JOHNSON MATTHEY
TECHNOLOGY REVIEW

www.technology.matthey.com

Emacs as a Tool for Modern Science

The use of open source tools to improve scientific workflows

Timothy Johnson
Johnson Matthey, Blounts Court, Sonning Common, Reading, RG4 9NH, UK

Email: timothy.johnson@matthey.com

PEER REVIEWED

Received 14th July 2021; Revised 10th September 2021; Accepted 14th September 2021; Online 15th September 2021

It is human nature to prefer additive problem solving even if removal may be the more efficient solution. This heuristic has wide ranging implications when dealing with science, innovation and complex problem solving. This is compounded when dealing with these issues at an institutional level. Additive solutions to workflows with extra software tools and proprietary digital solutions can impede work without offering any advantages in terms of Findable, Accessible, Interoperable, Reusable (FAIR) data principles or productivity. This viewpoint highlights one possible workflow and the mentality underpinning it with an aim to incorporate FAIR data, improved productivity and longevity of written documents while improving workloads within industrial research and development (R&D).

Introduction

FAIR data principles have been held as the gold standard for ensuring data across the sciences and across individual institutions is generated and kept in as sustainable a way as possible (1). FAIR data principles unlock powerful 'data lake' workflows that allow for multiple interactions, machine learning and deep insight to be gained, adding value to already collected data (2). Reports and peer reviewed publications are needed to share knowledge with others at both an inter- and intra-institution level.

One nemesis to this approach is the use of proprietary software and proprietary data standards. It has been suggested that all research software should be free open source software (FOSS) and that closed source software should be the exception (3). The use of FOSS and open source hardware has been shown to offer flexibility and insight in a range of practical applications within chemical R&D (4–6).

A wealth of new software is available every year including productivity tools, document management, data analysis suites and code produced *via* individuals or research groups. One recent report showed that ~51,000 publications in the life sciences had 25,900 unique pieces of software cited (7). In addition to the wealth of new software offerings humans are keenly biased towards additive problem solving (8). Adding to an existing system rather than taking away in order to solve a problem is seen across sectors, job roles and in the digital tools used to enable science. An exemplar of this type of approach in software was seen with the introduction of the ribbon into Microsoft Office. Those more experienced with the software were more likely to be dissatisfied and impeded by the addition of the ribbon into the Office suite (9). Frustration stemming from unclear error messages, poor wording and lack of training lead to a loss of as much as 40% of a user's time trying to solve software related issues (10).

As we train the next generation of scientists, and during the course of professional development, it is imperative that individuals reflect on and take control of the digital tools used to plan, conduct and share work. Frustration can be avoided if the tool being used is understood. Ideally, any skills learned during any part of an individual's scientific career should be transferable. This is not possible if proprietary software solutions are used as there is no guarantee the software will be available at a new role either due to funding, dropped support or incompatibility with other systems.

One part of the solution to this, as demonstrated clearly by software projects like GNU/Linux (referred to herein as Linux), is the use of open source plain file formats like text files. Text files are human and computer readable, have demonstrable longevity and, crucially, are free and open source. Coupling this with tools that allow users to build, maintain and deploy their own solutions could resolve many of the frustrations seen with modern computer use.

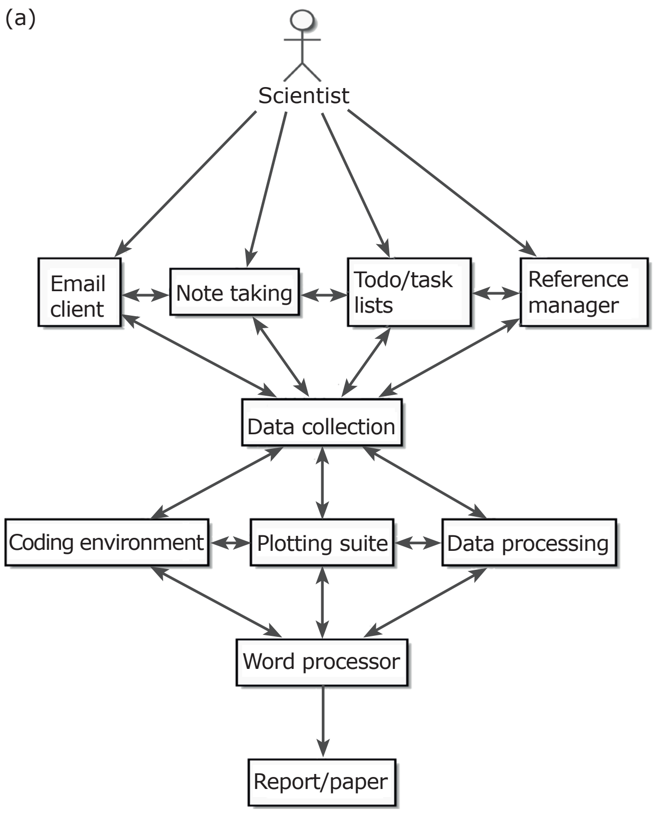
Herein a demonstration of a workflow using a single tool, working with just text files, that can be used to radically change the workflow of a modern, flexible and agile scientist. The key benefits are increased productivity, return on investment, cost and environment, health and safety *via* improved ergonomics. In this viewpoint it will be demonstrated that such a solution exists and how it can be used in the context of corporate R&D.

