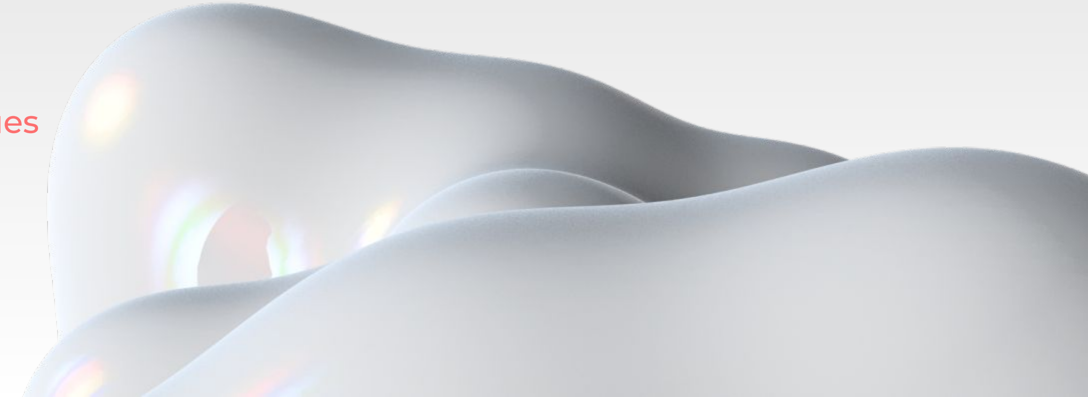
Portable wave functions with TREXIO

Evgeny Posenitskiy[*], Anthony Scemama[**]

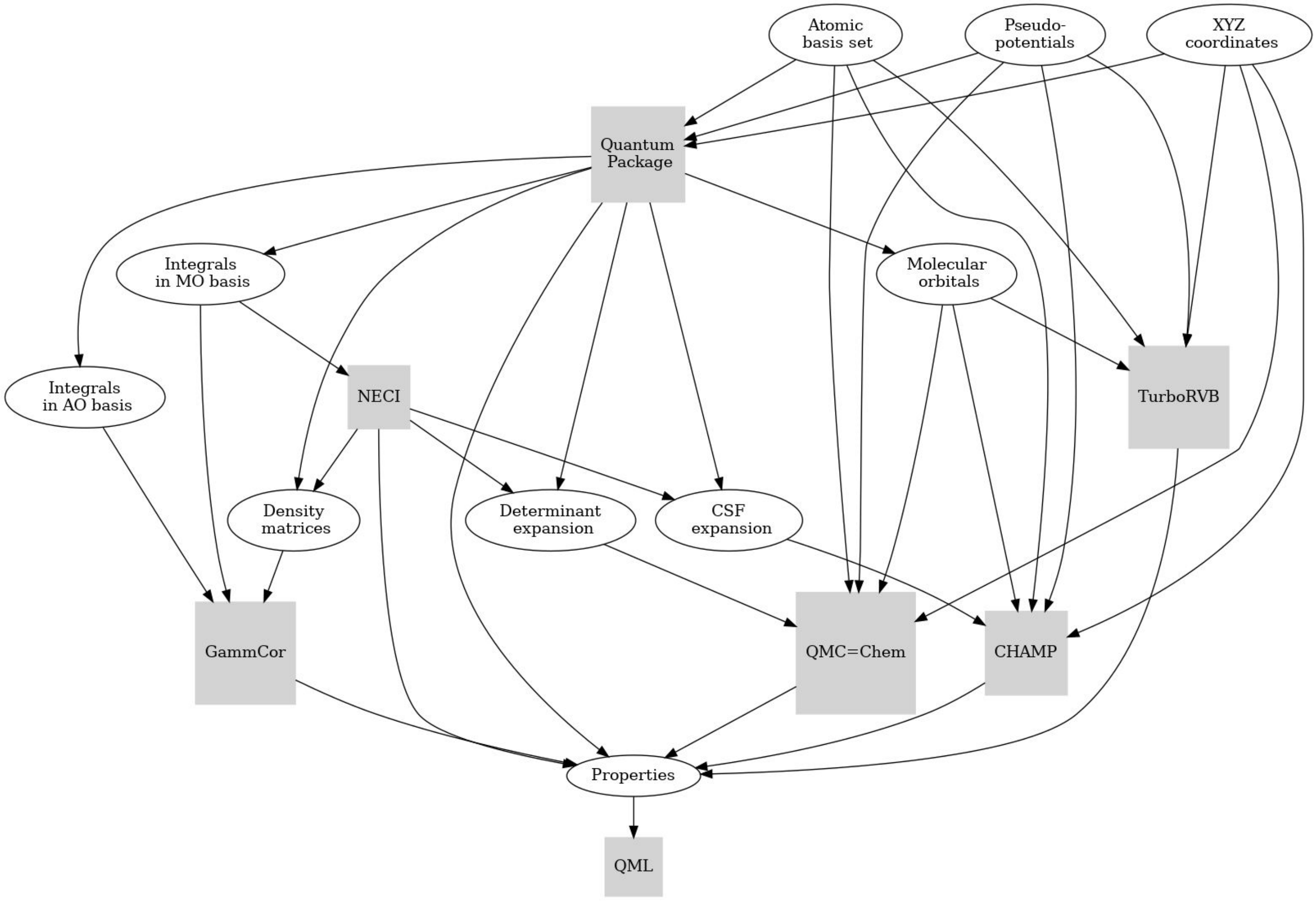
[*] Qubit Pharmaceuticals | Paris, France
[**] Laboratoire de Chimie et Physique Quantiques (LCPQ) @ CNRS/UPS | Toulouse, France

