mwfn: A strict, concise and extensible format for electronic wavefunction storage and exchange

Tian Lu,[a],* Qinxue Chen,[a]

[a] Beijing Kein Research Center for Natural Sciences, Beijing 100022, P. R. China (http://www.keinsci.com)

* Correspondence author. E-mail: sobereva@sina.com

ORCiD
Tian Lu: 0000-0002-1822-1229
Qinxue Chen: 0000-0003-0155-2387

Abstract: Analysis of electronic wavefunction generated by quantum chemistry codes has crucial role in exploring nature of electronic structure and providing valuable information of chemical interest. A file containing wavefunction information is inevitably needed as a communicator between wavefunction analysis codes and quantum chemistry programs. There have been many available formats designed for recording wavefunction, such as .fch, .molden, .wfn, .wfx and so on, however they all have different flaws and thus bringing evident inconvenience for development of new wavefunction analysis codes. To overcome this problem, in this article we define a new format "mwfn" (acronym of "Multiwfn wavefunction file") for electronic wavefunction storage and exchange purposes. This format is strict, concise, extensible, and able to provide all kinds of information for common wavefunction analyses. Since the "mwfn" format fully eliminates all shortcomings of existing formats, we expect it will become a standard for recording wavefunction in the field of wavefunction analysis and quantum chemistry.

Note: The format of "mwfn" described in this document is **<u>version 1.1</u>**. This format may be extended in the future according to actual requirements.

1. Introduction

Analysis of electronic wavefunction generated by quantum chemistry codes has crucial role in exploring nature of electronic structure and providing valuable information of chemical interest. A file containing wavefunction information is inevitably needed as a communicator between wavefunction analysis codes and quantum chemistry programs.

There have been many available formats designed for recording wavefunction, such as .fch,[1] .molden,[2] .wfn, .wfx[3] and so on. A brief overview and comparison of various known formats for recording wavefunction is given in Appendix. As can be seen, all of the existing formats have different flaws, for example:

• The popular .wfn format does not record any virtual orbital and orbital expansion coefficients with respect to basis functions, thus imposing severe limitations on realization of analysis methods.

• The .fch file defined by Gaussian[4] does not contain a field used to record orbital occupation numbers, leading to difficulty in recording certain type of orbitals, such as natural orbitals and adaptive natural density partitioning (AdNDP) orbitals.[5]

• The well-known .molden format (input file of Molden visualization program) was loosely defined, thus making it difficult for wavefunction analysis codes to be fully compatible with .molden files generated by different quantum chemistry programs. Furthermore, this kind of file does not explicitly record atomic nuclear charges and orbital spins, which results in a lot of troubles in implementing many analysis methods.

The Multiwfn program[6,7] developed by one of us supports most important wavefunction analysis methods, it has become one of the most popular codes in the field of wavefunction analysis. During the long-term development of this code, many shortcomings of existing wavefunction recording formats often bring great inconvenience to us. We believe there is a great need to define a format as a new standard for bridging wavefunction analysis codes and quantum chemistry programs. According to our experience, an ideal new format should simultaneously satisfy below conditions:

a) **Comprehensive**: The format should contains sufficient information that needed by almost all the most important and popular analysis methods in all cases.

b) **Strict**: In order to avoid any possibility of incompatibility between wavefunction analysis codes and quantum chemistry programs, strict definition must be given for all parts of the format.

c) **Concise**: The format should be as concise and compact as possible to avoid waste of file size. Redundant information should be avoided as much as possible.

d) **Extensible**: There should be no limitation on the type and size of the content (*e.g.* angular moment of basis functions and number of atoms and orbitals). Besides, the file should allow to record additional information that not originally defined by the standard format, so that quantum chemistry program can pass special information to wavefunction analysis code for uncommon kind of analysis.

e) **Easy to read and write**: The format should be designed in a way that developers can easily write subroutines to efficiently read information from and write data to the file.

f) **Humanly readable**: The content of the file should be manually identifiable, so that researchers can easily add, delete and edit its content via common text editors.

