Received: 24 July 2009, Revised: 24 November 2009, Accepted: 21 January 2010, Published online in Wiley Interscience: 2010

(www.interscience.wiley.com) DOI: 10.1002/cem.1298

Wiki based management of chemometric research projects

Bjørn K. Alsberg[a]* and Amanda Clare[b]

As research projects grow in size, there is a tendency for the number of management tasks to increase even faster. There are many different data objects and relations which need to be managed as part of the whole project: raw data, script files, function files, results, plots, tables etc.

Here it is suggested that a combination of wiki and version control software can be used as very effective and efficient tools in the management of chemometric research projects. A wiki provides a system for editing, viewing and navigating hyperlinked documents using web browsers. Easy syntax allows fast construction of web pages and hyperlinks without having to write HTML. Version control systems provide shared storage for source code and data (such as m-files, Python and Perl scripts, plot and diagram source files, LaTeX files etc). Items stored in version control are given version numbers and can be retrieved, worked upon and updated by multiple users in different locations without conflict.

At the top level of a project wiki the main ideas and goals are described whereas at the bottom level there are scripts, tables and raw data. The structure suggested by the different levels supports a common understanding of the need to record information and of which information to record. All levels can rapidly be accessed by following hyperlinks in the wiki. Combining information into new wiki pages is also very easy in this framework. Copyright © 2010 John Wiley & Sons, Ltd.

Keywords: wiki; version control; Subversion; management of projects; MediaWiki

1. INTRODUCTION

The management of scientific research projects in general can be challenging. A wide range of various issues are involved such as writing of grant proposals, inviting and hiring people, collecting data, building databases, building and running instruments, converting file formats, performing calculations, system administration of operating systems and collaborating with other scientists. The list of various tasks and operations that may be involved in research projects is almost endless and the important question is what tools and methods can be employed for making the management of such projects more efficient and effective. There are existing software tools for management of projects in general, however they may not be optimal for scientific purposes. Firstly, they tend to be based on commercial software which limits the availability for academic groups. The price may be too high and there is no possibility of modifying the source code to better fit the goals of a scientific project. Secondly, the tools may not contain all features necessary for dealing with scientific projects. This could be visualisation of 2D and 3D plots or manipulation of mathematical equations.

One tool which is increasingly being used to facilitate the management of projects is **wiki** [1,2] which was invented by the American computer programmer Howard G. Cunningham in 1994. Wiki means "fast" in Hawaiian and the basic idea is to provide websites where collaboration between many people is made as easy as possible. It has been demonstrated to be a powerful tool for handling knowledge management in many areas such as education [3–6], health and medicine [2,7,8], scientific collaboration and sharing of data [9–12], management of research groups [13] and software development [14–17]. The use of wiki to handle knowledge management and collaboration is related to the increasing trend of web-based dissemination which is referred to as "Science 2.0". This includes the use of Open Notebook Science which is the "the practice of making the entire primary record of a research project publicly available online as it is recorded" [18]. The UsefulChem project [19] is one of the first examples of Open Notebook Science and represents the chemistry research of the Bradley laboratory as an openly accessible wiki. The other interesting aspect of the emerging Science 2.0 area is the idea of social networking in science. MyExperiment [20] is a project that allows scientists to share workflows in the domain of bioinformatics, suitable for use in the Taverna system [21]. The aim is to reuse, combine and extend the scientific workflows of others to provide benefits similar to those obtained from the reuse of open source software.

It is here suggested that a wiki and version control system can be used as a tool for facilitating the management of chemometric research projects in particular. However there are project management tasks for which a wiki is not optimal. In

* *Correspondence to: B. K. Alsberg, Norwegian University of Science and Technology (NTNU), Department of Chemistry.*
E-mail: alsberg@nt.ntnu.no

a *B. K. Alsberg*
Norwegian University of Science and Technology (NTNU), Department of Chemistry

b *A. Clare*
Aberystwyth University, Department of Computer Science

particular these tasks relate to management of finances, budgets and personnel handling which will not be further discussed here. Our focus is on the management tasks which concern experiments, data analysis and reporting.

In order for a wiki and version control system to be used in research project management, information handling protocols at different steps in the project workflow are proposed. This is a suggested method for how to handle scientific information and data with respect to a wiki system and how to facilitate communication between people working on or associated with a project. The proposed approach is based on two main software technologies: wiki and version control. The latter refers to software systems which ensure control of different versions of source text for multiple collaborators in a project. Source text files at different stages in development are tracked by the version control software using a database such that the users easily can trace the complete history of all files and inspect who made changes and contributions. As the number of files and collaborators increase on a project, there is an increased risk of confusion and multiple incompatible copies of documents which need to be resolved. Version control software has been developed in computer science to handle such management tasks in larger software development projects. The tools are general and can be adopted to most scientific projects without any problems. In the approach presented, the version control software is used to handle all program and document source files. Binary data files are in general not stored through version control even though it is technically possible. For large amounts of binary data, relational or Resource Description Framework (RDF) databases should be used. Assuming that a web based query interface is available, it is easy to link to the database from a wiki.

The main idea presented is to create a system which effectively fuses together a wide range of information and data sources into a common software framework such that they become easily accessible to all members of a research project. As most scientists discover, even the smallest of projects can consume an enormous amount of time in data and information handling which could have been better spent on the main scientific focus of a project. A wiki based approach to project management may be therefore an effective tool so that more time and attention can be allocated to the scientific problems we want to study.

2. MATERIALS AND METHODS

2.1. Wikis

A wiki provides a web-based system for editing, viewing and navigating hyperlinked documents. Wikis are accessed through web browsers, making them available to multiple people at the same time on almost any platform. They have intuitive interfaces that enable fast construction of web pages and hyperlinks, using a much more simple syntax than HTML.