Emacs and Org-Mode

Figure 1 shows two simplified workflows. **Figure 1(a)** shows the current state for many scientists. Each box in this flow represents a separate piece of software. These often have different shortcut keys, require many open programs and limit the user in terms of customisability and automatic flows. Each box may represent a different piece of software with separate associated upkeep costs, adding to both R&D expenditure and cost to monitor and ensure compliance with licenses. **Figure 1(b)** shows one possible solution where a single software solution replaces all the programs in a digital workflow. This workflow is possible with the open source and free program: Emacs.

Emacs

Emacs is a fully programmable and extensible text editor. It is used widely in the IT and programming fields. Originally developed in the 1970s, the version used today (GNU Emacs - referred to herein as Emacs) was developed in the 1980s by



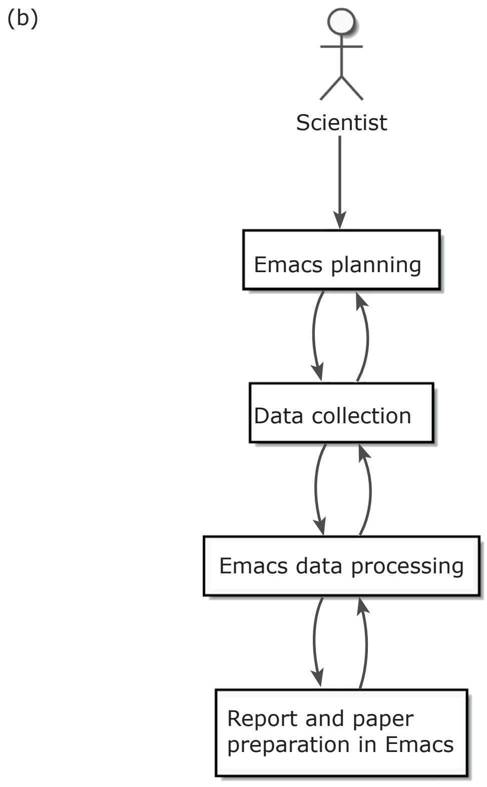


Fig. 1. Two simplified scientific workflows using: (a) current offerings; and (b) Emacs

Richard Stallman. It may seem retrograde that a decades old software solution can compete with newer offerings, but its longevity speaks to its utility. Emacs has been maintained and updated throughout this period with versions available across Windows, macOS and Linux.

Out of the box Emacs is a blank canvas. The decades of use mean that many contributors have written, maintained and updated a large number of packages that can be downloaded and used for free. These packages are completely user customisable and self-documenting. Emacs allows the user to employ these packages to build what is needed from the ground up. The below examples demonstrate how this approach can be used in a range of tasks in corporate R&D. This was built and personalised in-house with speed and ease of use being key. By building this tool from the ground up there is no bloat or incompatibilities that come with other, long lived, commercial solutions.

Figure 2(a) and **2(b)** shows the software loaded in either its unmodified form or after the application of one of the many distributions, in this case Spacemacs. These distributions come preconfigured for ease of use and with many quality of life features. It is possible for a user to use one of these or to build their own version.

Because the below use cases can be achieved from within one piece of software, productivity and focus can be retained with the use of suite-wide shortcuts and hot keys. This reduces the possibility of fragmented work which can reduce productivity (11). Emacs is also fully controllable from the keyboard, again improving speed, productivity and ergonomics.

