

# Lab data handling using EdelweissData™

Manual on handling of *in vitro* data generated in the laboratory and use of data management systems for storage and analysis

**Latest update:** 2019-09-17

**Authors:** Lucian Farcas, Tomaž Mohorič

**Contact:** [lab@edelweissconnect.com](mailto:lab@edelweissconnect.com)

## Table of Contents

<b>Summary</b>	<b>2</b>
<b>Use case 1. Handling of tabular data generated using a microplate reader</b>	<b>3</b>
Raw data generation and temporary storage	4
Enrichment of data with metadata	6
Data file	7
Upload of data to EdelweissData	9
Data visualisation and data API generation in Edelweiss Data Explorer	10
Data analysis in Jupyter notebooks	11
Analysis of dose-response curves and calculation of BMDs using data API	11
<b>Annexes</b>	<b>13</b>

## Summary

The aim of this workflow and related use cases is to support the automation of data handling that is generated by laboratories (research, CROs, etc.) within *in vitro* studies. The workflow is supported by EdelweissData™ and Jupyter notebook tools and addresses the data files processing, storage, analysis and reporting. The data consist of experimental results enriched with metadata (details on how the data was produced). This approach follows FAIR data principles and aims to enhance the accessibility and usability of the data, make it accessible in real-time, well annotated and reliable (qualitative data).

Following the execution of the laboratory assays, the data is generated using the adequate equipment for that particular method applied. This manual includes guidance for different types of data:

- ☒ **Tabular data** generated by a microplate reader (example of GloMax® Explorer Multimode Microplate Reader<sup>1</sup>)
- ☐ **Images** generated using an inverted microscope (example of ZEISS Axio Vert.A1 Inverted Microscope<sup>2</sup> *(not yet included)*)
- ☐ **Transcriptomics data** (complex tabular data) *(not yet included)*

### Documentation:

- **GitHub:** <https://github.com/DouglasConnect/jupyter> (the most relevant folders are 'DC Lab' and 'Edelweiss use cases/in-house data/')
- **Google Drive** (G:\My Drive\Projects\Laboratory\Lab Data Management):  
[https://drive.google.com/drive/u/1/folders/1NHOBol3uNzipLa\\_Dj-Wd3G8Eq7RnVK82](https://drive.google.com/drive/u/1/folders/1NHOBol3uNzipLa_Dj-Wd3G8Eq7RnVK82)

---

<sup>1</sup>

<https://ch.promega.com/products/fluorometers-luminometers-multimode-readers/multimode-readers/glomax-explorer-system/?catNum=GM3500>