"Bridging Quantum Monte Carlo and High-Performance Simulations" Symposium
February 7, 2024

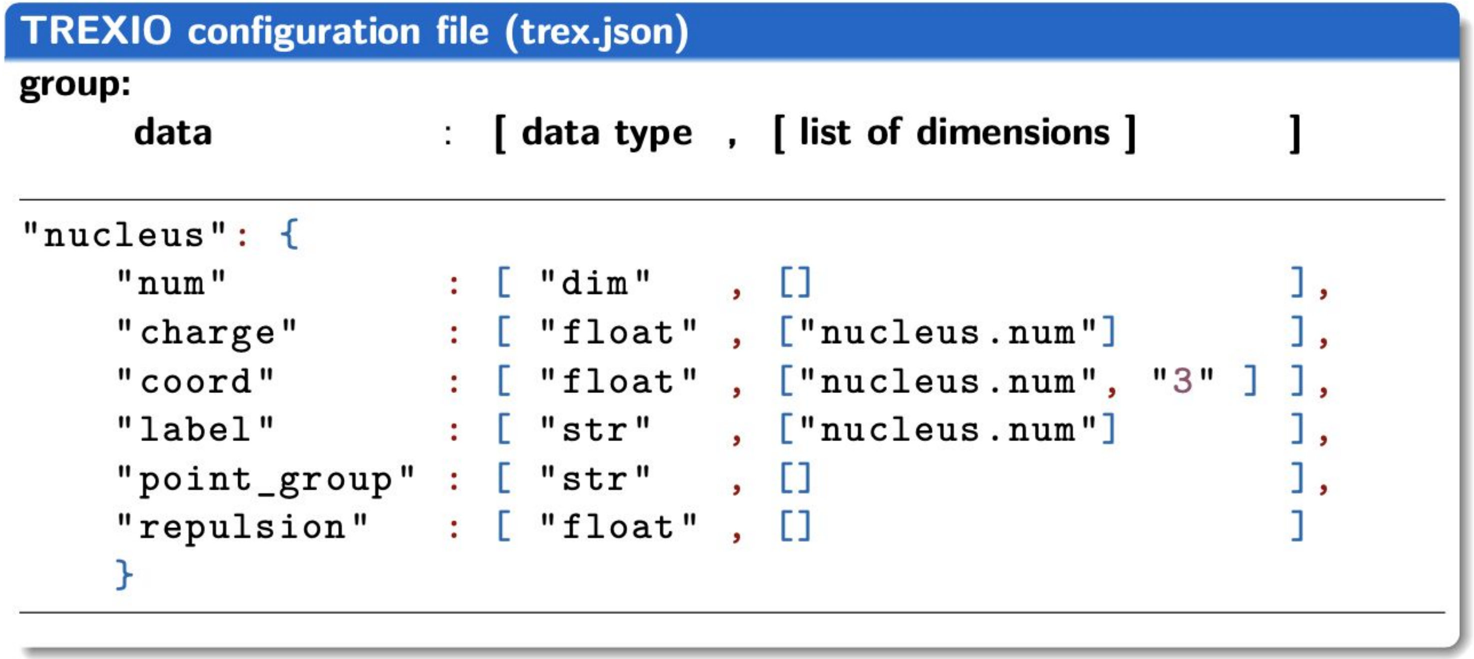


TREXIO as I/O format

Back in 2020



Atomic basis set
Pseudo-potentials
XYZ coordinates
Quantum Package
Integrals in MO basis
Molecular orbitals
Integrals in AO basis
NECI
TurboRVB
Density matrices
Determinant expansion
CSF expansion
GammCor
QMC=Chem
CHAMP
Properties
QML



TREXIO configuration file (trex.json)

group:

```
    data          : [ data type , [ list of dimensions ]        ]
```

```
"nucleus": {
    "num"         : [ "dim"    , []                         ],