Org-Mode

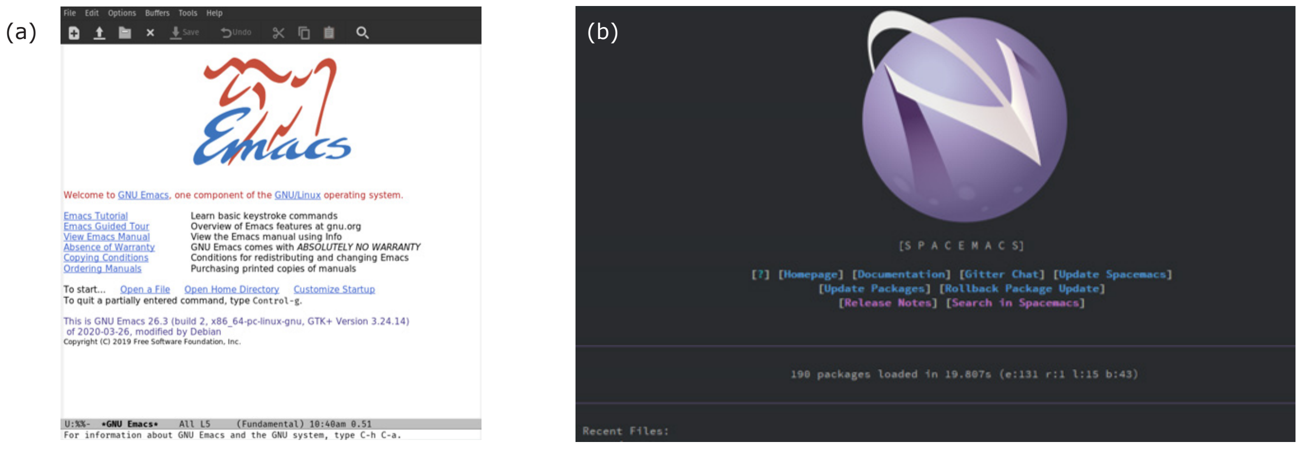
Org-mode is a major mode (a set of instructions for how certain files should be handled) for Emacs which was developed in 2003 by astrophysicist Carsten Dominik. Initially as a way to organise Dominik's work, it has grown into a full suite. Allowing for everything from 'todo' task management to note taking and scientific manuscript preparation.

Importantly, it allows for a single document to contain data, working code and prose (12). Org mode has several minor modes (options that can be turned on or off) which can unlock advanced features impossible with other free or commercial solutions. These will be discussed in the following sections.

Scientific Overhead

Data generation does not happen in a vacuum. A scientist's work day includes 'scientific overheads' that can dramatically lower the time spent by an individual on the act of conducting high quality science (13). Indeed only ~40% of young researchers' time in academia is spent on research, with the majority of the remaining time spent on writing and administration (14).

This is represented pictographically as the first set of software in **Figure 1(a)**. This can be thought of as everything up to the act of experimentation along with all the administration tasks associated with modern knowledge work. Emails, meetings and conferences all add to the overhead workers face. The following section is not an exclusive list of what can be done but aims to demonstrate a few case studies of how Emacs can remove the



```
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial          Learn basic keystroke commands
Emacs Guided Tour       Overview of Emacs features at gnu.org
View Emacs Manual       View the Emacs manual using Info
Absence of Warranty     GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions      Conditions for redistributing and changing Emacs
Ordering Manuals        Purchasing printed copies of manuals

To start...   Open a File   Open Home Directory   Customize Startup
To quit a partially entered command, type Control-g.

This is GNU Emacs 26.3 (build 2, x86_64-pc-linux-gnu, GTK+ Version 3.24.14)
 of 2020-03-26, modified by Debian
Copyright (C) 2019 Free Software Foundation, Inc.

U:%%-  *GNU Emacs*    All L5     (Fundamental) 10:40am 0.51
For information about GNU Emacs and the GNU system, type C-h C-a.
```

```
[?] [Homepage] [Documentation] [Gitter Chat] [Update Spacemacs]
      [Update Packages] [Rollback Package Update]
         [Release Notes] [Search in Spacemacs]

    190 packages loaded in 19.807s (e:131 r:1 l:15 b:43)

Recent Files:
```

Fig. 2. (a) Emacs splash screen; (b) Spacemacs splash screen burden of scientific overheads by consolidation of tasks with Emacs and Org-mode.

Daily Planning

The act of producing, reviewing and executing a plan is an essential component of problem solving (15). Time management behaviours improve job satisfaction and health while negatively impacting stress (16). Org-mode allows for easy task management and planning from within the Emacs environment.