<sup>2</sup>

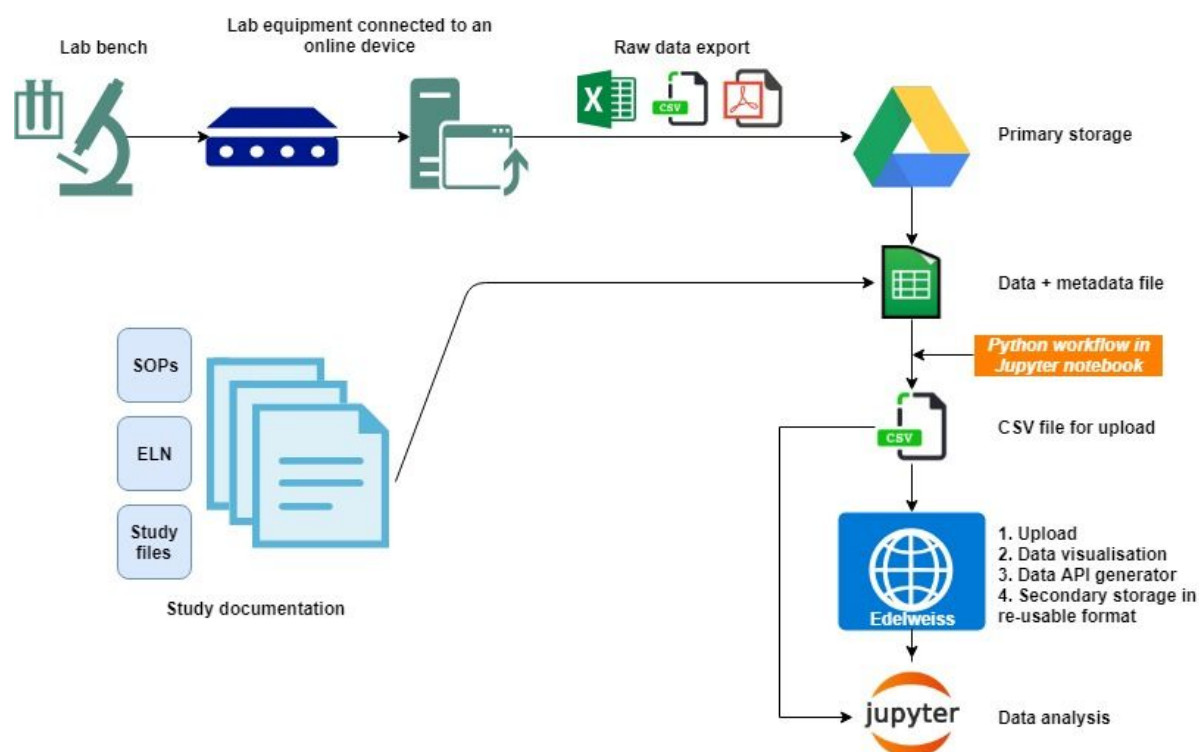
<https://www.zeiss.com/microscopy/int/products/light-microscopes/axio-vert-a1-for-biology.html>

---

## Use case 1. Handling of tabular data generated using a microplate reader

This section includes a practical example on data generated using the GloMax® Explorer Multimode Microplate Reader. The data is represented by absorbance, fluorescence or luminescence intensity measurements applied in different assays, e.g. cell viability, cytotoxicity, etc. The raw data generated is enriched with metadata from the available resources used for study documentation (SOPs, electronic lab book or other study files).

In brief, the following steps are involved: **Raw data generation** (data in different formats: .csv, .xls and pdf) > Automatic **export and storage** in Google Drive > Automatic **creation of data file for upload** (.csv) by merging the raw data and the metadata file (this step requires a metadata file containing all information regarding the identification of the study and the procedures applied) > **Upload the data** using EdelweissData™ CSV to API tool > **Visualise the data** in EdelweissData™ Explorer > **Select and copy the data and relevant data API** > **Use the data API URL for analysis** in Jupyter notebook (this step requires a pre existing workflow template, written in python or R that is able to recognise and use the data API).



*Semi-automatic data workflow from lab bench to web and analysis*

## Raw data generation and temporary storage

How is the data generated, exported and stored (temporarily) in Google Drive?

1. Set-up the reading protocol in the GloMax® Explorer software (step depending on the experimental design, e.g. endpoint, plate type, assay output, etc.).
  - a. The protocol can be saved and reused
  - b. The software offers predefined templates that can be also used

2. Run the measurement

3. The raw data is saved automatically in Google Drive:

<G:\My Drive\Projects\Laboratory\Lab Data Management\DC Lab Data\GloMax Explorer>

4. Data files exported:
  - a. **CSV file** including different details on the measurement (e.g. *unique ID, Well Position, Timestamp(ms), Optical Density*);
  - b. **Excel file** including three sheets *i)* minimal metadata on the reading, instrument used, etc. *ii)* measurement results (raw data) arranged as the original plate format (Figure 3), *iii)* measurement results (raw data) arranged by well position;
  - c. **PDF file** including similar information as in the Excel file;
  - d. **GloMax® raw file** that can be read and reused in the GloMax® software.