To meet all of above requirements, combined with our development experience of wavefunction analysis code, we designed a new format named "mwfn" (acronym of "Multiwfn wavefunction file") for recording wavefunction information, as will be described in detail in the next section. The .mwfn file has already been supported as input file of Multiwfn since version 3.7, and Multiwfn can also export .mwfn file based on wavefunction loaded from other kinds of files (*e.g.* .fch and .molden). Due to the important advantages of "mwfn" format compared to the existing ones as well as the impact of Multiwfn code, we believe this format will be a new recognized standard for exchanging and storing wavefunction purposes.

2. Definition of "mwfn" format

Basics

The file suffix of "mwfn" format files must be ".mwfn".

This file totally consists of below five fields, only the last one is optional

1. System information
2. Atom information
3. Basis function information
4. Orbital information
5. (Optional) Various matrices or other information

There should be at least one blank line between two fields. Each field consists of one or more subfields, which can record either scalar or vector/matrix/list data. The appearance order of various fields and subfields is fixed.

The subfield recording scalar data occupies one line, the value is followed by the label, and there must be an equal sign at the end of the label. For example:

```
nindbasis=      25
```

The subfield recording vector/matrix/list data occupies multiple lines, the first line is the label with "$" at its beginning, the other lines record data. The number of values in each line is arbitrary. For example:

```
$Shell types
  0  0  1  0  1  2  0  0  1  0  0  1
```

All labels and string data are case sensitive. There must be at least one space between two data or between data and label.

Unless otherwise specified, all data are recorded in free format, therefore there is no limitation on record accuracy. Although there is no clear requirement on the maximal length of a line, it is highly suggested to record data in a proper format so that the line length does not exceed 80 characters.

Between different fields or between different subfields, comment lines can be inserted with "#" at the beginning of the line.

The standard .mwfn file is designed for recording the wavefunctions with Gaussian type functions (GTFs) as basis functions. Wavefunctions in other types of basis functions, such as numerical basis and Slater type orbital (STO) are not formally supported.

Definition of the five aforementioned fields is described below in turn. Dark blue and purple texts correspond to subfield name recording scalar and vector/matrix/list data, respectively. Additional fields or subfields may be supplemented to the .mwfn file according to special purposes as long as they do not break basic format requirements.

Field 1: System information

♦ **Wfntype** (integer, scalar): Wavefunction type. Possible values:

0: Restricted closed-shell single-determinant wavefunction (*e.g.* RHF, RKS)

1: Unrestricted open-shell single-determinant wavefunction (*e.g.* UHF, UKS)

2: Restricted open-shell single-determinant wavefunction (*e.g.* ROHF, ROKS)

3: Restricted multiconfiguration wavefunction (*e.g.* RMP2, RCCSD)

4: Unrestricted multiconfiguration wavefunction (*e.g.* UMP2, UCCSD)

♦ **Charge** (float, scalar): Net charge of the system

♦ **Naelec** (float, scalar): Number of alpha electrons in the system

♦ **Nbelec** (float, scalar): Number of beta electrons in the system

Note that since the above three subfields are float types, it is possible to record wavefunction generated by partial occupation calculation.

♦ **E_tot** (float, scalar): Electronic energy of the system. It should be 0.0 if it is not available

♦ **VT_ratio** (float, scalar): Virial ratio (-V/T) of the system. It should be 0.0 if it is not available

Field 2: Atom information

♦ **Ncenter** (integer, scalar): Number of centers in the system. Commonly, the center refers to real atom, however it can also be dummy atom

♦ **$Centers** (*Ncenter* elements): Each line in this subfield corresponds to information of a center, for example water:

```
a b   c   d        e            f            g
1 O   8   8.0    0.00000000    0.00000000    0.11976353
2 H   1   1.0   -0.00000000    0.76149590   -0.47905410
3 H   1   1.0   -0.00000000   -0.76149590   -0.47905410
```

