
Data format and Standards for Computational Chemistry

The importance of standards for science

content

- Data management for comp science
 - Size
 - Portability among different systems
 - Text $\leftarrow \rightarrow$ binary
 - Endianess
 - Portability among different users
 - Plain $\leftarrow \rightarrow$ Annotated
- Different technologies for annotated files with examples
 - XML (\rightarrow CML, CSX)
 - HDF5 (\rightarrow Q5cost)

Data are kept in files

- Visualisation needs data
- For computational scientists data are contained into “files” and are the result of simulations
- Output files of Computational Chemistry tools (Gaussian, ADF, Gromacs, QE, ...)
- It is important to understand the nature of the files and difficulties related to them
 - Dimension of the file
 - Portability among different platform (ASCII/Binary, Endianess, ...)
 - Portability among different users (annotated/non annotated data)

File dimension

Moving data

- With growing power of computers, researchers tend to create larger files
- Hundreds of TB. What I can do?
 - Storage using Tier-0 Machine is limited in time (e.g. PRACE Project data can be stored for 3 Month)
 - Data analysis can be time consuming (eyen years)
 - I don't want to delete data
 - I have enough storage somewhere else?
 - How can I move my data?

Moving data: theory

- BW requirements to move Y Bytes in Time X

Bits per Second Requirements

10PB	25,020.0 Gbps	3,127.5 Gbps	1,042.5 Gbps	148.9 Gbps	34.7 Gbps
1PB	2,502.0 Gbps	312.7 Gbps	104.2 Gbps	14.9 Gbps	3.5 Gbps
100TB	244.3 Gbps	30.5 Gbps	10.2 Gbps	1.5 Gbps	339.4 Mbps
10TB	24.4 Gbps	3.1 Gbps	1.0 Gbps	145.4 Mbps	33.9 Mbps
1TB	2.4 Gbps	305.4 Mbps	101.8 Mbps	14.5 Mbps	3.4 Mbps
100GB	238.6 Mbps	29.8 Mbps	9.9 Mbps	1.4 Mbps	331.4 Kbps
10GB	23.9 Mbps	3.0 Mbps	994.2 Kbps	142.0 Kbps	33.1 Kbps
1GB	2.4 Mbps	298.3 Kbps	99.4 Kbps	14.2 Kbps	3.3 Kbps
100MB	233.0 Kbps	29.1 Kbps	9.7 Kbps	1.4 Kbps	0.3 Kbps
	1H	8H	24H	7Days	30Days

Moving data: some figures

- Moving outside CINECA
 - ✓ `scp` → 10 MB/s
 - ✓ `rsync` → 10 MB/s
 - ✓ `(gridftp` → 100 MB/s)
- I must move 50TB of data:
 - ✓ Using `scp` or `rsync` → 60 days (or 6)
- Bandwidth depends on network you are using (these figures using a 1Gb network)
- Could be better, but in general is even worse!!!

Moving data: some hints

- **Size matters:** moving many little files cost more then moving few big files, even if the total storage is the same!
- Moving file from Fermi to a remote cluster via gridftp

Size	Num. Of files	Mb/s
10 GB	10	227
100 MB	1.000	216
1 MB	100.000	61

- ✓ You can loose a factor 4, now you need 25 days instead of 6 to move 50TB!!!!!!

moving data: some hints

- ✓ Plan your data-production carefully
 - ✓ Clean your dataset from all unnecessary stuff
 - ✓ Compress all your ASCII files
 - ✓ Use **tar** to pack as much data as possible
 - ✓ Organize your directory structure carefully
 - ✓ Synchronize with **rsync** in a systematic way
 - ✓ One example:
 - We had a user who wants to move 20TB distributed over more than 2'000'000 files...
 - **rsync** asks many hours (about 6) only to build the file list, without any synchronization at all
-

Portability 1

ASCII vs Binary

Endianness

ASCII vs. binary/1

- ASCII is more demanding respect binary in term of disk occupation ...
 - Numbers are stored in bit (SP floating point number → 32 bit)
 - 1 single precision on disk (binary) → 32 bit
 - 1 single precision on disk (ASCII) → 80 bit
 - 10 or more **char** (**1.23456e78**)
 - Each char asks for 8 bit
 - Not including spaces, signs, return, ... moreover there are rounding errors, ...
- ... as well as computational time
 - Transferring data from memory to disk (binary) is simply matter of moving data
 - Transferring data from memory to disk (ASCII) requires a translation process