*Raw data files exported from the GloMax® Explorer software to Google Drive*

	1	2	3	4	5	6	7	8	9	10	11	12
A	3.79E-02	2.74E-02	3.16E-02	3.50E-02	3.41E-02	3.54E-02	3.75E-02	3.45E-02	3.24E-02	3.62E-02	3.58E-02	3.58E-02
B	4.09E-02	2.70E-02	8.31E-01	1.56E+00	2.00E+00	3.50E+00	3.49E+00	2.62E+00	3.00E+00	3.32E+00	2.78E+00	3.58E-02
C	4.26E-02	2.53E-02	9.23E-01	1.38E+00	2.23E+00	3.38E+00	3.47E+00	3.21E+00	3.59E+00	3.39E+00	2.82E+00	3.24E-02
D	4.87E-02	2.20E-02	8.01E-01	1.72E+00	2.55E+00	3.58E+00	3.41E+00	3.06E+00	3.21E+00	3.27E+00	2.48E+00	3.12E-02
E	4.35E-02	1.75E-02	9.73E-01	1.63E+00	2.68E+00	3.69E+00	3.31E+00	3.47E+00	3.03E+00	3.26E+00	2.65E+00	2.82E-02
F	4.66E-02	2.29E-02	3.14E-01	1.96E+00	2.55E+00	3.67E+00	3.42E+00	3.51E+00	3.44E+00	2.49E+00	2.84E+00	3.33E-02
G	4.31E-02	1.83E-02	9.29E-01	1.55E+00	2.45E+00	3.40E+00	2.84E+00	2.61E+00	3.47E+00	2.98E+00	2.74E+00	2.91E-02
H	4.01E-02	3.20E-02	3.28E-02	3.62E-02	3.28E-02	3.28E-02	3.24E-02	3.20E-02	3.24E-02	2.82E-02	3.50E-02	3.20E-02

Measurement results (raw data) arranged as the original plate format (example of a 96 well plate)

5. In the main data saving folder, the study owner can organise and move the files in study-specific subfolders
6. The data is ready for use in the next steps

## Enrichment of data with metadata

The metadata refers to information on the study (identification of the study) and experimental conditions applied. This information is extracted (currently manually) from the primary source of information (e.g. SOPs, electronic lab notebooks (ELN) like SciNote, files exported from the GloMax® Explorer plate reader, other study files).

The following categories of metadata is required (Figure 4):

- General information
- Test items
- Test system
- Exposure conditions
- Endpoint
- Quality control