The meaning of different columns:

a: Center index, must start from 1

b: Center name. For real atoms, it corresponds to element name, the first character should be in uppercase, the second character should be in lowercase. The name should be in line with the NIST periodic table.[8] The name for dummy atom should be "X"

c: Element index in periodic table. For dummy atom, this index should be 0

d: Nuclear charge. If for an atom effective core potential (ECP) was not employed, this value should correspond to element index. If ECP was employed, this value should correspond to the number of electrons that not represented by ECP. For dummy atom, this value should be 0.0

e, f, g: X, Y, Z coordinates of the center, respectively. The unit is Å

Field 3: Basis function information

♦ **Nbasis** (integer, scalar): Number of basis functions

♦ **Nindbasis** (integer, scalar): Number of linearly independent basis functions. Most quantum chemistry programs automatically eliminate linearly dependent basis functions when linearly dependent problem is detected, which is usually caused by use of large amount of diffuse functions

♦ **Nprims** (integer, scalar): Number of primitive functions

♦ **Nshell** (integer, scalar): Number of basis function shells

♦ **Nprimshell** (integer, scalar): Number of primitive shells

♦ **$Shell types** (integer; *Nshell* elements): Types of basis function shells. 0/1/2/3/4... correspond to S/P/D/F/G..., there is no upper limit. For shells with angular moment higher than P, there are two cases: Index ≥ 2 corresponds to Cartesian shell, index ≤ 2 corresponds to pure shell (also known as spherical-harmonic shell). Shell type of -1 is undefined, and the SP shell utilized by *e.g.* Gaussian is not formally supported

♦ **$Shell centers** (integer; *Nshell* elements): Center indices of basis function shells

♦ **$Shell contraction degrees** (integer; *Nshell* elements): Contraction degrees of basis function shells

♦ **$Primitive exponents** (float; *Nprimshell* elements): Exponents of primitive shells in basis function shells

♦ **$Contraction coefficients** (float; *Nprimshell* elements): Contraction coefficients of primitive shells in basis function shells

Note on order of basis functions

It is of great importance to explicitly define the order of different types of basis functions in each type of shell.

For Cartesian shells, the order is in line with the .fch file of Gaussian. The case of the first few types of shells are given below

➢ S shell: S
➢ P shell: X, Y, Z
➢ D shell: XX, YY, ZZ, XY, XZ, YZ
➢ F shell: XXX, YYY, ZZZ, XYY, XXY, XXZ, XZZ, YZZ, YYZ, XYZ
➢ G shell: ZZZZ, YZZZ, YYZZ, YYYZ, YYYY, XZZZ, XYZZ, XYYZ, XYYY, XXZZ, XXYZ, XXYY, XXXZ, XXXY, XXXX
➢ H shell: ZZZZZ, YZZZZ, YYZZZ, YYYZZ, YYYYZ, YYYYY, XZZZZ, XYZZZ, XYYZZ, XYYYZ, XYYYY, XXZZZ, XXYZZ, XXYYZ, XXYYY, XXXZZ, XXXYZ, XXXYY, XXXXZ, XXXXY, XXXXX

the order of other types of shells can be easily derived according to the rule of H shells.

For pure shells, the order is
➢ D shell: D 0, D+1, D-1, D+2, D-2
➢ F shell: F 0, F+1, F-1, F+2, F-2, F+3, F-3
➢ G shell: G 0, G+1, G-1, G+2, G-2, G+3, G-3, G+4, G-4

similar for other types of shells. Transformation between basis functions in Cartesian shell and those in pure shell is discussed in Ref. [9]. For easy programming, the tables containing transformation coefficients for S, P, D, F, G, H shells are explicitly provided in the supplemental material.

Field 4: Orbital information

This field records information of all orbitals. For restricted wavefunctions (Wfntype=0, 2, 3), there should be *Nindbasis* orbitals, while for unrestricted wavefunctions (Wfntype=1, 4), there should be 2*Nindbasis orbitals.