I/O: ASCII vs. binary/2

- ASCII is worst (in terms of space and time) but ...
- ASCII is readable by humans and also by ALL computers (it is a standard!)
 - A chkpoint file from Gaussian produced on a X85 based computer, if ASCII coded, can be moved on an IBM Power based system and used as input file of another Gaussian run.
- Binary is strictly related to the computers who produced it
 - A chkpoint file from Gaussian produced on a X85 based computer, if binary coded, must be converted (or translated into ASCII, before it can be moved on an IBM Power based system and used as input file of another Gaussian run.

Endianness

- Fortunately, IEEE standard set rules for floating point operations, so now also for binary data a standard is available
- The only difference among different computers is today related to “data storage”
- Single precision FP: 4 bytes (**B0**,B1,B2,B3)
 - ✓ Big endian (IBM): **B0** B1 B2 B3
 - ✓ Little endian (INTEL): B3 B2 B1 **B0**
- Solutions:
 - ✓ Hand made conversion
 - ✓ Compiler flags (intel, pgi)
 - ✓ I/O libraries (HDF5)

Portability 2

Plain vs «Annotated» files

Data Interoperability

- Share data among different researchers
 - different tools (e.g. visualization tools)
 - different systems
 - different analysis/post processing
- Only who wrote the data knows how they are written!
 - No problem if the file is an “internal” file
 - No problem if the programmers of the different tools know each other or all the tools are well documented
- A standard format for data was never imposed for Comp Chem, each program has its own data format, no program strong enough to impose a de-facto standard

Data Interoperability

- Possible solution: to include the description of the data in the file (metadata) and agree on a common “language”
- Or to produce a “translator” tool able to translate each format into the other one (Babel)
- Several experiences to produce specific “languages” for data in Computational Chemistry, all of them based on already defined (general) tools for annotation data
 - CML - (based on XML)
 - CSX - A Standard Data Format for Computational Chemistry:
 - Q5Cost (based on HDF5)

CML – Chemical Markup Language

- ❑ CML is an XML-based language for representing chemical data
- ❑ More precisely, CML is the application of XML for the representation of molecules and molecular representation, crystallography and spectra
- ❑ CML evolved in the chemical industry to solve the needs of exchanging molecular and other information for publishing Web-based documents for patent applications, standards committees, and other organizations
- ❑ CML does not cover all chemistry but focuses on molecules (and similar structures representable by a formula)
- ❑ CML does represent molecules, atoms, and bonds


CSX - Common Standard for eXchange

- It is a structured data container design to hold CC result data and additional metadata, it is a XML schema
 - Developed by Neil Ostlund then part of a SBIR grant (US Dept of Energy). Version 2.0 is currently under development.
 - With respect of CML its focus is mainly on complex structures e.g. residues and CC results
-

Q5cost

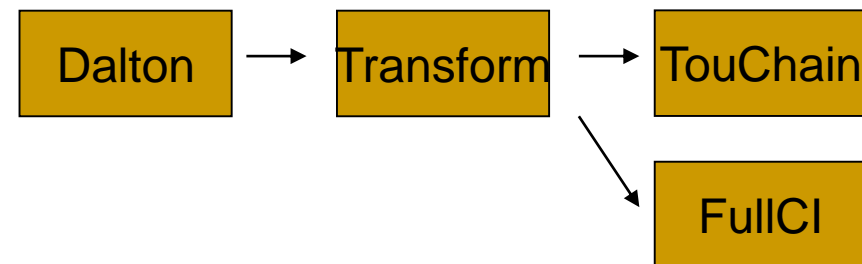
Common format for Quantum Chemistry interoperability

Where we started from

- Realisation of a distributed environment for making **code interoperability** possible and easy.
 - Definition of a consensus strategy for making codes communicate
 - Definition of a Common data Format for Quantum Chemistry
- 

