Best Practices in Research Data Management

Hauke Sonnenberg & Michael Rustler

2018-06-12 10:00:53

Contents

Chapter 1

Introduction

"These days, data trails are often a morass of separate data and results and code files in which no one knows which results were derived from which raw data using which code files."

— Professor Charles Randy Gallistel, Rutgers University

This document is the outcome of the KWB project FAKIN (Forschungsdatenmanagement an kleinen Instituten = research data management at small institutes).

It defines best practices for research data management. It is mainly based on the personal experiences of the authors having worked in many different research projects at KWB.

The document is outlined as follows:

- Chapter 2 defines Best Practices for different topics.
- Chapter 3 gives overviews about KWB projects and lists how different data related tasks have been solved within these projects.

This document is assumed to be a "living" document. We highly appreciate any comments and suggestions for improvements. What are your experiences with research data managment tasks? Can you provide solutions for specific tasks?

Example blocks:

TIP

NOTE

WARNING

CAUTION

IMPORTANT

Chapter 2

Best Practices

2.1 Data-Related Project Controlling

At the start of a research project

- Choose a project acronym and store it in `PROJECTS.txt`.
- Check if the organisations that you expect to get data from are listed in `ORGANISATIONS.txt` and extend this file if necessary.
- Create a subfolder for your project and subfolders for the organisations in the rawdata folder structure.

At the start of a project or if an employee or trainee enters the project

- Give an introduction to our research data management as described in this document.

Regularly during the project

- Check if the folder structure within your project's rawdata subfolder still complies with the rawdata folder structure and clean the structure, if not.

2.2 Naming Conventions

2.2.1 Acronyms

Acronyms are unique, clear names for objects. They should

- be short but meaningful and easy to remember,
- be all lowercase,
- consist of only alphanumeric letters (`a-z`, `0-9`) or the hyphen (`-`).

2.2.1.1 Acronyms for Projects

At the start of a project we define a project acronym. This acronym is intended to be used in file and folder names.

Whenever we want to indicate the relation to a certain project in a file or folder name, we use the project acronym in exactly the typing that was defined. This is important as we want to distinguish between raw data, processed data and project results in our Folder Structures.

The project acronyms are defined in a simple text file `PROJECTS.txt` in the `//server/projects$` folder, see Project Folder Structure.

2.2.1.2 Acronyms for Organisations

It is very important to know the owners of data. Therefore we define unique acronyms for the owners of data that we use. The acronyms are defined in a special file `ORGANISATIONS.txt`

2.3 Folder Structures

2.3.1 Project Folder Structure

We said that we want to concentrate on the folder structures within the project folders. Nevertheless, we would like to give a recommendation on how the project folders could be organised within its top level folder. In this structure, there are no subfolders for the different departments any more.

```
//server/projects$
- PROJECTS.txt
- project-1/
- project-2/
- project-3/
```

In the `projects$` folder

- each subfolder name should appear in the file `PROJECTS.txt`
- there should not be any folder on the top level that does not represent a project.
- there should be no other files on the top level as the files that are described in this documentation.
- there are no subfolders representing departments any more. The mapping of projects to departments is done in the file `PROJECTS.txt`

2.3.2 Rawdata Folder Structure

We will create a network folder `//server/rawdata$` in which all files have set the read-only property. We suggest to store raw data by project first and by the organisation that owns (i.e. generated, provided) the data second. This could look like this:

```
//server/rawdata$
- ORGANISATIONS.txt
- PROJECTS.lnk [Symbolic Link to PROJECTS.txt in //server/projects$]
- flusshygiene
  - bwb
  - kwb
  - uba
  - ...
- ogre
  - kwb
  - bwb
  - uba
  - ...
- ...
```

Restrictions/Conventions:

- Each top-level folder should represent a project, i.e. should be defined in the top level file `PROJECTS.txt`.

- Each possible owner should be defined in the top level file `ORGANISATIONS.txt`.
- The naming convention for the organisations is the same as for projects.

2.4 Raw Data

As raw data we define data that we receive from a device or from a project partner.