    "charge"      : [ "float"  , ["nucleus.num"]            ],
    "coord"       : [ "float"  , ["nucleus.num", "3" ]      ],
    "label"       : [ "str"    , ["nucleus.num"]            ],
    "point_group" : [ "str"    , []                         ],
    "repulsion"   : [ "float"  , []                         ]
    }
```

More details in the TREXIO documentation*

* *https://trex-coe.github.io/trexio/trex.html*

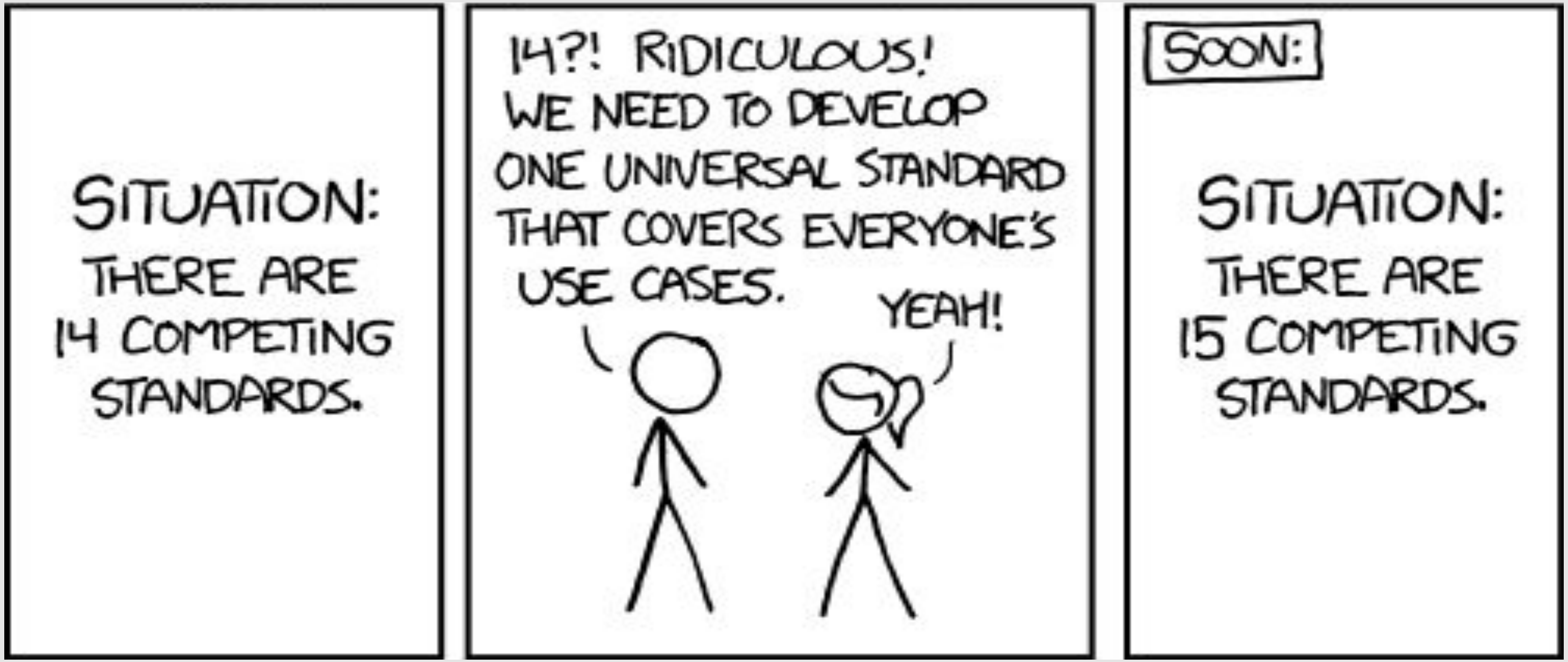

SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Image source: https://imgs.xkcd.com/comics/standards.png

