

Tips and advice when creating a python software for lab members to use in academia



PyData Global 2021

Jeremy Selva    

Images by Amonrat Rungreangfangsai

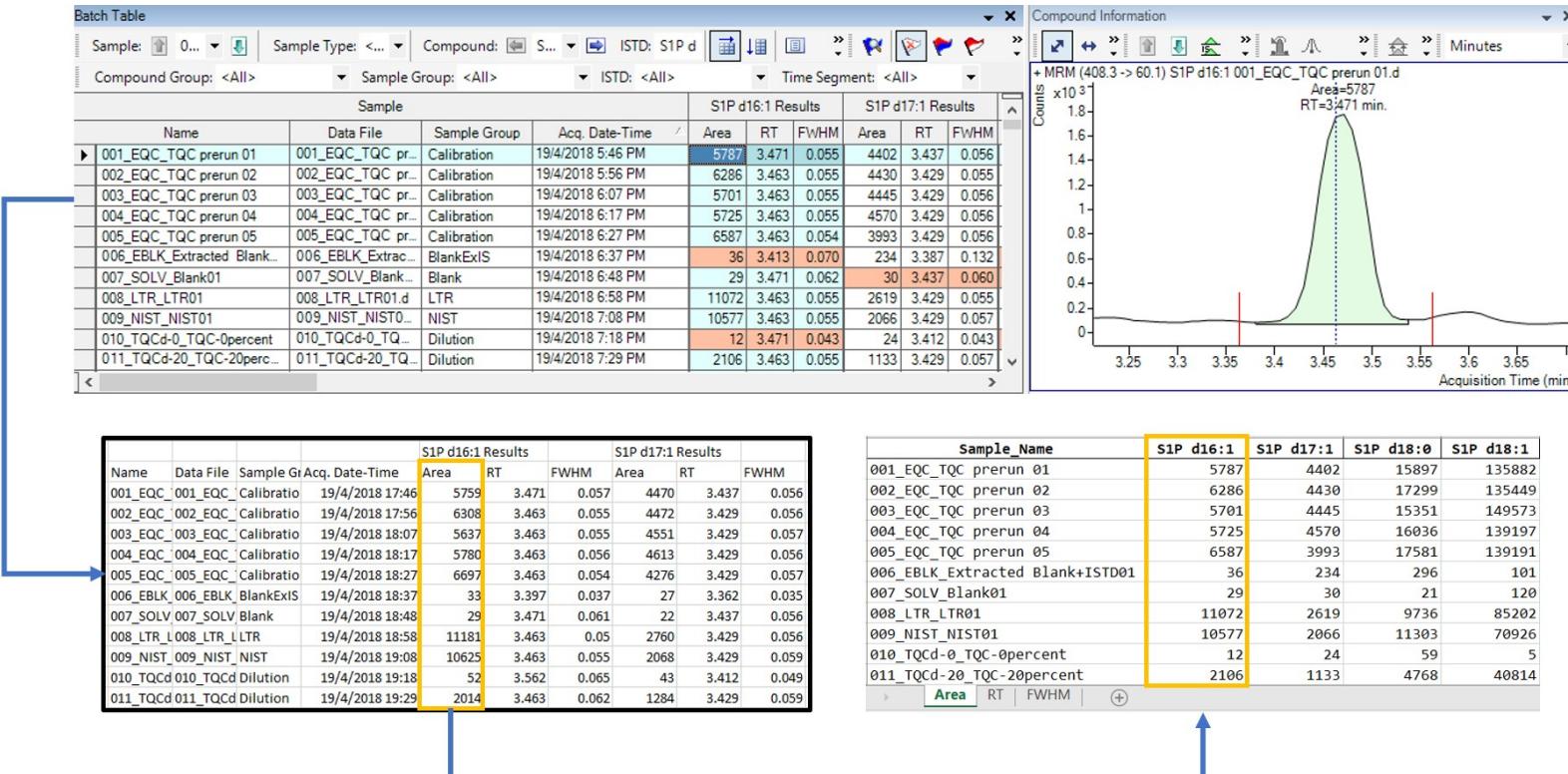
Xaringan Slide Template by Sharla Gelfand 

<https://jauntyjjs.github.io/PyDataGlobal2021Talk> 

Introduction

It started with a python script used to organise Mass Spectrometry data for my project.

Export
as CSV



Introduction

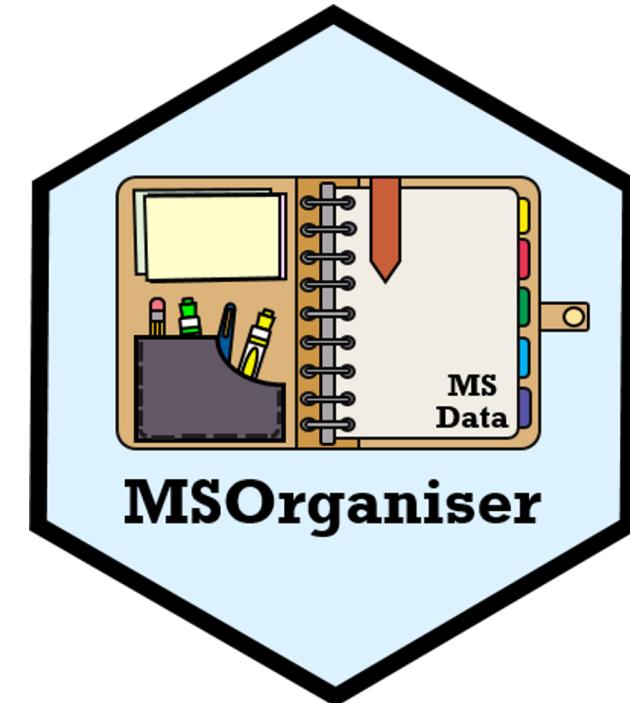
Many asks me to fix/run similar scripts created by past lab members.

- *"This script does not work on my computer anymore. I don't know why."* 🤔



Introduction

I combined all these scripts into one Windows executable software in Python called **MSOrganiser**.



<https://github.com/SLINGhub/MSOrganiser>

Introduction

Looking back, I realised that things have not been easy...



Introduction

In academia, most of us are not trained in coding or software development. It is hard for us to write software for our own research paper, it is even more challenging to create software for others to use.

[Daniel Lemire's blog](#)

On the quality of academic software

- Most academic researchers write software for themselves. As Cook put it: “People who have only written software for their own use have no idea how much work goes into writing software for others.” Cooking your own food is a lot easier than being a chef in a restaurant. The difference between the two is at least an order of magnitude, if not two.

There is very little appreciation for this fact in academia. The default is to write throw-away code: you write the code, you use it and you forget about it. Issues like maintenance and documentation are discussed at length in some classes, but it is rarely put in practice within academia.



Introduction

In this talk, I decide to share some things that have helped me tremendously.



