# Data management procedures for reproducible research pipelines

Ivan C. Hanigan

**Abstract**

This unpublished working paper was written to accompany the material included in the PhD thesis 'Using Reproducible Research Pipelines to Help Disentangle Health Effects of Environmental Changes from Social Factors'. It sets out the key data management and analysis principles that were found to be most effective for the reproducibile synthesis and integration of heterogeneous datasets for analysis and reporting. The draft was last updated August 18, 2016. The version submitted with the thesis is available on the phd_appendix branch of the Github repository: https://github.com/swish-climate-impact-assessment/swish_data_management_procedures.

## Contents

# 1 Introduction

There is a need for developing an evidence-based set of best practice guidelines for data management procedures that support reproducibility in all fields of computational data analysis (Long, 2008; Noble, 2009; Peng, 2015). Reproducibility is the ability to recompute the results of a data analysis with the original data (as distinct from replication which involves analysing independently collected data (Peng, 2011)). The examples drawn together in this report come from experiences and use-cases found from implementing reproducible research pipelines in an eco-social epidemiologic research context. This emerging paradigm mixes environmental and social epidemiology and is inherently concerned with complex systems. To do this work integration of heterogenous data sources, and synthesising new datasets, is required. Then analyses that aim to recognise subtle and complicated patterns in the environmental and social determinants of health must be rigorously and transparently conducted (McMichael, 2013). This document outlines a suite of data management procedures that have been found to effectively assist the development of reproducible research pipelines in this context.

## 1.1 The 'reproducibility crisis'

All data analyses are reproducible to a varying degree of difficulty. A data analysis might be reproducible but require thousands of hours of work. A primary challenge for reproducible data analysis is to make analyses that are *easy* to reproduce.

In essence this requires attention to be turned to the issue of how the data and analytical steps amassed – toward a reality where this is archived and there is a good understanding all round as to how the study were set up and conducted. Different assumptions or different treatment of the data could conceivably lead to different inferences and conclusions being drawn, such as in the example shown by Silberzahn and Uhlmann (2015) in which 29 research teams were given the same dataset but reached a wide variety of conclusions using different methods on the same dataset to answer the same question.

This is partly because of an underlying complexity in the information drawn from complex systems involving multi-causality, and partly because of different assumptions and different backgrounds and viewpoints. A finding that a variable does or does not cause a disease, might be drawn honestly from the same set of data.

## 1.2 A common (flawed) approach for generating statistical reports

A common approach (with inherent flaws that make it error-prone) was identified by Scott (2010), and the examples are paraphrased here. First: the data entry, cleaning, preparation and possibly statistical analyses are conducted by 'point-and-click' procedures using spreadsheet software such as Microsoft (MS) Excel. This introduces well-known issues with handling of dates, coloured cells, hidden columns, missing data, poor algorithms and unreliable results (McCullough and Heiser, 2008). In cases where data are imported to a program such as STATA or SPSS, further point-and-click data preparation and statistical analyses often occur. Post-analysis, spreadsheets such as MS Excel are regularly used to record or format the desired results, and generate figures. Finally, the results (text, tables and figures) from the data analysis system are inserted into a word processor (eg, MS Word) using 'copy-and-paste' procedures (or typed by hand).

Problems with this common, flawed approach according to Scott (2010) are:

- You sit down to finish writing your manuscript. You realize that you need to clarify one result by running an additional analysis. You first re-run the primary analysis. Major problem: the primary results don't match what you have in your paper.
- When you go to your project folder to run the additional analysis, you find multiple data files, multiple analysis files, and multiple results files. You can't remember which ones are pertinent.
- Youv́e just spent the week running your analysis and creating a results report (including tables and figures) to present to your collaborators. You then receive an email from your PI asking you to regenerate the report based on a subset of the original data set and including an additional set of analyses – she would like it by tomorrowś meeting.
- With point and click programs there is no way to record the steps performed that generated the documented results.
- Common to keep analysis code, results, and reports as separate files and to save various versions of each of these as separate files. After several modifications of one or more of the files involved, becomes unclear which version of the files exactly correspond to the desired analysis and results.
- Every time analyses and/or results change, have to regenerate the results report by hand – very time consuming.
- Very easy for human error to creep into results report (eg, typing in results by hand, copying/pasting the wrong tables/figures).

### 1.3 Reproducible research reports: A better alternative

It is widely recommended that a better approach is to create Reproducible Research Reports (RRR) (Healy, 2013). This embeds the analysis into the report so that the code to clean and prepare the data or to perform the desired statistical analysis is included in the document that contains the documentation and text of the report. Solutions have been developed that combine both the data analysis code and the descriptive prose that constitutes the publishable report into a compendium (Gentleman and Temple Lang, 2004; Schulte et al., 2012).

Reproducible research reports are written using a scripting language for statistical computing and graphics. The report is made up of ordinary text written in a suitable format that enables the computational process to recognise it as text. An example is the Rmarkdown format which is very similar to text used when authoring word processor documents (http://rmarkdown.rstudio.com). There are also chunks of pure statistical programming code (such as R codes) that perform data manipulations and analyses when the document is 'evaluated'. When the processing stage is run a report document is generated that includes both content as well as the output of any embedded computer code 'chunks' within the document. These are distinguished from the regular text by a special delimiter at their beginning and end. An example using the R language is presented below:

````
---
title: "Reproducible report example"
author: "Ivan C. Hanigan"
output:  pdf_document
---
# Some exploratory analysis
In this section we do some exploratory analysis of the NMMAPS data for
deaths in Chicago 1987-2000. The code, messages and intermediary
results are hidden in the resulting report document.
```{r, echo = FALSE, message = FALSE}
## We begin by reading in the data file:
## If using our own data we would use 'read.csv' or a similar tool to import data to R
# my.data <- read.csv('data/sampledata.csv',header=TRUE)
## for this example use data that are included in the dlnm package
library(dlnm)
# look at the structure of the data
# str(chicagoNMMAPS)
# summary(chicagoNMMAPS)
```
We made a simple scatter plot shown below
```{r, echo = FALSE, message = FALSE}
## make some plots. first by day
# with(chicagoNMMAPS, plot(date, cvd, type = "l"))
# we suspect a relationship between temperature and deaths
with(chicagoNMMAPS, plot(temp, cvd, pch = 16, cex = .6))
title(main = "A scatter plot of daily temperatures against deaths")
```
We ran some exploratory models. A Poisson GAM with smooth functions
on temperature and time was compared to a linear fit on temperature.
```{r, echo = FALSE, message = FALSE}
library(mgcv)
fit1 <- gam(cvd ~ s(temp) + s(time), data=chicagoNMMAPS, family = "poisson")
# we can access post-estimation summary statistics
summary(fit1)
# do model testing to confirm that the error terms are distributed as assumed
gam.check(fit1)
# or just plot the exposure-response function
plot(fit1, select = 1)
title(main = "The exposure-response function estimated using MGCV")
aic1 <- AIC(fit1)
# compare this to a linear term for temperature
aic0 <- AIC(gam(cvd ~ temp + s(time), data=chicagoNMMAPS, family = "poisson"))
# calculate the delta aic
aici <- aic1 - aic0
```
The result can be automatically inserted to the text. This model has
a delta AIC of `r round(aici,1)` (smoothed minus linear term).
````

