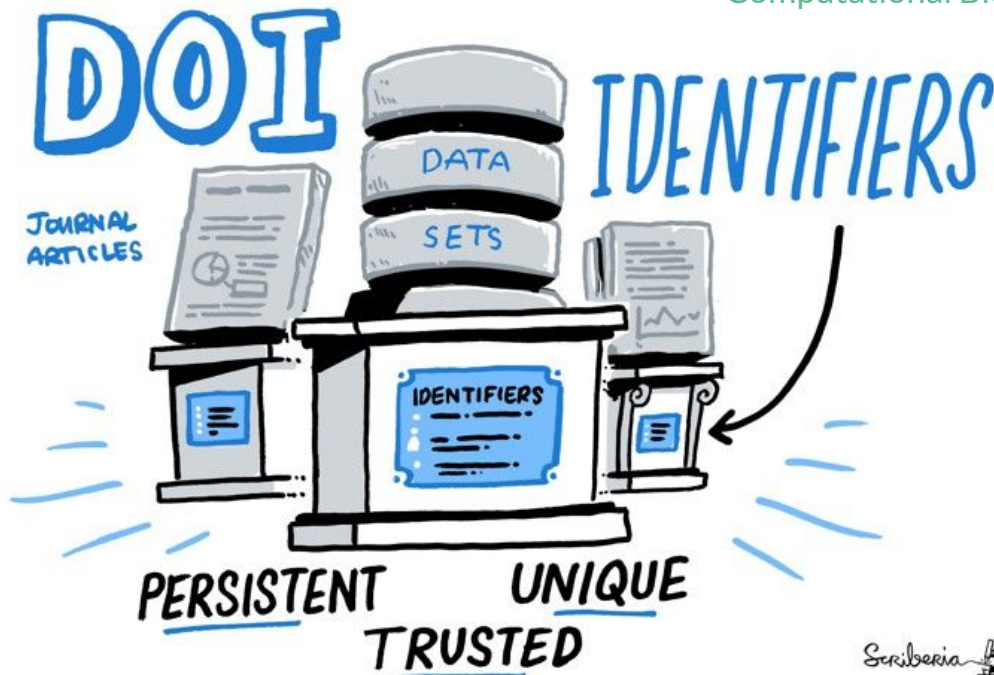


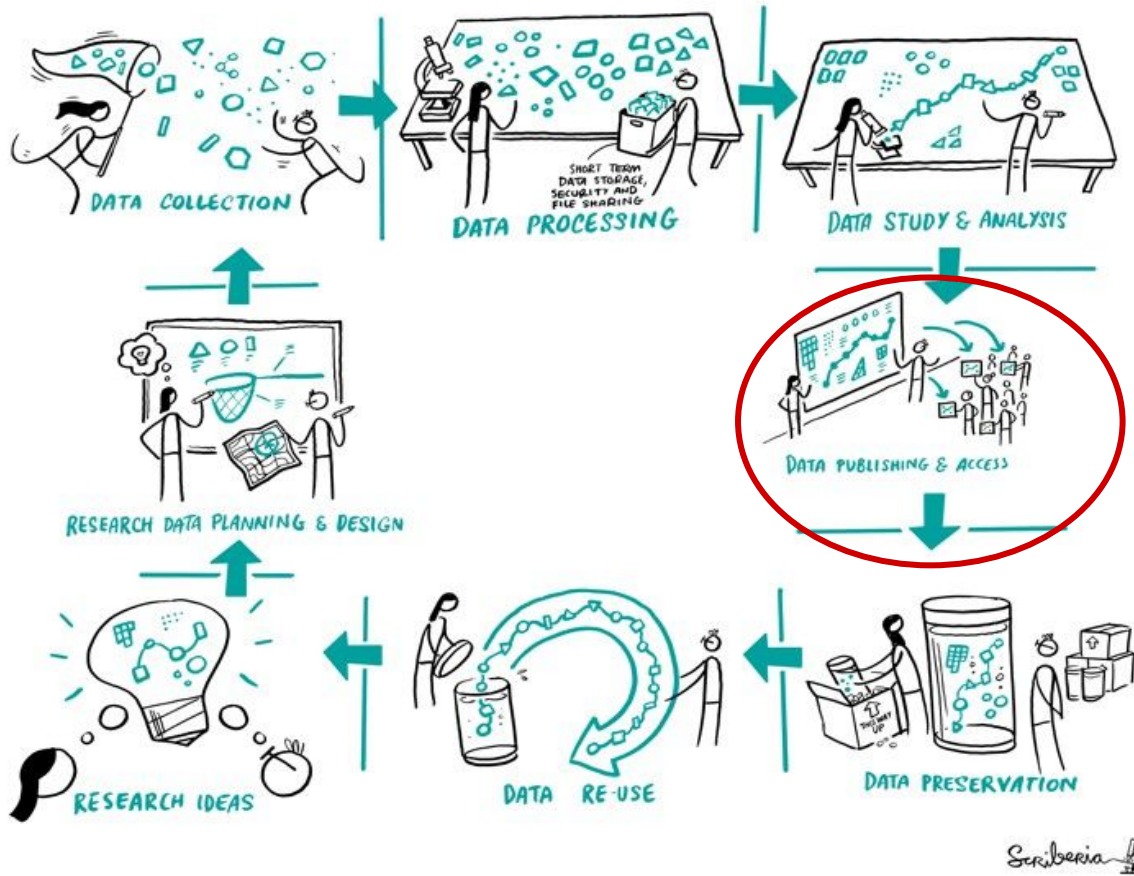
# Make Your Computational Analysis Citable