Enhancements compared to other wave function formats:

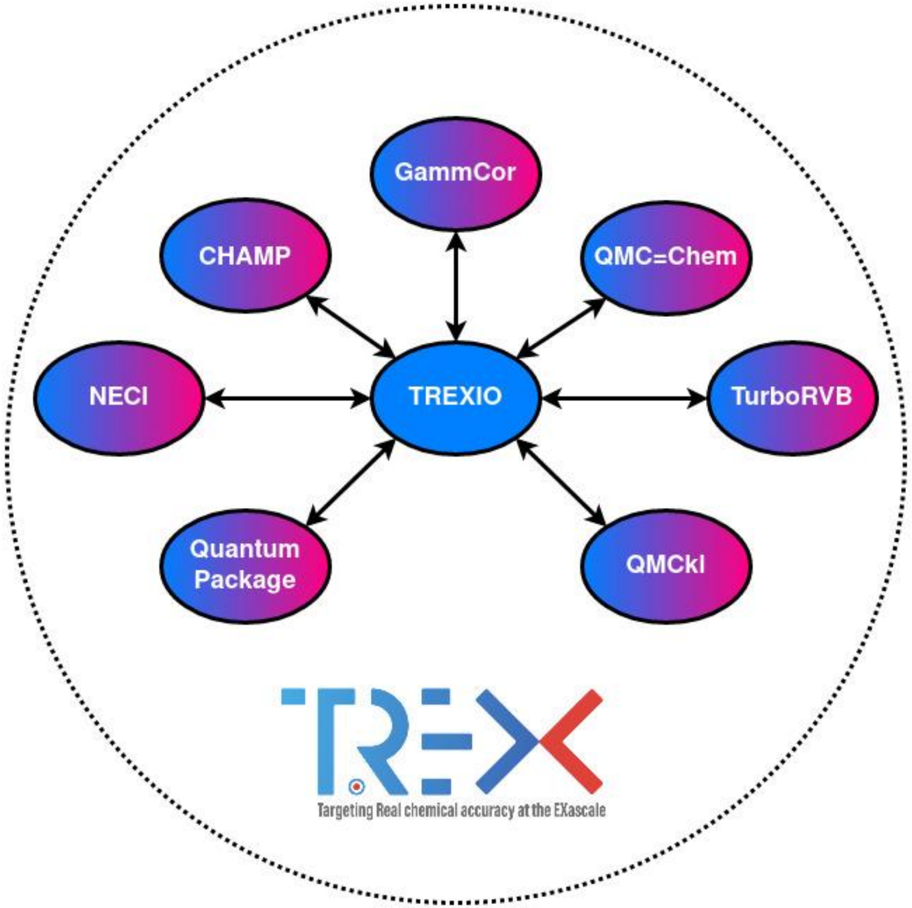
- Fully self-consistent, i.e. no external (code-specific) knowledge is required
- Exhaustive list of normalization parameters to cover existing ambiguities
- AOs support for Cartesian, spherical and numerical representations
- Compact storage of quantities like 2e integrals and CI determinants

Enhancements compared to other wave function formats:

