VarPy: A python library for volcanic and rock physics data analysis

Rosa Filgueira
Andrew Bell
Malcolm Atkinson
Ian Main

University of Edinburgh, School of Informatics
University of Edinburgh, School of Geosciences

Python and Scientific Communities

- High-level programming language, interpreted with capabilities for object-oriented programming
- Very easy to learn for non-programmers
- Well structured: Easy to read and understand
- Well-known language with a big community
- Free and open source

Python and Scientific Communities

- Rich scientific computing libraries
  - **Obspy**: Simplifies the usage of Python programming for *seismologists*.
  - **SymPy**: Python library for *symbolic mathematics*
  - **Geopy**: Python *Geocoding* Toolbox
  - **Numpy**: Array processing for numbers, strings, records, and objects
  - **Scipy**: Scientific Library for Python
- Most Popular libraries: http://www.s-anand.net/blog/the-most-popular-scientific-python-modules/

Overview of Obspy

- Open-source Python toolbox for **seismological data processing** → http://obspy.org/

- Parsers for common file formats and seismological signal processing routines:
  - **seismological time series (waveform)**
- Besides**:**
  - It has many users
  - Well documented
  - Provides user support
  - It grows progressively adding new features

Volcanology and Rock Physics Communities

- Increase of digital instrumentation in volcanology and rock physics
  - Huge amount of seismicity data that need for computational analyses and models.
- Not library designed specifically for volcanic earthquake and rock physics data.
- Each researcher develops each own codes

Proposal: VarPy

- New open-source python toolbox
- Aim: facilitate rapid application development for those communities
  - Focus in seismicity deformation data
  - Full repertoire of commonly required actions
    - Analysis and modeling in real time and retrospective
    - Capabilities for data exploration, data analysis, quality check
  - Users can define their own workflows to develop
    - models, analyses and visualizations

Expected Benefits

- Easy method to analyze **seismicity data**
- Standardize different tasks/procedures
- Using the same functions by different researches → easier to compare results and performance of models
- Cost of maintaining the library is shared among the community

VarPy design

- Library style from **ObsPy**
- VarPy does not attempt to replace the functions provided by other python libraries (NumPy, SciPy) → complementary of them
- VarPy & Ipython notebooks