By Batool Almarzouq (She/her)  
Computational Biologist @batool664  
OSCSA, KAIMRC, UoL

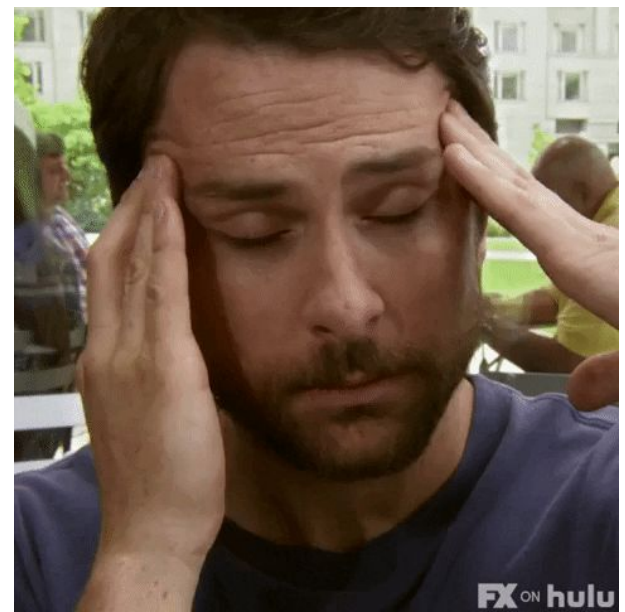


Scriberia

The Turing Way project illustration by Scriberia. Used under a CC-BY 4.0 licence. DOI: [10.5281/zenodo.3332807](https://doi.org/10.5281/zenodo.3332807)



Not every computational analysis is published!



The Turing Way project illustration by Scriberia. Used under a CC-BY 4.0 licence. DOI: 10.5281/zenodo.3332807.



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# Author: Batool Almarzouq
# Date: 2021-06-01
# Description: Metagenomic Analysis to the soil in Saudi Arabia

# Import the necessary libraries
import os
import sys
import subprocess
import time
import argparse
import glob
import re
import shutil
import logging
import random
import string
import math
import itertools
import collections
import multiprocessing
import concurrent.futures
import itertools

# Define the main function
def main():
    # Parse the command line arguments
    parser = argparse.ArgumentParser(
        description="Metagenomic Analysis to the soil in Saudi Arabia")
    parser.add_argument(
        "-i", "--input", type=str, required=True,
        help="Input directory containing the raw sequencing data")
    parser.add_argument(
        "-o", "--output", type=str, required=True,
        help="Output directory for the analysis results")
    parser.add_argument(
        "-t", "--threads", type=int, default=1,
        help="Number of threads to use for parallel processing")
    parser.add_argument(
        "-v", "--verbose", action="store_true",
        help="Enable verbose logging")
    parser.add_argument(
        "--help", action="help",
        help="Show this help message and exit")
    args = parser.parse_args()

    # Set up logging
    logging.basicConfig(
        level=logging.INFO,
        format="%(asctime)s - %(levelname)s - %(message)s",
        filename=os.path.join(args.output, "metagenomic_analysis.log"))

    # Create the output directory if it does not exist
    os.makedirs(args.output, exist_ok=True)

    # Get the list of input files
    input_files = glob.glob(os.path.join(args.input, "*"))

    # Check if there are any input files
    if not input_files:
        logging.error("No input files found in the specified directory")
        sys.exit(1)

    # Initialize the variables for the analysis
    threads = args.threads
    verbose = args.verbose

    # Start the analysis
    logging.info("Starting the metagenomic analysis")
    start_time = time.time()

    # Run the analysis in parallel
    with concurrent.futures.ThreadPoolExecutor(max_workers=threads) as executor:
        # Submit the analysis tasks
        futures = [
            executor.submit(
                run_analysis,
                input_file,
                args.output,
                verbose)
            for input_file in input_files
        ]