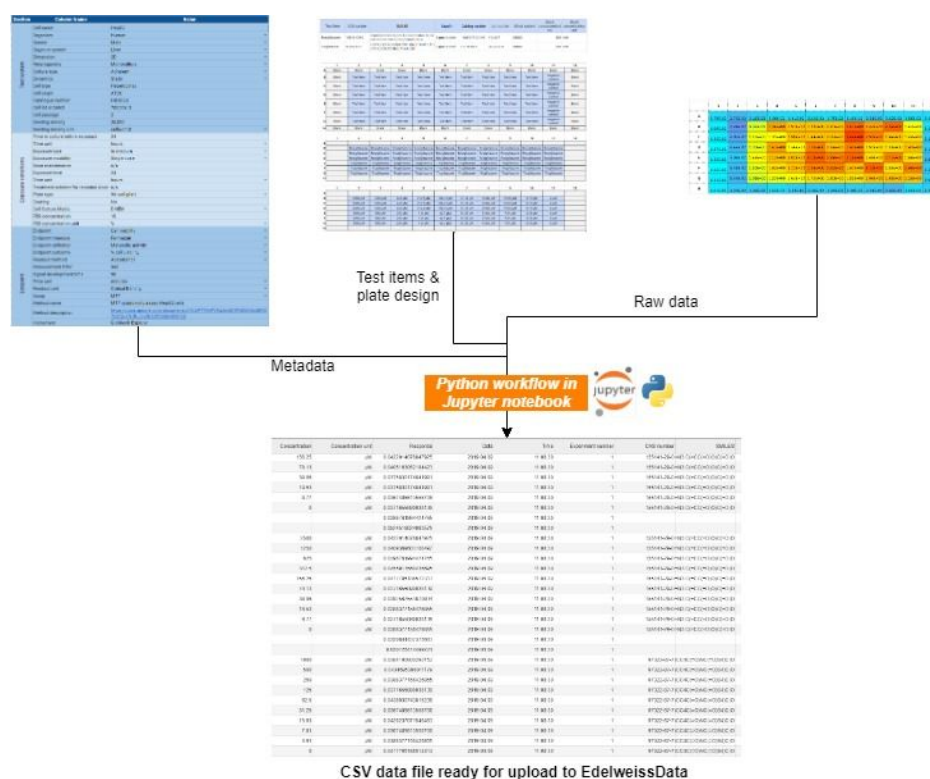
Section	Column Name	Value
Test system	Cell name	HepG2
	Organism	Human
	Gender	Male
	Organ or system	Liver
	Dimension	2D
	Heterogeneity	Monoculture
	Culture type	Adherent
	Dynamics	Static
	Cell type	Hepatocytes
	Cell origin	ATCC
	Catalogue number	HB-8065
	Cell lot or batch	70007613
	Cell passage	2
	Seeding density	30,000
	Seeding density unit	cells/cm2
Exposure conditions	Time in culture before exposure	24
	Time unit	hours
	Exposure type	In medium
	Exposure modality	Single dose
	Dose maintenance	n/a
	Exposure time	24
	Time unit	hours
	Treatment scheme for repeated dose	n/a
	Plate type	96-well plate
	Coating	No
	Cell Culture Media	DMEM
Endpoint	FBS concentration	10
	FBS concentration unit	%
	Endpoint	Cell viability
	Endpoint measure	Formazan
	Endpoint definition	Metabolic activity
	Endpoint outcome	% cell viability
	Readout method	Absorbance
	Measurement Filter	560
	Signal development time	90
	Time unit	minutes
	Readout unit	Optical Density
	Assay	MTT
	Method name	MTT cytotoxicity assay: HepG2 cells
	Method description	<a href="https://docs.google.com/document/d/10dP7T9hPVNz3wGji1PhDWh3x8BYZ7c3rQuZNDUJo-c0/edit?usp=sharing">https://docs.google.com/document/d/10dP7T9hPVNz3wGji1PhDWh3x8BYZ7c3rQuZNDUJo-c0/edit?usp=sharing</a>
	Instrument	GloMax® Explorer

*Categories and example of metadata*

## Data file

An example of [data file](#) (Named: [Cell name] [Assay] [Exposure time] [Test item] [Execution date YYYYMMDD]) is available. The spreadsheet contains all steps followed in order to prepare the file for upload (Figure 5):

- **Metadata** collected as shown above,
- **Test items** description,
- **Plate design** that shows the distribution of compounds and their concentrations in the plate type used for the experiment,
- **Raw data** generated by the GloMax® Explorer plate reader (multiple replicates).



*Creation of data file using a python workflow run in Jupyter notebook*

The merging and exporting of the metadata, data schema and raw data into a single csv file that can be uploaded to EdelweissData™ is done using a [python workflow](#) run in Jupyter notebook<sup>3</sup>. Further the upload to EdelweissData is performed (optional step).

The workflow contains the following steps:

- Set the filename of the original dataset
- Import libraries
- Check the pandas version
- Read the raw excel file & extract assay plate design, concentrations, responses
- Collect plates with responses for different experiments
- Collect properties of test items (sort of meta data)

<sup>3</sup> <https://jupyter.prod.openrisknet.org/>



- Build final dataframe

Optional steps:

- Upload to EdelweissData
- Visualize assay plates

### Edelweiss use case: storing the in-house generated raw data (version 2.0)

**Goal:** start with the laboratory generated excel file, which contains metadata and measurements of the assay plate. The script creates a csv file, which is ready to be uploaded to Edelweiss.