The result can be seen at this link (https://github.com/swish-climate-impact-assessment/swish_data_management_procedures/blob/phd_appendix/Reproducible_report_example.pdf). In this example the Rmarkdown engine is used to construct the final report by 'weaving' together the prose and the code. The prose is written in 'markdown' which is a simple way to use 'markup' commands to tell the program to do formatting on the inputs. For example the first 'hash' (#) symbol tells the program that this line should be written in the style of heading-1. The three 'backtick' marks (') tell the program that the following text should be interpreted as R code. Inside these 'chunks' the 'hash' symbol is interpreted as a comment and the line is not executed.

If an analysis is published as a reproducible report, readers can have greater confidence in the work that was done, and verify this for themselves if questions remain. However, it has also been recognised that reproducible research can still be wrong and a prevention approach has been recommended that incorporates evidence-based data analysis tools and techniques into such a pipeline (Leek and Peng, 2015a).

## 1.4 Reproducible Research Pipelines (RRP) defined

The techniques of pipelines described here are targeting the integrity of the process of data selection, the robustness and suitableness of the methods used, a commonsense and well-argued selection of health outcomes and environmental or social exposures, and the clarity and transparency of the methods used.

To achieve this, a guiding principle is that analysts should effectively implement 'pipelines' of method steps and tools. Standardised and evidence-based methods based on conventions developed from many data analysts approaching the problems in a similar way should be used, rather than each analyst configuring a pipeline to suit particular individual or domain-specific preferences (Borer et al., 2009; White et al., 2013).

Noble (2009) points out that 'the principles behind organizing and documenting computational experiments are often learned on the fly, and this learning is strongly influenced by personal predilections'. Leek and Peng (2015b) describe this as data analysis being 'taught through an apprenticeship model, and different disciplines develop their own analysis subcultures'. By codifying what an appropriate pipeline would contain, data analysis will be more robust. According to Peng (2015), there should not be a 'lonely data analyst' coming up with their own method. If a researcher conducted an analysis using an evidence-based reproducible research pipeline 'you could at least have a sense that something reasonable was done' (Peng, 2015).

## 1.5 The core components of a pipeline

The core concepts and flow of steps in the method are shown in Figure 1 (after Peng et al. (2006) and Sólymos and Fehér (2008)). In this model there are two main actors: the author and the reader. The author moves from left to right, from initial hypothesis and study design, through data collection and pre-processing, to analysis and reporting. The aim is to conduct all steps of the analysis work in such a way that the key dataset and code script can be sent through the distribution mechanism and the reader can easily move from right to left. Thus the reader can start with the published results and then dig deeper by assessing the analysis code and analytic data to gain full understanding of the methods steps.

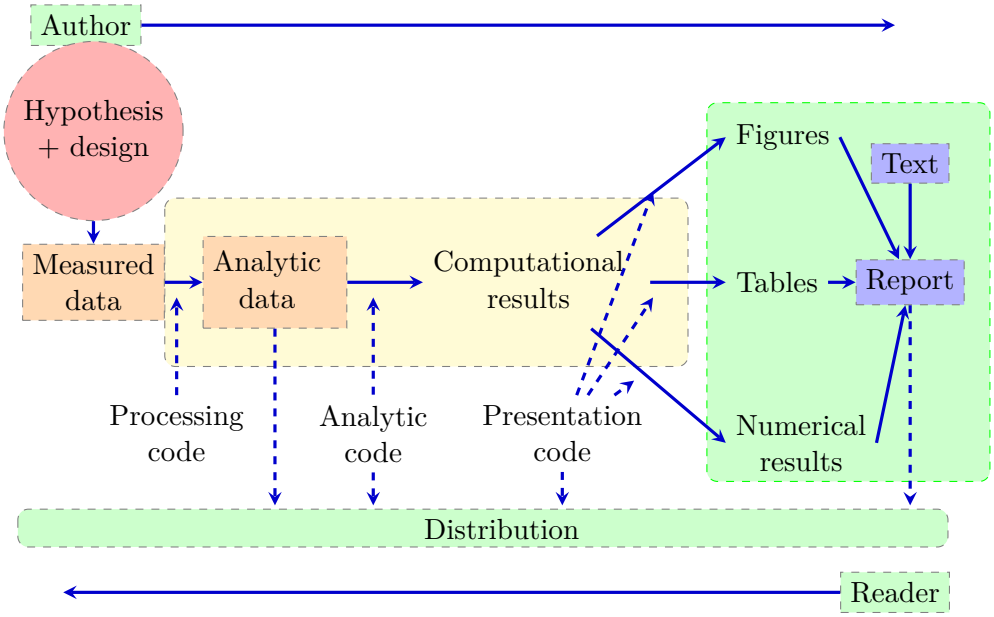


Figure 1: A schematic diagram representing the reproducible research pipeline

Peng et al. (2006) distilled a core set of components for reproducibility from earlier work including that of Schwab et al. (2000). These are:

- Hypothesis and design
- Data (measurement, pre-processing, analytic)
- Analysis Methods
- Documentation (of all steps)
- Distribution (of the paper, data and code).