All orbitals in this file should be recorded consecutively, and there must be one and only one blank line between information of two neighbouring orbitals. Definitions of all subfields for each orbital are shown below, the order of appearance should not be altered.

♦ **Index** (integer, scalar): Orbital index, starts from 1. Beware that the value must occupy 10 characters, for example: `Index=□□□□□□□□33`. This rule allows program quickly locates to the reading position of an orbital of interest. Evidently, to locate to the beginning of the entire orbital information field, one simply needs to locate to the line containing `Index=□□□□□□□□□1`

♦ **Type** (integer, scalar): Orbital type
0: Alpha+Beta (*i.e.* spatial orbital)
1: Alpha
2: Beta

♦ **Energy** (float, scalar): Orbital energy in Hartree

♦ **Occ** (float, scalar): Orbital occupation number. This field can also be used to record eigenvalues of natural transition orbitals (NTO),[10] natural adaptive orbitals[11] and so on.

Note that in any case that there is only one set of spatial orbitals with partial occupation numbers, the *Wfntype* should be 3, while in any case that there are Alpha and Beta spin orbitals with partial occupation numbers, the *Wfntype* should be 4. Thus, for example, if present file contains NTOs for a closed-shell system, the *Wfntype* should be set to 3, since eigenvalues of NTOs are non-integer and they are recorded in "Occ" subfield.

♦ **Sym** (string, scalar): Irreducible representation of orbital. Only the first character should be capitalized. No number should appear at the beginning. For example, "A1", "B1u", "T2g", "E" are acceptable, while "a1" and "45B1" are inacceptable. If irreducible representation is not available, the value should simply be "?"

♦ **$Coeff** (float; *Nbasis* elements): Expansion coefficients of all basis functions in orbital

Note on order of orbitals

Although there is no mandatory requirement on the order of orbitals, for orbitals with definite occupation number (*e.g.* molecular orbitals, localized molecular orbitals[12] and natural orbitals), below convention should be followed if there is no special reason. Note that the aforementioned *Index* property is shared by all orbitals, thus for unrestricted wavefunction, the *i*th Alpha orbital has index of *i*, while the *i*th Beta orbital has index of *Nindbasis*+*i*.

- Restricted closed-shell wavefunction (Wfntype=0)
    [Doubly occupied Alpha+Beta orbitals]
    [Unoccupied Alpha+Beta orbitals]
- Unrestricted open-shell wavefunction (Wfntype=1)
    [Singly occupied Alpha orbitals]
    [Singly occupied Beta orbitals]
- Restricted open-shell wavefunction (Wfntype=2)

      [Doubly occupied Alpha+Beta orbitals]

      [Singly occupied Alpha orbitals]

      [Unoccupied Alpha+Beta orbitals]

- Restricted multiconfiguration wavefunction (Wfntype=3)

      [Alpha+Beta orbitals] (occupation numbers are sorted from high to low)

- Unrestricted multiconfiguration wavefunction (Wfntype=4)

      [Alpha orbitals] (occupation numbers are sorted from high to low)

      [Beta orbitals] (occupation numbers are sorted from high to low)

Field 5: Various matrices

This field is fully optional, it is used to record various matrices that may be used in certain wavefunction analyses. Standard names of common matrices in .mwfn file are shown below, all the listed matrices are symmetric with float type of data, the dimensions are all *Nbasis×Nbasis*.

    $Total density matrix

    $Alpha density matrix

    $Beta density matrix

When density matrix is to be recorded, for restricted closed-shell wavefunctions (Wfntype=0 and =3), only "Total density matrix" should be presented; while for other types of wavefunctions, only "Alpha density matrix" and "Beta density matrix" should be simultaneously given.

    $1-e Hamiltonian matrix

    $Alpha 1-e Hamiltonian matrix

    $Beta 1-e Hamiltonian matrix

Effective single-electron Hamiltonian matrix, such as Fock matrix and Kohn-Sham matrix. When this kind of matrix is to be recorded, for restricted closed-shell and restricted open-shell wavefunctions (Wfntype=0, =2 and =3), only "1-e Hamiltonian matrix" should be presented; while for unrestricted wavefunctions, only " Alpha 1-e Hamiltonian matrix" and "Beta 1-e Hamiltonian matrix " should be simultaneously given.