A wiki can contain either a single web page or a whole site, and in principle anybody can modify the page contents. In its original form, the wiki employs a very democratic attitude towards the contents on its web pages. The philosophy is that over time, the contents of a wiki page will be inspected and modified to become of a much higher quality than could be achieved when only one person is contributing. However this is not desired for all types of pages and therefore restriction control is therefore possible on many wiki systems. Page history is maintained by the wiki, so that previous edits are not lost, and a record is kept of who made changes and when these changes occurred. There is a wide variety of wiki software available, and many of these systems are free and open source. WikiMatrix [22] provides a useful comparison of their features (for example, the number of languages supported, the cost of the system, the type of storage provided, the plug-ins available and whether user access rights can be specified).

Perhaps the best known wiki in public use is Wikipedia, a freely available encyclopedia to which anyone can contribute. Wikis are also beginning to be used as specialised scientific resources, for example Protopedia [23], NMRWiki [24], Blue Obelisk [25], but as of 2009, these are still rare.

2.1.1. *MediaWiki*

We have chosen to use MediaWiki [26] which is the same system as that used for Wikipedia. Some of the reasons for choosing MediaWiki are:

- It is open source and freely available
- It can easily display mathematical symbols and formulas using a syntax similar to LaTeX

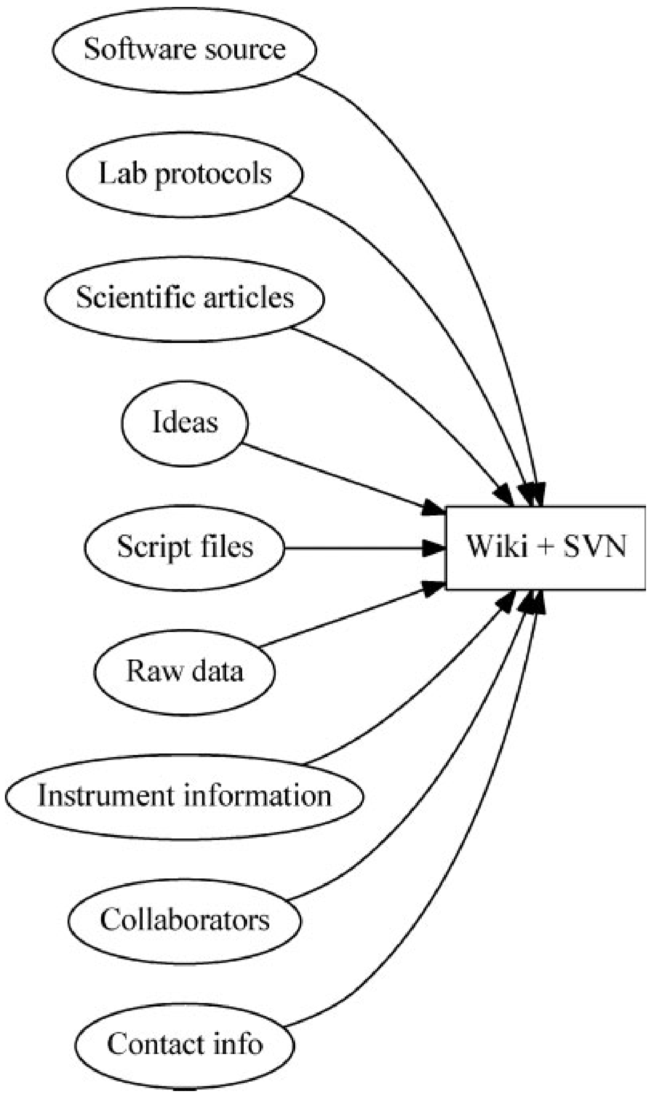


Figure 1. All project and group information is handled by one system.

- Use of source text from Wikipedia becomes easier
- It contains many commands for creating subject categories and subcategories
- Pages can be converted to PDF
- Images are easily included
- Complex tables are supported
- It has a wide user base, providing support and fixing bugs
- Many plug-ins exist (and if the ideal plug-in does not yet exist, then it can be written)
- Contains tools for access control to the wiki

MediaWiki uses a simple rich text editor that allows the user to edit wiki markup directly, or to highlight text and click a button to apply markup (such as bold, italic, heading, etc). The wiki markup is simple and is written using plain text. The following is an example of a table used to describe chemometrics scripts, which is located under the subsection "Data preparation":

```
== Testing of tables ==

{| class="wikitable" border="1"
|-
! Systematic file name
! Description
! Link to comments/results
|-
| [https://someplace.org/svn/Scripts/Test/file1.m file1] ||
This file is for testing (1) || [[file1_ScoreImages | Result 1]]
|-
| [https://someplace.org/svn/Scripts/Test/file2.m file2] ||
This file is for testing (2) || [[file2_ScoreImages | Result 2]]
|-
| [https://someplace.org/svn/Scripts/Test/file3.m file3]||
This file is for testing (3) || [[file3_ScoreImages | Result 3]]
|-
|-
|}
```

Figure 2 shows how the table will appear after parsing of this code by the wiki system. The special markup text is wiki code for making a table and hyperlinks. Some of the syntax is similar to what is found in HTML for producing tables. The `\{|` and `|\}` mark the beginning and end of a wiki table. `[[ ]]` is used to indicate hyperlinks and `||` separates two columns in the table. The == is used to indicate a subsection. A subsubsection is made by increasing the number of = characters: `=== Subsubsection ===`.

2.2. Version control

Version control systems provide shared storage for source code and data (such as m-files, Python and Perl scripts, plot and diagram source files, HTML source, Rich Text and LaTeX document files etc). Updates to the files stored in a version control system can be committed along with a descriptive comment, and then retrieved by other users. Most version control systems save the original file in full, and thereafter only save the changes that have been made, rather than re-saving the whole document. For this reason they are most suited to reasonably small text-based files. Larger data and binary files (such as images) are usually not stored in version control systems as this is not efficient.