### 1.5.1 Hypothesis and design

The first stage of the pipeline is hypothesis generation and study design. In this stage documentation should explain the literature base supporting the study, the decisions made in selection of explanatory factors for inclusion, decisions made such as the experimental unit, observational unit, measurement method, as well as spatial or temporal extent. This information will also be needed for ethical review and approval.

### 1.5.2 Data

The data that were measured should be well managed, however the requirements for accessing the original raw data are less important than for the analytical dataset. Descriptions of how the measured data were transformed into the analytic data should be available. Public data repositories or institutional services such as university libraries should be used to ensure longevity of the data storage.

### 1.5.3 Methods

The software code underlying the principal results needs to be made available. In addition, the computer environment necessary to execute that code should be described adequately to 'deploy' a new computer set-up that can reproduce the computations needed.

### 1.5.4 Documentation

Adequate documentation of the code and data should be available to enable others to repeat the analyses and to conduct other similar ones. This can take the form of metadata, reports, journal papers or even books (Peng and Dominici, 2008). Indeed textbooks on statistical methods can benefit greatly from being accompanied by data and analytical code to enhance their pedagogic functions (Barnett et al., 2014; Barnett and Dobson, 2010).

Inadequate documentation can lead to data misuse. This may be due either to honest misunderstandings about data attributes (no dataset is perfect and self-explanatory, see Michener et al. (1997)) or intentional misuse for malicious or selfish reasons (for example the misuse of data by Bjorn Lomborg to support the argument that environmental health conditions are actually improving. See Bodnar et al. (2004) for a discussion on Lomborgś misuse of data). There have also been notable examples of mistakes in data analyses used for climate change science. See Cai et al. (2010) for a discussion of one such case. The careful storage and curation of datasets is also critical because data from many studies are lost (Pullin and Salafsky, 2010; Vines et al., 2014).

An important underpinning to reproducible research is the reproducible report. This is the ultimate form of documentation because the information that represents the outputs of the research is written alongside the code that performs the computations that are being described. There has been many recent advances made in terms of tools for reproducible reports such as R markdown and knitr (Xie, 2014).

Metadata should be created and maintained as a priority task at all stages of the data analysis process. An international standard should be preferred over selectively choosing what information one collects and what fieldnames one uses to describe each item of documentation. Ecological Metadata Language (EML) and the Data Documentation Inititative (DDI) are two such standards that offer useful semantic constructs for describing epidemiological data.

Time and effort may be saved by considering metadata requirements at the commencement of a study, rather than trying to recall all the details later. If metadata adheres to a standard schema, it can be used in catalogues to enable fast searching and retrieval, or in machine-to-machine data queries that assist data access and use.

### 1.5.5 Distribution

Distribution or dissemination of the material needs to use a standard method if they are to be used by others. It is not enough just to provide access to the software and data, but also adequate documentation is required to explain and potentially assist downstream users to piece these together.

## 2 Principles for organising projects, datasets and files

For data to be reused in the future, files and related documents need to be carefully managed to allow future users (including the original collector) to find and understand them. A formalised approach to data management should be developed, and following a widely agreed 'standard' if possible. This example shows the use of the Ecological Metadata Language (EML) concepts of Projects, Datasets and Entities. EML was developed primarily for creating metadata and allows sufficient detail to describe the collection process and record decisions that were made during the creation of the data. In EML the elements of any dataset can be seen as a nested hierarchy at three levels shown in figure 2.

1. The Project level: this is an overarching grouping of data. It might be indicative of the principal investigator or organisation who provided the data, or a programme of research studies (sub-projects).
2. The Dataset level: this is a distinct grouping of data that might be organised around a particular time period or geographical region.
3. The Entity level: This grouping of data includes data files (such as tables in CSV or Excel, shape-files and raster images) or documents (such as metadata descriptions or related publications).

This conceptual framework can be very useful for the organisation of the work constituting a single pipeline, as well as when working with multiple pipelines within several projects.

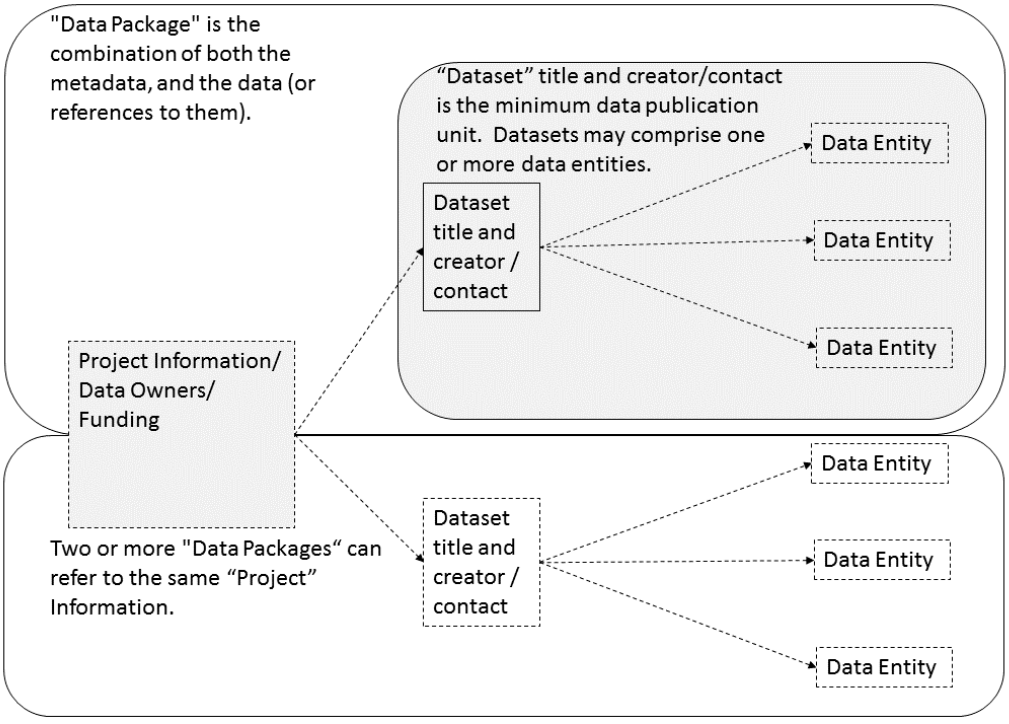


Figure 2: The EML approach to managing Projects, Datasets and Entities

### 2.1 Procedures when conducting a reproducible research analysis

Having defined above the principle components for a pipeline there are procedural questions about how to go about compiling those. The key steps include:

- Data Management Plans and Data Inventories
- Planning and implementing a pipeline
- Tracking method steps

These three topics will be explored in each of the next three sections of this report.

