python-openbabel)
  [ 3] Automatic generation for transition metal complexes (using python-openbabel)
  [ 4] Mixed manual/automatic generation (using python-openbabel)
  [ 5] Automatic generation by element (using python-openbabel)
  [ 6] Leave empty and fill out later
Choice: 2 <ENTER>
! If you do not have python-openbabel use [6] and the at_lists shown later.

Coordinate file (coor_file):
Choice (autocomplete enabled): [qchem.out] coord.mol <ENTER>
Detected file type: mol

Format of coordinate file (coor_format):
Choice: [mol]  <ENTER>

*** Fragment composition ***
  Fragment 1: C8 H6
  Fragment 2: C H
  Fragment 3: C H

Checking whether the at_lists definition is valid ...
at_lists= [[1, 6, 2, 3, 9, 4, 10, 5, 11, 12, 7, 8, 13, 16], [14, 17], [15, 18]]
  3 lists with individual numbers of entries:
[14, 2, 2]
  18 total entries, with maximal value 18

Formula for Omega matrix computation
   0 - simple, 1 - Mulliken, 2 - Lowdin (Om_formula):
Choice: [2]  <ENTER>

Omega descriptors to be computed:
  [ 1] Standard set
  [ 2] Transition metal complex
  [ 3] None
Choice: [1]  <ENTER>

Print-out of electron/hole populations
  [ 1] None
  [ 2] For fragments
  [ 3] For fragments and individual atoms
Choice: [1]  <ENTER>

Perform natural transition orbital (NTO) analysis? (comp_ntos):
Choice (y/n): [y]  <ENTER>

Perform analysis of domain NTOs and conditional densities? (comp_dntos):
Choice (y/n): [n] y <ENTER>

NTOs as Jmol script? (jmol_orbitals):
Choice (y/n): [y]  <ENTER>

NTOs in Molden format (molden_orbitals):
Choice (y/n): [n]  <ENTER>

NTOs in Cube file format (requires orbkit) (cube_orbitals):
Choice (y/n): [n] y <ENTER>

Create VMD Network for NTOs (vmd_ntos):
Choice (y/n): [n]  <ENTER>

Calculation of Particle/Hole density (requires orbkit)? (comp_p_h_dens):
Choice (y/n): [n] y <ENTER>

Create VMD Network for p/h densities (vmd_ph_dens):
Choice (y/n): [n]  <ENTER>

Compute conditional densities as cube files?
 0 - no, 1 - hole, 2 - electron, 3 - both (comp_dnto_dens):
Choice: [0] 1 <ENTER> ! We are choosing to fix the hole on each fragment
                      ! and observe the resulting conditional electron density

Write conditional densities to fchk file
 0 - no, 1 - hole, 2 - electron, 3 - both (fchk_dnto_dens):
Choice: [0] 1 <ENTER> ! Use this option if you want to view the result in IQMol

Calculation of transition densities between ground state and excited states (requires orbkit) (comp_rho0n):
Choice (y/n): [n]  <ENTER>

Number of CPUs for orbkit calculations (numproc):
Choice: [4] 2 <ENTER> ! naphthalene is relatively small - 2 CPUs should be enough

Perform exciton analysis?
Choice (y/n): [y] n <ENTER>

Adjust detailed output options?
Choice (y/n): [n]  <ENTER>

Name of input file
Choice: [dens_ana.in]  <ENTER>
```

The following should be written to the dens_ana.in file:

```
rtype='fchk'
rfile='qchem.fchk'
coor_file='coord.mol'
coor_format='mol'
at_lists=[[1, 6, 2, 3, 9, 4, 10, 5, 11, 12, 7, 8, 13, 16], [14, 17], [15, 18]]
Om_formula=2
eh_pop=0
comp_ntos=True
comp_dntos=True
jmol_orbitals=True
molden_orbitals=False
cube_orbitals=True
vmd_ntos=False
```

```
comp_p_h_dens=True
vmd_ph_dens=False
comp_dnto_dens=1
fchk_dnto_dens=1
comp_rho0n=False
numproc=2
prop_list=['Om', 'POS', 'PR', 'CT', 'COH', 'CTnt', 'PRNTO', 'Z_HE']
```

8.2 Transition density matrix analysis

Now call: > `theodore analyze_tden`

TheoDORE writes cube files according to, for this three fragment example, the hole being on fragment 1, 2 or 3.

The following states are printed out:

```
state       dE(eV)     f      Om     POS     PR     CT     COH    CTnt  PRNTO   Z_HE
--------------------------------------------------------------------------------
T_B3u_1     4.361      -   1.003  1.308  1.573  0.287  1.387  0.008  2.358  2.746
T_B3u_2     5.176      -   1.005  1.284  1.502  0.356  1.508 -0.004  2.134  2.352
S_B3u_1     5.401  0.000  1.004  1.284  1.503  0.358  1.514  0.003  2.115  2.314
S_B3u_2     7.360  1.724  1.018  1.286  1.510  0.309  1.433  0.003  2.314  3.047
```

8.3 Plotting of the orbitals

Using Jmol, the shortcuts used earlier are available to view the orbitals. Simply run:

> `jmol -n dnto_hole_jmol.spt`   > `jmol -n dnto_elec_jmol.spt`

Then the results can be viewed by looking at the .html files in a browser.

A more compact representation is obtained by processing the cube files generated via VMD. Run:
> `theodore vmd_plots rho*.cb`
When prompted, do the following:

```
Compute volume integrals over cube files for isovalues? (do_vol):
Choice (y/n): [n] y <ENTER>

Use special DNTO mode? (dnto):
Choice (y/n): [n] y <ENTER>

Volume integral for conditional density (iso1):
Choice: [0.750000]  <ENTER>

Volume integral for probe density (iso2):
Choice: [0.750000]  <ENTER>

VMD Material for conditional density (mat1):
Choice: [AOShiny]  <ENTER>

VMD Material for probe density (mat2):
Choice: [Glass1]  <ENTER>

Width of images in output html file (width):
```

```
Choice: [400]  <ENTER>

Number of columns in the output html file (ncol):
Choice: [4] 4 <ENTER> ! This value should be one larger than the number of fragments you use

Adjust file names?
Choice (y/n): [n]  <ENTER>
```

The files required for visualisation will be created. Now do the following:

> `vmd coord.xyz`

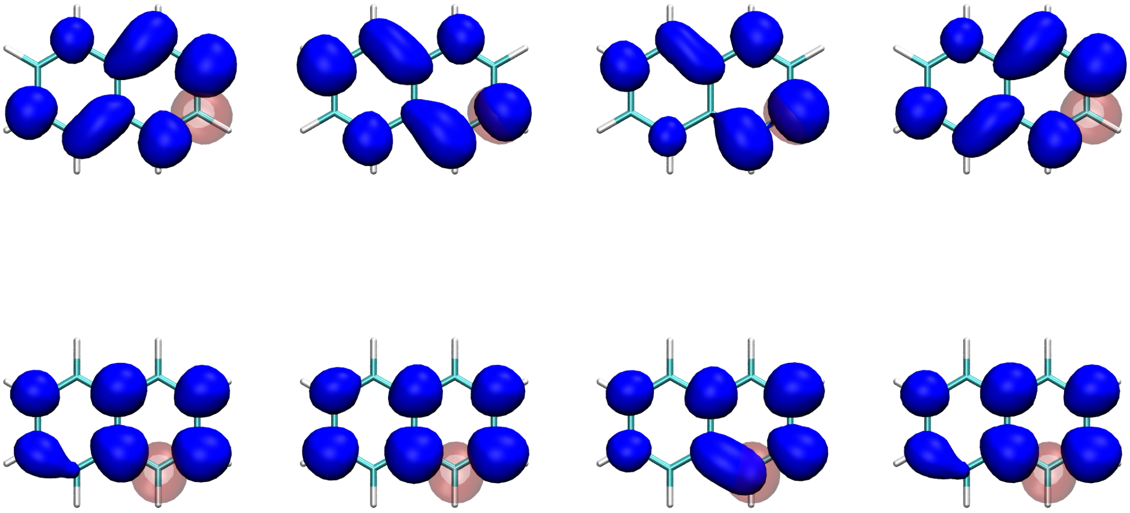
Within `VMD`:

```
1.   File - Load Visualization State - load_all.vmd
2.   Adjust the perspective
3.   File - Load Visualization State - plot_all.vmd
```

Close `VMD` and from call: > `bash convert.bash`

Finally open the images in a web browser > `firefox vmd_plots.html`

The key results you get for the naphthalene CH fragments as outlined in this tutorial are shown below. The red shading indicates the position of the hole.



You can also use PyMOL to visualise the conditional densities but note that you need to change the .cb file ending to .cube . Plots can be created via the QC_PYMOL *toolkit:* `https://github.com/felixplasser/qc_pymol`

9 Attachment/detachment analysis (Molcas - natural orbitals)

While the previous examples were focused on an analysis of the transition density matrices, TheoDORE can also analyze state- and difference-density matrices. These are most conveniently read in as natural orbital (NO) files in Molden format. The utility script `analyze_nos` can be used as a shortcut for analyzing NO files.

Also standard RHF/UHF and DFT (RKS/UKS) orbitals can be analyzed with this mode. It is for example possible to compute the different density between a closed-shell RKS ground state and a UKS triplet or ionized state.

9.1 NO analysis

Get the input files

```
> cp -r $THEODIR/EXAMPLES/STANDARD/fa2.rassi/QC_FILES/ fa2.rassi.tutorial
```

Then simply call

```
> theodore analyze_nos -u MOLDEN.1 MOLDEN.2 MOLDEN.3
```

The -u flag signifies unrestricted orbitals. Generally speaking the operation of analyze_nos depends on the correct specification of input flags. Call "theodore analyze_nos" without arguments to see a listing of the options.