Most of our research results are based on data. We acknowledge the importance of raw data by

- storing then in a special place where it is specially secured
- describing them with metadata

Rawdata are stored in the rawdata folder structure

2.5 Metadata

2.5.1 Special Files

We propose to define some special files that contain metadata related to files and folders. To indicate that these files have a special meaning, the file names are all uppercase.

2.5.1.1 File PROJECTS.txt

This file contains the project acronyms as we want to use them e.g. as top-level folder names in our project folder structure. The projects are grouped by department.

Possible content of `PROJECTS.txt`:

```
# Department SUW (Surface Water)
dswt: DSWT
flusshygiene: Flusshygiene
kuras: KURAS
mia-cso: MIACSO
monitor: MONITOR
ogre: OGRE
reliable-sewer: RELIABLE_SEWER
sema: SEMA
sema-berlin: SEMA Berlin
sema-berlin-2: SEMA Berlin 2
spree-2011: SPREE2011
spree-2011-2: SPREE2011 "reloaded"

# Department GRW (Groundwater)
optiwells: OPTIWELLS
optiwells-2: OPTIWELLS 2
wellma: WELLMA

# Department WWT (Wastewater Treatment)
...
```

In the file `PROJECTS.txt` the project acronyms appear in alphabetical order. They map the acronym to a project name or a project title and the year of the start of the project.

Question: Do we already have a place where "official" metadata about projects are stored? If yes, the acronym could be included there. But then, everybody should know about it!

2.5.1.2 File ORGANISATIONS.txt

Possible content of `ORGANISATIONS.txt`

```
bwb: Berliner Wasserbetriebe
kwb: Kompetenzzentrum Wasser Berlin
uba: Umweltbundesamt
```

2.6 Data Processing

2.6.1 Automatic Data Import into R

In the following we describe how data can be imported into the R-Programming Environment

2.6.1.1 Import Data From One Excel File

- Save the original file in the rawdata zone.

2.6.1.2 Import Data From Many Excel Files

2.6.1.2.1 Files Are In the Same Format

Import Excel files of the same format by

- defining a function that is able to read the data from that file
- calling this function in a loop for each file to import.

2.6.1.2.2 Files Are In Different Formats

We developed a general approach of importing data from many Excel files in which the formats (e.g. more than one table area within one sheet, differing numbers of header rows) differ from file to file.

Chapter 3

Projects

3.1 Spree2011 (2007)

Used data by source (data formats in parentheses)

- KWB:
  - water level and discharge at one monitoring site (Text/CSV)
  - rain (Text/CSV)
- BWB:
  - pumping rates in the pumping stations (Excel)
  - water levels in the pumping stations (Excel)
  - rain at some gauges near the monitoring site (Excel)

Tasks and methods by topic

- Dry-weather and wet-weather calibration of a sewer network model (Infoworks)
  - InfoWorks: Creating rain input files
  - InfoWorks: Creating RTC input files

Questions that arose:

- Where to store presentations (trainee vs. employee)?
- Where to store the raw data (personal drive of the trainee)?
- How does Infoworks interpret timestamps, how do BWB provide timestamps? -> metadata

3.2 MIACSO (2009)

Monitoring

- sites: one site in the sewer (monitoring container), more sites in the river
- variables: water quantity and quality
- devices: online sensors

Modelling

- Sewerage: Infoworks
- River hydraulics: Hydrax
- River quality: QSim

Data storage

- High amount of data -> extra server: `moby`
- We put some effort in planning good folder structures for the data. Nevertheless the structure at the end of the project is not as clean as it was planned.
- Data that we received from project partners was stored in `Daten/EXTERN`.
- Raw data was stored in a folder `Daten/RAW` which was write-protected and required a special user-account for storing new data.

```
Daten/
  ACCESS/ # MS Access databases, containing raw data
  EXTERN/ # External data (by organisation)
  META/   # MS Access databases, containing metadata
          # (about calibration, maintenane, sites, variables)
  RAW/    # Text files containing raw data, from KWB-own devices only, by site
```