1: Convert scripts to a web service or executable software

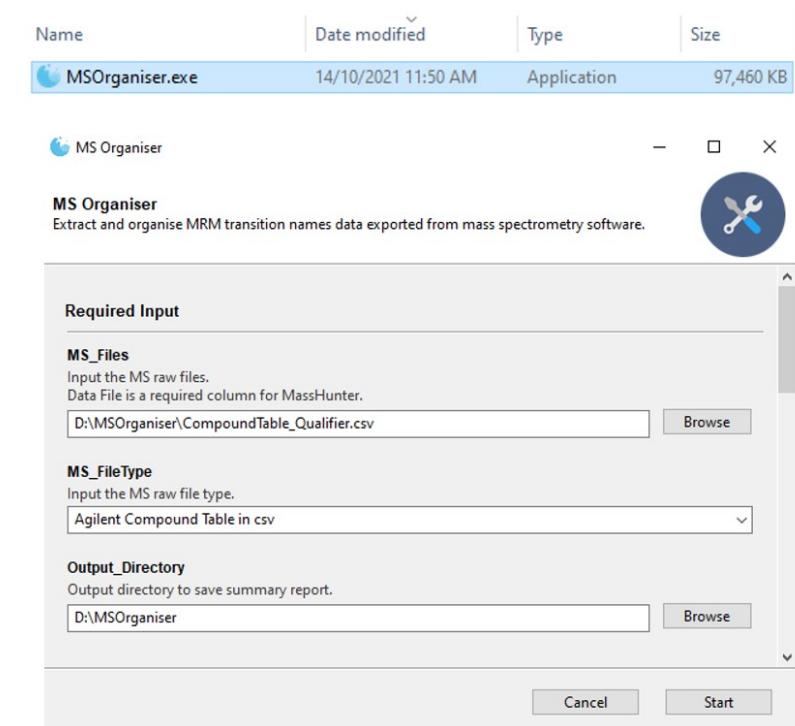
Users make less error when clicking buttons compared to typing in command lines

Attempt 1 ... Need to install Python and libraries

Attempt 50 ... Figure out how the script works

Attempt 100 ... Finally

Attempt 1 ... Double-click, fill in parameters, click Start



VS

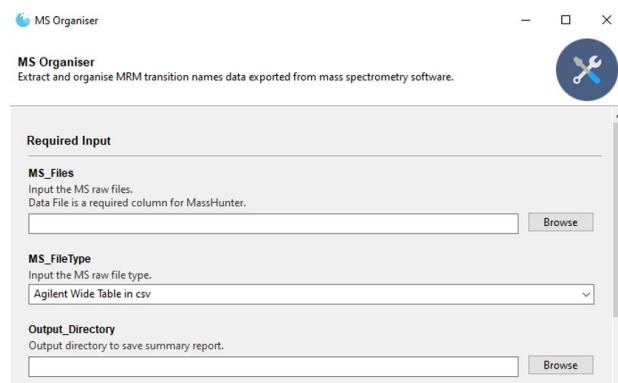
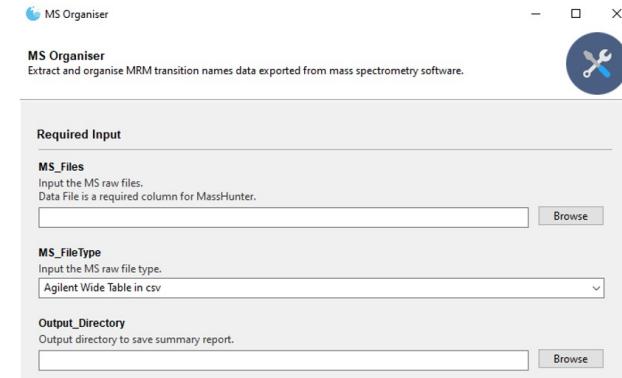
Advice 1 Tips

- Chriskiehl's [Gooey](#) to create a simple GUI interface quickly.
- [Pyinstaller](#) to convert the scripts to a stand alone executable program.

```
#Required Arguments
#nargs="+" is needed to turn multiple input into a list.
required_args.add_argument('--MS_files',
                           required=True,
                           nargs='+',
                           help="Input the MS raw files.\nData File is a required column for Mi
                                 widget='MultiFileChooser',
                                 default=stored_args.get('MS_Files')))

required_args.add_argument('--MS_FileType',
                           required=True,
                           choices=['Agilent Wide Table in csv',
                                    'Agilent Compound Table in csv',
                                    'Multiquant Long Table in txt'],
                           help='Input the MS raw file type.', default=MS_FileType)

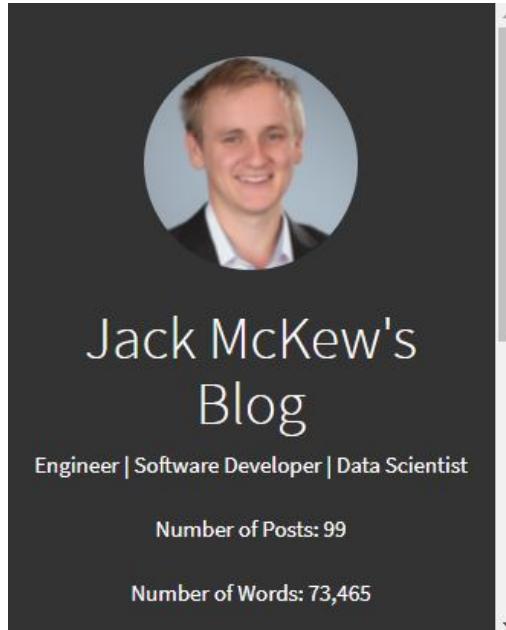
required_args.add_argument('--Output_Directory',
                           required=True,
                           action='store',
                           help="Output directory to save summary report.",
                           widget="DirChooser", default=stored_args.get('Output_Directory'))
```



Name	Date modified	Type	Size
MSOrganiser.exe	14/10/2021 11:50 AM	Application	97,460 KB

Advice 1 Tips

Jack McKew's [blog](#) helps me make use of the two tools to get what I need.

A screenshot of a blog post page. At the top right, there is a navigation bar with links: HOME, ARCHIVES, CATEGORIES, TAGS, SITEMAP, and ATOM. To the right of the navigation bar is a small black cat icon. The main title of the post is "Making Executable GUIs with Python, Gooey & Pyinstaller", centered above the content. Below the title, the text "Posted on Fri 01 November 2019 in [Python](#) • 4 min read" is shown. The main body of the post begins with the text: "Today we will go through how to go from a python script to packaged executable with a guided user interface (GUI) for users. First off we still start by writing the scripts that we would like to share with others to be able to use, especially for users that may be uncomfortable in a programming environment and would feel at home with a GUI." Further down, another paragraph starts with "My personal favourite part about [Gooey](#), is that you are essentially creating a command line interface (CLI) tool, which Gooey then uses to generate a GUI. This eliminates having two separate code bases to facilitate CLI & GUI users, which can be very painful at times." A vertical scrollbar is visible on the right side of the page.