By setting up 'Org-capture', a package that works with Org-mode, todos can be captured and stored centrally from anywhere within Emacs. This makes capturing and recording tasks without interruption to flow trivial. Agendas and todo lists can be automatically populated from multiple sources (for example, reading list, meeting notes, project files). Importantly this approach works well with systems like 'getting things done' while staying flexible enough to allow for individual customisation (17). Examples of todo management as well as automatically generated agenda views can be found in **Figure 3(a)** and **3(b)** respectively.

Administration

Additionally other tedious tasks can be automated. The use of tools like 'Yasnippet' allow for chunks of text to be stored and pasted into a document with only a few key presses. The production of meeting notes, for example, can be sped up by producing a template which can be imported. These can be exported *via* a .tex file and rendered into a PDF using LaTeX. This may seem arduous but, once set up, this is completely automated.

Macros can also be recorded and called when needed. If any task is done repeatedly then tools with Emacs can be used to automate that process. This reduction of overheads frees up a scientist to allow them to do what generates value for companies and academic institutions alike.

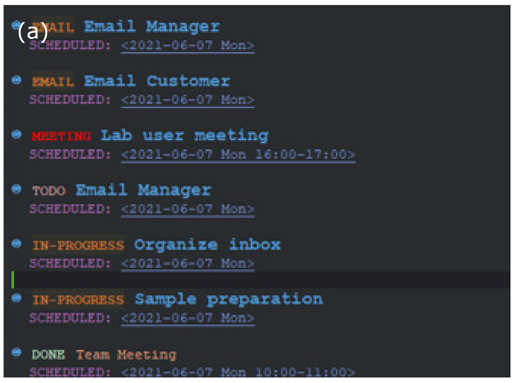
In a world where scientists are not just expected to produce data but be fully fledged knowledge workers, tools like this are invaluable. Their flexibility and utility can be tailored to the user's workflow, enabling high productivity work to be conducted.

Reference Management

The act of collecting, reading and making notes on reference materials is a key aspect of scientific work. Importantly any possible solution to digitalise this should allow for citations to be placed within documents as well as easy access to referencing styles. This is possible with commercial solutions and even some open source options. Where an Emacs workflow outshines all is that the reference manager, note taking, citation tools and writing program are all one.

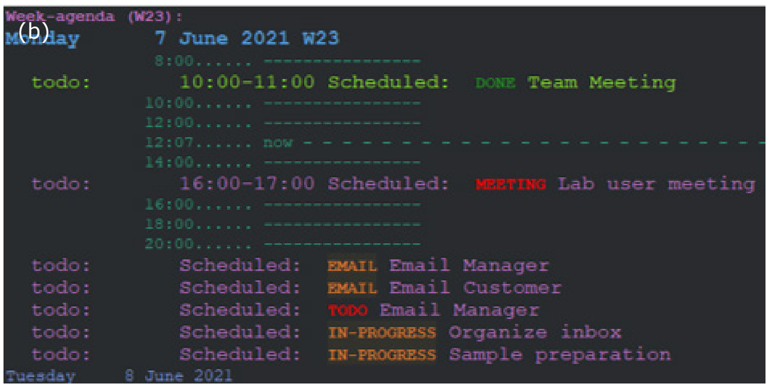
Packages like 'Org-ref' allow for import of PDFs from digital object identifiers (DOIs) allowing for fast import and conversion into a defined bibtex file (the plain text file used by LaTeX to generate citations). Notes can be accessed quickly using a package like 'Interleave' or 'Org-noter' which allows for automated note taking during the reading of a document, **Figure 4**.

Linking of notes and PDFs is extremely powerful and a rarity in the reference manager space. Due to the notes being in plain text they are also searchable unlike PDF highlighting or other, non-text or paper based, approaches.



(a)

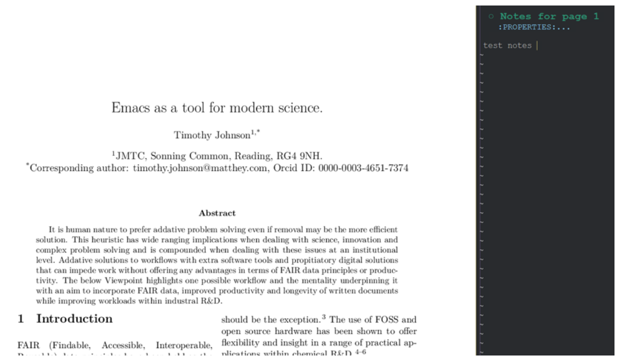
```
EMAIL Email Manager
SCHEDULED: <2021-06-07 Mon>

EMAIL Email Customer
SCHEDULED: <2021-06-07 Mon>

MEETING Lab user meeting
SCHEDULED: <2021-06-07 Mon 16:00-17:00>

TODO Email Manager
SCHEDULED: <2021-06-07 Mon>

IN-PROGRESS Organize inbox
SCHEDULED: <2021-06-07 Mon>

IN-PROGRESS Sample preparation
SCHEDULED: <2021-06-07 Mon>

DONE Team Meeting
SCHEDULED: <2021-06-07 Mon 10:00-11:00>
```