    $Overlap matrix: (i, j) element is defined as $\langle \chi_i | \chi_j \rangle$, where χ denotes basis function, similarly hereinafter

    $Kinetic energy matrix: (i, j) element is defined as $\langle \chi_i | -\frac{1}{2}\nabla^2 | \chi_j \rangle$

    $Potential energy matrix: (i, j) element is defined as $\langle \chi_i | \sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} | \chi_j \rangle$, where Z_A and $\mathbf{R}_A$ are nuclear charge and coordinate of atom A, respectively; $\mathbf{r}$ is integral coordinate

    $X electric dipole moment matrix: (i, j) element is defined as $\langle \chi_i | -x | \chi_j \rangle$

    $Y electric dipole moment matrix: (i, j) element is defined as $\langle \chi_i | -y | \chi_j \rangle$

    $Z electric dipole moment matrix: (i, j) element is defined as $\langle \chi_i | -z | \chi_j \rangle$

where x, y, z are three Cartesian components of position vector $\mathbf{r}$.

For saving space, all symmetric matrices should be recorded in lower triangular form. For example, the record order of the elements of the matrices mentioned above should be (1,1), (2,1), (2,2), (3,1), (3,2), (3,3) ... (*Nbasis, Nbasis*), and thus totally *Nbasis×(Nbasis+1)/2* values.

For clarity, additional descriptions should be added after matrix label. For example, overlap matrix should be recorded as

```
$Overlap matrix, dim=   19   19  lower= 1
   1.00000000E+00   2.33689858E-01   1.00000001E+00   7.77761507E-50   4.26647475E-49
   9.99999999E-01   1.73004161E-49   3.73334539E-49   3.96633889E-97   9.99999999E-01
   ... [ignored]
```

where the two values after "dim=" are numbers of rows and columns, respectively. The 1 after "lower=" explicitly indicates that the matrix is recorded in lower triangular form.

For a general, non-symmetric matrix with N_R rows and N_C columns, its elements must be recorded in the following order,

(1,1), (1,2) ... (1,N_C), (2,1), (2,2) ... (2,N_C), (3,1), (3,2) ... (3,N_C) ...... (N_R,1), (N_R,2) ... (N_R,N_C)

and the two values after "dim=" should be N_R and N_C, respectively, the value after "lower=" should be 0.

Sanity test

In order to confirm that a .mwfn file satisfies aforementioned format requirements and the content is reasonable, performing sanity test using Multiwfn is highly recommended. There are two useful ways to check sanity (the matrices recorded in field 5 are not taken into account):

(1) Load the .mwfn file into Multiwfn, enter main function 100 and select subfunction 4, then choose "Electron density" as integrand. Electron density over the whole space will be calculated based on Becke's multicenter integration method.[13] If the result is very close to the actual number of electrons, that means the .mwfn file is reasonable. (In rare cases, especially when there are very heavy atoms, the result may deviate from integer detectably even if the .mwfn file is fully reasonable, in this case one can enlarge "sphpot" and "radpot" parameters in "settings.ini" file of Multiwfn to respectively increase number of angular and radial integration points, and then check if the result could be improved)

(2) Load the .mwfn file into Multiwfn, enter main function 1000 (a hidden function) and select subfunction 100. This function will check satisfaction of normalization condition of all orbitals, and show you the maximum deviation to unity and integer. If both the values are evidently larger than zero, then there must be some severe problems in the .mwfn file.

Example files

Four sample .mwfn files are provided in supplemental material, their descriptions are given in the "README.txt" document. These files can be used to better understand the format definition and check if your own code is able to normally load files in "mwfn" format.