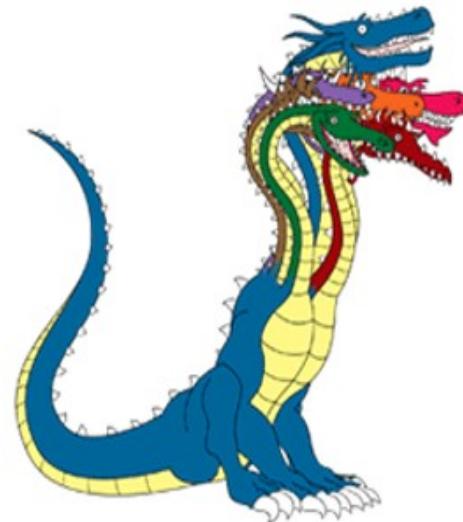
I forgot about new users

Simple program

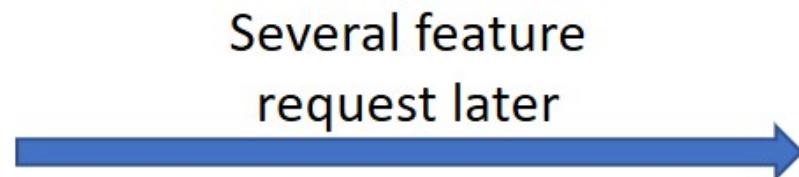


3 page
user manual

Complex program



30 page
user manual



I forgot about new users

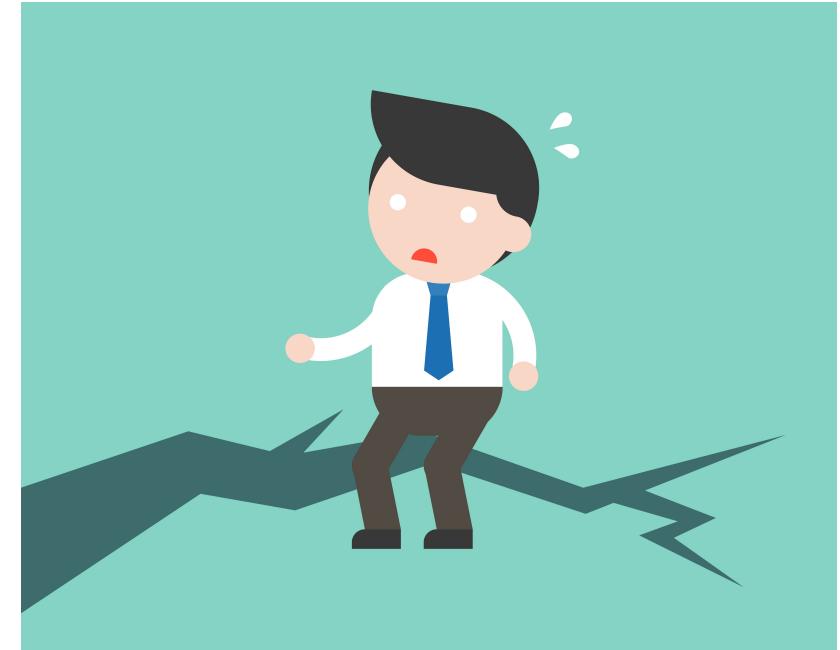
New users typical reaction to complex program

- Scared 😨 and Confused 😳
- "Maybe this tool is not for me" 😐



Cartoon Hydra by how-to-draw-cartoons-online.com

Images by [Amonrat Rungreangfangsai](#)



2: Give a software overview using cheatsheet

MSOrganiser Summary

Input Raw data

Table of valid MRM data formats

MRM transition names data form	Compulsory column names	Restricted column names	
Without Qualifiers	With Qualifiers		
Agilent's WideTable form	Sample • Data File Compound Results • Area	Qualifier Results • Area	Sample • Quantification Message Columns from • ISTD Compound Methods • ISTD Compound Results
Agilent's CompoundTable form	Sample • Data File Compound Method • Name Compound Results • Area	Qualifier Method • Transition Qualifier Results • Area	

Example

Agilent's Wide Table form

Sample	Agilent's WideTable Results	Car 010/1/01/01 Results	Car 010/1/01/02 Results	Car 010/1/01/03 Results	Car 010/1/01/04 Results	Car 010/1/01/05 Results	Car 010/1/01/06 Results
Conditioning_d	70	10	40	10	30	20	300
Conditioning_d	70	20	40	10	30	20	300
Conditioning_d	60	10	40	10	30	20	300
Conditioning_d	60	10	40	10	30	20	300
Sample_Batch_02_d	40	10	30	10	20	10	300
Sample_Batch_02_d	40	10	30	10	20	10	300
Sample_Batch_02_d	40	10	30	10	20	10	300
Sample_Batch_02_d	40	10	30	10	20	10	300

Agilent's Wide Table form with Qualifier

Sample	Agilent's WideTable Results	Car 010/1/01/01 Results	Car 010/1/01/02 Results	Car 010/1/01/03 Results	Car 010/1/01/04 Results	Car 010/1/01/05 Results	Car 010/1/01/06 Results
Conditioning_d	237	34	40	20	42	20	290
Conditioning_d	237	34	40	20	42	20	290
Conditioning_d	149	91	40	20	349	91	290
Conditioning_d	149	91	40	20	349	91	290
Sample_Batch_02_d	105	48	22	34	18	34	290
Sample_Batch_02_d	105	48	22	34	18	34	290
Sample_Batch_02_d	105	48	22	34	18	34	290
Sample_Batch_02_d	105	48	22	34	18	34	290

Agilent's Compound Table form

Compound Method	ISTD_01000000	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000

Agilent's Compound Table form with Qualifier

Compound Method	ISTD_01000000	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000
LPC 14:0	10/10/2010 21:00	ISTD_01000000	ISTD_01000000	ISTD_01000000

Additional Features

Different output format for the organised data.

Output_Format

Select specific file type to output

Excel
Select Option
Excel
CSV

Set Transpose_Results to True to let the sample name be the columns instead of the transition names.

Transpose_Results

Select this option to True to let the samples be the

False
Select Option
True
False

Select the following concatenation options to combine multiple input files to one result file.

Concatenate

Concatenate multiple input files into one output file.

No Concaten.
Select Options
Concatenate
Concatenate along Sample Name (rows)
Concatenate along Transition Name (columns)

Advice 2 Tips

RStudio cheatsheet examples and template and advice

RStudio Cheatsheets

The cheatsheets below make it easy to use some of our favorite packages. From time to time, we will add new cheatsheets. If you'd like us to drop you an email when we do, click the button below.

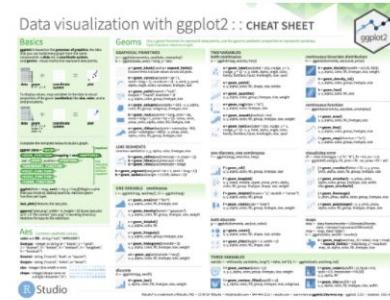
[SUBSCRIBE TO CHEATSHEET UPDATES](#)

- [CONTRIBUTED CHEATSHEETS](#)
- [TRANSLATIONS](#)
- [HOW TO CONTRIBUTE](#)

Data visualization with ggplot2 cheatsheet

The `ggplot2` package lets you make beautiful and customizable plots of your data. It implements the grammar of graphics, an easy to use system for building plots. Updated August 2021.

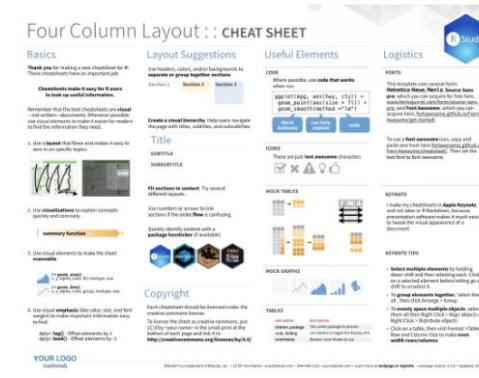
[DOWNLOAD](#)



Contribute a New Cheat Sheet

Want to contribute a cheat sheet of your own?