## 3 Data management plan and data inventory

In eco-social epidemiology there is a need for a data management plan and a data inventory that enables individual scientists, or multidisciplinary teams of scientists, to manage large and heterogeneous collections of disparate data sources efficiently. Keeping track of all the elements of a linked health, social and environmental database is very challenging, despite major improvements in data management software, web-portals and virtual laboratories (Fleming et al., 2014).

Effective data management policies and procedures are essential in managing data-related risk. Such risks include data loss or corruption, technological obsolescence, breaches of privacy or copyright, and errors or misuse.

Data management plans are needed for developing procedures and processes to keep data safe. There is an issue when ensuring that all relevant data are collected in deciding what is relevant. Keeping an up-to-date data inventory and careful organisation of all folders and files helps mitigate these problems.

Whether data management is the responsibility of the individuals collecting or collating it, or of the lead scientist, clarity on how and where data are stored and who manages it is vital, as is a 'succession plan' that sets out the vision of the data collections preservation and re-use into the future.

### 3.1 Data storage and access

Some datasets such as sensitive personal information about suicide or climate change scenarios with restrictions due to privacy and confidentiality rules, or because of protected intellectual property, need to be accessed in a restricted way. This complicates the implementation of the method of pipelines which dictates that all the steps, models and assumptions need to be made transparent and available for scientific debate even though the datasets may require authorisation to access. Restrictions around access to data have increased recently in Australia. As an example the custodians of the national mortality database made it virtually impossible to access these data for several years after the discovery of an incident in which Australian population health researcher Dr Stephen Begg was reported to have hacked into the database in an illegal act (O'Keefe, 2007). The subsequent investigation by the data custodians led to a wide ranging modification to the procedures for approval and provision of these data that make the access much more restricted. Appropriate access to data is therefore required to address this issue. In the work reported in the conference presentation in this thesis, a range of available workflow tools for data management and analysis were investigated and developed.

The key components of the data management system as identified above are shown graphically in figure 3. A project can contain many datasets (folders), which in turn may hold many entities (files). This organisation model for a project can be described in terms of 'general' projects (about general contextual data, accessible to the entire group) or 'specific' projects (about a specific Health/Exposure study and will have a specific subset of authorised people who can access it). General data can go into the main storage folder however specific data needs to have more secure storage.

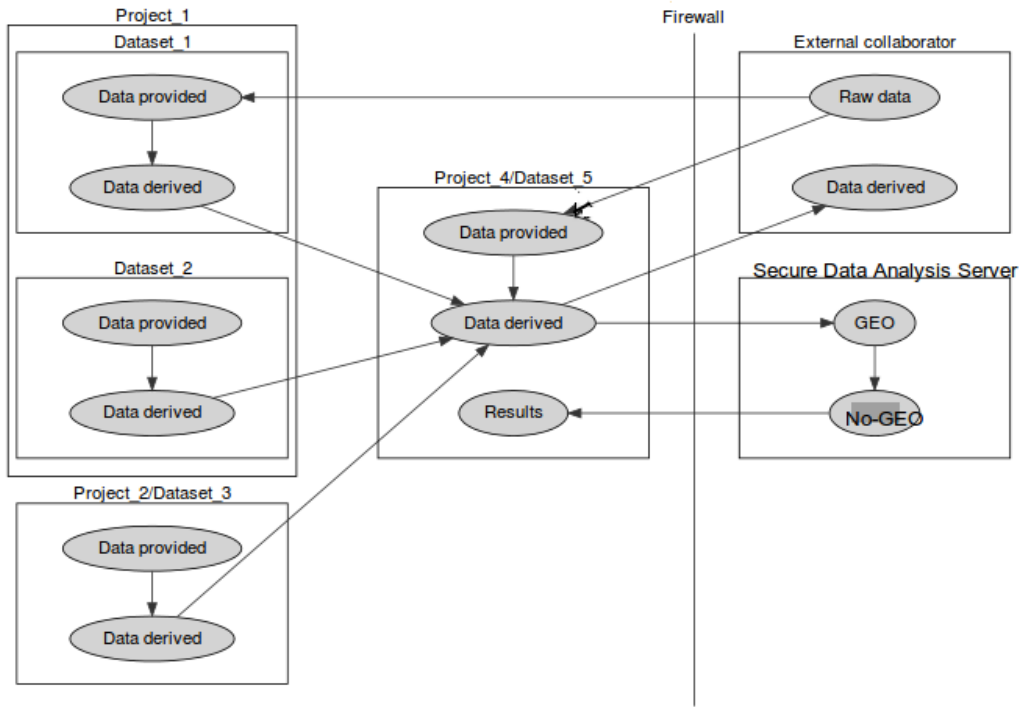


Figure 3: A schematic diagram of the management of large multi-institute collaborative project

### 3.2 Case study 1: EML and folder structure

In figure 4 the conceptual framework described above is implemented in a standardised folder structure. The main storage location for this data collection is called 'Research data'. This computer drive is then structured in a simple hierarchy of projects (folders), datasets (sub-folders) and entities (sub-sub-folders or individual files). Entities may be individual files, or groups of files in a sub-sub-folder to cater for data structures such as those where there are a collections of files that make up a single entity such as the Shapefile or Raster Image dataset as used in Geographical Information Systems (GIS) software. As seen in figure 3 it might make sense to group all entities into folders that delineate those files provided as raw data and those that were derived by some process within the project.

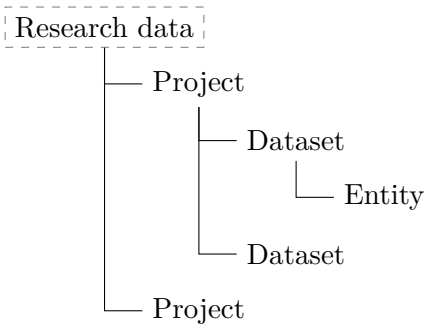


Figure 4: Conceptual framework for grouping data files (entities) within datasets and projects

## 4 Planning and implementing a pipeline

It can be much easier to conceptualise a complicated data analysis method than to implement this as a reproducible research pipeline. The most effective way to implement a pipeline is by methodically tracking each of the steps taken, the data inputs needed and all the outputs of the step. If done in a disciplined way then the analyst or some other person could 'audit' the procedure easily and access the details of the pipeline they need to scrutinise.