(b)

```
Week-agenda (W23):
Monday      7 June 2021 W23
               8:00...... ----------------
  todo:         10:00-11:00 Scheduled:  DONE Team Meeting
              10:00...... ----------------
              12:00...... ----------------
              12:07...... now - - - - - - - - - - - - - - - - - - - - - - - - -
              14:00...... ----------------
  todo:         16:00-17:00 Scheduled:  MEETING Lab user meeting
              16:00...... ----------------
              18:00...... ----------------
              20:00...... ----------------
  todo:         Scheduled:  EMAIL Email Manager
  todo:         Scheduled:  EMAIL Email Customer
  todo:         Scheduled:  TODO Email Manager
  todo:         Scheduled:  IN-PROGRESS Organize inbox
  todo:         Scheduled:  IN-PROGRESS Sample preparation
Tuesday     8 June 2021
```

Fig. 3. (a) Todo lists; (b) agenda views



Emacs as a tool for modern science.

Timothy Johnson[1,*]

[1]JMTC, Sonning Common, Reading, RG4 9NH.
[*]Corresponding author: timothy.johnson@matthey.com, Orcid ID: 0000-0003-4651-7374

Abstract

It is human nature to prefer addative problem solving even if removal may be the more efficient solution. This heuristic has wide ranging implications when dealing with science, innovation and complex problem solving and is compounded when dealing with these issues at an institutional level. Addative solutions to workflows with extra software tools and propitiatory digital solutions that can impede work without offering any advantages in terms of FAIR data principles or productivity. The below Viewpoint highlights one possible workflow and the mentality underpinning it with an aim to incorporate FAIR data, improved productivity and longevity of written documents while improving workloads within industral R&D.

1 Introduction

FAIR (Findable, Accessible, Interoperable,

should be the exception.[3] The use of FOSS and open source hardware has been shown to offer flexibility and insight in a range of practical applications within chemical R&D.[4–6]

```
○ Notes for page 1
  :PROPERTIES:...

test notes
```

Fig. 4. An example of note taking while viewing a PDF using Interleave

Post Experiment Workload

Data Analytics

One of the benefits of multidisciplinary teams is learning about best practices outside of one's field. One concept that has taken hold in the computer science world is that of literate programming. Literate programming is the idea that written code should not just tell a computer what to do but that it is imperative that the code also informs a human about what is running (18).

This approach should be common to scientists. The aim of written reports, manuscripts and presentations is to display complex data and analysis in an easy to understand form for humans. The problem, as we approach more complex analysis, is that: (a) the analysis is split from the final report or manuscript which leads to loss of reproducibility; or (b) that the analysis is hidden in proprietary software that does not conform to FAIR principles nor the longevity principles a large corporate or academic institution may expect.

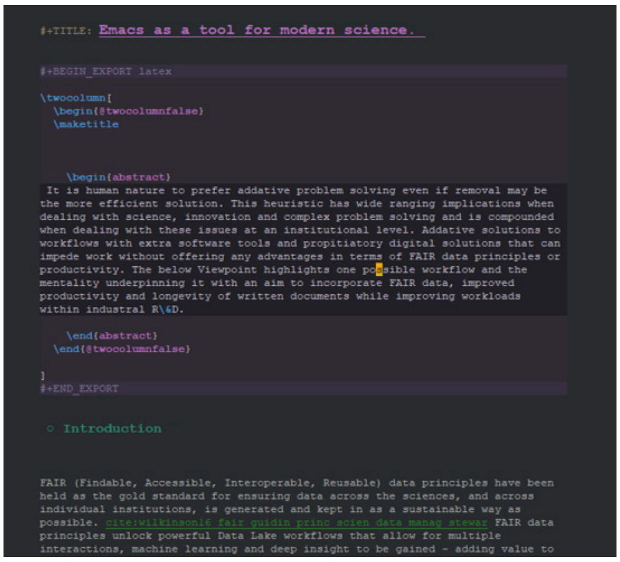
Org-mode, by utilising 'Org-babel', allows for chunks of code to be written and executed from within a single document. Variables can be extracted from these code blocks and then embedded in the text or fed into other code blocks. There are clear parallels between this type of approach and that of the IPython/Jupyter notebooks. These notebooks offer similar advantages in combining prose and code, allowing for reproducibility in data analytics. Both Emacs Org-mode and IPython/Jupyter notebooks offer parallelisation as a feature within the language. These notebooks do, however, suffer from the same issues described above as they form part of a fragmented software solution. As will be described below, they also lack the ability to embed analysis to a final manuscript.