We'd like to help you make and share high quality cheat sheets on R topics. The template below provides a useful starting place. It contains tips for designing a three or four column cheat sheet, as well as reusable elements to build your sheet with. [Check out the README for more tips on making good cheat sheets.](#)

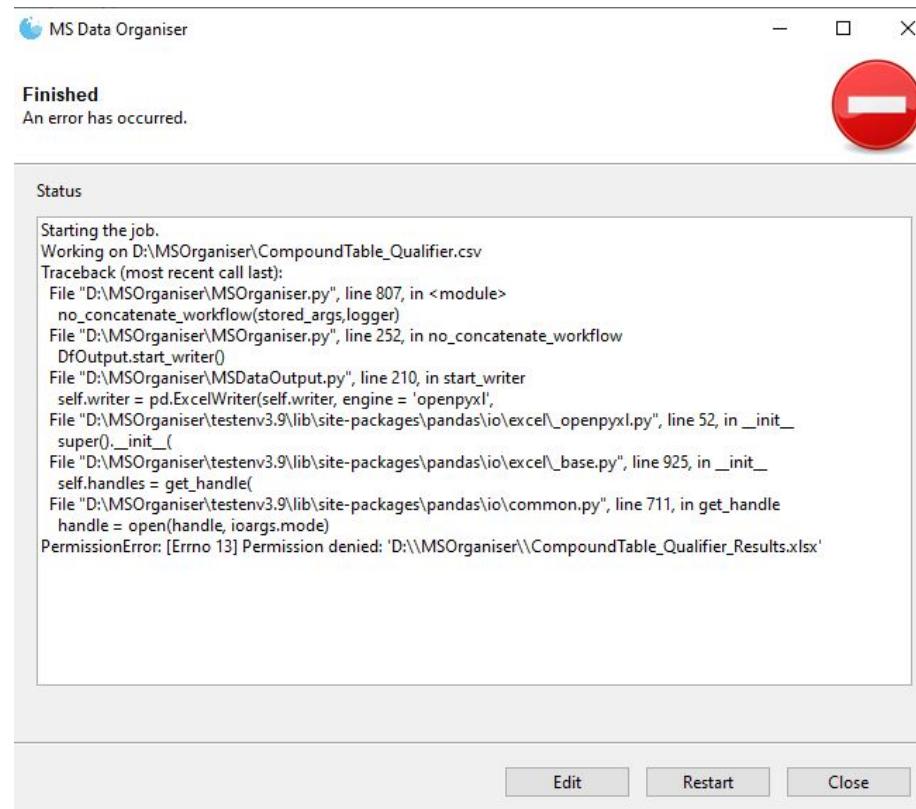


[Download Template for Keynote](#)

[Download Template for Powerpoint](#)

3: Give helpful messages when users make a mistake.

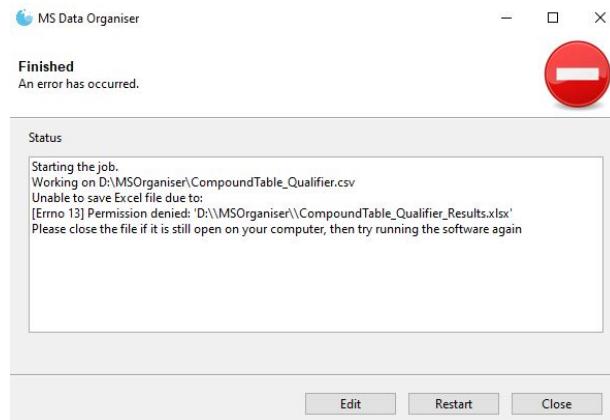
Users and programmers make mistakes sometimes. Software error messages are unavoidable.



3: Give helpful messages when users make a mistake.

However, if you can make them helpful, it does goes a long way.

```
try:  
    self.writer = pd.ExcelWriter(self.writer, engine = 'openpyxl',  
        #options = options,  
        engine_kwargs = {'options': {'strings_to_formulas': False,  
            'strings_to_urls' : False  
        }  
    )  
  
except IOError as i:  
    if self.inguui:  
        print('Unable to save Excel file due to:',flush=True)  
        print(i,flush=True)  
        print('Please close the file if it is still open on your computer, ' +  
            'then try running the software again',flush=True)  
    if self.logger:  
        self.logger.error('Unable to save Excel file due to:')  
        self.logger.error(i)  
        self.logger.error('Please close the file if it is still open on your computer, ' +  
            'then try running the software again')  
    sys.exit(-1)
```



Advice 3 Tips

Saadia Minhas' [blog](#) provides some good tips

4. Be Humble — Don't Blame User

A good error message is humble. It conveys the issues gracefully to its user without blaming him for his actions.

The user can perform an incorrect action again and again. But the design's responsibility is to inform him about his mistakes in a good way.

"A good way to incorporate more human tone to your error messages is to think about explaining it out loud to someone. How does it sound when you speak it in conversation." — [Sonia Gregory](#)



Micheal Lynch's [tip](#) to show openness to criticism

11. Artfully solicit missing information

Sometimes code review notes leave too much room for interpretation. When you receive a comment like, "This function is confusing," you probably wonder what "confusing" means, exactly. Is the function too long? Is the name unclear? Does it require more documentation?

For a long time, I struggled to clarify ambiguous notes without sounding defensive. My instinct was to ask, "What's confusing about it?" but that comes across as grouchy.

Once, I unintentionally sent a vague note to my teammate, and he responded in a way that I found fantastically disarming:

What changes would be helpful?

I love this response because it signals a lack of defensiveness and openness to criticism. Whenever a reviewer gives me unclear feedback, I always respond with some variation of, "What would be helpful?"

Another useful technique is to guess your reviewer's intent and proactively edit your code based on that assumption. For a note like, "this is confusing," give your code a second look. Usually, there's something you can do to improve clarity. A revision communicates to your reviewer that you're amenable to change, even if it's not the one they had in mind.

4: Output preprocessing reports to show accountability.

Academia journals are getting more demanding with regards to software, wanting reusability besides reproducibility.

nature computational science

Explore content ▾ About the journal ▾ Publish with us ▾

[nature](#) > [nature computational science](#) > [editorials](#) > article

Editorial | Published: 23 July 2021

But is the code (re)usable?