Items stored in version control are given version numbers and can be retrieved, worked upon and updated by multiple users in different locations without conflict. Some systems (for example RCS) manage multiple users by insisting on file locking (only one person can work on one file at once), whereas others (for example CVS [27], SVN [28]) allow multiple users access to the same file and then provide support for merging changes if conflicts occur. Previous versions of files can be resurrected or inspected, providing transparency, provenance and backup.

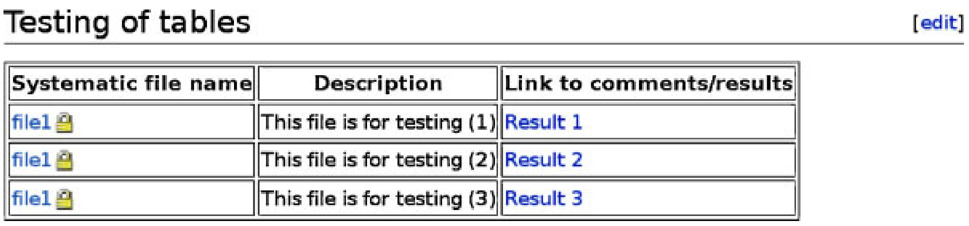
The older, well established version control systems (CVS, SVN) are often based around a single central repository that holds all files and to which users commit and retrieve data. Recently there have been several new version control systems created (for example git, darcs), which assume that each user has his/her own local repository, which are all equally valid repositories, and patches (updates) can be pushed or pulled from one repository to another.

2.2.1. *Subversion*

For version control or code management, we have decided to use the Subversion [28] (SVN) software system. SVN was originally created by CollabNet Inc and made to be almost compatible with the older Concurrent Versions System (CVS) software. It is based on an open source license and is used for version control in many well known open source projects such as KDE, GNOME, Apache, Ruby, Python and FreeBSD.

Some of the features of SVN are:

- It is open source and freely available
- Version control of directories which are treated just like files
- All copying, deleting, and renaming are versioned
- Version control of symbolic links


Testing of tables [edit]

Systematic file name	Description	Link to comments/results
file1	This file is for testing (1)	Result 1
file1	This file is for testing (2)	Result 2
file1	This file is for testing (3)	Result 3

Figure 2. This shows the appearance of the simple table formatted by the wiki syntax shown in Section 2.1.1.

- All output from SVN is designed to be readable by humans in addition to being accessible to software parsing
- SVN allows multiple users access to the same file, and provides tools for resolving conflicting changes of files and directories
- SVN also efficiently handles version control of binary files
- The time required for a SVN operation is proportional to the size of the changes resulting from that operation, not to the absolute size of the project in which the changes are taking place
- Bindings exist to popular programming languages such as Python, Perl, Java, and Ruby
- Plug-ins exist for integrated development enviroments such as Eclipse
- Clients exist for many platforms (Windows, Mac, Linux, Solaris, etc)
- It has a large user community for help, bug fixes and support

In the following examples are included to demonstrate some of the operations allowed by SVN. Here we add all m-files to the system:

```
svn add *.m
A       script1.m
A       script2.m
A       script3.m
```

If these files are stored under a directory `mfiles`, they can be retrieved (or "checked out", shortened to `co`) from another location by the following command:

```
svn co file:///somedirectory/svn/mfiles
A     mfiles/script1.m
A     mfiles/script2.m
A     mfiles/script3.m
```

3. PROJECT COMPONENTS AND STRUCTURE

3.1. Project components

Let us first identify what are the main components of a chemometric research project. Once these have been identified it is possible to discuss how these components can be structured in a wiki system. The most common components which appear in chemometric research projects include:

- Experimental design and planning
- Sample preparation
- Software and script development
- Instrument handling
- Data collection
- Data analysis
- Article/report/thesis writing
- Lecture/poster/talk creation

At the higher level of the research group it is likely that useful pages will contain:

- Project member information
- List of ongoing projects
- List of previous projects

Experimental design and planning. Here information is stored pertaining to:

- Description of the most important factors
- What design was chosen
- How many replicates were used
- Description of the response(s) and relations between the factors
- Reason for selection of factors
- Design matrices
- Description of the main hypotheses

Sample preparation. Usually the samples being analysed have been prepared in some way and this information needs to be accessible to all project members. If e.g. samples prepared under different factory conditions are tested, then this should be included in a separate wiki page:

- Experimental protocol followed for preparation of samples
- Who performed the experiments and when
- At what location (lab, factory, outdoors etc) the experiment was performed
- Chemicals and biological components used

Software and script development. The different types of softwares and scripts being used for data preparation, calibration, visualisation and analysis should be included and properly commented. How to set up a simple system for handling e.g. Matlab and other higher language scripts for chemometric analysis is described in a separate section below. However, data analytical scripts are not the only scripts that should be included. Sometimes people use Perl, Python or bash scripts to extract data from instruments, databases or the internet. These scripts should also be linked to and commented. As all these source files are under version control it is possible to inspect older versions of the source if necessary. This also prevents loss of important source by accidental deletion and also from errors included at a later stage. An investigator might include a wrong constant in a program or read an incorrect file. In both cases, a version control system makes it easy to see when the change was made and also to retrieve an older version without the added errors.

There is another very important reason for insisting that all group members should include their software scripts and functions into the same wiki system: It makes it much easier to share tools within the group. Often it is seen that even within the same group, people construct the same tools again and again because they have no way of knowing what other group or project members are doing. Therefore, it is here suggested that for Matlab and other similar languages, various toolbox areas are created where general tools can be rapidly accessed. It may be necessary to include local toolboxes for a project which is not of a general interest. However it should be stressed to collaborating members that if they think a script or function could be of general interest, then it should be placed in a toolbox/library.