**Description:** the excel file contains:

- "Metadata" sheet with info regarding the study
- "Test items" sheet with info on compound properties (SMILES, CAS, etc.)
- "Plate design" sheet with three assay plate layouts (one for description of the well content, one for the compound name and one for the concentration)
- "Results raw data #" sheet(s) with the number in the name corresponding to the number of the experimental run

**To do:** generate the schema json file to support the dataset upload.

Set the filename of the original dataset

```
[1]: filename = "HepG2 MTT 24h Test item 2019009.xlsx"
```

Import libraries

```
[2]: import pandas as pd
```

Read the raw excel file

```
[3]: df = pd.read_excel(filename, sheet_name=None)
```

Define a function that adds the names for rows and columns in the assay plate.

```
[4]: def addNamesToThePlate(plate):  
    plate.index.name = "Row"  
    plate.columns.name = "Column"  
    return plate
```

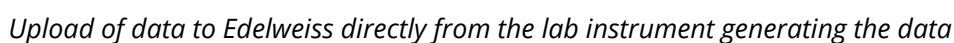
Extract the assay plate design, annotations and measurements from the raw dataset `df`.

```
[5]: plateDesign = addNamesToThePlate(df['Plate design'].iloc[0:8, 0:12])  
    plateTestItems = addNamesToThePlate(df['Plate design'].iloc[10:18, 0:12])  
    plateConcentrations = addNamesToThePlate(df['Plate design'].iloc[20:28, 0:12])
```

*Screenshot from Jupyter notebook workflow created to merge metadata and data files into a csv file ready for upload*



**Option 1.** Use of Edelweiss CSV-to-API tool<sup>4</sup> to upload the csv file generated in the previous step.



```
Collection created
Dataset created
Dataset uploaded
Successfully published dataset
TAKE ME TO DATASET
```

### Result of the automatic upload of csv file to EdelweissData

<sup>4</sup> <https://csvtoapi.edelweiss.douglasconnect.com/>

# Data visualisation and data API generation in Edelweiss Data Explorer

URL: <https://dataexplorer.edelweiss.douglasconnect.com/>

Select visible columns

Get API link

Round values

Display images

Showing: 30 of 30 results

Exposure time

Plate type

Endpoint

Assay

Execution date

Concentration

Concentration unit

Response

24h	96wp	Cell viability	MTT	11/2/2018	0	microM	1.45
24h	96wp	Cell viability	MTT	11/2/2018	0	microM	1.51
24h	96wp	Cell viability	MTT	11/2/2018	0	microM	1.44
24h	96wp	Cell viability	MTT	11/2/2018	3.91	microM	1.52
24h	96wp	Cell viability	MTT	11/2/2018	3.91	microM	1.64
24h	96wp	Cell viability	MTT	11/2/2018	3.91	microM	1.44
24h	96wp	Cell viability	MTT	11/2/2018	7.81	microM	1.59
24h	96wp	Cell viability	MTT	11/2/2018	7.81	microM	1.76
24h	96wp	Cell viability	MTT	11/2/2018	7.81	microM	1.45
24h	96wp	Cell viability	MTT	11/2/2018	15.63	microM	1.71
24h	96wp	Cell viability	MTT	11/2/2018	15.63	microM	1.76
24h	96wp	Cell viability	MTT	11/2/2018	15.63	microM	1.51
24h	96wp	Cell viability	MTT	11/2/2018	31.25	microM	1.47
24h	96wp	Cell viability	MTT	11/2/2018	31.25	microM	1.53
24h	96wp	Cell viability	MTT	11/2/2018	31.25	microM	1.5
24h	96wp	Cell viability	MTT	11/2/2018	62.5	microM	0.733
24h	96wp	Cell viability	MTT	11/2/2018	62.5	microM	0.688
24h	96wp	Cell viability	MTT	11/2/2018	62.5	microM	0.465
24h	96wp	Cell viability	MTT	11/2/2018	125	microM	0.121
24h	96wp	Cell viability	MTT	11/2/2018	125	microM	0.118
24h	96wp	Cell viability	MTT	11/2/2018	125	microM	0.103
24h	96wp	Cell viability	MTT	11/2/2018	250	microM	0.06
24h	96wp	Cell viability	MTT	11/2/2018	250	microM	0.05
24h	96wp	Cell viability	MTT	11/2/2018	250	microM	0.147
24h	96wp	Cell viability	MTT	11/2/2018	500	microM	0.0698
24h	96wp	Cell viability	MTT	11/2/2018	500	microM	0.0674
24h	96wp	Cell viability	MTT	11/2/2018	500	microM	0.0685
24h	96wp	Cell viability	MTT	11/2/2018	1000	microM	0.0891
24h	96wp	Cell viability	MTT	11/2/2018	1000	microM	0.0839
24h	96wp	Cell viability	MTT	11/2/2018	1000	microM	0.0835