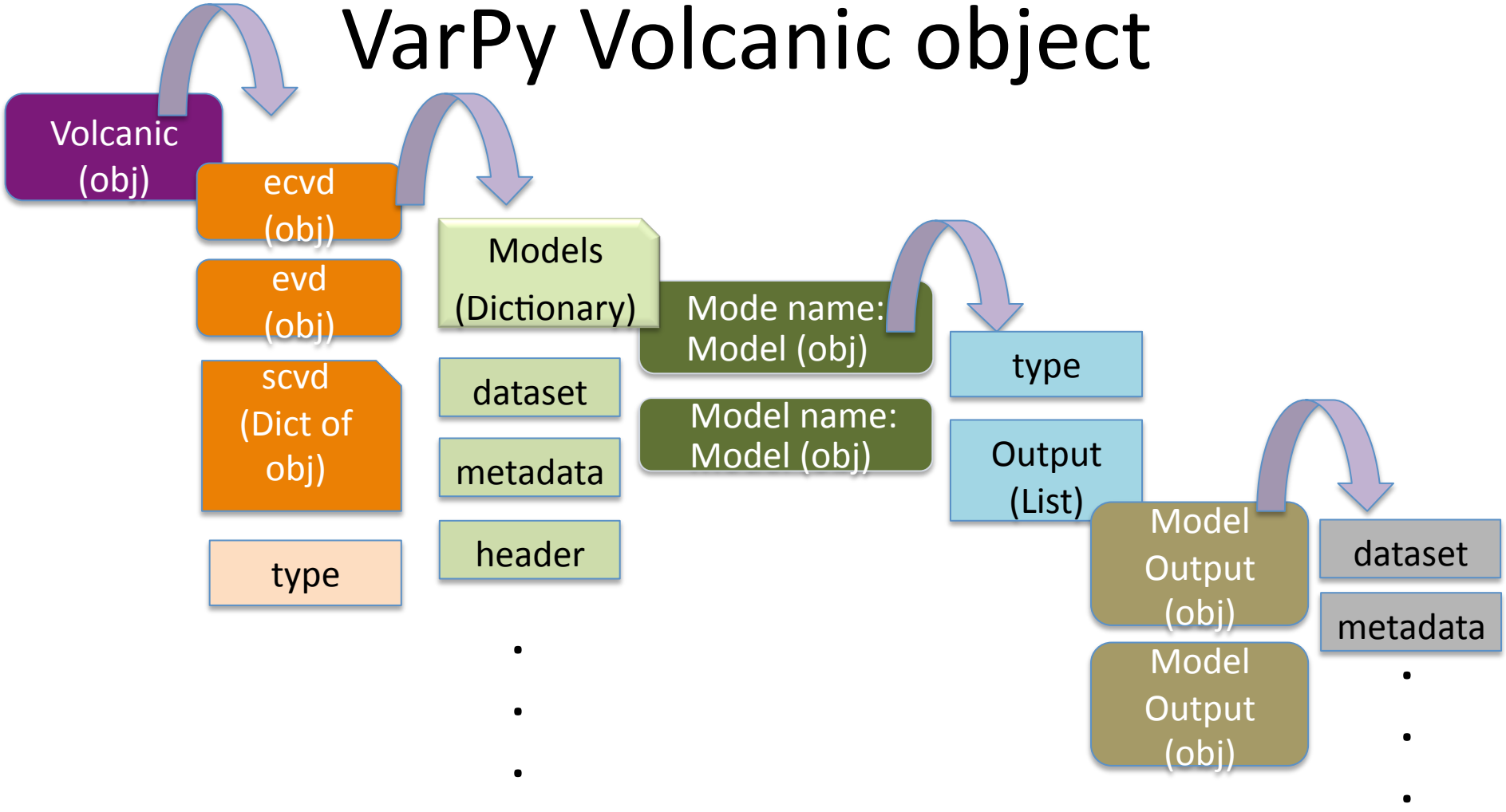
VarPy datatypes

- **Seismicity deformation: strain and stress data**
- Time-series data of two classes
  - *Event catalogue data* (ECD)
    - Series of events (acoustic emissions, earthquakes, volcanic eruptions)
    - Occur at discrete times
    - Specific attributes (location, depth, magnitude, duration)
  - *Sampled continuous data* (SCD)
    - Series of times at which a continuous variable has been measured, and the value of that variable
    - Sample times are defined by the instruments' operator
    - May (or may not) be evenly spaced (daily, every second).
- Volcanic observatories and rock physics laboratories can produce data of both classes in a single experiment.