Instrument handling. Many chemometric groups have various analytical instruments in their laboratories such as hyperspectral cameras, near infrared (NIR) spectrometers, mass spectrometers and nuclear magnetic resonance spectrometers. For every instrument there is a protocol for how to use it. A separate page for this should be included since this is very valuable for new members of the research group. In addition, if new parts or experiences with the instrument are made, this can easily be added by any member of the group for everybody to share. Typical information which are useful to add here are:

- Basic protocol for use
- Links to the company (or companies) which sells the instrument and spare parts, service history and maintenance contract details
- Links to articles describing the instrument and the underlying theory
- Images of the instrument and its parts
- A log of who used the instrument, and when
- A link to database(s) of raw data recorded by the instrument

Data collection. This is a typical project level page which may be unique to a particular project. The page typically will contain information related to:

- Who recorded the data, and when
- Information on the different samples
- Information on the experimental and instrumental parameters
- Where the raw data are stored (with links)
- Possible comments from experimenter related to each recording

Data analysis. The data analysis usually forms the main focus in chemometric research, either as a point in itself or as part of a method development project. In all cases, various algorithms are applied to data under different conditions and data sets. A central idea here is that **all experiments must be encoded in stand-alone scripts**. As part of the proposed protocol for data analysis and preprocessing (see below) is that the investigator should never fall for the temptation to generate results which are used as part of an article only interactively in e.g. Matlab/Octave or Python. The reason is that if a complex sequence of operations has been performed, it might be difficult and time consuming to later try to recreate what was performed at an earlier stage.

It is important to realise that the word “script” should here be interpreted in the broadest sense. It does not have to be a formalised language such as C++, Matlab or Java. It can be any set of instructions which enable the accurate reproduction of a certain experiment or task performed. Many chemometricians use software which are mainly based on a graphical user interface (GUI) without the ability to make use of a scripting language. However, even in such cases, the investigator must ensure that all relevant GUI operations are accurately recorded, either automatically or manually.

Writing articles, reports, talks etc. Since all relevant information of a project is accessible through the wiki system, it may be a good idea to start the article/report/talk[†] also in the wiki. The main reason for this is the fact that many scientific presentations are collaborations between people at different locations in the world. Sending e.g. Word or LaTeX files by e-mail is of course possible, however slightly cumbersome. In particular is this the case when multiple authors need to work on the same presentation. It is also possible just to use SVN for the document source files, however there is one main drawback with this. This is that the user must either start a separate program to view the content (such as when using Word) or compile the source before viewing (as is the case in LaTeX). The wiki approach on the other hand allows the presentation content to be viewed immediately. This also makes it easier for other group members and collaborators who are not directly involved in the presentation to inspect and comment on the comments if needed.

A much simpler approach is to start writing the presentation as a wiki page. The system already has its own version control software and updates are visible to everybody immediately. Links to pages with results and other types of information are very easy to create. As the presentation wiki page matures, there comes a point when it is time to move the content to the chosen document preparation system which will be used to create the final version of the presentation. Some of the advantages of starting the presentation wiki page are:

- Facilitates communication between authors
- The presentation wiki page can be made available on public web pages without worrying about copyright issues when the the final version is published
- Makes it easy to enter the wiki source into Wikipedia in order to inform others about the research performed by the group
- Makes it easier for students to access the written material produced in the group

It is possible to make a parser program which converts the wiki code into the chosen documentation code. For Microsoft Word users, the best option would probably be to parse the wiki code into Rich Text Format (RTF) or some other ASCII based format which can be imported into Word. Another possibility is to parse the wiki code into LaTeX markup which was done in this case. LaTeX code and associated figure files are generated by our wiki2latex parser which can later be compiled to PDF or PostScript.

Centralisation and legacy systems. The flexibility of a wiki allows a research group to decide which of the above project components it wants to centralise and which it prefers to keep separate. There is no pressure to integrate or link to everything, particularly if data from legacy systems are not available on networked machines. A research group can gradually move towards more extensive use of a wiki and version control system by integrating more project components when appropriate.

3.2. Project structure

Now that we have identified the main components of chemometric research projects, the next question is how this can be implemented using wiki and version control software. Two aspects of structure are easily represented in a wiki: hierarchical and hyperlinked structure. A version control system will store a hierarchy of files. The structure will depend on the natural relationships between the research project components. Rather than representing directly the project workflow (see below) the structure tends to reflect the project components that go naturally together.

In general a chemometrics project may be described with a top wiki page containing a title and short description of the project. Then subsections, lists and hyperlinked pages can be used to divide the top page into the different components of the project (Instrument Handling, Data Collection, Experimental Design, etc), see Figure 3. At the lowest level of the hierarchy the wiki page is likely to link to code and data stored in the version control

[†] Henceforth referred to as a **presentation**

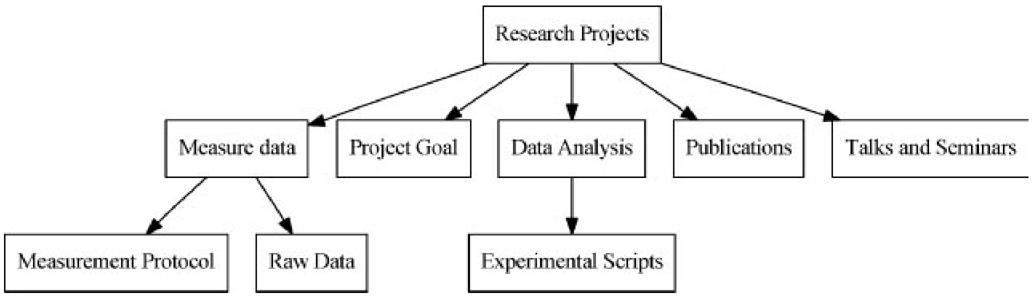


Figure 3. This shows the wiki structure from the project level. For visualisation purposes not all project components have been included.

system. This organisation of the wiki pages would suggest that the different project components are hierarchical, however this cannot always be followed. In those cases, hyperlinks between arbitrary wiki pages will be used. Free hyperlinking between pages creates a flexibility which makes the wiki more user friendly. For instance, such linking makes it possible for common subcomponents to be shared by many projects. A typical example is handling of instrument information which is likely to be identical for several projects that use the same instruments. A page describing the correct initialisation procedure for e.g. a hyperspectral camera or a near infrared spectroscope may be linked to several projects at the appropriate levels in their hierarchies, and also to an "instrument handling" hierarchy of pages.