[Nature Computational Science](#) 1, 449 (2021) | [Cite this article](#)

852 Accesses | 9 Altmetric | [Metrics](#)

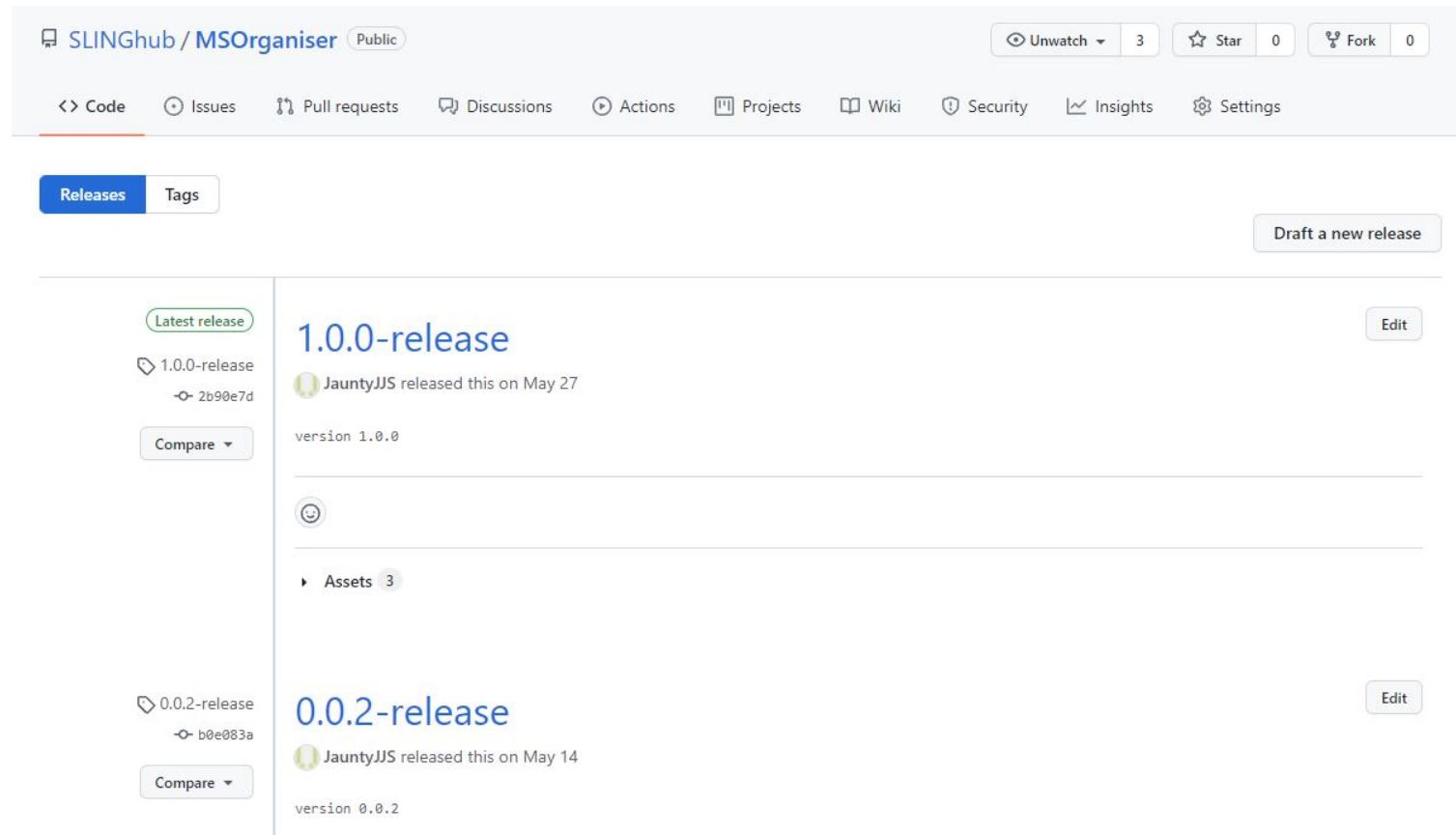
While it is crucial to guarantee the reproducibility of the results reported in a paper, let us also not forget about the importance of making research artifacts reusable for the scientific community.

We editors at *Nature Computational Science* do our best to ensure that the source codes associated with our papers have the aforementioned elements to make them reusable as much as possible. In addition to asking our referees about reusability during the code peer-review process, we also ask authors to fill out a [code and software submission checklist](#) that covers reusability requirements. But we also urge you, the author, to start thinking more about reusability when you are working on a project and submitting a paper. For instance, make sure that you have detailed instructions, making the code easier to install whenever possible, and making it easier to run with new data and parameters as well. A useful tip is to ask a colleague from your lab to try out your code to see whether they can successfully follow the instructions and run your code on a potentially different machine.

<https://www.nature.com/articles/s43588-021-00109-9>

Advice 4 Tips

Tag a label on different software version and encourage users to cite not just the software name but the version number as well.



Advice 4 Tips

Create a report file/table (pdf or excel) that store the user's input parameters

Parameter Report

Parameters	Value
Input_File	CompoundTable_Qualifier.csv
Input_File_Type	Agilent Compound Table in csv
Output_Options	Area
Output_Format	Excel
Concatenate	No Concatenate
Allow_Multiple_ISTD	False
Transpose_Results	False
Long_Table	False
Long_Table_Annot	False

Chris Moffitt's example using WeasyPrint

Mon 16 February 2015

Creating PDF Reports with Pandas, Jinja and WeasyPrint

Posted by Chris Moffitt in articles

Matt Clarke's example using Gilfoyle

How to create PDF reports in Python using Pandas and Gilfoyle

To save time, I created a Python package for generating PDF reports and presentations. Here's how you can use it to create automated ecommerce and marketing reports.

Advice 4 Tips

Show pre-processing results to explain how the software calculate the final results. They are also helpful when there is a need to troubleshoot for logical errors

Optional Settings

Testing
 Testing mode will generate more output tables.

normArea by ISTD = (Area / ISTD_Area)

1	Sample_Name	LPC 20:5	LPC 14:0	LPC 20:0 (IS)
2	05_TQC	0.247414206	1.008425235	1
3	06_TQC_02	0.249451066	1.003607277	1
4	07_TQC_03	0.234231379	1.013153724	1
5	08_TQC_04	0.23028757	0.979591837	1
6	09_TQC_05	0.237342767	0.973820755	1
7	10_TQC_06	0.237774167	1.021283509	1
8	11_TQC_07	0.248128958	1.024426351	1

equals

1	Transition_Name	Transition_Name_ITSD
2	LPC 14:0	LPC 20:0 (IS)
3	LPC 20:0 (IS)	LPC 20:0 (IS)
4	LPC 20:5	LPC 20:0 (IS)
5		

1	Sample_Name	LPC 20:5	LPC 14:0	LPC 20:0 (IS)
2	05_TQC	3612	14722	14599
3	06_TQC_02	3181	12798	12752
4	07_TQC_03	2956	12786	12620
5	08_TQC_04	2979	12672	12936
6	09_TQC_05	3019	12387	12720
7	10_TQC_06	2927	12572	12310
8	11_TQC_07	3017	12456	12159

divided by

1	Sample_Name	LPC 20:5	LPC 14:0	LPC 20:0 (IS)
2	05_TQC	14599	14599	14599
3	06_TQC_02	12752	12752	12752
4	07_TQC_03	12620	12620	12620
5	08_TQC_04	12936	12936	12936
6	09_TQC_05	12720	12720	12720
7	10_TQC_06	12310	12310	12310
8	11_TQC_07	12159	12159	12159

My documentation issue

I started my documentation by trying to follow some advice provided by Lee's [article](#)