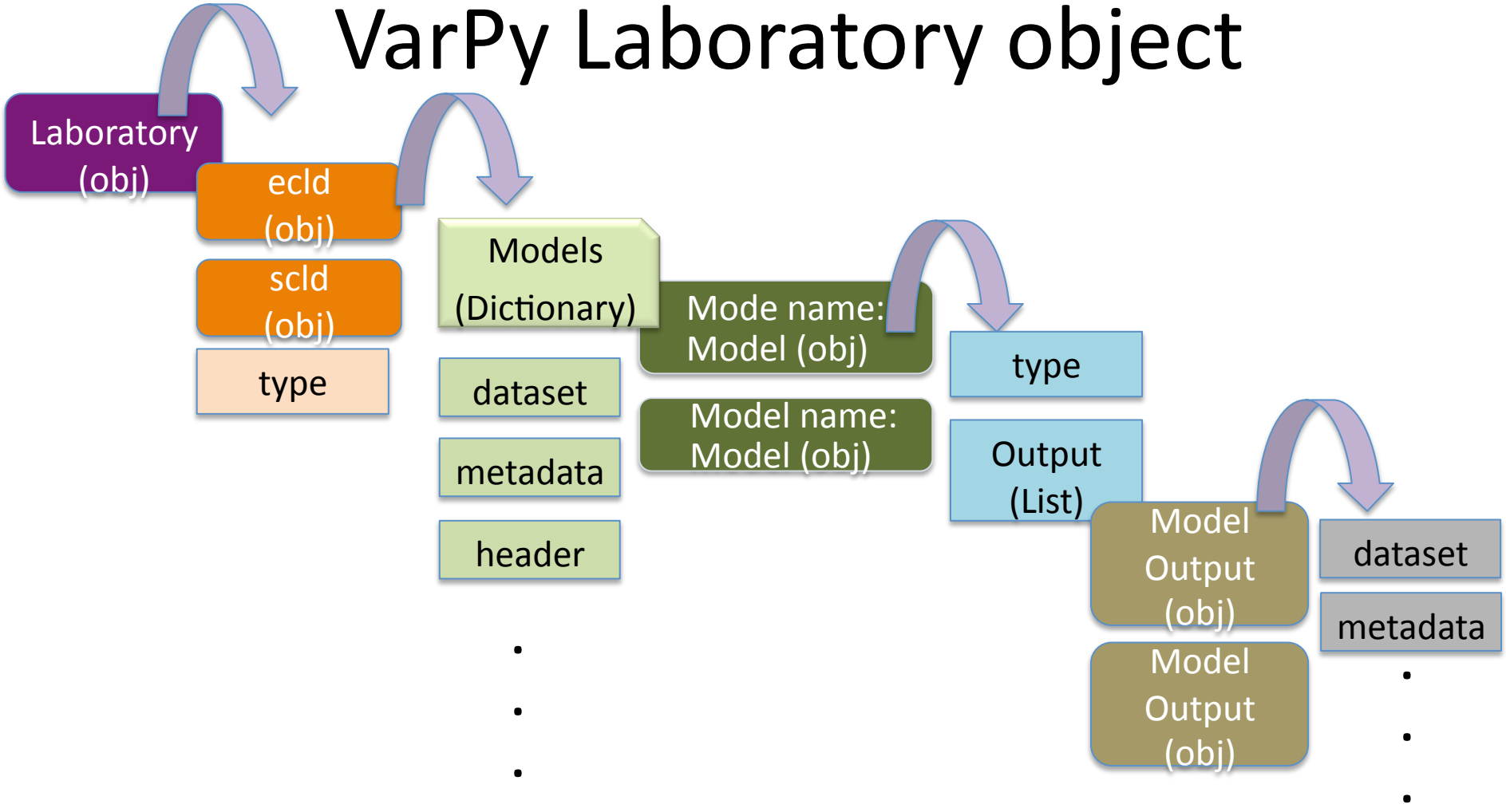
VarPy datatypes

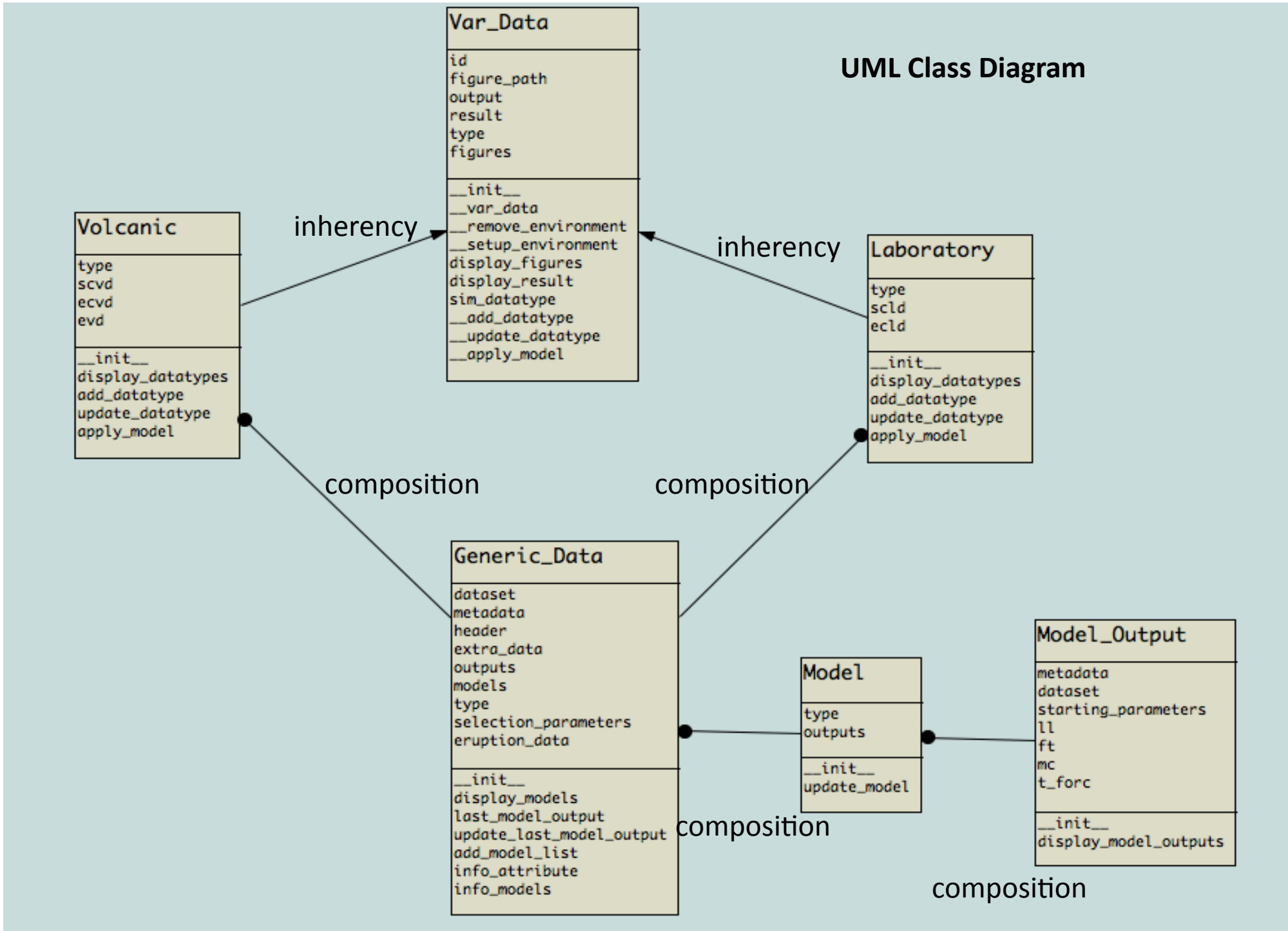
- Representation by using 4 different datatypes
  - *Event catalogue laboratory data* (ECLD)
  - *Event catalogue volcanic earthquake data* (ECVD)
  - *Sampled continuous laboratory data* (SCLD)
  - *Sampled continuous volcanic earthquake data* (SCVD)
- Volcanic data also have:
  - Eruption volcanic data (EVD):
    - Time-series data of volcanic eruptions, intrusions, or other events
    - with descriptions (type, size, duration,)