One can also use Multiwfn to generate .mwfn file based on wavefunction information loaded from other kinds of files. To do so, a .fch or .molden or .gms file (renamed GAMESS-US or Firefly output file) should be loaded into Multiwfn, then enter subfunction 2 of main function 100, one will find an option used to export current wavefunction to .mwfn file.

Appendix: Overview and comparison of formats for recording wavefunction

Basic characters of all well-known formats that carry wavefunction information are collected in below table for comparison.

File format	Contained information									
	Basis functions	PGTFs[a]	Atoms coordinates	Nuclear charges	Unocc. orbitals[b]	Orbital spins	Orbital occupations	Orbital IRREP[c]	Density matrix	Fock/KS matrix
wfn	×	√	√	√	×	×	√	×	×	×
wfx	×	√	√	√	×	√	√	×	×	×
fch/fchk	√	√	√	√	√	√	×	×	√	×
molden	√	√	√	×	√	√	√	√	×	×
NBO .47	√	√	√	√	√	×	×	×	√	(√)[d]
mkl	√	√	√	×	√	√	√	√	×	×
mwfn	√	√	√	√	√	√	√	√	(√)[c]	(√)[d]

[a] Primitive Gaussian type functions. Note that this kind of information can be converted from basis function information

[b] Unoccupied orbitals

[c] Irreducible representation

[d] Optional information

The formats in the above table are briefly introduced below.

● **AIM wavefunction file (.wfn)**: This is a very old format firstly introduced by Bader's atoms-in-molecules analysis program AIMPAC, and currently supported by a lot of mainstream quantum chemistry softwares, such as Gaussian, ORCA, GAMESS-US, Firefly, Q-Chem and NWChem. The information in .wfn file include atomic indices/names/coordinates/charges, orbital energies, occupation numbers, orbital expansion coefficients with respect to Cartesian primitive Gaussian type functions (PGTFs). Despite its popularity, its shortcomings are quite obvious:

(a) Angular momentum of PGTFs is only formally supported up to f. Therefore, such as wavefunctions generated under cc-pVQZ[14] or def2-QZVP[15] basis set for organic systems cannot be recorded in this format.

(b) Generally this format does not contain any virtual orbital. Therefore it is impossible to perform many kinds of analyses based on this file, such as the hole-electron analysis[16,17] proposed by us, which inevitably involves virtual orbitals. However, we note that in Gaussian, it is possible to export all unoccupied orbitals to this file by using *IOp(99/18=-1)* keyword.

(c) Orbital spins are not explicitly recorded, therefore analysis program must guess spins in open-shell cases. However, the guessed spins may be sometimes wrong and thus lead to misleading result.

(d) Basis function information is not included in this format, therefore many commonly used analyses, such as Mulliken population and orbital composition analysis,[18,19] Mayer bond order analysis[20] and charge decomposition analysis[21,22] cannot be performed.

● **AIM extended wavefunction file (.wfx)**: This format is defined as an extension of the .wfn format, it was firstly supported by Gaussian 09 since B.01 revision. .wfx file supports infinitely high angular moment of PGTFs and explicitly records orbital spins, thus fully removes the aforementioned drawbacks (a) and (c) of the .wfn format. Unfortunately, due to overusing XML style tags, the .wfx format is no longer as compact as the .wfn format.

A special point of the .wfx format is the newly introduced electron density function (EDF) field,[23] which employs multiple GTFs with fitted parameters to approximately mimic the density of the core electrons represented by effective core potential (ECP). This feature makes result of electron density analysis for the wavefunction using ECP very close to the that for full electron wavefunction. Although the EDF field is valuable and included in the .wfx format, we decide not to make our "mwfn" format explicitly support this feature, the reason is that there has been a well constructed and open-source EDF library,[24] which has been embedded into Multiwfn and can also be easily incorporated into other analysis codes. When Multiwfn loads a file containing wavefunction generated by a calculation employing ECP, Multiwfn will automatically extract proper EDF data from the library and assign them to the atoms using ECP. Therefore we believe there is no necessity to explicitly record EDF information in wavefunction file.