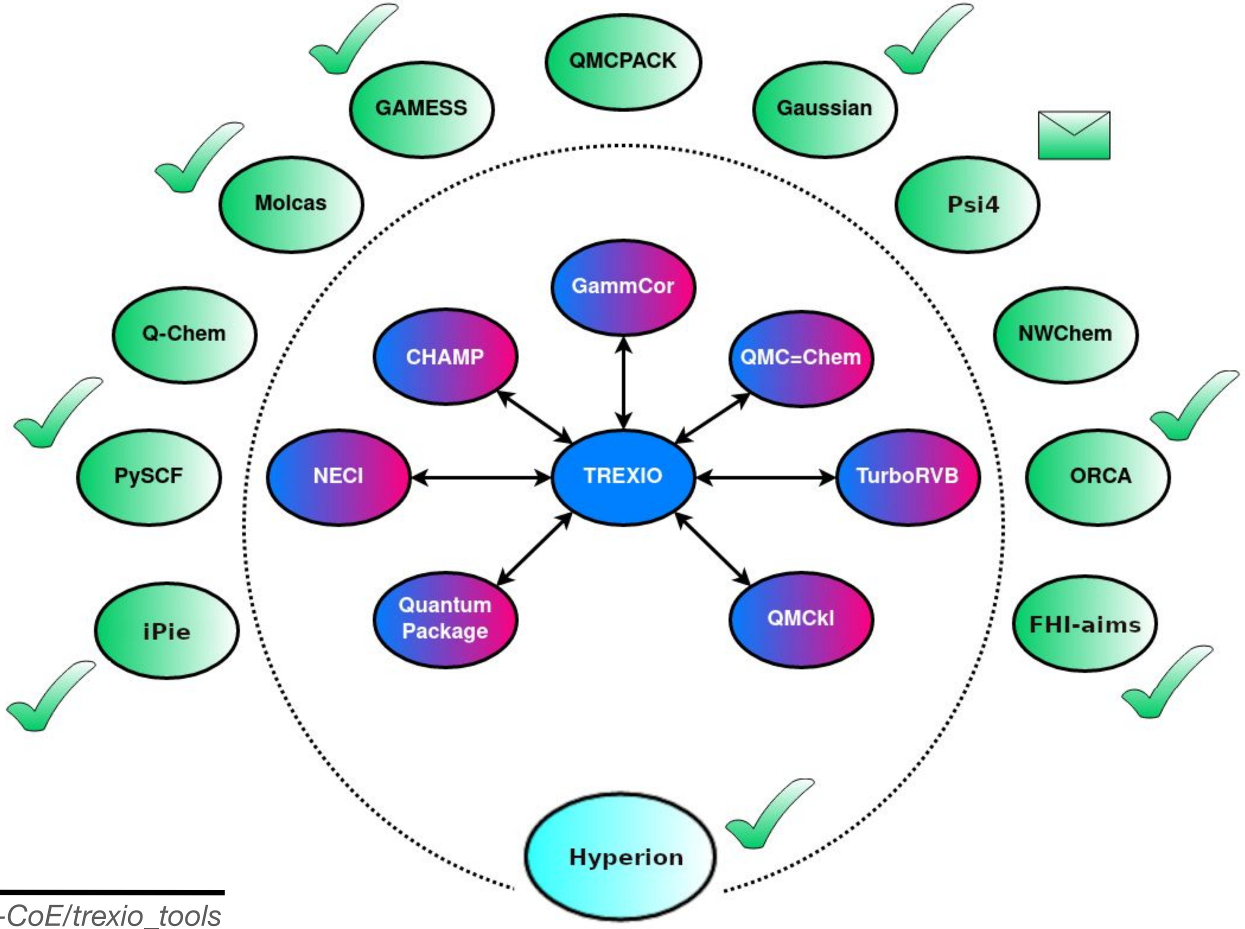
- Fully self-consistent, i.e. no external (code-specific) knowledge is required
- Exhaustive list of normalization parameters to cover existing ambiguities
- AOs support for Cartesian, spherical and numerical representations
- Compact storage of quantities like 2e integrals and CI determinants

- **No custom text-based formatting** - forget about the typos!