VarPy Volcanic object



VarPy Laboratory object




UML Class Diagram
Var_Data
id
figure_path
output
result
type
figures
__init__
__var_data
__remove_environment
__setup_environment
display_figures
display_result
sim_datatype
__add_datatype
__update_datatype
__apply_model
Volcanic
type
scvd
ecvd
evd
__init__
display_datatypes
add_datatype
update_datatype
apply_model
Laboratory
type
scld
ecld
__init__
display_datatypes
add_datatype
update_datatype
apply_model
inherency
inherency
composition
composition
Generic_Data
dataset
metadata
header
extra_data
outputs
models
type
selection_parameters
eruption_data
__init__
display_models
last_model_output
update_last_model_output
add_model_list
info_attribute
info_models
Model
type
outputs
__init__
update_model
Model_Output
metadata
dataset
starting_parameters
ll
ft
mc
t_forc
__init__
display_model_outputs
composition
composition

Example of VarPy objects

- Volcanic object with *ecvd* datatype:

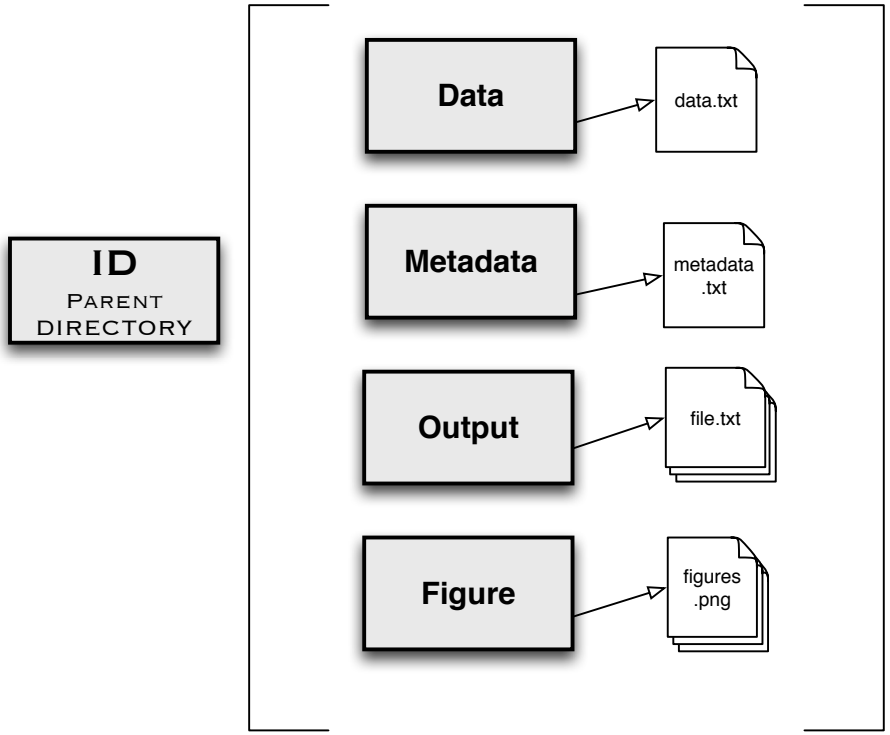
```
from varpy.management import core
ID = 'Tjornes_ex1'
ecvd_data_file = 'Iceland_IMO_C1_95-onwards.txt'
ecvd_metadata_file = 'Iceland_IMO_C1_meta.txt'
d1 = core.Volcanic(ID)
d1.add_datatype('ecvd',ecvd_data_file,ecvd_metadata_file)
```