Data API URL for the current selection:

https://registry.edelweiss.douglasconnect.com/data

Data visualisation in Edelweiss and data API generation

## Data analysis in Jupyter notebooks

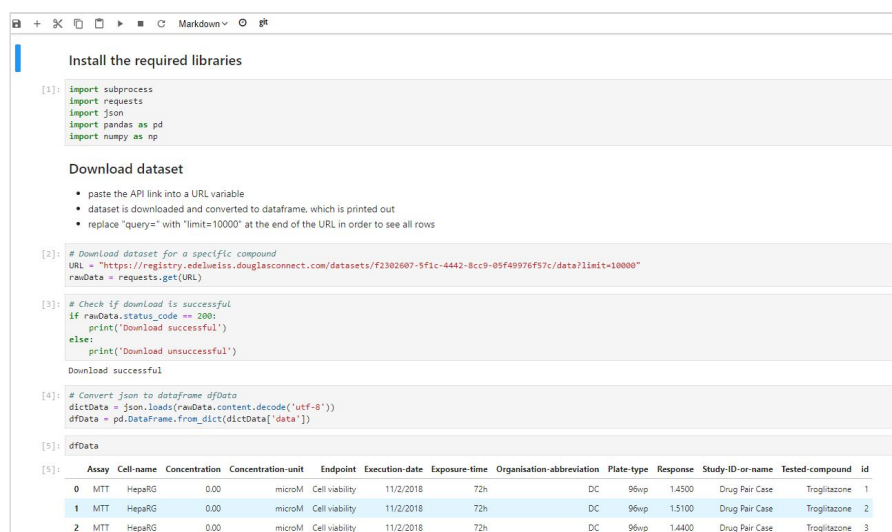
### Analysis of dose-response curves and calculation of BMDs using data API

A benchmark dose (BMD) is a dose or concentration that produces a predetermined change in response rate of an adverse effect. This predetermined change in response is called the benchmark response (BMR)<sup>5</sup>.

The analysis is performing in Jupyter notebook using drc package<sup>6</sup>, adapted to Python. The data API is generated in Edelweiss Data Explorer.

The workflow and codes used to achieve this is documented in GitHub<sup>7</sup>. The process includes the following steps (each with the relevant code):

- Install the required libraries
- Download dataset (paste the API data from link Edelweiss Data Explorer into a URL variable) and check if the download is successful (**Figure 8**)
- Prepare dose-response data: selection of the columns/rows from the dataset that will be used. The first column should contain **<concentration>** or **<doses>** and the second column should contain **<responses>**. Concentrations and responses should be in a format that is accepted by the drc script
- Prepare arguments for the R script (enter the BMR levels list for which the BMD should be calculated)
- Run the R script and generate the results in two formats (see **Figure 9** and **Figure 10**)



```
[1]: import subprocess
import requests
import json
import pandas as pd
import numpy as np

Download dataset
• paste the API link into a URL variable
• dataset is downloaded and converted to dataframe, which is printed out
• replace "query=" with "limit=10000" at the end of the URL in order to see all rows

[2]: # Download dataset for a specific compound
URL = "https://registry.edelweiss.douglasconnect.com/datasets/f2382687-5f1c-4442-8cc9-05f49976f97c/data?limit=10000"
rawData = requests.get(URL)

[3]: # Check if download is successful
if rawData.status_code == 200:
    print('Download successful')
else:
    print('Download unsuccessful')

Download successful

[4]: # Convert json to dataframe dfData
dictData = json.loads(rawData.content.decode('utf-8'))
dfData = pd.DataFrame.from_dict(dictData['data'])

[5]: dfData
```