Pages that are linked to from several projects encourage information reuse and reliability. The obvious potential disadvantage of sharing editable information is that any changes made to a page will retrospectively seem to apply to all projects that have linked to this page. If we e.g. change the protocol for a scientific instrument, then we must be careful that all projects which previously used a different protocol still have access to the original information. MediaWiki provides a "What links here" facility so that the researcher can see the sphere of influence of the changes that they are about to make.

Hierarchies and hyperlinks have become topics of interest within the Web community as researchers try to understand how best to categorise and manage data (for example [29]). The discussion about whether hyperlinks undermine hierarchies (causing helpful structure to be obscured or bypassed) will continue. For most people, the benefits of hyperlinks far outweigh the potential for untidiness caused by too much flexibility. Most wikis provide more support for hyperlinks than hierarchies. MediaWiki does allow hierarchical structuring via sections and lists, and will summarise the outline of a page as a table of contents, but does not enforce a hierarchical structure; this is left to the design of the research group members.

In the following section we suggest a structure that has been helpful to our own chemometrics projects. It ensures that the research group members know what information they are expected to document for the different stages in their research projects.

4. PROJECT WORKFLOW

4.1. Chemometric workflows

A workflow is here defined to be the dynamics between the different research components. Questions like what should be done, in what order and in what way are central to the workflows.

In general there are two main research workflow types in the field of chemometrics:

- Method development
- Application of chemometric methods

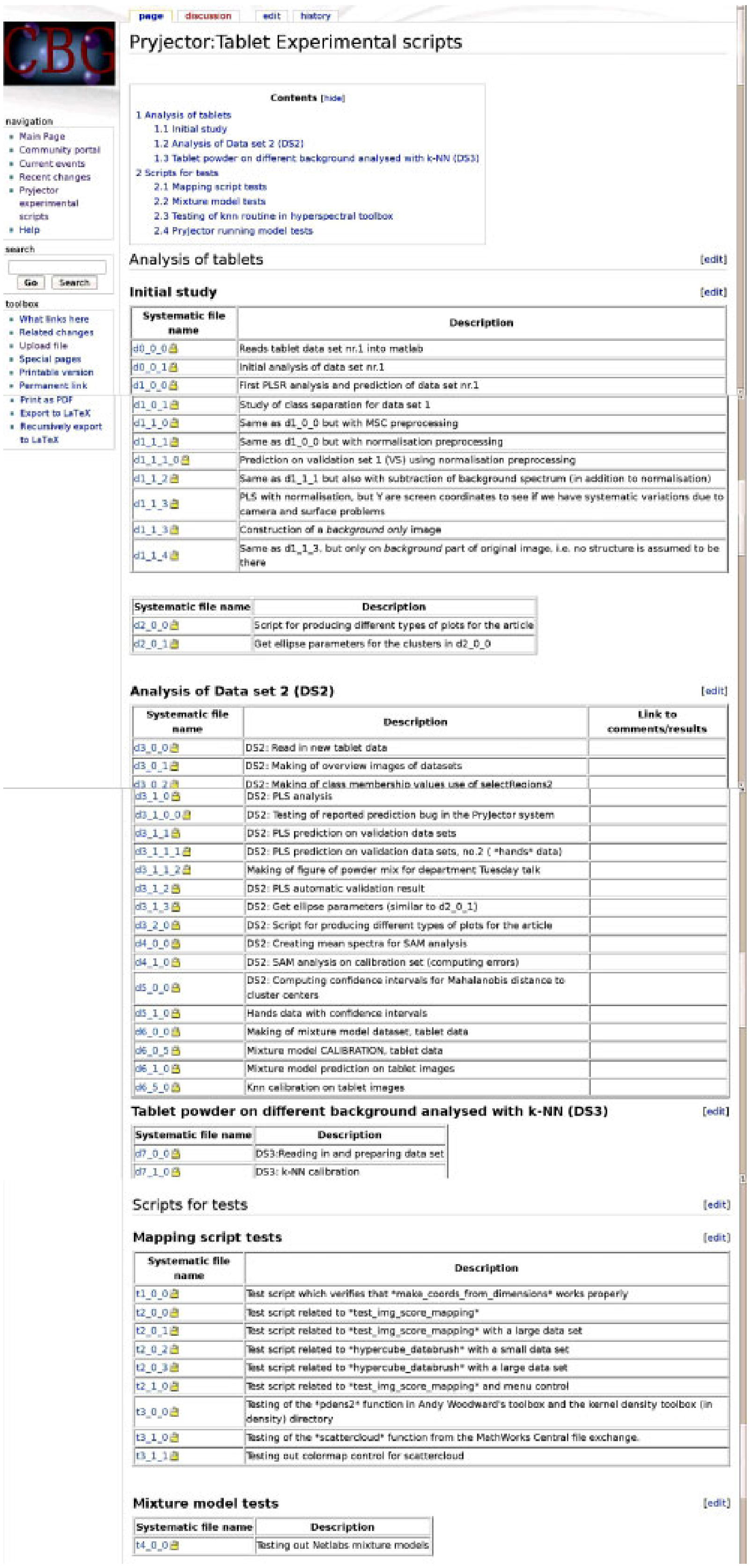
This is of course a simplification of what types of chemometrics research projects exist, however it serves here as a usful categorisation.