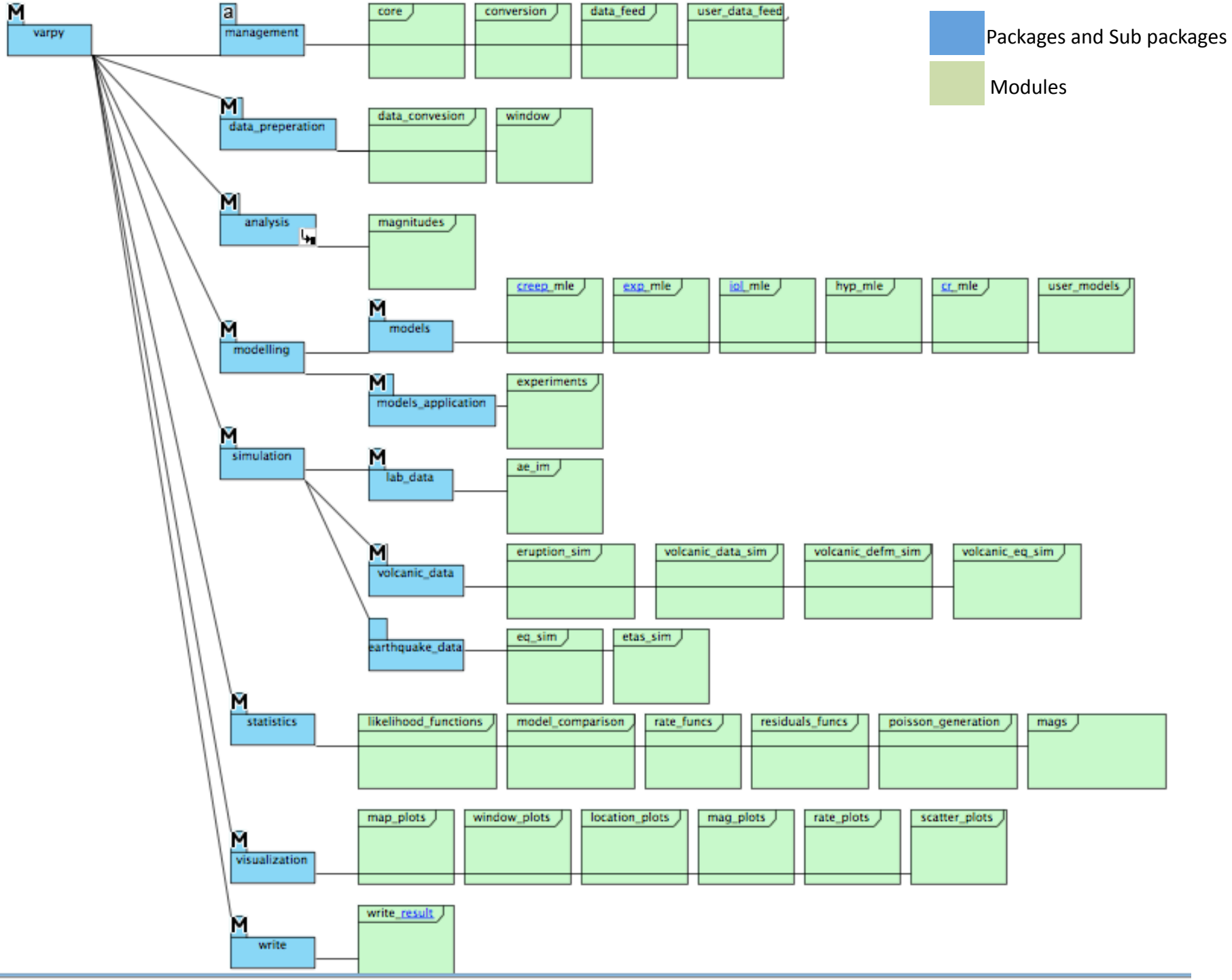
- Laboratory object with *scld* datatype:

```
from varpy.management import core
ID = 'UCL_Lab'
scld_data_file = 'UCL-exp1.txt'
scld_metadata_file = 'UCL-exp1-meta.txt'
d2 = core.Laboratory(ID)
d2.add_datatype('scld',scld_data_file,scld_metadata_file)
```

VarPy environment

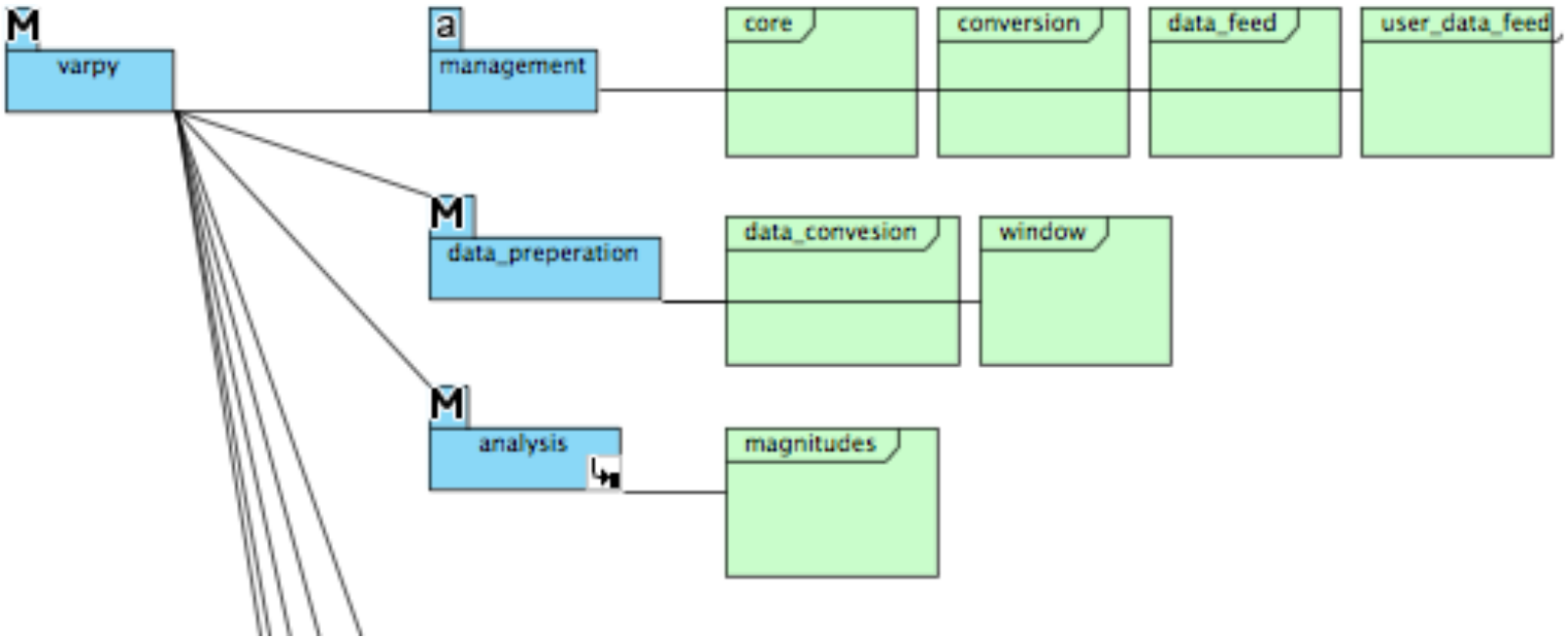
- `d1 = core.Volcanic(ID) or`
- `d1 = core.Laboratory(ID)`
- Creates a tree-directory in the current directory