	Assay	Cell-name	Concentration	Concentration-unit	Endpoint	Execution-date	Exposure-time	Organisation-abbreviation	Plate-type	Response	Study-ID-or-name	Tested-compound	id
0	MTT	HepaRG	0.00	microM	Cell viability	11/2/2018	72h	DC	96wp	1.4500	Drug Pair Case	Troglitazone	1
1	MTT	HepaRG	0.00	microM	Cell viability	11/2/2018	72h	DC	96wp	1.5100	Drug Pair Case	Troglitazone	2
2	MTT	HepaRG	0.00	microM	Cell viability	11/2/2018	72h	DC	96wp	1.4400	Drug Pair Case	Troglitazone	3

*Screenshot of the Jupyter notebook workflow showing the first steps and the data downloaded*

5

[https://www.chemsafetypro.com/Topics/CRA/What\\_Is\\_Benchmark\\_Dose\\_\(BMD\)\\_and\\_How\\_to\\_Calculate\\_BMDL.html](https://www.chemsafetypro.com/Topics/CRA/What_Is_Benchmark_Dose_(BMD)_and_How_to_Calculate_BMDL.html)

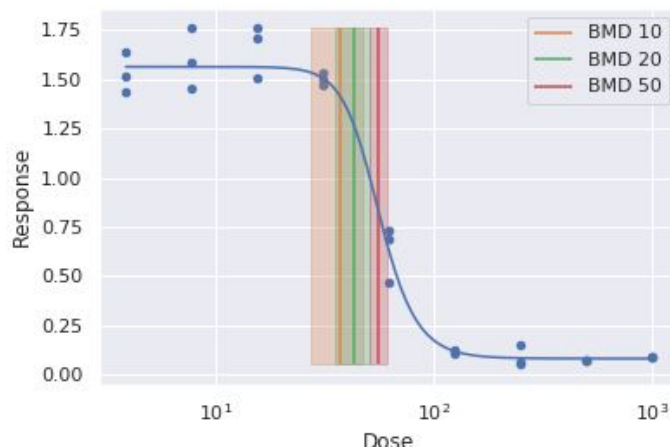
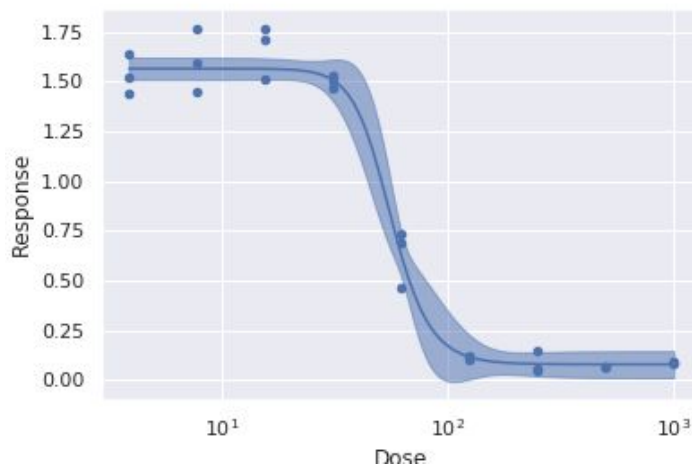
6 <https://cran.r-project.org/web/packages/drc/>

7 <https://github.com/DouglasConnect/jupyter/tree/master/DC%20Lab>

The columns used for analysis (i.e. dose response curves and calculation of benchmark doses (BMDs)) are: **<Concentration>** and **<Response>** (details on the analysis are presented below).

BMR	BMD	BMDL	Std. Error	BMDU
10	37.29482	27.59429	4.710049	46.99535
20	43.21991	35.07983	3.952378	51.35998
50	56.18424	51.1425	2.447997	61.22599

*Example of BMD results generated in Jupyter notebook (BMR = benchmark response, BMD = benchmark dose, BMDL = benchmark dose (lower confidence limit), BMDU = benchmark dose (upper confidence limit))*