### 4.1 A standardised data analysis pipeline framework

One method that was selected for use in the papers of this thesis was the concept of the Load-Clean-Functions-Do (LCFD) framework. This was first proposed by Josh Reich on the open-source software discussion forum called 'stack overflow' (http://stackoverflow.com/a/1434424), and then encoded into the 'makeProject' R package (http://cran.r-project.org/web/packages/makeProject/makeProject.pdf). The approach is demonstrated in case study 2 below.

### 4.2 Case study 2: Simple pipeline using the makeProject package

```
# in an interactive R session at the command line choose your project directory
setwd("~/projects")
# load the required functions from the makeProject package
library(makeProject)
# use the makeProject function to
makeProject("my_first_pipelines_project")

### gives
/my_first_pipelines_project/
    /code/**.R
    /data/
    /DESCRIPTION
    /main.R

# in main.R you put these lines into the script
# and run them as the steps of the pipeline evolve
source("code/load.R")
source("code/clean.R")
source("code/func.R")
source("code/do.R")

# Reporting is then a matter of choice
## If using the rmarkdown approach there would be an Rmd file that contained the prose
## and turned into a PDF, HTML or Word document with a line such as
rmarkdown::render("My-Pipeline-Report.Rmd", "pdf_document")
```

### 4.3 File organization and naming

In many stages of a pipeline, an analyst will want to include details of the settings or what dataset they started out with. Rather than saving a folder or file name that is long and uninformative there are many different ways to organizing folders and files.

Key techniques for this are available and known in the data analysis community as 'Tidy Data' guidelines. In the words of Wickham (2014) the order that data should be arranged in follows some generic principles:

> 'A good ordering makes it easier to scan the raw values. One way of organizing variables is by their role in the analysis: are values fixed by the design of the data collection, or are they measured during the course of the experiment? Fixed variables describe the experimental design and are known in advance. Computer scientists often call fixed variables dimensions, and statisticians usually denote them with subscripts on random variables. Measured variables are what we actually measure in the study. Fixed variables should come first, followed by measured variables, each ordered so that related variables are contiguous. Rows can then be ordered by the first variable, breaking ties with the second and subsequent (fixed) variables.'

### 4.3.1 An exemplar

The following protocol was developed for an ecology and biodiversity database that the author of this PhD thesis was involved with. The naming convention relied heavily on a sequence of information being used to order the names of folders, subfolders and files. This is:

1. The project name (and optional sub-project name)
2. Data type (such as experimental unit, observational unit, and/or measurement methods)
3. Geographic location (State, Country)
4. Temporal frequency and coverage (such as annual or seasonal tranches).

### 4.3.2 The concepts of slow moving dimensions and fast moving variables

The concept of dimensions and variables can be useful here, and especially for deciding on filenames. Dimensions are fixed or change slowly while variables change more quickly. By 'change', this means that there are more of them. For example the project name is 'fixed', that is it does not change across the files, but the sub-project name does change, just more slowly (say there may be 2-3 different sub-projects within a project). Then there may be a set of data types, and these 'change' more quickly than the sub-project name. Then the geographic and temporal variables might change quickest of all.

So a general rule for the order of things can be stated. The fixed and slowly changing variables should come first (those things that don't change, or don't change much), followed by the more fluid variables (or things that change more across the project). List elements can then be ordered so that the groups of things that are similar will always be contiguous, and vary sequentially within clusters.

An example is shown in Table 1 to describe this and make it easier to understand. Here is a set of file names that were constructed for an ecological field sites project that I worked on (http://www.supersites.net.au/). That project involved ecological data sampled at plot-based measurement locations. At the begining of the procedure a controlled vocabulary of data types and their acronyms was created.

Table 1: An example of standardised filename conventions to simplify tracking complicated datasets

| Filename | Title |
|---|---|
| asn_fnqr_soil_charact_robson_2011.csv | Soil Data, Far North Queensland Rainforest SuperSite, Robson Creek, 2011 |
| asn_fnqr_soil_pit_robson_2012.csv | Soil Pit Data, Water Content and Temperature, Far North Queensland Rainforest SuperSite, Robson Creek, 2012 |
| asn_fnqr_veg_seedling_robson_2010-2012.csv | Seedling Survey, Far North Queensland Rainforest SuperSite, Robson Creek, 2010-2012 |
| asn_fnqr_veg_seedling_transect_coord_robson_2010-2012.csv | Seedling Survey, Far North Queensland Rainforest SuperSite, Robson Creek, 2010-2012 |
| asn_fnqr_core_1ha_robson_2014.csv | Soil Pit Data, Soil Characterisation, Far North Queensland Rainforest SuperSite, Robson Creek, Core 1 ha plot, 2014 |
| asn_fnqr_fauna_biodiversity_ctbcc_2012.csv | Vertebrate Fauna Biodiversity Monitoring, Far North Queensland Rainforest SuperSite, CTBCC, 2012 |
| asn_fnqr_fauna_biodiversity_ctbcc_2013.csv | Vertebrate Fauna Biodiversity Monitoring, Far North Queensland Rainforest SuperSite, CTBCC, 2013 |
| asn_fnqr_fauna_biodiversity_ctbcc_capetrib_2014.csv | Avifauna Monitoring, Far North Queensland Rainforest SuperSite, Cape Tribulation, 2014 |
| asn_fnqr_fauna_biodiversity_ctbcc_lu11a_2014.csv | Vertebrate Fauna Biodiversity Monitoring, Far North Queensland Rainforest SuperSite, CTBCC, LU11A, 2014 |
| asn_fnqr_fauna_biodiversity_ctbcc_lu7a_2014.csv | Vertebrate Fauna Biodiversity Monitoring, Far North Queensland Rainforest SuperSite, CTBCC, LU7A, 2014 |
| asn_fnqr_fauna_biodiversity_ctbcc_lu7b_2014.csv | Vertebrate Fauna Biodiversity Monitoring, Far North Queensland Rainforest SuperSite, CTBCC, LU7B, 2014 |

# 5 Tracking method steps: Visualisation techniques

## 5.1 Make a list of steps, inputs and outputs

A very simple example of a pipeline is shown in Table 2. The steps and data listed in Table 2 can be visualised using the `newnode` function described below in case study 3. This creates the graph of this pipeline shown in Figure 5. As the analysis progresses through the phases of testing, refinement and final versions. The linked table and graphical depiction can be very helpful for reference by the analyst. The optional setting to define a status of each step (TODO, DONE, WONTDO) can be used to add colour, and show steps that remain to be done. The addition of short summary descriptions are also very useful for orienting oneself to the required tasks and their priorities. Such flow chart diagrams can be printed up on large sheets of paper and stuck on the wall beside a computer workstation for use in day-to-day work.