When it comes to the wiki and the version control software, it is important that they are set up in such a way that the workflow for a project becomes as effective and efficient as possible. As well as assisting with the structure of the project, the wiki and version control can help with the management and workflow of the project. It can be used for the active workflow (what is to be done next, what should be done), and to keep a historical record of what has been done, recording the data provenance and the methods applied so that experiments can be repeated, checked or disseminated. Both of these aspects are important, and dedicated tools exist to help manage scientific workflow (such as Taverna [21], Kepler [30–32]). These tools provide excellent support for scientific workflow in their respective domains (often including graphical languages for describing workflow, and allowing executable workflows). However, they require the project team to learn to use a special purpose environment which does not necessarily integrate well with the rest of the project management information. A wiki-based workflow is simple, but general-purpose and flexible, and is relatively easy to encourage project members to use.

The active workflow can be managed by having a common understanding of the components that should be part of each page (either by using page templates, or more informally). In this way, the users know what information they need to produce and record. Some wiki systems, such as Trac [17], also have tickets that can be raised to assign tasks to project members. Trac and other similar systems are in particular useful for project workflows where programming is of central importance, i.e. the project is developing one or several software components which in themselves are of central importance. In those cases, the workflow will much more resemble the ones we find in general software development.

4.2. Information protocols

To enable an effective and efficient workflow, a set of rules for how information is stored and handled must be agreed on in beforehand. If everybody on a project could decide for themselves in what way the information should be inserted into the wiki and version control software there is a risk the system



Pryjector:Tablet Experimental scripts

Contents [hide]

1 Analysis of tablets
1.1 Initial study
1.2 Analysis of Data set 2 (DS2)
1.3 Tablet powder on different background analysed with k-NN (DS3)
2 Scripts for tests
2.1 Mapping script tests
2.2 Mixture model tests
2.3 Testing of knn routine in hyperspectral toolbox
2.4 Pryjector running model tests

Analysis of tablets

Initial study

Systematic file name	Description
d0_0_0	Reads tablet data set nr.1 into matlab
d0_0_1	Initial analysis of data set nr.1
d1_0_0	First PLSR analysis and prediction of data set nr.1
d1_0_1	Study of class separation for data set 1
d1_1_0	Same as d1_0_0 but with MSC preprocessing
d1_1_1	Same as d1_0_0 but with normalisation preprocessing
d1_1_1_0	Prediction on validation set 1 (VS) using normalisation preprocessing
d1_1_2	Same as d1_1_1 but also with subtraction of background spectrum (in addition to normalisation)
d1_1_3	PLS with normalisation, but Y are screen coordinates to see if we have systematic variations due to camera and surface problems
d1_1_3	Construction of a *background only* image
d1_1_4	Same as d1_1_3. but only on *background* part of original image. i.e. no structure is assumed to be there

Systematic file name	Description
d2_0_0	Script for producing different types of plots for the article
d2_0_1	Get ellipse parameters for the clusters in d2_0_0

Analysis of Data set 2 (DS2)

Systematic file name	Description	Link to comments/results
d3_0_0	DS2: Read in new tablet data	
d3_0_1	DS2: Making of overview images of datasets	
d3_0_2	DS2: Making of class membership values use of selectRegions2	
d3_1_0	DS2: PLS analysis	
d3_1_0_0	DS2: Testing of reported prediction bug in the Pryjector system	
d3_1_1	DS2: PLS prediction on validation data sets	
d3_1_1_1	DS2: PLS prediction on validation data sets, no.2 (*hands* data)	
d3_1_1_2	Making of figure of powder mix for department Tuesday talk	
d3_1_2	DS2: PLS automatic validation result	
d3_1_3	DS2: Get ellipse parameters (similar to d2_0_1)	
d3_2_0	DS2: Script for producing different types of plots for the article	
d4_0_0	DS2: Creating mean spectra for SAM analysis	
d4_1_0	DS2: SAM analysis on calibration set (computing errors)	
d5_0_0	DS2: Computing confidence intervals for Mahalanobis distance to cluster centers	
d5_1_0	Hands data with confidence intervals	
d6_0_0	Making of mixture model dataset, tablet data	
d6_0_5	Mixture model CALIBRATION, tablet data	
d6_1_0	Mixture model prediction on tablet images	
d6_5_0	Knn calibration on tablet images	

Tablet powder on different background analysed with k-NN (DS3)

Systematic file name	Description
d7_0_0	DS3:Reading in and preparing data set
d7_1_0	DS3: k-NN calibration

Scripts for tests

Mapping script tests

Systematic file name	Description
t1_0_0	Test script which verifies that *make_coords_from_dimensions* works properly
t2_0_0	Test script related to *test_img_score_mapping*
t2_0_1	Test script related to *test_img_score_mapping* with a large data set
t2_0_2	Test script related to *hypercube_databrush* with a small data set
t2_0_3	Test script related to *hypercube_databrush* with a large data set
t2_1_0	Test script related to *test_img_score_mapping* and menu control
t3_0_0	Testing of the *pdens2* function in Andy Woodward's toolbox and the kernel density toolbox (in density) directory
t3_1_0	Testing of the *scattercloud* function from the MathWorks Central file exchange.
t3_1_1	Testing out colormap control for scattercloud

Mixture model tests

Systematic file name	Description
t4_0_0	Testing out Netlabs mixture models

Figure 4. Screen shot of script tables for ongoing project at CBG, NTNU.

would become chaotic and unsuitable for project management. To avoid this we need to agree on a a predefined procedural method or **protocol** for how information is handled.

The level of details in a protocol will depend on the group and what type of projects they have. For instance, a research group performing experiments with biohazard level 4 pathogens will obviously need very strict protocols related to handling of materials for safety reasons, whereas a group working mostly with hyperspectral imaging of e.g. wood may not require the same level of detail and conformity. The important message is that the protocols are there to be useful and facilitate the overall workflow, not becoming obstacles themselves. The UsefulChem wiki gives an example of a very minimal guide for writing experimental descriptions [19]. However, we would like to suggest here an approach that is more suitable for chemometrics and integrates well with the processes that chemometrics researchers follow.