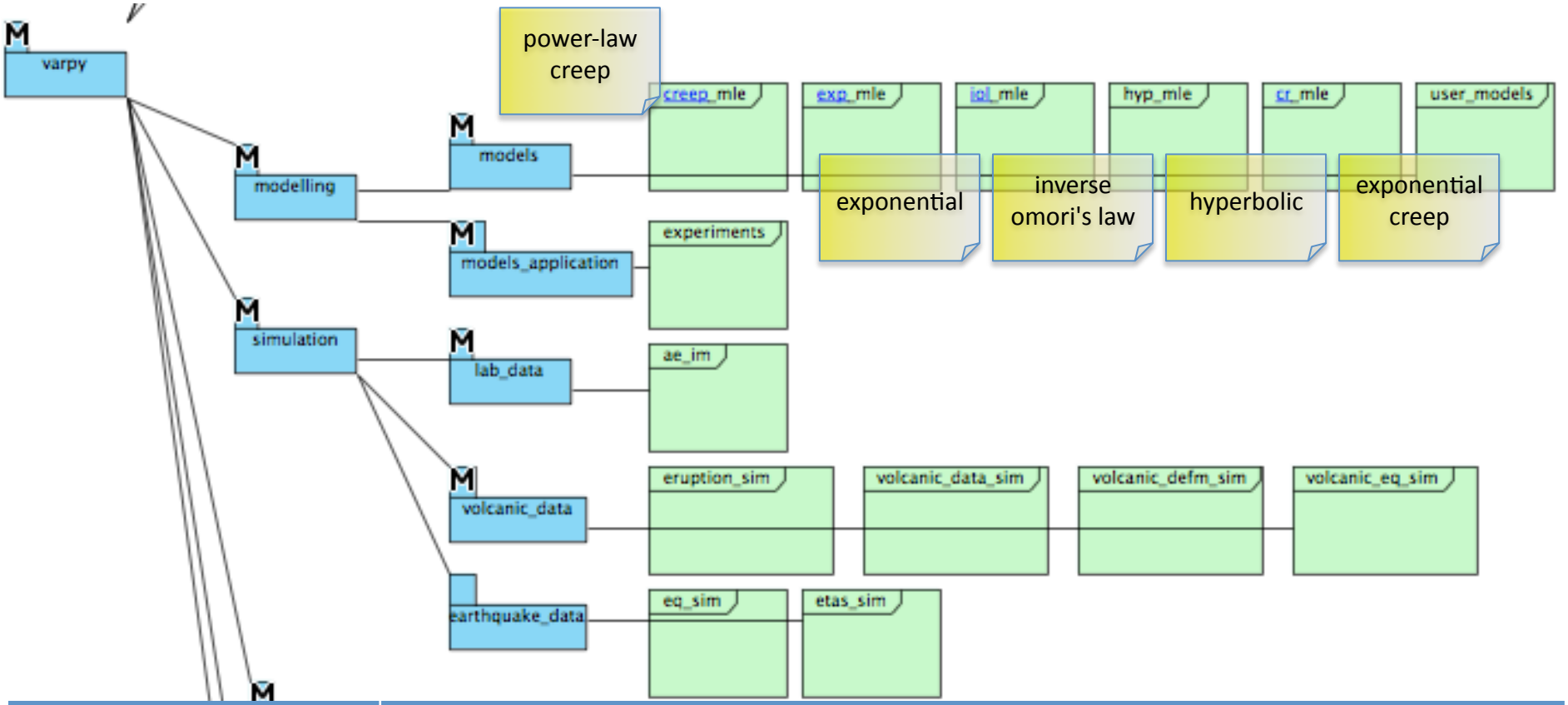

varpy
management
core
conversion
data_feed
user_data_feed
data_preperation
data_convesion
window
analysis
magnitudes
modelling
models
creep_mle
exp_mle
iol_mle
hyp_mle
cr_mle
user_models
models_application
experiments
simulation
lab_data
ae_im
volcanic_data
eruption_sim
volcanic_data_sim
volcanic_defm_sim
volcanic_eq_sim
earthquake_data
eq_sim
etas_sim
statistics
likelihood_functions
model_comparison
rate_funcs
residuals_funcs
poisson_generation
mags
visualization
map_plots
window_plots
location_plots
mag_plots
rate_plots
scatter_plots
write
write_result
Packages and Sub packages
Modules