Table 2: A table with the steps of a simple data analysis pipeline

| STEP | INPUTS | OUTPUTS | DESCRIPTION | STATUS |
|---|---|---|---|---|
| Step1 | Source 1, Source 2 | Derived 1, QC check | This might be latitude and longitude of sites | DONE |
| Step2 | Source 3 | Derived 2 | This might be weather data | DONE |
| Step3 | Derived 1, Derived 2 | Derived 3 | Merging these data means they can be analysed | TODO |
| Step4 | Derived 3 | Model selection | | TODO |
| Step5 | Model selection | Sensitivity analysis | | TODO |

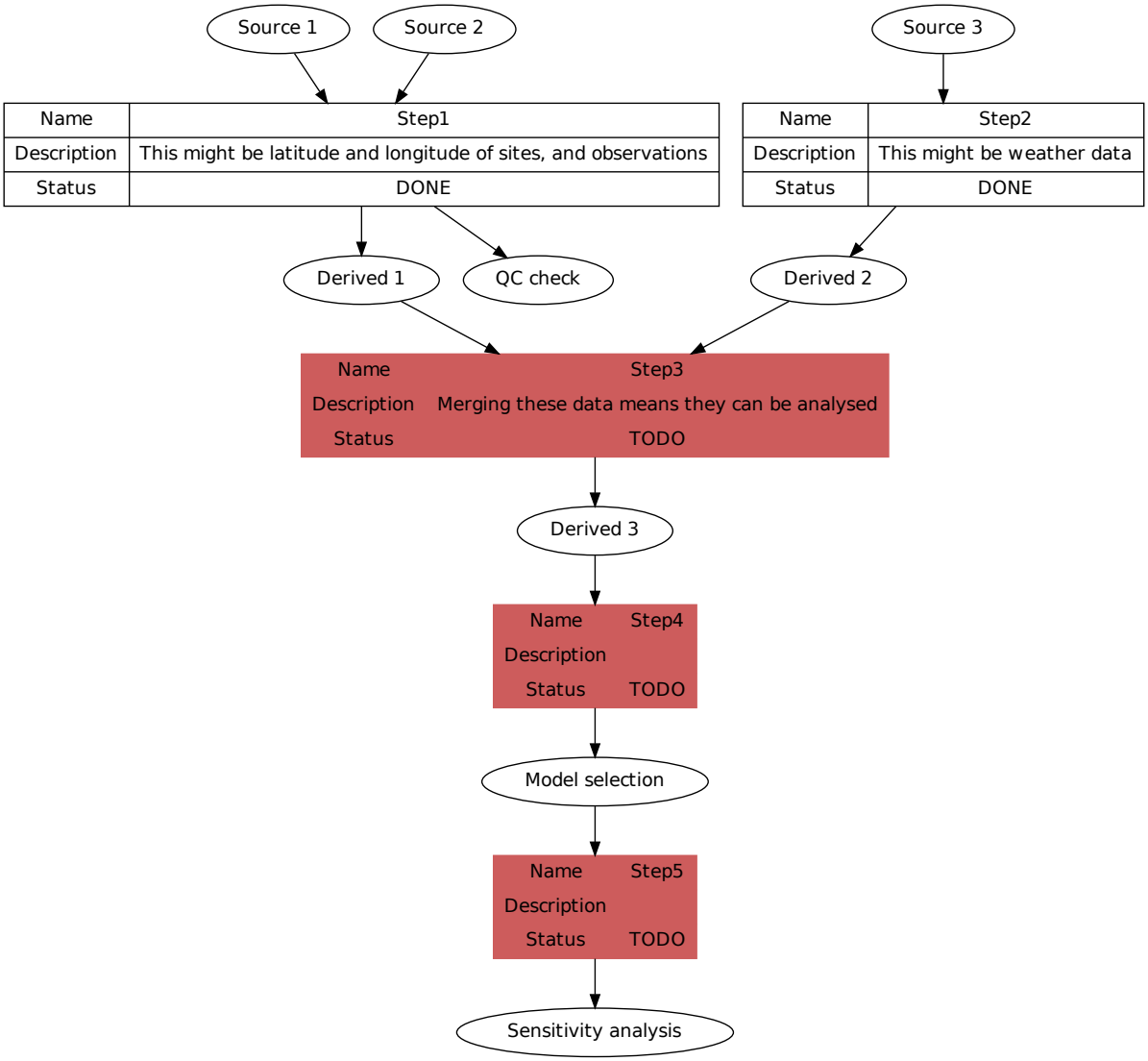


Figure 5: A visualisation of a data analysis pipeline showing the use of colour

As an example of the kinds of tangible steps such a workflow might entail a schematic diagram has been created and is shown in Figure 6.