A central part of most chemometric research projects is the performing of analysis and preprocessing of data. We will therefore focus in particular on a protocol for handling information related to these tasks that we have found to very useful. The protocol is based on the use of an **Experimental scripts page** (ESP) to keep a record of the data analysis and preprocessing workflow of a chemometrics project. It should be noted that the script page approach is not optimal for GUI (graphical user interface) based software since the user has to create his own way of recording the various GUI tasks. Due to this problem, we focus only on script based projects using tools like Matlab, Octave, SciLab, Mathematica, Python etc. Note that this will also include GUI based software which can be used in a "recording mode", i.e. where a session can be recorded as an ASCII sequence of operations which effectively work as a script.

The experimental script protocol. This proposed protocol is based on breaking away from the interactive mode of work allowed for in high level languages like Octave, Matlab, Python, R and Mathematica. Performing important operations interactively in any of these systems is discouraged as it becomes easy to forget important command sequences and later review and reuse what was performed. Instead the protocol suggested is based on starting any work sequence by creating a script file designated for a limited set of actions. A draft of the script file is first generated, followed by execution. Any modification are made in the script file, not interactively to ensure that proper command sequences are stored and documented. Such a process is somewhat slower in the beginning, but will over time increase the overall efficiency. The reasons for this are as follows:

- When the user returns to a project, the time required to get into the details again is minimized.
- Creating scripts with comments for every relevant task performed in a project will effectively create a documentation of what was done at different stages during the project. This will minimize introduction of errors and make every project much more transparent for others to inspect.
- Sets of scripts can later be recalculated in various combinations to handle new data and/or different parameter settings. Scripts for generating figures and tables for articles should be made. Such scripts can later be run again with other parameters with very little extra work.
- Sets of scripts makes it much easier for collaborators to understand each others work and reuse code for other purposes.

The edit-script-execute-it cycle is intended for all computations and experiments, also those that are regarded as tests and not directly related to the main problem.

However we need a way for handling the collection of scripts and this is where the "Experimental scripts page" (ESP) plays an important role. In some sense the ESP may be regarded as the most important wiki page in the whole project as it contains all the links and information for managing the experimental scripts. The ESP traces out the flow of results generated from the project that will form the basis for subsequent publications. The ESP is built as follows:

- The page consists of different subsections, each dealing with a specific topic. Examples of such topics are:
  - Reading of raw data
  - Filtering to remove noise
  - Analysing data X with chemometric method Y
- Within each subsection there will be a **script table** of scripts that appear to belong together
- The script table contains the following information:
  - Name of script source file (usually a **systematic name**)
  - Link to the script source file (controlled by version control software)
  - A short description of what the script is doing
  - An optional link to a page which contains results from the script

For each subsection, there should also be a short description of the intention behind the scripts. This ensures that current ideas and thought processes when writing the script are captured.

The ESP is made in the following way: First, the meta information about the script is put into the script table. A systematic name is chosen, followed by a brief description about what this script should do. This is a top-down approach such that the new script can be seen in relation to other scripts and experiments. Then the actual script is created in an editor followed by testing, e.g. by running the corresponding m-file in Matlab/Octave.

We may view this way of working as an ongoing documentation of the research. It will provide a detailed history of the project which can be inspected at different details levels. Even if the script did not produce the correct results, it should be kept and perhaps given a special label in the script table. When the project is finished, all results can be recreated by starting a series of scripts.

5. PROJECT STAFF

5.1. Project member information

As a research group becomes larger and evolves over time, many people have been part of different projects. Every time a student, postdoc, technician, visiting scientist or a collaborator is associated with the group, a web page with the name, e-mail address(es), phone number(s) and mail address(es) for that person should be made. When a member leaves the group for another location, it can be stated that it would be nice if the person could ensure the personal information is updated on the webpage in the future. All that is needed is for the person to log in to the project web page and make changes. In this way it is easy for the group leader (and others connected to the group) to get in contact with all persons associated with a project and also to convey contact information to others. It is surprising how much time may be lost in trying to collect contact information for e.g. old PhD students and postdocs.

5.2. Initiating/training new group members

When a new person arrives in a research group to work as a student, postdoc, visiting scientist etc it is often time consuming to get that person active in the group. The information the new group member needs can range from where to rent or buy houses to review relevant papers for the project. Over time, it is possible to see that the same problems tend to repeat and therefore wiki pages should be constructed specifically for getting new group members up to speed. As a general rule, it should be the responsibility of all group members and collaborators to update the information on the wiki pages. In particular, it would be most useful if the new members of the group allocate some of their time to document in the wiki what new or updated information is needed. In this way, the experiences and knowledge of the group members are better utilised.

5.3. Managing communication

A web based wiki system provides a way for ensuring effective communication about the state of projects and other relevant information to all members and collaborators. E-mails can of course be used, however over time such messages tend to be buried in the mailboxes of individuals and can be difficult to retract when needed. By using a wiki, everybody can share the same information. It is also possible to attach a **blogging system** to the wiki where various messages and information are made available continuously.

6. DISCUSSION

We have described a wiki plus version control system for managing a chemometrics research project. This approach has many advantages and some shortcomings. We first list the shortcomings and the progress still to be made in this area, and follow this by summarising the advantages.