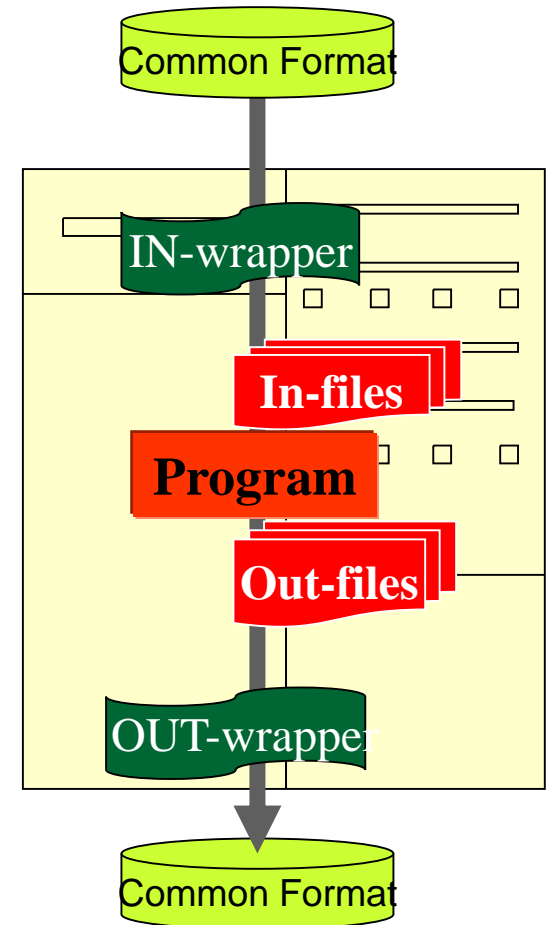
The problem

- The problem arises from a very common situation in the community:
a scientific problem needs several programs to be solved
 - open source or commercial programs used for producing the standard quantities (COLUMBUS, DALTON, MOLCAS, MOLPRO, ...)
 - in-house codes written to test computational experiments
- None of them shares the same data format
- Tricky procedure
 - Get all the programs
 - Install them on the local computer
 - **Translate the data**



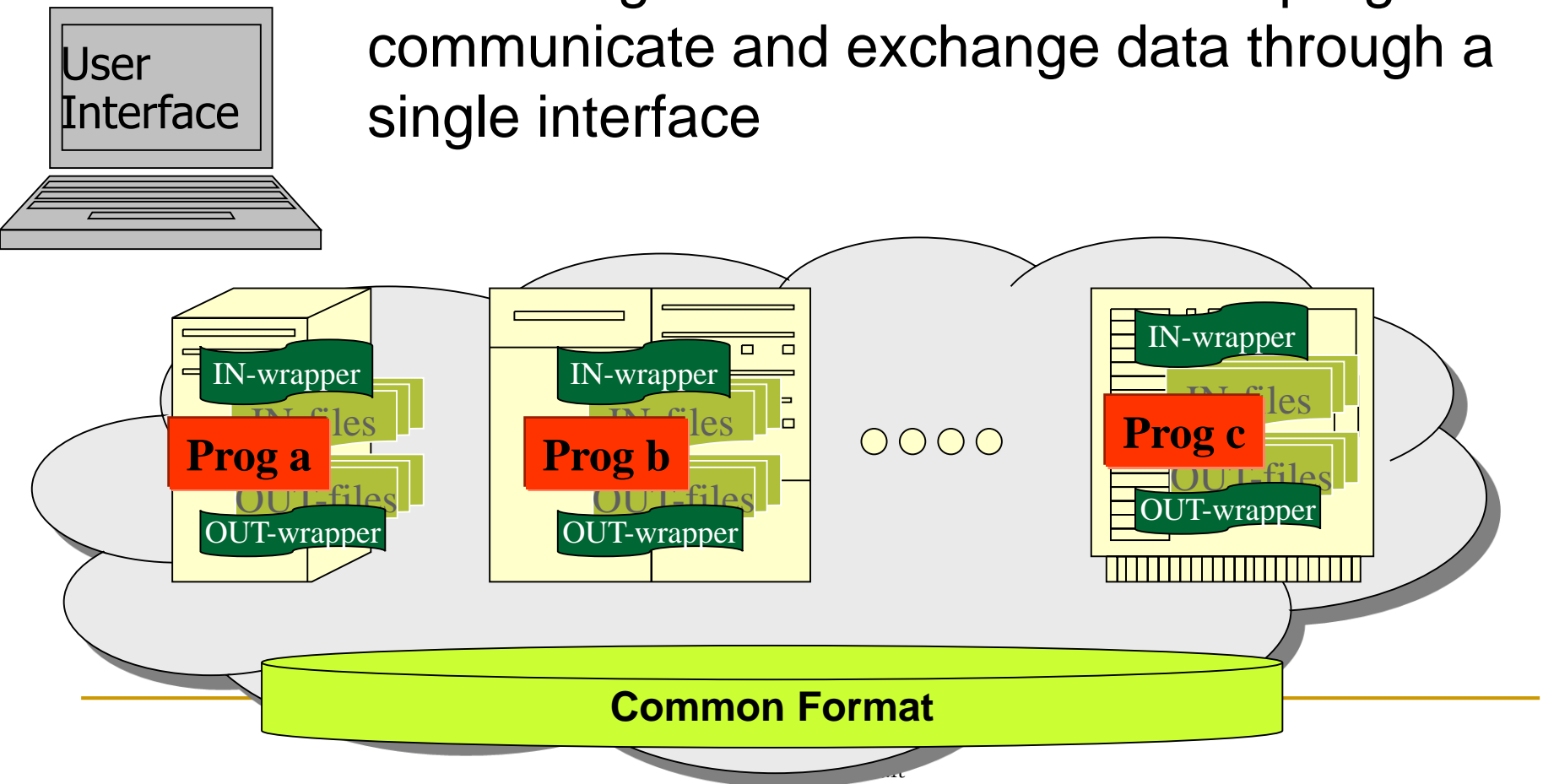
The solution (1/2)