PLOS COMPUTATIONAL BIOLOGY

OPEN ACCESS

EDITORIAL

Ten simple rules for documenting scientific software

Benjamin D. Lee 

Published: December 20, 2018 • <https://doi.org/10.1371/journal.pcbi.1006561>

Article	Authors	Metrics	Comments	Media Coverage
▼				
Introduction				
Rule 1: Write comments as you code				
Rule 2: Include examples (and lots of them)				
Rule 3: Include a quickstart guide				

Citation: Lee BD (2018) Ten simple rules for documenting scientific software. PLoS Comput Biol 14(12): e1006561. <https://doi.org/10.1371/journal.pcbi.1006561>

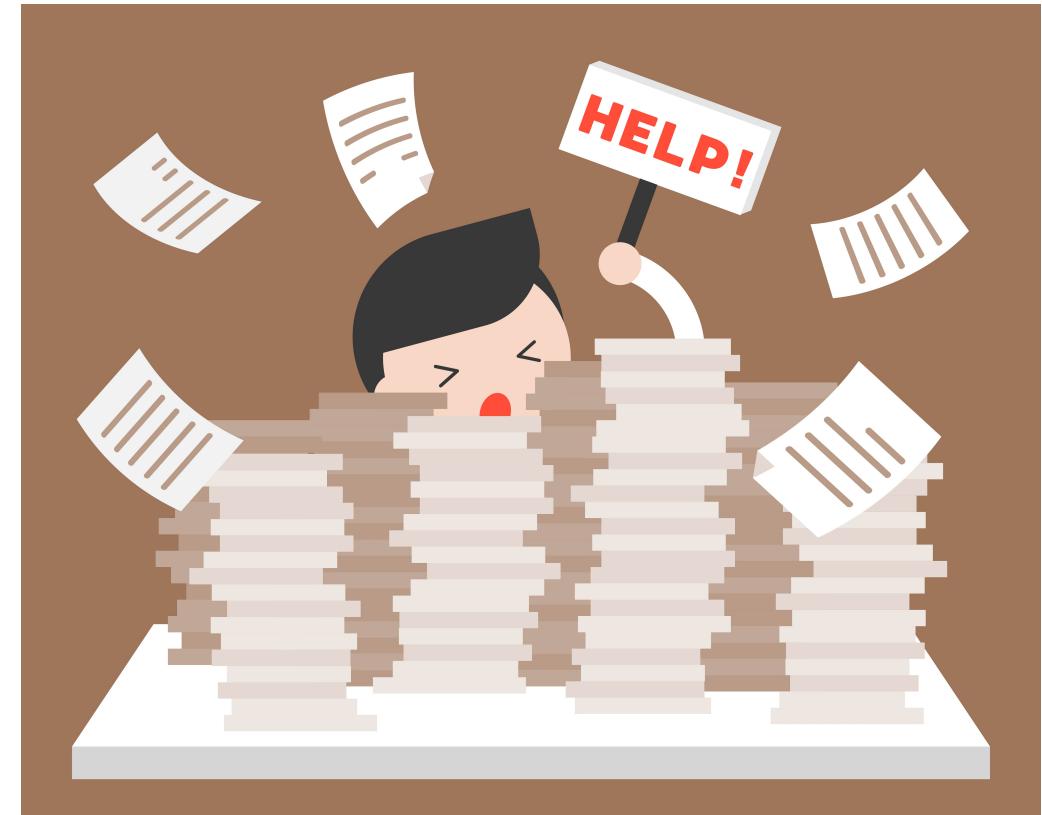
Editor: Scott Markel, Dassault Systemes BIOVIA, UNITED STATES

Published: December 20, 2018

Copyright: © 2018 Benjamin D. Lee. This is an open access article distributed under

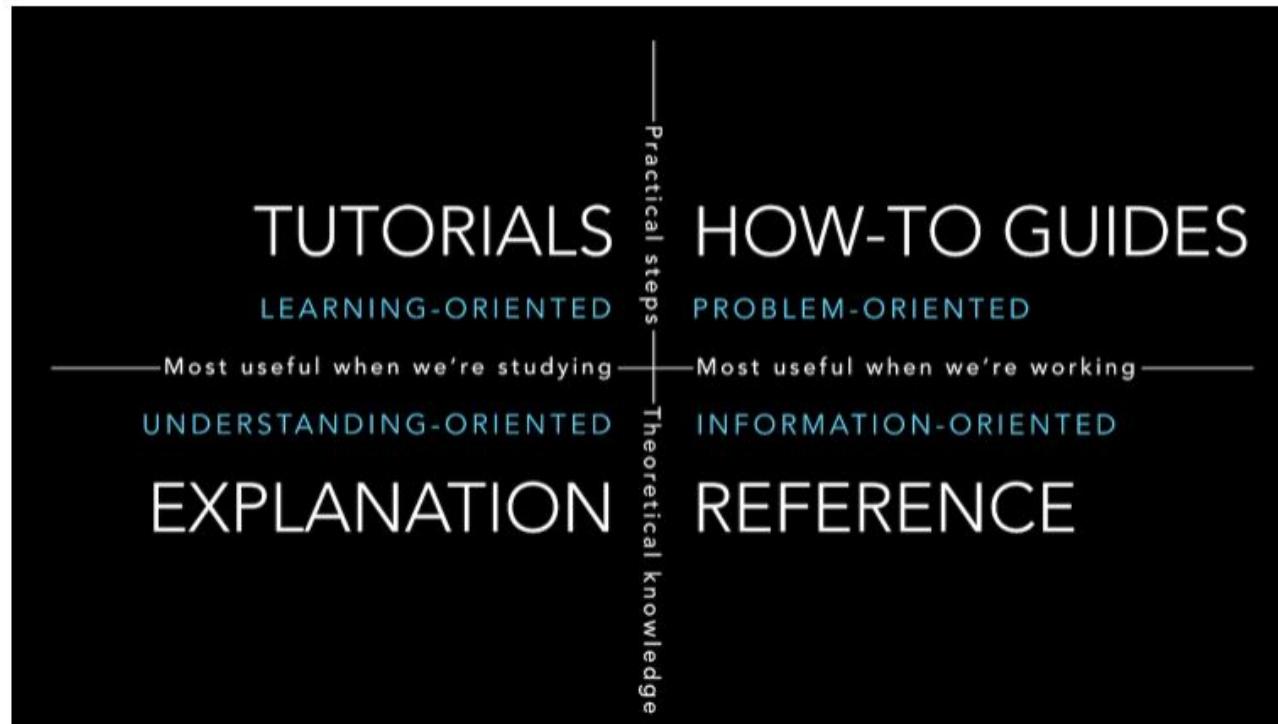
My documentation issue

As my documentation starts to get longer and more complex, people find it hard to understand.



5: Organise your documentation into specific structures

Thankfully, I came across DIVIO [website](#) that encourages me to split my documentations into manageable categories.



The documentation system outlined here is a simple, comprehensive and nearly universally-applicable scheme. It is proven in practice across a wide variety of fields and applications.

5: Organise your documentation into specific structures

Following in its footsteps, this is what I come up with



Advice 5 Tips

Tutorial: Github README

README.md

MSOrganiser

License MIT Maintainer JauntyJJS Python 3.9.4

MSOrganiser is created to provide users a convenient way to extract and organise MRM transition names data exported from mass spectrometry software into an Excel or csv file in a few button clicks.