General Packages

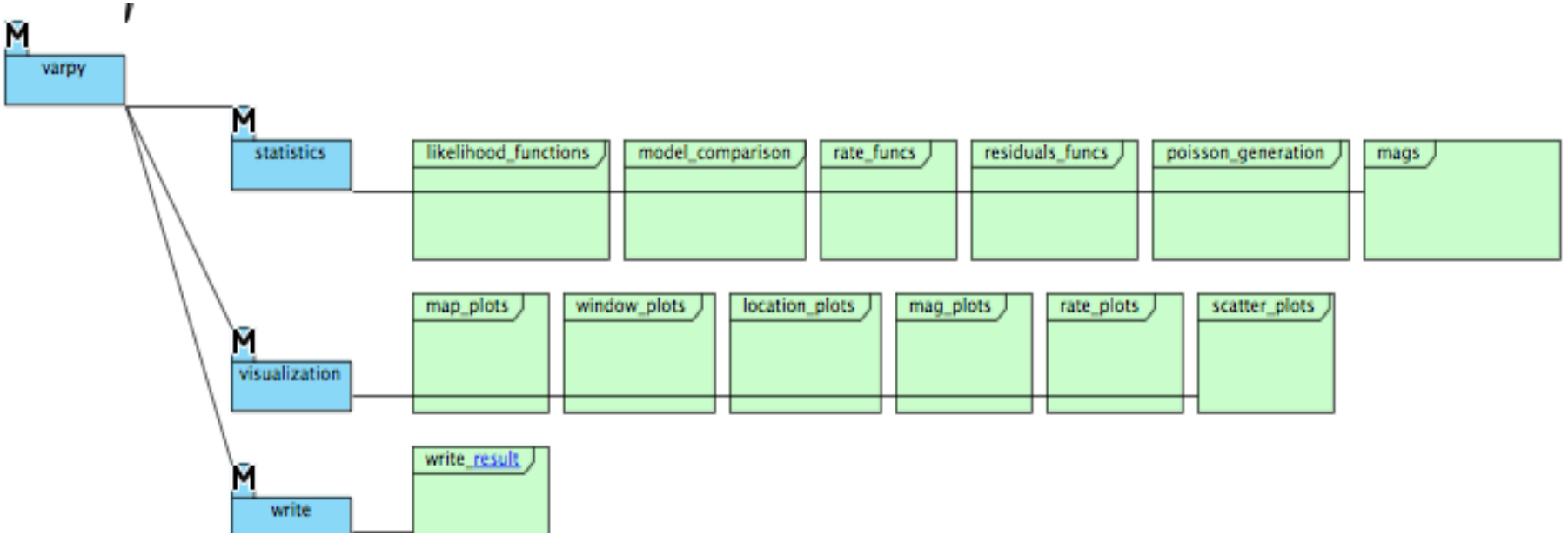
Package	Functionality
varpy.mangement	Core classes
varpy.data_preparation	Filtering routines
varpy.analysis	Analyzing filtered data routines
varpy.modeling	Modeling routines
varpy.simulation	Simulating routines of seismic data
varpy.statistics	Statistical routines for assisting with other routines
varpy.visualization	Plotting routines
varpy.write	Writing results routines



Module	Functionality: Module for
core	Handling varpy objects and methods.
conversion	Converting dates and time into different formats.
data deed & user_data_feed	Importing metadata and storing it into the varpy object
data_conversion	Converting values to another type of data (AE energy to magnitude)
window	Selecting a smaller sample based on a single (time window between two given dates) or on a combination of variables
magnitudes	Analyzing the filtered data (calculates completeness magnitude)



Sub-package	Functionality: Contains modules with
models	Various models which can be fitted to data.
model_application	Experiments with data & models: single analysis and multiple analysis
lab_data	Rock physics seismic data simulators
volcanic_data	Volcanic seismic data simulators
earthquake	Earthquake data data simulators



Package	Functionality: Contains modules for
statistics	Assisting with the generation of synthetic data, fitting models and comparing models
visualization	Plotting filtered data, the results of analyses, simulated data and model
write	Writing the results of analyses and experiments into text files