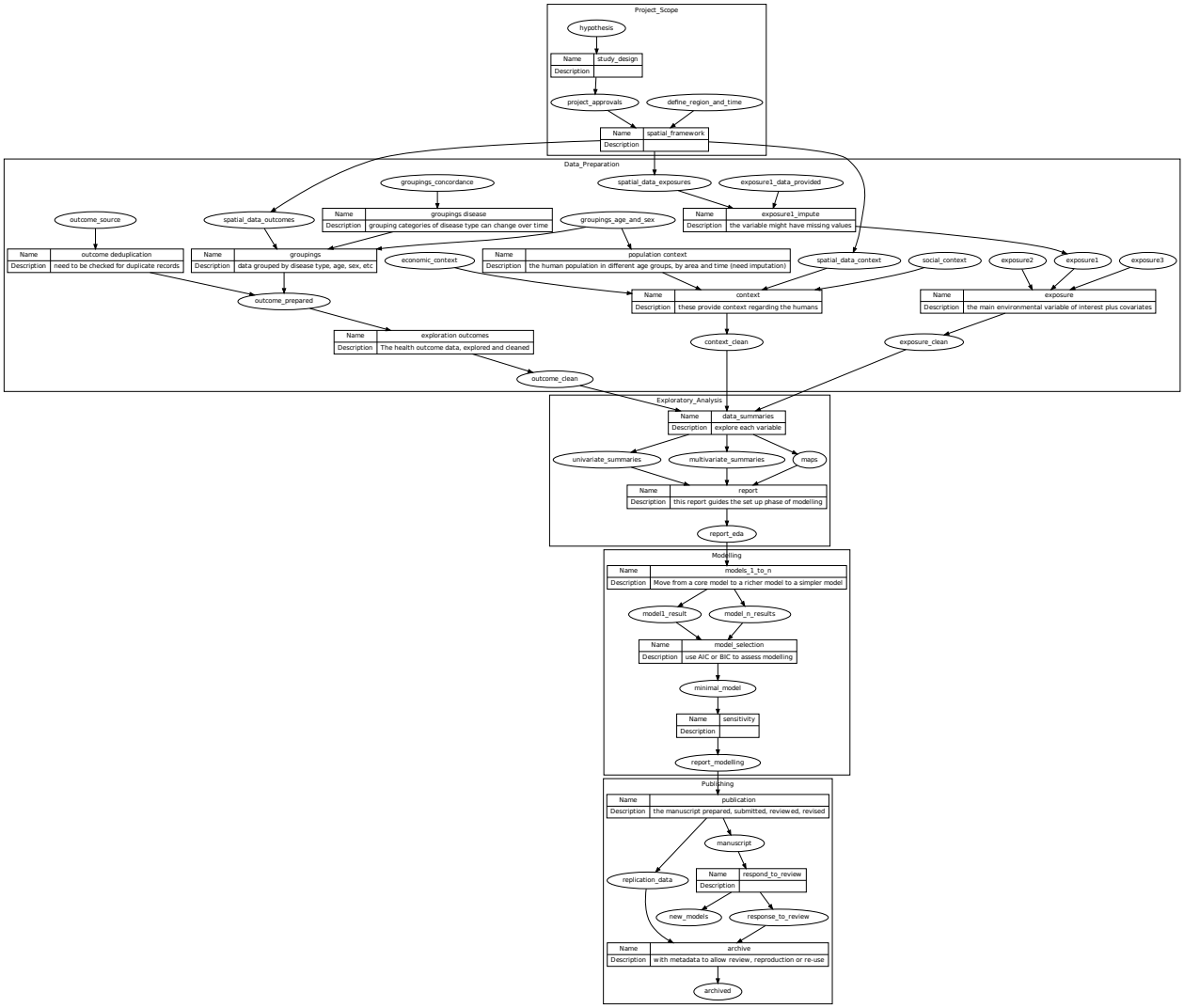


Figure 6: A schematic flow chart showing the steps required to prepare and conduct an analysis of health, environmental and social data.

A high resolution version of this image is available online at https://github.com/swish-climate-impact-assessment/swish_data_management_procedures/blob/phd_appendix/images/envepi_data_pipeline.pdf

## 5.2 Case study 3: Visualisation of methods steps using bespoke software

The method step is the key atomic unit of a scientific pipeline. It consists of inputs, outputs and a rationale for why the step is taken.

A simple way to keep track of the steps, inputs and outputs is shown in Table 3.

The steps and data listed in Table 3 can be visualised. To achieve this an R function was written

Table 3: A simple table to track method steps, data inputs and outputs

| STEP | INPUTS | OUTPUTS |
|---|---|---|
| Step1 | Input 1, Input 2 | Output 1 |
| Step2 | Input 3 | Output 2 |
| Step3 | Output 1, Output 2 | Output 3 |

as part of this PhD project and is distributed in the author's own R package available on Github (https://github.com/ivanhanigan/disentangle). This is the `newnode` function. The function returns a string of text written in the `dot` language which can be rendered in R using the `DiagrammeR` package, or the standalone `graphviz` package. This creates the graph view shown in Figure 7. Note that a new field was added for Descriptions as these are highly recommended.

```
library(disentangle); library(stringr); library(readxl)
steps <- read_excel("steps_basic_workflow.xlsx")
nodes <- newnode(indat = steps, names_col = "STEP",
                 in_col = "INPUTS",out_col = "OUTPUTS")
DiagrammeR::grViz(nodes)
```

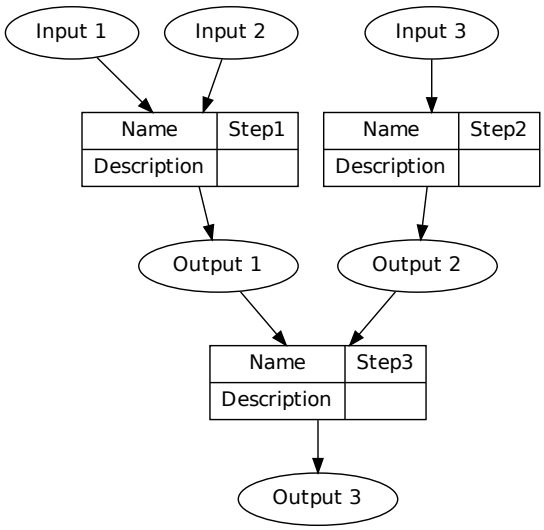


Figure 7: A graphical view of the steps that comprise a simple data analysis pipeline

## References for appendix 2


Adrian G. Barnett and Annette J. Dobson. *Analysing Seasonal Health Data*. Springer, Berlin, Heidelberg, Germany, 2010.

Adrian G. Barnett, Peter Baker, and Annette J Dobson. Package 'season': Analysing seasonal data R functions. R package version 0.3-5, 2014.