Some examples of shortcomings that could be improved upon for future scientific project management systems are:

- A wiki can become untidy if project members do not conform to some agreed structure, such as that described above. This is both a strength and a weakness; the strength is that the relative freedom and flexibility encourages project members to use the systems and imposes little overhead, but the weakness is that good organisation of the information relies on their good intentions to use the agreed structure.
- A project member can update a wiki page without checking which other pages link to the page they have updated. This is perfectly acceptable for an online encyclopedia, but not always for a scientific record. The history of a page is always recorded for posterity and rollback. However, it would be better to have an system extension that would warn the user of the potential for changing information that is linked from previous work, and suggest solutions such as subdividing a page.
- A wiki mixes content with formatting so the two are inseparable (for example, information expressed in tables can't be easily reformatted to become new pages).
- The data are not semantically annotated. Although we may use words such as "script", "hyperspectral" and "conclusion" that are clear to the project members, they are not available for machine understanding and further processing of the data. The data can be exported as XML but the markup will be minimal (titles, authoring and revision details) but most text is exported as a single unmarked block.
- Version control is ideal for storing textual data (results, scripts, latex documents, etc.), but not for binary data (diagrams, powerpoint talks, etc). Version control systems record the changes from one version of the data to the next, and this is not easy to make efficient for binary data.
- The best ways to provide integration of the information in a wiki system with large amounts of structured data stored in project databases is still a research issue.

Scientific projects are increasingly becoming collaborative, with people across the globe working on the same project. The use of the internet and the Web is therefore natural in such cases. With a wiki system, the different project members can easily access information and log in to share information or edit existing pages. The webpage frontend to collaboration is natural and intuitive to use. As demonstrated above, the different information levels in a project can be rapidly accessed. Through the use of hyperlinks, the members are able to access everything from raw data to article figures through hyperlinks. Source code for scripts and functions is also available, enabling everybody to have a much deeper access to all information levels in the project.

However the most important result of the the proposed wiki based management approach is that it allows for complete openness about research projects, within a project team and externally. When a project is finished it is possible to release all relevant project information such that others can inspect, extract and learn from what has been done. This will be what Donoho *et al* [33] refer to as "reproducible research", where all the necessary information is available for others to replicate your experiments. Too often, the scientific article is not sufficient to provide all the necessary information about a project.

There are of course incentives for scientists to withhold information from completely open access (for example, scientists may have concerns about being scooped or that information published can destroy patenting opportunities). However, the benefits of using a wiki and version control approach for good reporting practice will still hold within the research group, even if the details are not to be made public.

Good record keeping in scientific research is essential for reproducibility, reliability and to maximise the use of the knowledge that has been discovered. We therefore encourage chemometrics researchers to make use of wikis and version control to assist them in this aim.

Acknowledgements

BKA thanks the Department of Chemistry at NTNU for financial support of this work. AFC is funded by a fellowship from the RAEng/EPSRC.

REFERENCES

1. Nelson R. *Visual Studies* 2008; **23**(3): 275–279.
2. Keim B. *Nature Medicine* 2007; **13**(3): 231–233.

3. Reinhold S, Abawi D. *Technologies For E-learning and Digital Entertainment, Proceedings* 2006; **3942**: 755–767.
4. Wheeler S, Yeomans P, Wheeler D. *British Journal of Educational Technology* 2008; **39**(6): 987–995.
5. Chu S. *Online Information Review* 2008; **32**(6): 745–758.
6. Luce-Kapler R. *Curriculum Matters* 2008; **4**: 85–101.
7. Trachsel S, Breitenstein J, Eberle B. *Swiss Medical Weekly* 2008; **138**(41–42): 16S–16S.
8. Bastida R. *International Journal of Mental Health Nursing* 2008; **17**: A2–A2.
9. Wagner C. *Journal of Database Management* 2005; **16**(2): I–VIII.
10. Kramer-Flecken A, Landgraf B, Krom J. *Fusion Engineering and Design* 2008; **83**(2–3): 375–381.
11. Stokes T, Torrance J, Li H, Wang M. *BMC Bioinformatics* 2007; **9**.
12. Lee T. *Nature* 2008; **455**(7212): 461–461.
13. Mooney S, Baenziger P. *Briefings in Bioinformatics* 2008; **9**(1): 69–74.
14. Kim S, Han H, Han S. *Innovative Approaches For Learning and Knowledge Sharing, Proceedings* 2006; **4227**: 646–651.
15. Peeters V, Schrier P. *Extreme Programming and Agile Processes in Software Engineering, proceedings* 2005; **3556**: 308–310.
16. Massar J, Travers M, Elhai J, Shrager J. *Bioinformatics* 2005; **21**(2): 199–207.
17. *The Trac Project*, http://trac.edgewall.org/.
18. *Open Notebook Science*, http://en.wikipedia.org/wiki/Open_notebook_science.
19. *UsefulChem Open Notebook Science*, http://usefulchem.wikispaces.com/.
20. *MyExperiment website*, http://www.myexperiment.org/.
21. *Taverna project website*, http://taverna.sourceforge.net.
22. *WikiMatrix, compare them all*, http://www.wikimatrix.org/.
23. *Protopedia, life in 3D*, http://www.proteopedia.org/.
24. *Open NMR Project NMR Wiki*, http://nmrwiki.org/.
25. *BlueObelisk.org*, http://blueobelisk.sourceforge.net/.
26. *MediaWiki*, http://www.mediawiki.org/.
27. *CVS - Concurrent Versions System*, http://www.nongnu.org/cvs/.
28. *Subversion, an open source version control system*, http://subversion.tigris.org/.
29. Shirky C. *Ontology is Overrated: Categories, Links, and Tags*, http://www.shirky.com/writings/ontology_overrated.html.
30. Ludascher B, Altintas I, Berkley C, Higgins D, Jaeger E, Jones M, Lee E, Tao J, Zhao Y. *Concurrency and Computation-practice & Experience* 2006; **18**(10): 1039–1065.
31. *The Kepler Project*, https://kepler-project.org/.
32. Zhang J, Pennington D, Michener W. *Computational Science - ICCS 2006, Pt 3, Proceedings* 2006; **3993**: 912–919.
33. Donoho D, Maleki A, Shahram M, Rahman I, Stodden V. *Computing in Science & Engineering* 2009; **11**(1): 8–18.