Plotting can be done in the same way with direct output to a number of image formats that can, in turn, be embedded into the Org file. If one simply wants a way to record one's work in an easy to follow format which is completely human readable then Org-mode makes that a simple task. Where the power of this approach becomes evident is when this is linked with manuscript or report production.

Manuscript and Report Preparation

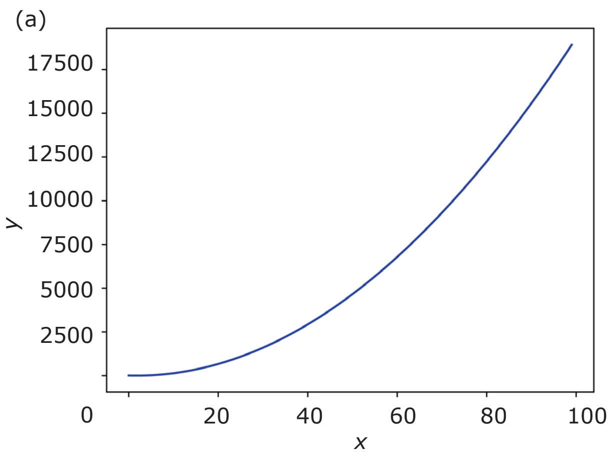
Org files are human readable with any text editor but Emacs unlocks many ways to quickly access the myriad of features not available outside Emacs. Importantly Org files can be exported in a range of formats including PDFs, markdown and open document formats. This manuscript was prepared as a Org file which was automatically processed into a .tex file and rendered into a PDF. Tools like 'Writeroom-mode' format documents to allow for a distraction-free writing experience, **Figure 5**.

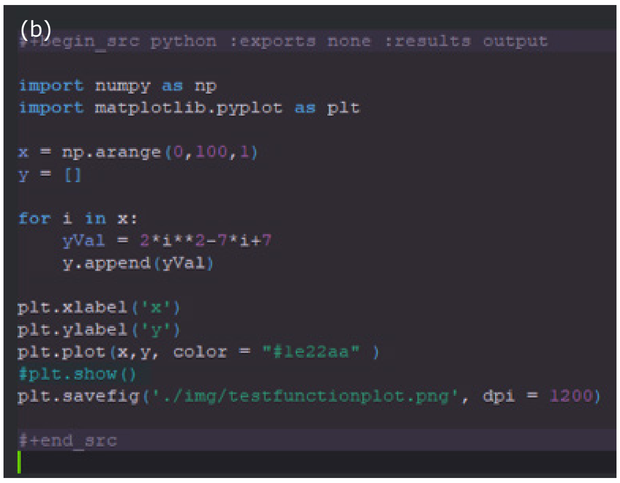
When it comes to reports and manuscripts written in Emacs and Org-mode it is trivial to produce literate documents. Data and analysis can all be included within the manuscript which is also machine accessible. This works well with FAIR principles allowing for a human readable document to also act as metadata and a repository for computer readable data. To demonstrate this **Figure 6(a)** is a plot rendered by Python code embedded in this document. The



```
#+TITLE: Emacs as a tool for modern science.

#+BEGIN_EXPORT latex

\twocolumn[
  \begin{@twocolumnfalse}
  \maketitle



    \begin{abstract}
 It is human nature to prefer addative problem solving even if removal may be
the more efficient solution. This heuristic has wide ranging implications when
dealing with science, innovation and complex problem solving and is compounded
when dealing with these issues at an institutional level. Addative solutions to
workflows with extra software tools and propitiatory digital solutions that can
impede work without offering any advantages in terms of FAIR data principles or
productivity. The below Viewpoint highlights one possible workflow and the
mentality underpinning it with an aim to incorporate FAIR data, improved
productivity and longevity of written documents while improving workloads
within industral R\&D.

    \end{abstract}
  \end{@twocolumnfalse}

]
#+END_EXPORT


 ○ Introduction


FAIR (Findable, Accessible, Interoperable, Reusable) data principles have been
held as the gold standard for ensuring data across the sciences, and across
individual institutions, is generated and kept in as a sustainable way as
possible. cite:wilkinson16 fair guidin princ scien data manag stewar FAIR data
principles unlock powerful Data Lake workflows that allow for multiple
interactions, machine learning and deep insight to be gained - adding value to
```