● **Gaussian formatted checkpoint file (.fch or .fchk)**: Binary checkpoint file of Gaussian program (.chk) can be converted to formatted checkpoint file (.fch or .fchk) via "formchk" utility in Gaussian package. Current version of Q-Chem and PSI4 programs can also export this kind of file. This format contains richer information than the .wfn and .wfx files, namely not only basis function information is recorded, but also virtual orbitals are available. Moreover, this format contains density matrix, which is useful in some analyses. In our opinion, the most evident drawback of this format for wavefunction analysis purpose is that it does not contain a field specific for recording orbital occupation numbers, this brings a great inconvenience in performing wavefunction for multiconfiguration wavefunctions, which are commonly represented as a set of natural orbitals, whose occupation numbers are non-integer and thus needed to be explicitly recorded.

● **Molden input file (.molden)**: This is input file of a historic visualization program Molden,[25] which was developed since the early 1990s. A wide variety of quantum chemistry packages, such as Molpro, Molcas, ORCA, Q-Chem, CFour, Turbomole, PSI4, MRCC and NWChem are able to produce this kind of file. This format records atomic information, definition of basis functions, information of all occupied and virtual orbitals (including expansion coefficients with respect to basis functions, occupation number, spin, energy and symmetry), meanwhile there is no information merely specific for Molden program. Due to this fact, it can be regarded as a general file format for carrying and exchanging wavefunction purposes. Unfortunately, there are a few prominent issues in this format:

(a) Nuclear charges are not recorded. Therefore when ECP is employed, analysis programs are unable to correctly identify actual nuclear charges, leading to completely wrong result of many analyses in which nuclear charges are involved, such as electrostatic potential analysis and atomic charge calculation.

(b) .molden format was not defined in a very strict way, as a consequence, although lots of quantum chemistry codes are able to generate this kind of file, the actual format varies from program to program. In fact, we found only very few programs are able to generate .molden file in a very standard form, the data in the .molden file exported by some programs are even

wrong. This evident issue makes developers of wavefunction analysis codes hardly to write a routine that fully compatible with .molden files generated by all programs. In contrast, our .mwfn format presents clear definition for every part of the file; furthermore, we provided Multiwfn code with special functions for checking the correctness of the file, thus the potential of creating non-standard .mwfn file is completely avoided.

(c) The data in .molden file is not as compact as .fch file. Due to this reason, despite these two kinds of files contain similar information, speed of loading wavefunction from .molden file must be slower than that from .fch file.

(d) .molden format only formally supports basis functions up to g angular moment.

● **NBO .47 file**: This is private input file format of stand-alone version of NBO code (GENNBO). It contains atom information and basis function information. In addition, many matrices such as Fock matrix, overlap matrix, LCAO matrix, dipole moment integral matrix are recorded in this file, which are needed by many analysis methods defined in the natural bond orbital (NBO) framework.[26]

● **Molekel file (.mkl)**: This file can be read by old version of Molekel visualization program, it carries similar information as .molden file. The "orca_2mkl" utility in ORCA package is able to generate this kind of file.

Finally, it is worth to note that until Multiwfn version 3.7, all aforementioned files except for .47 and .mkl can be loaded by Multiwfn for wavefunction analysis purpose, and all aforementioned files can be exported by subfunction 2 of main function 100 of Multiwfn. Relevant reading and writing Fortran code can be found in "fileIO.f90" file in Multiwfn source code package. Specifically, the routine name for reading and writing .mwfn file is "readmwfn" and "outmwfn", respectively.

References

[1]   Definition of fch format. http://gaussian.com/interfacing/ (accessed on Feb 18, 2020).

[2]   The Molden Format. http://cheminf.cmbi.ru.nl/molden/molden_format.html (accessed on Feb 18, 2020).

[3]   Format specification for AIM Extended Wavefunction Files (.wfx files). http://aim.tkgristmill.com/wfxformat.html (accessed on Feb 18, 2020).

[4]   M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, Williams, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, D. J. Fox, Gaussian 16. Wallingford, CT, **2016**.