Metadata

- many devices in the container -> meta data about device cleaning and maintenance important -> tool: `META_Maintenance.mdb`

Methods and Tools

- We imported most of the data from text files into MS Access databases in -> tool: `MiaCsoRawImport.mdb`
- We calibrated the sensors offline by using SQL queries to provide calibrated data from raw data -> tool: `MiaCsoMetaCalibControl.mdb`
- We used SQL queries to perform data processing -> tool: `MiaCsoStatAnalysis.mdb`
- Data validation (outlier detection) was done in a two step procedure:
  1. Automatic preselection using MS Access tool `MiaCsoStatAnalysis.mdb`
  2. Manual selection using self-developed graphical tool in Origin

Developed Tools:

- MS Access Applications
  - `MetaMaint.mdb`: Monitoring Metadata Management
  - `MiaCsoRawImport.mdb`: Text File Import to MS Access
  - `MiaCsoStatAnalysis.mdb` (project deliverable): Definition and automatic execution of sequences of SQL queries
- Origin extension to interactively select and store outliers graphically
- R packages
  - `kwb.mia.evalCritO2` (project deliverable): graphical evaluation of critical oxygen conditions in the river
  - `kwb.mia.iw`: Calculation of file sizes of InfoWorks result csv-files exported from InfoWorks.
  - `kwb.miacso`: functions used in MIA-CSO, for example for plotting data availabilities.

3.3 KURAS

Developed Tools:

- Frontend for KURAS Database of Rainwater Management Measures: `KURAS_DB_Acc2003_hs.mdb`
- R package `kwb.kuras`: Interface to KURAS database

3.4 OGRE

- Decision to use CUAHSI Community Observations Data Model (ODM)
- R script to import lab data from Excel to MS Access database implementing ODM

Developed Tools:

- R packages
  - `kwb.ogre`
  - `kwb.ogre.model`
  - `kwb.odm`
  - `kwb.odmx`

3.5 Flusshygiene

- Adaptation of free online monitoring data visualisation HydroServerLite
- Reusage of lab data import script developed in OGRE

3.6 DEMOWARE

Entstandene R Pakete:

- Grundwassermodellierung
  - kwb.hantush
  - kwb.vs2dh
  - kwb.demoware
- Quantitatives mikrobiologisches Risikomanagement
  - kwb.qmra: wird im Rahmen von AQUANES(#aquanes) weiter genutzt

3.7 OPTIWELLS

Created R packages:

- kwb.wtaq: Groundwater Modelling
- kwb.epanet: (Pressure)Pipe Network Simulation (EPANET)

3.8 RWE

(Semi)automatisierte Erstellung eines komplexen MODFLOW Modells (mehrere Layer, mehrere hunderte Entnahmebrunnen mit zeitlich variierender Entnahmemenge sowie Hinzufügen/Entfernen von Brunnen innerhalb des Simulationszeitraums) in Phyton mittels flopy sowie Entwicklung der Modellszenarien auf Github (siehe hier: Maxflow).

Input Christian !?!

3.9 AQUANES

Created R package:

- aquanes.report: data import, temporal aggregation, interactive visualisation of operational data of pilot facilities and joining with lab data

Sites (among others):

- Berlin-Tiefwerder
- Berlin-Schönerlinde
- Basel Lange-Erlen
- Haridwar

Challenges:

- zeitlich hoch aufgelöste Betriebsdaten (~ 10 Mio. Datenpunkte pro Monat) erforderten massive Performance Optimierung um die Visualisierung der Rohdaten im Tool über den Testbetriebszeitraum (18 Monate) auf Rechnern mit limitierten RAM Ressourcen (~ 8 GB) zu ermöglichen.
- Nutzung: wird von den Projektpartnern zum strukturierten Datenimport genutzt um dann darauf ggf. eigene weitere Analysen darauf aufzusetzen. Für die beiden Berliner Standorte wird zudem die Visualiserung der Betriebsdaten routinenäßig genutzt, um vom Regelbetrieb abweichende Zustände besser indentifizieren zu können.