Agnes Bodnar, Rosemary Castorina, Manish Desai, Paurene Duramad, Susan Fischer, Neil Klepeis, Song Liang, Sumi Mehta, Kyra Naumoff, Elizabeth M Noth, Morten Schei, Linwei Tian, Kathleen L Vork, and Kirk R Smith. Lessons learned from "the skeptical environmentalist": an environmental health perspective. *International journal of hygiene and environmental health*, 207(1):57–67, 2004. doi: 10.1078/1438-4639-00265.

Elizabeth T. Borer, Eric W. Seabloom, Matthew B. Jones, and Mark Schildhauer. Some simple guidelines for effective data management. *Bulletin of the Ecological Society of America*, 205–214, 2009. URL http://www.esajournals.org/doi/abs/10.1890/0012-9623-90.2.205.

Wenju Cai, Tim Cowan, Karl Braganza, David Jones, and James Risbey. Comment on 'On the recent warming in the Murray Darling Basin: Land surface interactions misunderstood' by Lockart et al. *Geophysical Research Letters*, 37(10):1–3, may 2010. doi: 10.1029/2009GL042254.

Lora Fleming, Andy Haines, Brian Golding, Anthony Kessel, Anna Cichowska, Clive Sabel, Michael Depledge, Christophe Sarran, Nicholas Osborne, Ceri Whitmore, Nicola Cocksedge, and Daniel Bloomfield. Data mashups: Potential contribution to decision support on climate change and health. *International Journal of Environmental Research and Public Health*, 11(2):1725–1746, 2014. doi: 10.3390/ijerph110201725.

Robert Gentleman and Duncan Temple Lang. Statistical Analyses and Reproducible Research. *Journal of Computational and Graphical Statistics*, 16(1):1–23, mar 2004. doi: 10.1198/106186007X178663.

Kieran Healy. Choosing Your Workflow Applications, 2013. URL https://github.com/kjhealy/workflow-paper.

Jeffrey T. Leek and Roger D. Peng. Opinion: Reproducible research can still be wrong: Adopting a prevention approach. *Proceedings of the National Academy of Sciences of the United States of America*, 112(6):1645–1646, 2015a. doi: 10.1073/pnas.1421412111.

Jeffrey T. Leek and Roger D. Peng. Statistics: P values are just the tip of the iceberg. *Nature*, 520 (7549):612–612, 2015b. doi: 10.1038/520612a.

J Scott. Long. *The workflow of data analysis using Stata*. Stata publishing, 2008. ISBN 9781597180474.

B. D. McCullough and David A. Heiser. On the accuracy of statistical procedures in Microsoft Excel 2007. *Computational Statistics and Data Analysis*, 52(10):4570–4578, 2008. doi: 10.1016/j.csda.2008.03.004.

Anthony J. McMichael. Impediments to comprehensive research on climate change and health. *International Journal of Environmental Research and Public Health*, 10(11):6096–6105, 2013. doi: 10.3390/ijerph10116096.

William K. Michener, James W. Brunt, John J. Helly, Thomas B. Kirchner, and Susan G. Stafford. Nongeospatial metadata for the ecological sciences. *Ecological Applications*, 7(1):330–342, 1997.

William Stafford Noble. A quick guide to organizing computational biology projects. *PLoS Computational Biology*, 5(7):1–5, 2009. doi: 10.1371/journal.pcbi.1000424.

B O'Keefe. Hackers pick up UQ cash prize, 2007. URL http://www.theaustralian.com.au/higher-education/hackers-pick-up-uq-cash-prize/story-e6frgcjx-1111113191659.

Roger D Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011. doi: 10.1126/science.1213847.

Roger D. Peng. *Report writing for data science in R*. Leanpub. Unpublished Draft (Accessed 22 Dec. 2015), 2015. URL https://leanpub.com/reportwriting.

Roger D Peng and F Dominici. *Statistical methods for environmental epidemiology with R. A case study in air pollution and health*. Springer Science & Business Media, New York, USA, 2008.

Roger D. Peng, Francesca Dominici, and Scott L. Zeger. Reproducible epidemiologic research. *American Journal of Epidemiology*, 163(9):783–789, 2006. doi: 10.1093/aje/kwj093.

Andrew S. Pullin and Nick Salafsky. Save the whales? Save the rainforest? Save the data! *Conservation Biology*, 24(4):915–917, 2010. doi: 10.1111/j.1523-1739.2010.01537.x.

E Schulte, D Davison, T Dye, and C Dominik. A multi-language computing environment for literate programming and reproducible research. *Journal of Statistical Software*, 46(3), 2012. URL http://www.jstatsoft.org/v46/i03/paper.

M Schwab, M Karrenbach, and J Claerbout. Making scientific computations reproducible. *Computing in Science and Engineering*, 2(6):61–67, 2000. doi: 10.1109/5992.881708.

Theresa A Scott. Reproducible Research with R, TEX and Sweave, 2010. URL http://biostat.mc.vanderbilt.edu/wiki/pub/Main/TheresaScott/ReproducibleResearch.TAScott.handout.pdf.

Raphael Silberzahn and Eric L. Uhlmann. Crowdsourced research: Many hands make tight work. *Nature*, 526(7572):189–191, 2015. doi: 10.1038/526189a.

P Sólymos and Z Fehér. The mefa package: a tool for reproducible data processing in biogeography, 2008. URL http://biogeography.blogspot.com.au/2008/04/mefa-package-tool-for-reproducible-data.html.

Timothy H. Vines, Arianne Y K Albert, Rose L. Andrew, Florence Débarre, Dan G. Bock, Michelle T. Franklin, Kimberly J. Gilbert, Jean-Sébastien Moore, Sébastien Renaut, and Diana J. Rennison. The availability of research data declines rapidly with article age. *Current Biology*, 24(1):94–97, 2014. doi: 10.1016/j.cub.2013.11.014.

Ethan P White, Elita Baldridge, Zachary T Brym, Kenneth J Locey, Daniel J Mcglinn, and Sarah R Supp. Nine simple ways to make it easier to (re)use your data. 2013. doi: 10.7287/peerj.preprints.7v2.

Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10):1–23, 2014. doi: 10.18637/jss.v059.i10.

Yihui Xie. Chapter 1. Knitr: A comprehensive tool for reproducible research in R. In V Stodden, F Leisch, and RD Peng, editors, *Implementing Reproducible Research*. CRC Press, Boca Raton, USA, 2014. ISBN 978-0-387-98140-6.