Today

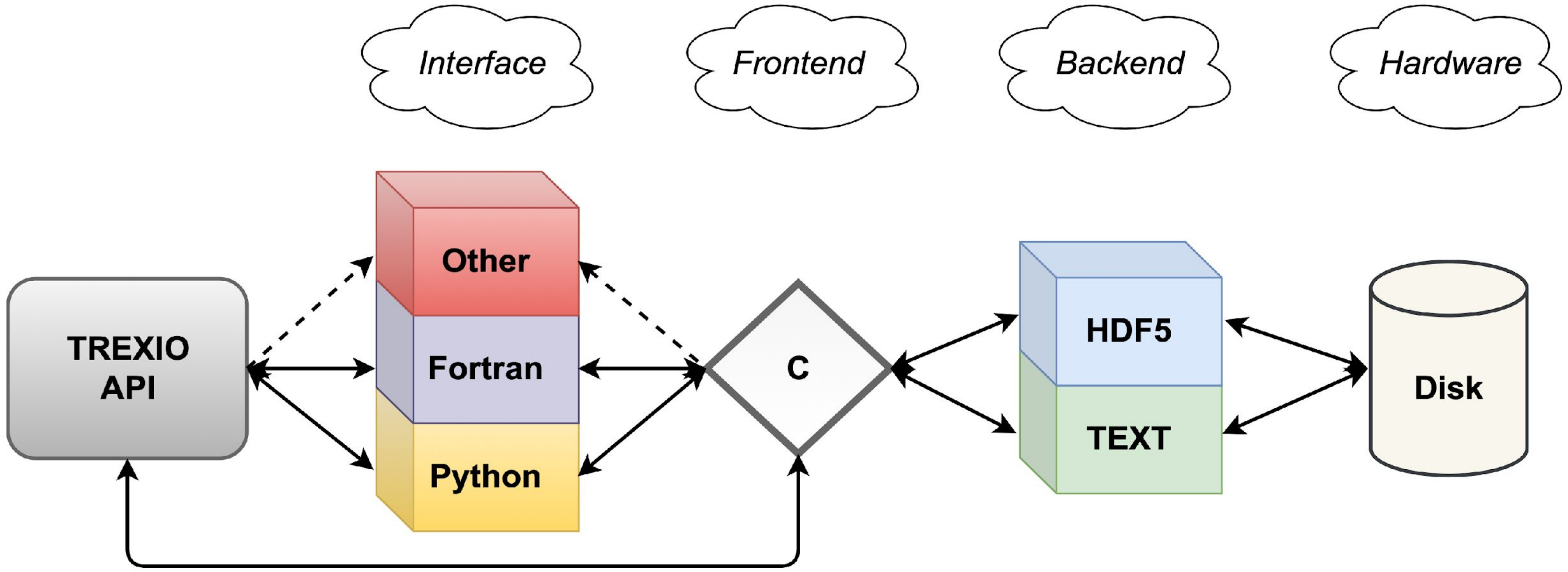

GammCor
CHAMP
QMC=Chem
NECI
TREXIO
TurboRVB
Quantum Package
QMCkl
TREX
Targeting Real chemical accuracy at the EXascale