[5]   D. Y. Zubarev, A. I. Boldyrev, Developing paradigms of chemical bonding: adaptive natural density

partitioning. *Phys. Chem. Chem. Phys.* **2008**, *10*, 5207.

[6] Website of Multiwfn program. http://sobereva.com/multiwfn (accessed on Feb 18, 2020).

[7] T. Lu, F. Chen, Multiwfn: A Multifunctional Wavefunction Analyzer. *J. Comput. Chem.* **2012**, *33*, 580.

[8] NIST Periodic Table of the Elements. https://www.nist.gov/pml/periodic-table-elements (accessed on Feb 18, 2020).

[9] H. B. Schlegel, M. J. Frisch, Transformation between Cartesian and pure spherical harmonic Gaussians. *Int. J. Quantum Chem.* **1995**, *54*, 83.

[10] R. L. Martin, Natural transition orbitals. *J. Chem. Phys.* **2003**, *118*, 4775.

[11] J. L. Casals-Sainz, A. Fernández-Alarcón, E. Francisco, A. Costales, Á. Martín Pendás, Bond Order Densities in Real Space. *J. Phys. Chem. A* **2020**, *124*, 339.

[12] T. Lu, Q. Chen, A simple method of identifying π orbitals for non-planar systems and a protocol of studying π electronic structure. *Theor. Chem. Acc.* **2020**, *139*, 25.

[13] A. D. Becke, A Multicenter Numerical Integration Scheme for Polyatomic Molecules. *J. Chem. Phys.* **1988**, *88*, 2547.

[14] J. T. H. Dunning, Gaussian Basis Sets for Use in Correlated Molecular Calculations. I. The Atoms Boron through Neon and Hydrogen. *J. Chem. Phys.* **1989**, *90*, 1007.

[15] F. Weigend, R. Ahlrichs, Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297.

[16] T. Lu, Multiwfn Manual, version 3.7(dev), Section 3.21.1, available at http://sobereva.com/multiwfn   (accessed on Feb 18, 2020).

[17] Z. Liu, T. Lu, Q. Chen, An sp-hybridized all-carboatomic ring, cyclo[18]carbon: Electronic structure, electronic spectrum, and optical nonlinearity. *Carbon* **2020**, *165*, 461.

[18] T. Lu, F. Chen, Comparison of Computational Methods for Atomic Charges. *Acta Phys. -Chim. Sin.* **2012**, *28*, 1.

[19] T. Lu, F. Chen, Calculation of Molecular Orbital Composition. *Acta Chim. Sinica* **2011**, *69*, 2393.

[20] I. Mayer, Charge, Bond Order and Valence in the AB initio SCF Theory. *Chem. Phys. Lett.* **1983**, *97*, 270.

[21] S. Dapprich, G. Frenking, Investigation of Donor-Acceptor Interactions: A Charge Decomposition Analysis Using Fragment Molecular Orbitals. *J. Phys. Chem.* **1995**, *99*, 9352.

[22] M. Xiao, T. Lu, Generalized charge decomposition analysis (GCDA) method. *J. Adv. Phys. Chem* **2015**, *4*, 111.

[23] T. A. Keith, M. J. Frisch, Subshell Fitting of Relativistic Atomic Core Electron Densities for Use in QTAIM Analyses of ECP-Based Wave Functions. *J. Phys. Chem. A* **2011**, *115*, 12879.

[24] W. Zou, Z. Cai, J. Wang, K. Xin, An open library of relativistic core electron density function for the QTAIM analysis with pseudopotentials. *J. Comput. Chem.* **2018**, *39*, 1697.

[25] G. Schaftenaar, J. H. Noordik, Molden: a pre-and post-processing program for molecular and electronic structures. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 123.

[26] F. Weinhold, Natural Bond Orbital Methods. In Encyclopedia of Computational Chemistry 2, P. v. R. Schleyer, Ed., John Wiley & Sons: West Sussex, **1998**, p. 1792.