Fig. 5. A view of a draft of this manuscript from within Emacs using Writeroom-mode





```
#+begin_src python :exports none :results output

import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0,100,1)
y = []

for i in x:
    yVal = 2*i**2-7*i+7
    y.append(yVal)

plt.xlabel('x')
plt.ylabel('y')
plt.plot(x,y, color = "#1e22aa" )
#plt.show()
plt.savefig('./img/testfunctionplot.png', dpi = 1200)

#+end_src
```

Fig. 6. Examples of: (a) plot produced from: (b) code written within an Org file

values have been calculated from data within the file. The code snippet for this can be seen in **Figure 6(b)**. If any changes are made to the analysis or the data, the plot is updated. This means that a single Org file can be provided and all data and analytics can be reproduced. It also makes the process of data analytics and report writing much easier. Any changes to the analysis will be updated in the text, either *via* plots or by embedded variables. This reduces the cognitive load associated with making requested changes, either during the peer review cycle or due to feedback from colleagues.

Previous reports have demonstrated how experimental data can be embedded into PDFs produced from Emacs allowing for a manuscript or report to contain all the data reported (19). The benefits of this are clear for both scientific integrity and rigour but also as a way to ensure a report or manuscript can be understood fully if an employee were to leave an institution, retaining the value of that work indefinitely.

Limitations

Emacs has a reputation of being difficult to learn and this should not be ignored. Emacs has a learning curve however this can be as steep or as shallow as the user needs. Emacs distributions like 'Spacemacs' or 'Doom Emacs' allow for mnemonics key bindings and other quality of life features. Vanilla Emacs has many of the graphical user interface aspects you would expect, such as menus, which allows for most of the functionality to be explored. Becoming proficient takes time however this comes slowly as utility is unlocked. As summarised by John Kitchin:

> "Scientific publishing is a career-long activity, and one should not shy away from learning a tool that can have an impact over this time scale." (19)

While this still holds true, the author feels it is imperative to add the same is true of all aspects of a scientist's workflow including productivity, reference management and data analytics.

Additionally, despite best efforts, all aspects of an Emacs workflow may not be possible. Email is possible within Emacs. However due to some institutions' policies, such as Azure Information Protection, it may not be possible to set up due to issues with accessing confidential information without support from the host organisation. In this case it would not be possible to utilise such a tool. Similarly, while FOSS software allows for flexibility and the ability to create one's own code, a user will be dependent on the software being correctly maintained. This lack of warranty is an inherent issue with FOSS. With repositories like GitHub (and similar), it is possible to access, fork and publish or maintain one's own repositories for tools at a personal or institutional level, providing licensing conditions allow.

The maintenance overhead should not be underestimated, especially when considering issues with business continuity. However, this is not a new problem and, if the value is seen, institutions can add resource to deliver long lasting FOSS solutions. Parallels can be drawn to the development of the

Linux kernel. Here private companies contribute extensively to the FOSS development because there is an understanding of the value of that project to their business interest (20).

While FOSS approaches offer great benefits, the use of proprietary or closed source software is preferable when that software offers utility not possible by other routes. Complex analysis using statistical software, complex peak fitting or databases requiring subscriptions are still a reality of the profession. When these tools are needed the approach outlined above still works providing the data can be exported from such a program into a plain text format. If this is not possible and FAIR principles cannot be upheld, the use of such a tool should be re-evaluated to determine if its use can facilitate long term and sustainable analysis.

Conclusions

Emacs is a powerful and versatile tool for modern science. It facilitates the production, handling and analysis of data in a FAIR fashion while allowing modern scientists to be as agile as possible. By using tools under one FOSS umbrella huge productivity gains can be realised along with improvements in ergonomics and associated cost benefits with the removal of proprietary software tools. The learning curve should be viewed in the context of a lifelong scientific career. With institutions understanding the value of data beyond a single scientist, applying (or supporting individuals who wish to apply) this type of workflow more widely would have a profound and long last effect beyond the career of just one scientist.