Today



https://github.com/TREX-CoE/trexio_tools

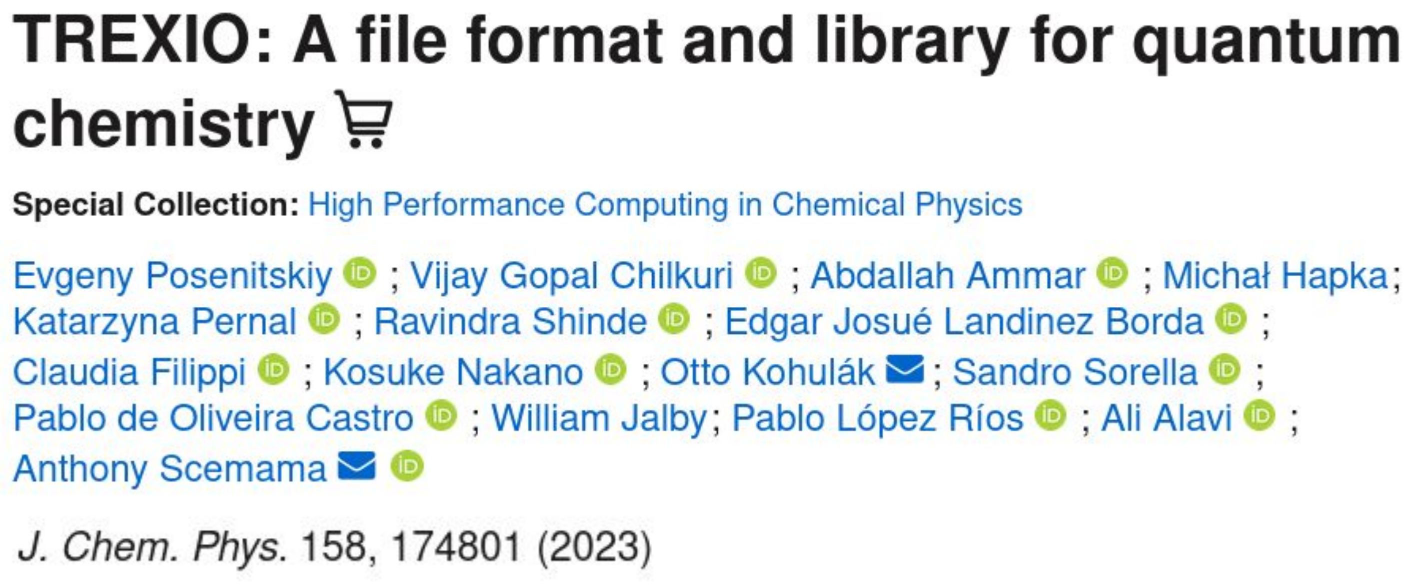
TREXIO as I/O library



- Source code in pure **C** (C99) for the best performance and portability
- Interfaces in **Fortran** (ISO_C_BINDING), **Python**, **OCaml**, **Rust**, **Julia**
- TREXIO passed all tests on **20 different hardware** architectures of the Debian build farm

https://github.com/TREX-CoE/trexio

https://arxiv.org/abs/2302.14793



TREXIO: A file format and library for quantum chemistry

Special Collection: High Performance Computing in Chemical Physics

Evgeny Posenitskiy ; Vijay Gopal Chilkuri ; Abdallah Ammar ; Michał Hapka; Katarzyna Pernal ; Ravindra Shinde ; Edgar Josué Landinez Borda ; Claudia Filippi ; Kosuke Nakano ; Otto Kohulák ; Sandro Sorella ; Pablo de Oliveira Castro ; William Jalby; Pablo López Ríos ; Ali Alavi ; Anthony Scemama

J. Chem. Phys. 158, 174801 (2023)

https://doi.org/10.1063/5.0148161

https://arxiv.org/abs/2302.14793

Conclusion

TREXIO format

- Flexible and fully **self-consistent representation**
- Programmatic access, **no need to learn new** keywords or text formatting

TREXIO format

- Flexible and fully **self-consistent representation**
- Programmatic access, **no need to learn new** keywords or text formatting

TREXIO library

- Portable and **easy to install** (Autotools/CMake, conda, Spack, Guix, pip)
- **High-performance I/O** back end based on the HDF5 library
- TREXIO tools: **out-of-the-box interface with community codes**

Adoption of TREXIO enabled

- Elimination of the I/O bottleneck in all TREX-CoE flagship codes
- QP ➪ TREXIO ➪ GammCor : SAPT with CIPSI density matrices
- QP ➪ TREXIO ➪ CHAMP : QMC with CIPSI wave functions
- QP ➪ TREXIO ➪ iPie : Auxiliary Field QMC with CIPSI wave functions
- QP ➪ TREXIO ➪ Hyperion : Quantum Computing (VQE) with CIPSI wave functions
- FHI-aims ➪ TREXIO : Quantum Chemistry with numerical orbitals
- trexio_tools ➪ TREXIO : free out-of-the-box interface with GAMESS, Gaussian, [Open]Molcas, PySCF, ORCA, FHI-aims

Acknowledgements

- **TREX-CoE**
- Kosuke Nakano (PySCF converter)
- Vijay Gopal Chilkuri (ORCA converter)
- Johannes Jan Guenzl (FHI-aims integration)

- **Qubit Pharmaceuticals**
- Diata Traore (Hyperion interface)
- César Feniou (VQE integration)

- TREXIO repository : https://github.com/TREX-CoE/trexio
- TREXIO helper tools : https://github.com/TREX-CoE/trexio_tools
- Documentation : https://trex-coe.github.io/trexio

Thank you for your attention!

```fortran
use trexio
integer(trexio_exit_code) :: rc
integer(trexio_t)         :: fhandle

fhandle = trexio_open(file_name, 'w', TREXIO_HDF5, rc)
call trexio_assert(rc, TREXIO_SUCCESS)
rc = trexio_write_nucleus_num(fhandle, 12)
call trexio_assert(rc, TREXIO_SUCCESS)
rc = trexio_close(fhandle)
call trexio_assert(rc, TREXIO_SUCCESS)
```

Fortran

```python
import trexio

with trexio.File(file_name, ‘w’, trexio.TREXIO_HDF5) as
fhandle:
    trexio.write_nucleus_num(fhandle, 12)
    assert trexio.has_nucleus_num(fhandle)
```

Python

https://github.com/TREX-CoE/trexio