- Each program
 - remains “at home”
 - maintains its own proprietary data format
- A set of “translation utilities” takes care of translating data from/to the proprietary formats and the common format



The solution (2/2)

A sort of “grid environment” makes programs communicate and exchange data through a single interface



The first problem: Data Format

- One of the main challenges is a **lack of community standards for data representation**

- Two main strategies:

1. To develop translators that map from one format to another. (theoretically unscalable, even if some example available)
2. Support the development of a **community standard** and develop translators towards the common formats. This is the strategy we decided to follow

Which kind of data for QC codes interoperability?

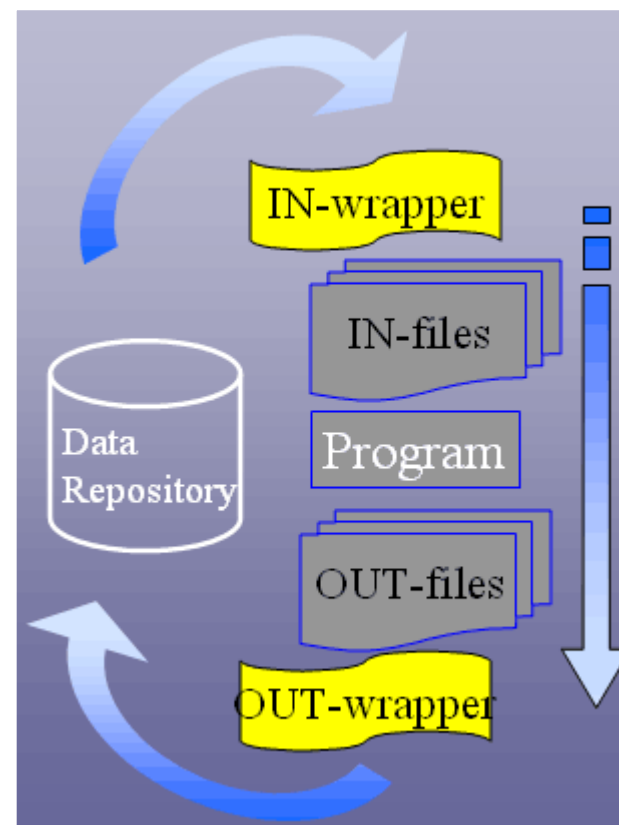
- We identified **two different kinds** of information
 - **small data** quantities, mainly ASCII coded, like atom labels, geometry, symmetry, basis sets and so on

□ **large datasets**, normally binary, like integrals and expansion coefficients.