With the addition of the `MSTemplate_Creator`, the software is also able to normalize the peak area with respect to the internal standard's peak area and calculate the concentration of the analytes.

The screenshot shows the MSOrganiser application. On the left, a 'Batch Table' window displays a list of samples with their corresponding compound names, acquisition dates, and peak areas. A blue arrow points from the 'Export as csv' button at the bottom left to a detailed view of a specific row in the table. On the right, a 'Compound Information' window shows a chromatogram for a sample named '001_EGC_TQC_prenum_01'. The plot shows a single sharp peak at approximately 3.47 minutes, with its area and retention time labeled. Below the plot is a table of peak data for this sample.

Name	Data File	Sample	Acq Date-Time	SIP d16:1 Results	SIP d17:1 Results
001_EGC_TQC_prenum_01	001_EGC_TQC.pr	Calibration	19/4/2018 10:45 PM	Area: 3.471, RT: 3.471, FWHM: 0.056	Area: 3.437, RT: 3.437, FWHM: 0.056
002_EGC_TQC_prenum_02	002_EGC_TQC.pr	Calibration	19/4/2018 11:26 PM	Area: 3.463, RT: 3.463, FWHM: 0.056	Area: 3.429, RT: 3.429, FWHM: 0.056
003_EGC_TQC_prenum_03	003_EGC_TQC.pr	Calibration	19/4/2018 11:27 PM	Area: 3.463, RT: 3.463, FWHM: 0.056	Area: 3.437, RT: 3.437, FWHM: 0.056
004_EGC_TQC_prenum_04	004_EGC_TQC.pr	Calibration	19/4/2018 11:17 PM	Area: 3.463, RT: 3.463, FWHM: 0.056	Area: 3.429, RT: 3.429, FWHM: 0.056
005_EGC_TQC_prenum_05	005_EGC_TQC.pr	Calibration	19/4/2018 10:48 PM	Area: 3.463, RT: 3.463, FWHM: 0.056	Area: 3.437, RT: 3.437, FWHM: 0.056
006_EBLK_Extracted_B	006_EBLK_Extrac	BlankEBL	19/4/2018 10:37 PM	Area: 3.471, RT: 3.471, FWHM: 0.056	Area: 3.437, RT: 3.437, FWHM: 0.056
007_SOVL_Blane#01	007_SOVL_Blane	BlankSOV	19/4/2018 10:48 PM	Area: 29.371, RT: 29.371, FWHM: 0.062	Area: 30.347, RT: 30.347, FWHM: 0.060
008_LTR_LTRO#01	008_LTR_LTRO	LTR	19/4/2018 10:58 PM	Area: 11072, RT: 11072, FWHM: 0.056	Area: 2619, RT: 2619, FWHM: 0.056
009_LTR_LTRO#02	009_LTR_LTRO	LTR	19/4/2018 10:59 PM	Area: 10571, RT: 10571, FWHM: 0.056	Area: 2619, RT: 2619, FWHM: 0.056
010_TQC#6-TQC-0percent	010_TQC#6-TQC	Dilution	19/4/2018 10:18 PM	Area: 12.341, RT: 12.341, FWHM: 0.043	Area: 24.342, RT: 24.342, FWHM: 0.043
011_TQC#20-TQC-20percent	011_TQC#20-TQC	Dilution	19/4/2018 10:29 PM	Area: 2106, RT: 2106, FWHM: 0.055	Area: 1133, RT: 1133, FWHM: 0.057

How To Guide: Cheatsheet

Organise MRM data to tidy form

1. Load MRM Files and specify MS_FileType and Output Directory

MS_Files
Input the MS raw files.
Data File is a required column for MassHunter
Sample Name and Component Name are required columns for Sciex

C:\Users\bchjjs\Desktop\MSOrganiser\tests\testdata\CompoundTableForm.csv;C:\U...

MS_FileType
Input the MS raw file type
Agilent Compound Table in csv

Select Option
Agilent Wide Table in csv
Agilent Compound Table in csv

Output_Directory
Output directory to save summary report
C:\Users\bchjjs\Desktop\MSOrganiser\tests

2. Choose Output options

Output_Options
Select specific information to output

Area
normArea by ISTD
normConc by ISTD
RT

3. Click start to organise data

By default, organised data will be output into an Excel file in wide table form

Sample_Name	LPC 20:5	LPC 14:0	LPC 18:3	LPC 16:1	LPC 22:6
05_TQC	3612	14722	3297	18932	4538
06_TQC_02	3181	12798	2779	16303	3919
07_TQC_03	2956	12786	2629	16283	3865
08_TQC_04	2979	12672	2562	15874	4006
09_TQC_05	3019	12387	2659	15419	3870
10_TQC_06	2927	12572	2657	15798	3704
11_TQC_07	3017	12456	2483	15386	3992
12_POC_08	2998	12024	2665	15696	3704

A report pdf file will also be provided to record the parameters used

Parameter Report

Parameters	Value
Input_File	WideTableForm.csv
Output_Format	Excel
Annex_Report	WideTableForm_Annotation.xlsx
Area	Area
Output_Options	normArea by ISTD
Output_Options	normConc by ISTD
Output_Options	RT

Advice 5 Tips

Explanation: User Documentation

5.7 Saving your settings with a Json file

When you open MSOrganiser for a second time, you will realise that MSOrganiser is able to remember your previous settings. This is due to the presence of a json file created when the software starts to run



To clear your settings, just delete the json file.

5.8 Log files

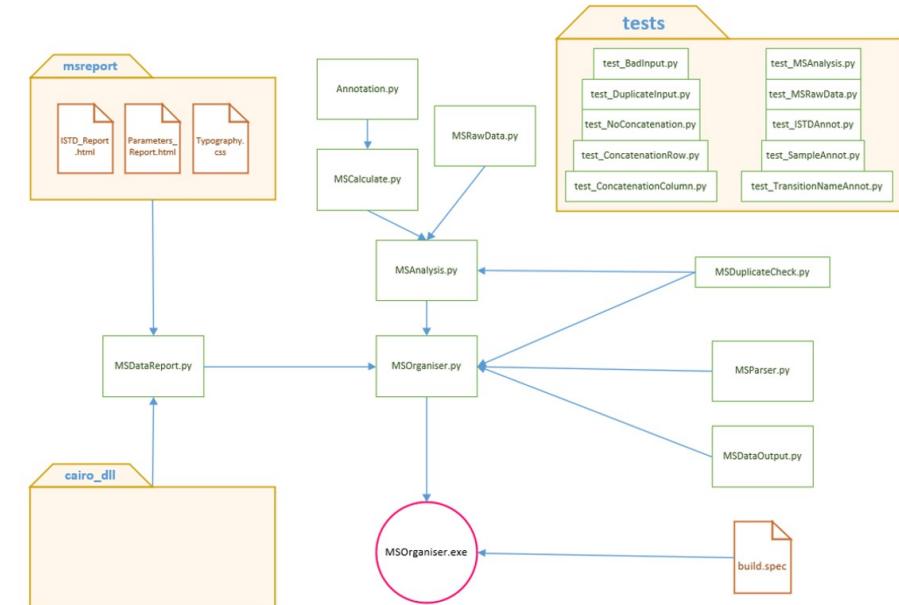
MSOrganiser also keep some log files to keep track of its work status and save any warnings that you have missed.

