UNIVERSITY OF Southampton

HDF5

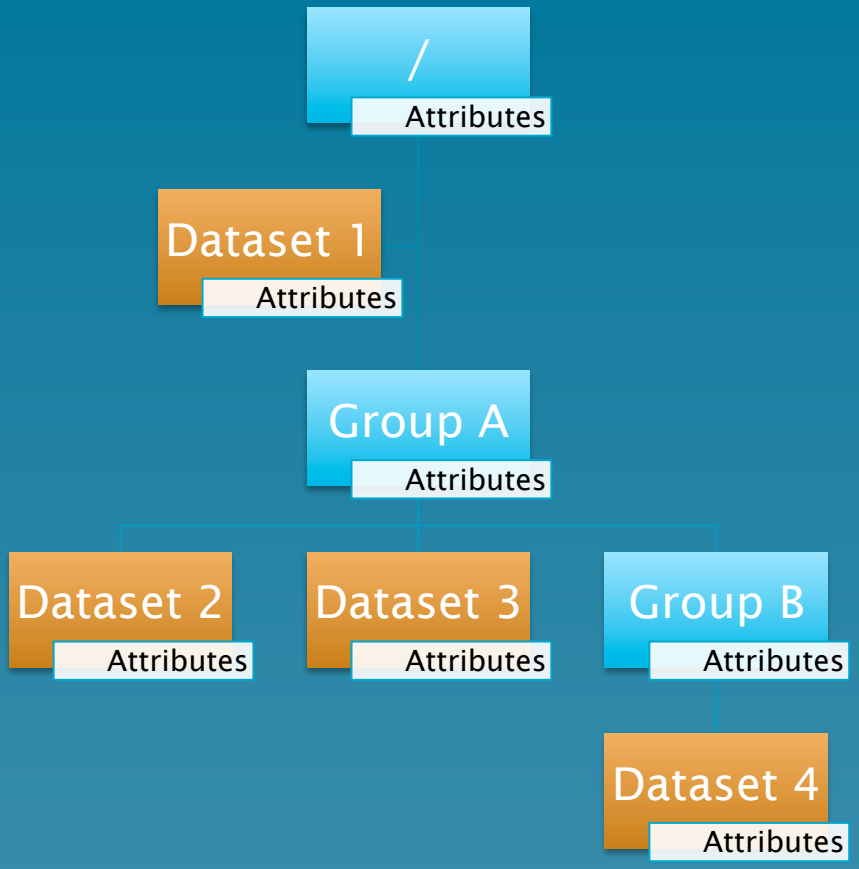
James Harrison & Rory Brown

jch1g10@soton.ac.uk & rob1g10@soton.ac.uk

NGCM

What is HDF5?

- A hierarchical file
- Structure:
  - Groups
  - Datasets
  - Attributes



Why use it?

- Store multiple datasets together
- Structural versatility
- Include metadata in attributes (self-describing)
- Can be read/written in parallel
- Interfaces in C, C++, Fortran, Java and Python

h5ls

- Similar to `ls` in terminal
- `h5ls file.h5/path`

Flag	Description
-r	Recursive
-v	Verbose
-d	Display data (unstructured)

h5dump

- Display contents of an HDF5 file

Flag	Description
-n	List objects in file
-n 1	List objects and attributes
-H	Display only header information (no data)
-A 0	Suppress display of attributes

h5dump

- Display contents of an HDF5 file

Flag	Description
-g /path/to/G	Display group G
-d /path/to/D	Display dataset D
-a /path/to/A	Display attribute A
-N path	Display any object matching path

h5copy

- Copy objects from one HDF5 file to another, or within a file

Flag	Description
-i	Input file
-o	Output file
-s	Source object
-d	Destination object
-v	Verbose
-p	Create parent groups

Exercise 1

- Copy datasets from exercise1.h5 to solution.h5
- Clues are found in the attributes
- Start from the attribute in the root group
- Run "`python codebreaker.py`" to check solutions

h5py

- In the second half of the presentation, the h5py module will be introduced
- Documentation: http://docs.h5py.org/en/latest/
- The following knowledge is assumed:
  - Basic Python (incl. dictionaries)
  - Numpy

h5py

- Python module for manipulating HDF5 files
  - Create HDF5 files
  - Change file structure
  - Read and write data
- Use "`import h5py`" to access module commands

Files and groups

- `f = h5py.File('filename', 'a')`
- `f.close()` when finished
- Groups work like dictionaries (keys and values)
- `grp = f.create_group('name')`
- `del group['subgroup']`

Datasets

- Multi-dimensional datasets
- `dset = group.create_dataset('name', size, dtype)`
  - `size`: tuple
  - `dtype` (optional): e.g. `'f'`, `'i8'`
- `dset = grp.create_dataset('name', data=x)`
  - `x`: numpy array

Reading and Writing Data

- `data = dset[…]` or `dset[…] = data`
- Numpy slicing syntax
  - e.g. `dset[3:5, :, 2]`
- `shape` / `len` / `dtype` / `size` (same as numpy)
- `dset.parent`: group containing `dset`

Attributes

- Can be attached to any group or dataset
- `object.attrs['name'] = data`
- data can be a scalar, string or numpy object

Exercise 2

- This exercise will be about creating your own HDF5 file
- Work through the exercises in exercise2.ipynb