Already valuable solutions for first type, nothing for the second one

The Common Data Format

- What do we need
 - Complete
 - Flexible
 - Near to chemists
 - Good performance on binary data
 - External
- How to use it
 - Library to be used for input/output wrappers for data conversion
- A format for **interchange**, not to be used as an internal format
- We are interested in **functionality** (that has to be general and complete). **Performance** and efficiency, although important, are not the main focus.



Large bin data

- We looked for a suitable technology that can merge
 - portability,
 - efficiency,
 - FORTRAN binding,
 - data compression, and
 - easy access to information.

HDF5 was considered the right technology

HDF

by NCSA/University of Illinois

Hierarchical Data Format

To develop, promote, deploy, and support open and free technologies that facilitate scientific data storage, exchange, access, analysis and discovery.

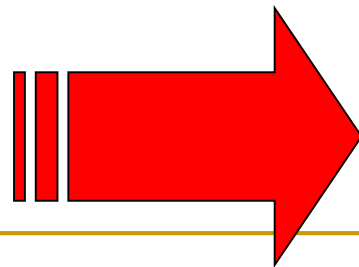
- Format and software for **scientific data**
- Stores images, **multidimensional arrays**, tables, etc.
- Emphasis on **storage** and **I/O efficiency**
- **Free** and commercial software support
- Emphasis on **standards**
- Users from **many** (engineering and) **scientific fields**

Consists in a Format definition, a library (C and Fortran) and tools ...

Much work already done

What else do we need?

- **Design the data format for QC**
- **Produce an hi-level library for I/O:**
addressing specific quantities and functions in the data format
- **The language of choice is Fortran, because**
“Chemists prefer Fortran”:
 - HDF5 comes with a Fortran interface
- **write “Wrappers”** using the new library:
each wrapper is a translator, specifically designed for a QC code in the chain, capable of retrieving information from, and writing information to, the file in accordance with the defined syntax.



Q5Cost

Q5cost

- a common **data format** for Quantum Chemistry, designed for interoperability.
- **A library** for the management of data produced by a generic QC code and its conversion from and to the common data format.
- The Q5cost format and library is based and takes advantage of **HDF5** (Hierarchical Data Format)

Who did contribute

- Activity carried on within Cost in Chemistry D23-D37
- Collaborating partners:

- Elda Rossi, Andrew Emerson – CINECA (**Coordinator**)
- Gian Luigi Bendazzoli – Università di Bologna
- Antonio Monari – was at Università di Bologna and IRSAMC
- Renzo Cimiraglia, Celestino Angeli, Chiara Pastore - Università di Ferrara
- Daniel Maynau, Stefano Evangelisti, Anthony Scemama – IRSAMC, Toulouse
- Vallet Valerie, JeanPierre Flament - University of Lille
- José Sanchez-Marin - Universitat de Valencia
- Peter Szalay, Attila Tajiti - Eötvös Loránd University, Budapest
- Kállay Mihaly - Budapest University of Technology and Economics
- Kim Baldrige - University of Zürich
- Kenneth Ruud - University of Tromsø
- Stefano Borini - was at ETH – CH

The model (Data \leftrightarrow Metadata)

- **Data:** the large binary quantities (integrals, properties and wave functions). Usually stored by matrices with an arbitrary number of indices (rank-n arrays), scale aggressively with the system size, normally accessed with a “chunked” approach (i.e., using well-defined blocks of data).
- **Metadata:** simple and small pieces of data that describes and better defines the data. They represent well-known chemical entities like nuclear energy, molecular orbital labels, and molecular symmetry and can be stored as scalars, vectors or matrices.
 - nuclear repulsion energy → floating point scalar
 - molecular orbitals coefficients → (N,M) floating point matrix
 - orbital energies → floating point vector
 - molecular orbital labels → vector of strings
 - ...