```
Mulliken populations ! Analysis of the ground state
MOLDEN.1
------------------------------------
  Atom     state        nu       nunl
------------------------------------
  C  1   5.92413   0.10075   0.03645 ! Gross population on the atom, and two measures for unpaired electrons
  C  2   5.92413   0.10075   0.03645
  O  3   8.35389   0.10030   0.03601
  O  4   8.35389   0.10030   0.03601
...
------------------------------------
        32.00002   0.40234   0.14493 ! Total number of electrons / unpaired electrons
------------------------------------

MOLDEN.2
--------------------------------------------------------
  Atom     state        nu       nunl        det       att
--------------------------------------------------------
  C  1   6.04465   0.44212   0.49413   0.03847  -0.15898 ! Also attachment/detachment populations
  C  2   6.04465   0.44212   0.49412   0.03847  -0.15898
  O  3   8.28628   0.67712   0.71951   0.45914  -0.39152
  O  4   8.28628   0.67712   0.71951   0.45914  -0.39152
...
--------------------------------------------------------
        32.00000   2.34492   2.54684   1.10119  -1.10117 ! sum: promotion number p
--------------------------------------------------------
```

```
Valence information ! Valence analysis following Ref. [6]
...

Bond order information ! Bond orders following Ref. [6]
 <at1>-<at2> : <bond order>
MOLDEN.1
  1=3  : 1.6330 ! Double bond between C=O
  1-5  : 0.9229
  1-7  : 0.9229
  2=4  : 1.6330
  2-6  : 0.9229
  2-8  : 0.9229
MOLDEN.2
  1-3  : 1.2048 ! Bond order reduced in excited state
  1-5  : 0.9145
  1-7  : 0.9145
  2-4  : 1.2048
  2-6  : 0.9145
  2-8  : 0.9145
```

9.2 Plotting of the orbitals

In the case of using JMOL, you can use the automatic functionality for creating the natural difference orbitals (NDOs) or natural orbitals (NOs)

```
> jmol -n ndo_jmol.spt
```

The NDOs are in general similar to the NTOs, only that they also contain contributions from double excitations and orbital relaxation [3].

9.3 Alternative

It is also possible to perform the state-density analysis using

```
> theodore theoinp
```

and

```
> theodore analyze_sden
```

10 VIST plot for visualizing aromaticity

This section is still under construction but you can try running the basic commands.

Get the input files

```
> cp -r $THEODIR/EXAMPLES/UTILS/BCyc_VIST/QC_FILES BCyc_VIST.tutorial
```

10.1 Simple example

```
> theodore plot_vist -p -o simple.vmd neutral.log
```

10.2 More complicated example

```
> theodore plot_vist -c -v '0 4' neutral.log triplet.log 2M.log
```

11 Contact

If you have any questions about this tutorial or about the TheoDORE program, please use the forum:

https://sourceforge.net/p/theodore-qc/discussion/bugs_questions/

You can also reach me via email: f.plasser at lboro.ac.uk

References

[1] F. Plasser *JCP* **2016**, 144, 194107. DOI: 10.1063/1.4949535.

[2] F. Plasser and H. Lischka *JCTC* **2012**, 8, 2777. DOI: 10.1021/ct300307c.

[3] F. Plasser, S. A. Bäppler, M. Wormit, A. Dreuw *JCP* **2014**, 141, 024107. DOI: 10.1063/1.4885820

[4] N. M. O'boyle, A. L. Tenderholt, K. M. Langner *J. Comput. Chem.* **2008**, 29, 839. DOI: 10.1002/jcc.20823

[5] S. A. Mewes, J.-M. Mewes, A. Dreuw, F. Plasser *PCCP* **2016**, 18, 2548. DOI: 10.1039/C5CP07077E

[6] I. Mayer *IJQC* **1986**, 29, 477. DOI: 10.1002/qua.560290108

[7] S. Mai, F. Plasser, J. Dorn, M. Fumanal, C. Daniel, L. González *Coordination Chemistry Reviews* **2018**, 361, 74. DOI: 10.1016/j.ccr.2018.01.019

[8] F. Plasser *ChemPhotoChem* **2019**, 3, 702. DOI: 10.1002/cptc.201900014