A screenshot of a Notepad window titled 'Test_Log - Notepad'. The window displays a log of events from 2021-10-18 16:35:32,790. The log includes INFO messages about starting the job and working on a CSV file, and ERROR messages about unable to save an Excel file and permission denied for an XLSX file. It also includes a warning to close the file if it is still open.

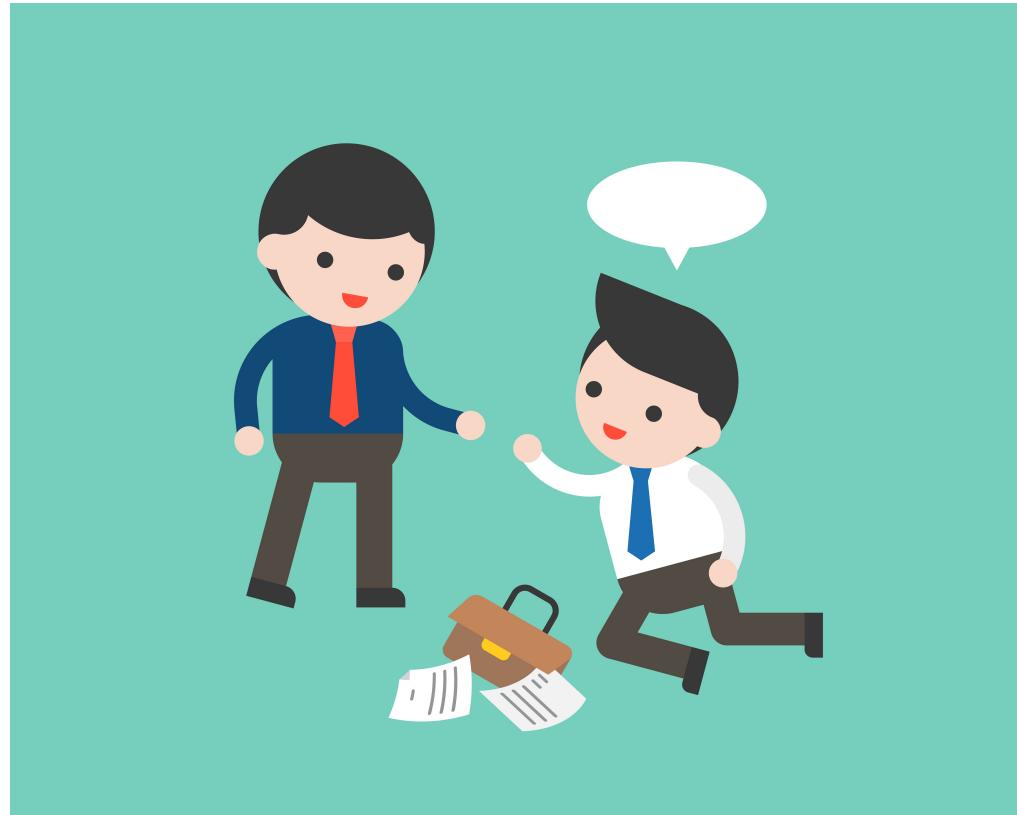
Currently MS Organiser is able to keep track of log files for three days.

Reference: Developer Documentation

11 Code Structure Outline

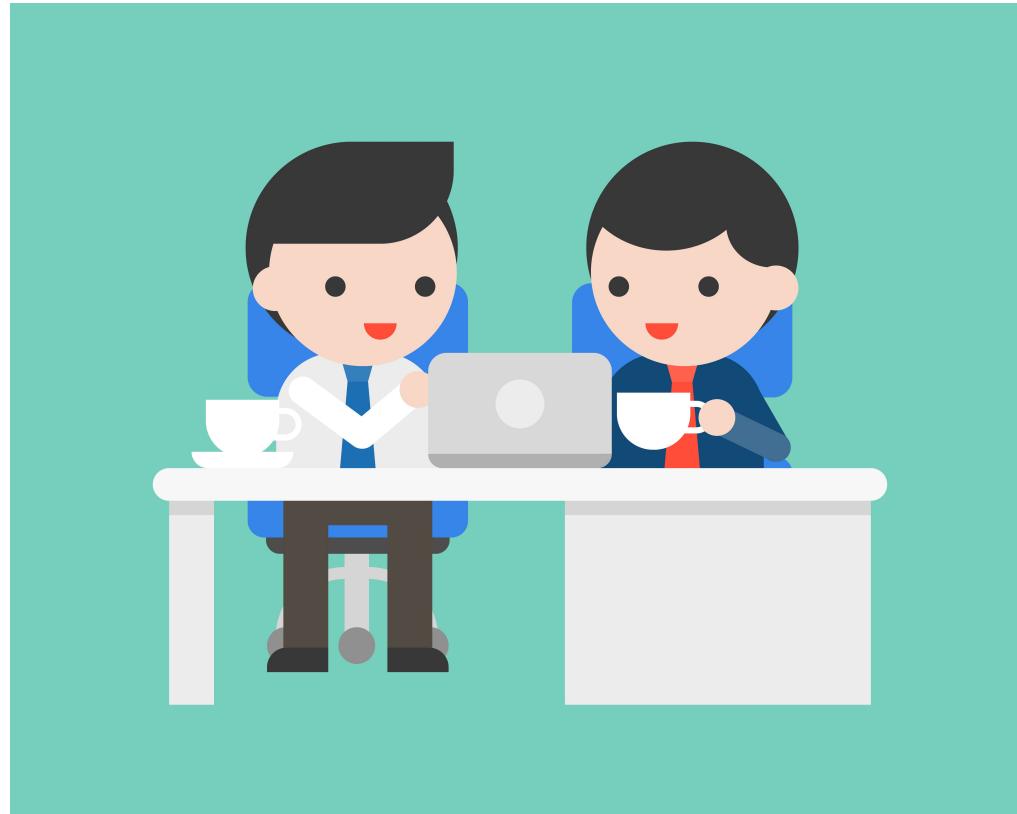


Take Home Advice: Create a software that gives a lasting impact.



- The main purpose for creating a software is not to make us popular, it is to **help others with their problems**.
- A problem no matter how small can be as annoying as big ones
- The more annoying the problem the software tries to solve, the more useful it is.

Take Home Advice: Create a software that gives a lasting impact.



- Don't feel discouraged when you are tasked to create a tool that does small and simple things.
- Instead "do (these) small things with great love." -- *Saint Teresa of Calcutta*
- "If you can impact a few people deeply, they will just shout from the rooftops for you. The breadth of the impact will be a matter of time". -- *Yihui Xie* [blog](#)



Impact: Depth or Breadth?

Yihui Xie / 2018-08-29

All the best...

Review of advice

- Convert scripts to a web service or executable software.
- Give a software overview using cheat sheets.
- Provide helpful messages when users make a mistake.
- Output preprocessing reports to show accountability.
- Organise your documentation into specific structures.

Take home advice

- Create a software that gives a lasting impact.



Images by [Amonrat Rungreangfangsai](#)

Xaringan Slide Template by [Sharla Gelfand](#)

<https://jauntyjjs.github.io/PyDataGlobal2021Talk>