The model: domains

- **System:** general info on the molecular system
 - Geometry
 - Symmetry
 - Basis set
- **Atomic Orbital (AO):** refers to the data defined on the AO basis, overlap, one-electron integrals, two-electron integrals and the generic property, i.e. any other property that can be described on the AO basis (dipole moment integrals, for example). This domain contains also the definition of the Basis Set.
- **Molecular Orbital (MO):** refers to the data defined on the MO basis, one- and two-electron integrals and the generic property. This domain contains also the transformation matrix needed to define the MO on the AO basis.
- **Wave Function (WF):** refers to the definition of the wave function.

Q5Cost 1.1
Extension with Density Matrix

Ctime (s)
Atime (s)
Q5version (s)

System

Title
Electrons
 α, β

Symmetry:
- num_sym
- labels

Geometry:
- charges
- coordinates
- labels
- nuclear_energy
- num_atom
- atomic_number

Basis

Coord system
Atom
Angular number
Magnetic number
Coeff/exp
Num of contracted
Num of primitives

AO

Name
Num_orb_sym
Num_orb_tot
Labels
Transformation

MO

AO_pointer
Name
Num_orb_sym
Num_orb_tot
Labels
Orbitals
SCF_energy
Classification
Occ_num
Symmetry

WF

MO_pointer
Name
Energy
Core Energy
Num_dets

overlap

oneint

twoint

densities

prop

Index/value

Index/value

Index/value

Index/value

Index/value

oneint

twoint

densities

prop

Index/value

Index/value

Index/value

Index/value

DetCoeff

Determinants/
coefficients

Name
Description
Rank
Symmetry
Real/Complex

The Q5Cost Library

- The library is written in F90, made of more than 100 routines, whose names are chemically significant and strictly recall the function of the routine itself.
- The routines belong to different classes, both referring to domains and data objects within the domain:

```
Q5Cost_init
Q5Cost_deinit
Q5Cost_file_*      Routines for housekeeping and file mangmnt
Q5Cost_Basis_*     Basis set
Q5Cost_System_*    General high level information
Q5Cost_Sys_Geometry_ Geometry of the molecule
Q5Cost_Sys_Symmetry_ Space symmetry of the molecule
Q5Cost_AO_         Atomic Orbitals
Q5Cost_AOOneInt_   Atomic Orbitals: one-e integrals
Q5Cost_AOTwoInt_   Atomic Orbitals: one-e integrals
Q5Cost_AOOverlap_  Atomic Orbitals: overlap integrals
Q5Cost_AODensity_  Density Matrix on the AO orbitals
Q5Cost_AODensityOne_ Density M. on the AO orbs: One-body
Q5Cost_AODensityTwo_ Density M. on the AO orbs: Two-bodies
Q5Cost_MO_         Molecular Orbitals
Q5Cost_MOOneInt_   Molecular Orbitals: one-e integrals
Q5Cost_MOTwoInt_   Molecular Orbitals: two-e integrals
Q5Cost_MODensity_  Density M. on the MO orbs
Q5Cost_MODensityOne_ Density M. on the MO orbs: One-body
Q5Cost_MODensityTwo_ Density M. on the MO orbs: Two-bodies
Q5Cost_Property_   Integrals for a generic operator
Q5Cost_WF_         Wave-function
Q5Cost_WFDetCoef_ Wave-function: Dets and coeffs
```

Some examples

- Routines working on the domains and on the metadata
`Q5Cost_AO_set_num_orb_sym`
- Routines working on the large QC datasets, for example:
`Q5Cost_AOOneInt_append`
- `Q5Cost_AOOverlap_read`
(`file_id`, `offset`, `howmany`, `idx`, `value`, `error`,
`user_tag`)

The .q5 file

- Data are stored into a file, usually identified by the “.q5” extension.
- It is a standard HDF5 file and can be accessed using the HDF5 tools (h5dump, h5ls, ...)
- **More specific tools** are available in the Q5cost distribution (q5dump, q5edit, ...).
- Each .q5 file always refers to a single molecular specie, a single geometry and a single choice for the basis set functions.
- If you need more geometries (like in geometry optimisation workflows) you need multiple .q5 files.
- Each .q5 file may contain data from different QC calculations
- The format was also adapted to Quantum Dynamics calculations thanks to the collaboration with Perugia University