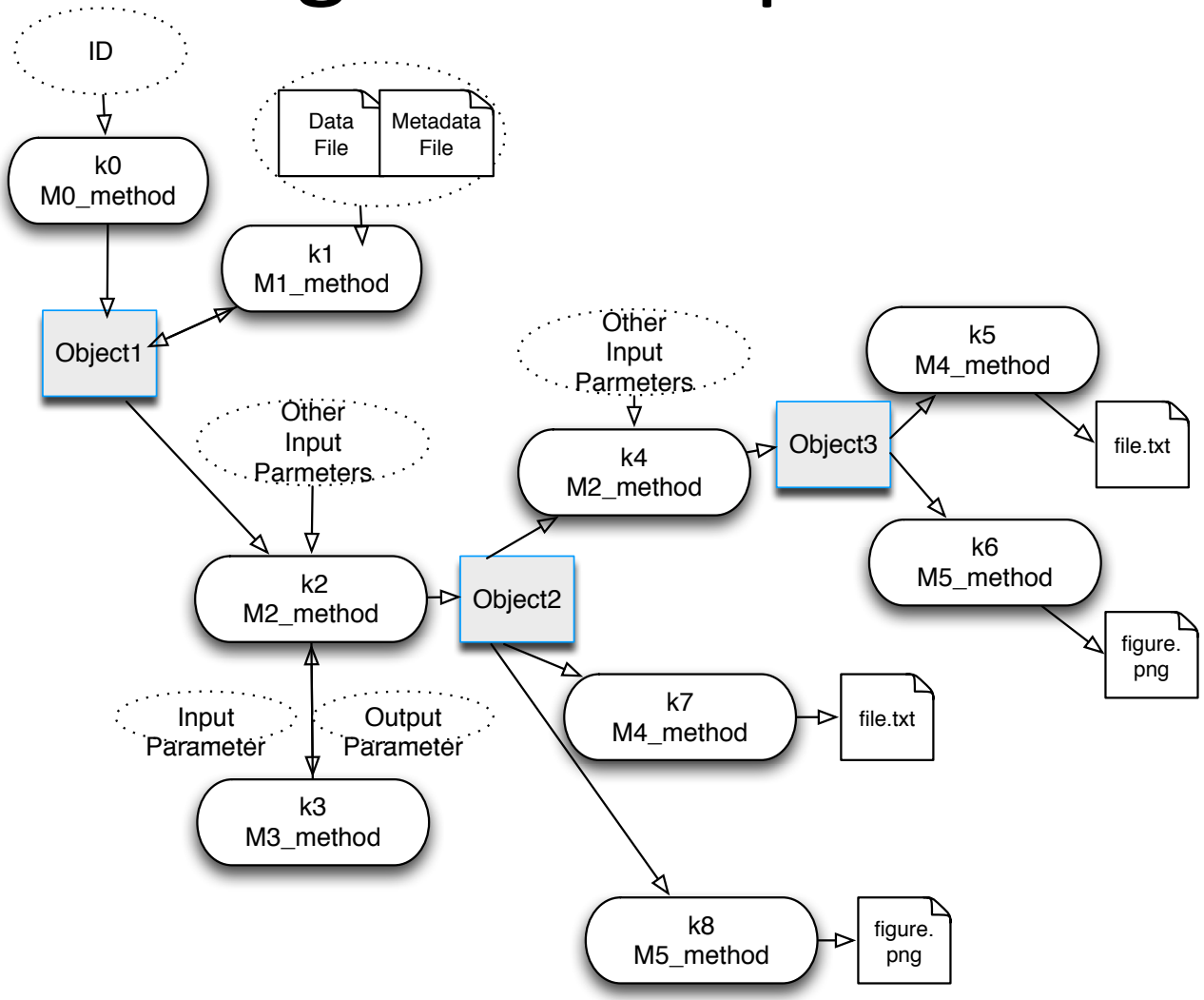
Varpy Experiments

- Experiments configuration: Model + Filtered data + Parameters
- Experiment types:
  - Single Analysis → Apply once a model
    - Retrospective Analysis:
      - Known failure/eruption time (ft)
      - Output: How well the model explains the data
    - Single Forecast
      - ft is not known
      - Output: Prediction of the failure/eruption.
  - Multiple Analysis → Apply several time a model
    - Prospective Forecast
      - ft not known
      - Output: Prediction in real time of the failure/eruption.
    - Retrospective Forecast:
      - Simulate ft not known
      - Output: Prediction of the failure/eruption

Type of methods

- M0 method
  - Aim: to create a volcanic or laboratory object
  - Input: ID
  - Output: new object with a tree-directory.
- M1 method:
  - Aim: to add an attribute to an object
  - Input: data and metadata files
  - Ouput: Add the attribute to the object, and copy the files into the tree-directory
- M2 method
  - Aim: to modify an attribute (or several attributes) of a object.
  - Input: object
  - Return: new object (copy of the input object) with the modifications. NO creates another tree-directory
- M3 method
  - Aim: to transform data.
  - Input: data
  - Output: data transformed
- M4 method
  - Aim: to write an attribute of an object to file.
  - Input: attribute of the object
  - Output: store a new file in the tree-directory
- M5 method
  - Aim: to plot into a figure an attribute of an object.
  - Input: attribute of the object
  - Output: store new figure the tree-directory

Algebra of operations



Allows us to keep the previous status of the object every time that we want to perform an update/modification of one or some attributes of the object.

VarPy Iptyhon Notebooks

- Notebook-1
  - Example of data exploration and visualization based on the Tjornes fracutre zone (Iceland)
- Notebook-2
  - Example of applying forecasting methods using volcanic data from Mt. Etna

VarPy contributions

- VarPy already allows:
  - Data exploration
  - Quality check
  - Data analysis
- Researchers could contribute:
  - Developing standards for:
    - Data format
    - Methodology for processing data
  - New models, simulators of seismic data, filters …

EFFORT and VarPy

- Multi-disciplinary
- **EFFORT aims**:
  - Determine predictability of brittle failure of rock samples
  - Determine how predictability scales with geo-system complexity
  - Provide facility for developing and testing codes:
    - Encourage data and model sharing through EFFORT gateway
- EFFORT gateway: http://effort.is.ed.ac.uk
- VarPy will help run the user's models in the gateway automatically.
- Two variants of the VarPy with many of the functions identical: Gateway & Developers versions

Questions

- Contacts:
  - Rosa Filgueira: rosa.filgueira@ed.ac.uk
  - Andrew Bell: a.bell@ed.ac.uk