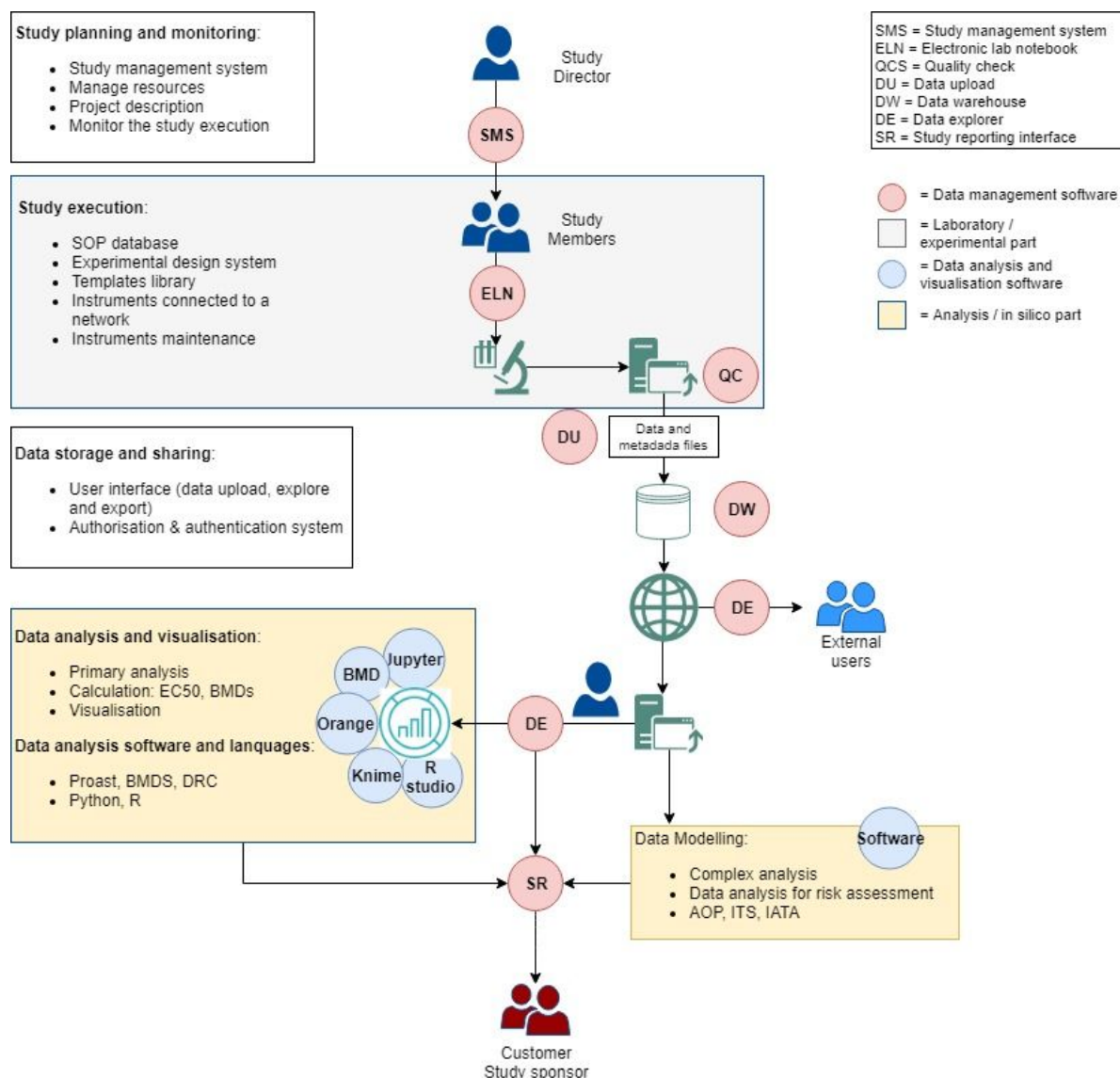


*Example of chart (dose-response curve) generated in Jupyter notebook within BMDs calculation workflow*

Workflow documentation in **GitHub**:

<https://github.com/DouglasConnect/jupyter/tree/master/Laboratory/Data%20API>

# Annexes



Overall data management from study planning, execution, data analysis and reporting