Acknowledgements

The author would like to thank Ed Wright, Ludovic Briquet, Carl Tipton and Cristina Estruch Bosch of Johnson Matthey for fruitful discussion and feedback during the drafting process.

Mac and macOS are trademarks of Apple Inc, registered in the USA and other countries and regions. Microsoft, Azure, Office and Windows are trademarks of the Microsoft group of companies. All other trademarks are the property of their respective owners.

References

1. M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons, *Sci. Data*, 2016, **3**, 160018
2. R. Hai, S. Geisler and C. Quix, 'Constance: An Intelligent Data Lake System', in "SIGMOD '16: Proceedings of the 2016 International Conference on Management of Data", Association for Computing Machinery, New York, USA, June, 2016, pp. 2097–2100
3. W. Hasselbring, L. Carr, S. Hettrick, H. Packer and T. Tiropanis, *Inform. Technol.*, 2020, **62**, (1), 39
4. F. Massingberd-Mundy, S. Poulston, S. Bennett, H. H.-M. Yeung and T. Johnson, *Sci. Rep.*, 2020, **10**, 17355
5. M. D. M. Dryden, R. Fobel, C. Fobel and A. R. Wheeler, *Anal. Chem.*, 2017, **89**, (8), 4330
6. N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch and G. R. Hutchison, *J. Cheminform.*, 2011, **3**, 33
7. D. Schindler, B. Zapilko and F. Krüger, 'Investigating Software Usage in the Social Sciences: A Knowledge Graph Approach', 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, 31st May–4th June, 2020, "The Semantic Web", eds. A. Harth, S. Kirrane, A.-C. N. Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase and M. Cochez, Lecture Notes in Computer Science, Vol. 12123, Springer, Cham, Switzerland, 2020, pp. 271–286
8. G. S. Adams, B. A. Converse, A. H. Hales and L. E. Klotz, *Nature*, 2021, **592**, (7853), 258
9. 9th WSEAS International Conference on Data Networks, Communications, Computers (DNCOCO '10), University of Algarve, Faro, Portugal, 3rd–5th November, 2010, "Advances in Data Networks, Communications, Computers", eds. N. E. Mastorakis and V. Mladenov, World Scientific and Engineering Academy and Society Press, Athens, Greece, 2010
10. J. Lazar, A. Jones and B. Shneiderman, *Behav. Inform. Technol.*, 2006, **25**, (3), 239
11. A. N. Meyer, L. E. Barton, G. C. Murphy, T. Zimmermann and T. Fritz, *IEEE Trans. Software Eng.*, 2017, **43**, (12), 1178
12. E. Schulte and D. Davison, *Comput. Sci. Eng.*, 2011, **13**, (3), 66

13. M. L. Pace, *Limnol. Oceanogr. Bull.*, 2020, **29**, (1), 20
14. B. Maher and M. S. Anfres, *Nature*, 2016, **538**, (7626), 444
15. D. J. Simons and K. M. Galotti, *Bull. Psychon. Soc.*, 1992, **30**, (1), 61
16. B. J. C. Claessens, W. van Eerde, C. G. Rutte and R. A. Roe, *Person. Rev.*, 2007, **36**, (2), 255
17. F. Heylighen and C. Vidal, *Long Range Plan.*, 2008, **41**, (6), 585
18. D. Cordes and M. Brown, *Computer*, 1991, **24**, (6), 52
19. J. R. Kitchin, *ACS Catal.*, 2015, **5**, (6), 3894
20. D. Homscheid, J. Kunegis and M. Schaarschmidt, 'Private-Collective Innovation and Open Source Software: Longitudinal Insights from Linux Kernel Development', 14th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E, Delft, The Netherlands, 13th–15th October, 2015, "Open and Big Data Management and Innovation", eds. M. Janssen, M. Mäntymäki, J. Hidders, B. Klievink, W. Lamersdorf, B. van Loenen and A. Zuiderwijk, Lecture Notes in Computer Science, Vol. 9373, Springer, Cham, Switzerland, 2015, pp. 299–313

The Author

Timothy Johnson (PhD, MChem, CChem) is a Senior Scientist who has worked at Johnson Matthey since 2016. His research interests focus on the production, characterisation and testing of porous materials for industrial applications. He is passionate about understanding workflows to reduce workloads and improve productivity.