        # Wait for all tasks to complete
        executor.wait()

    # Calculate the execution time
    end_time = time.time()
    execution_time = end_time - start_time

    # Print the execution time
    logging.info(f"Execution time: {execution_time} seconds")

    # Print the results
    logging.info("Analysis completed successfully")

# Define the run_analysis function
def run_analysis(input_file, output_dir, verbose):
    # Extract the file name and extension
    file_name = os.path.basename(input_file)
    file_ext = os.path.splitext(file_name)[1]

    # Create the output file name
    output_file = os.path.join(
        output_dir,
        f"{os.path.splitext(file_name)[0]}_analysis_{file_ext}")

    # Run the analysis
    logging.info(f"Running analysis for {input_file}")
    # ... (Detailed analysis steps would go here) ...

# Run the main function
if __name__ == "__main__":
    main()
```



From <https://github.com/BatoolMM/MetagenomicsAnalysis>



Batool Almarzouq. (2021, June). Metagenomic Analysis to the soil in Saudi Arabia.  
Zenodo. <http://doi.org/10.5281/zenodo.4942110>

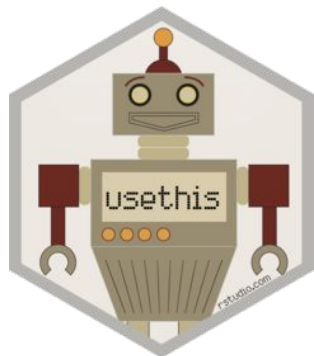


Zenodo is a general-purpose open-access repository developed under the European OpenAIRE program



# STEP 1 >

## Create LICENSE.md



snyk	Copyleft					Permissive			
	GPL	AGPL	LGPL	CC-BY	CC-BY-NC	Apache	MIT	BSD	Unlicense
Permissions in addition to commercial use, distribution, modification:									
Patent use	●	●	●	●	●	●	●	●	●
Patent use	●	●	●	●	●	●	●	●	●
Conditions									
Disclose source	●	●	●	●	●	●	●	●	●
License & copyright notice	●	●	●	●	●	●	●	Source	●
Network use is distribution	●	●	●	●	●	●	●	●	●
Same license	●	●	Library	●	File	●	●	●	●
State changes	●	●	●	Some	●	●	●	●	●
Limitations/Disclaimers									
Liability	●	●	●	●	●	●	●	●	●
Warranty	●	●	●	●	●	●	●	●	●
Trademark use	No explicit limitation				●	●	●	●	●

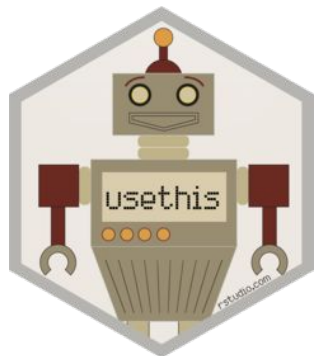
Taken from <https://snyk.io/learn/open-source-licenses/>

More details can be found in my blog:

<https://batool-blabber.netlify.app/posts/2021-06-23-make-your-computational-analysis-citable/>

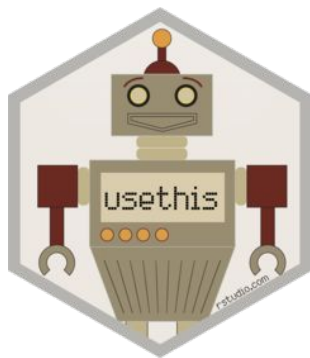
STEP 1 >

Create LICENSE.md



STEP 2 >

Use git (version control)



STEP 3 >

Use Zen4R package and add CITATION.md

eblondel/zen4R

zen4R - R Interface to Zenodo REST API



3 Contributors 3 Issues 24 Stars 8 Forks

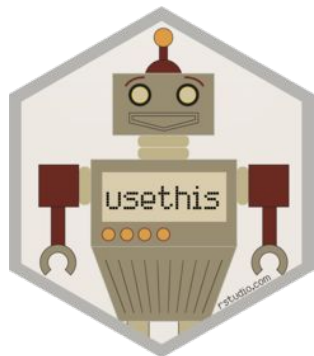
Batool Almarzouq. (2021, June). Metagenomic Analysis to the soil in Saudi Arabia. Zenodo.  
<http://doi.org/10.5281/zenodo.4942110>

More details can be found in my blog:

<https://batool-blabber.netlify.app/posts/2021-06-23-make-your-computational-analysis-citable/>

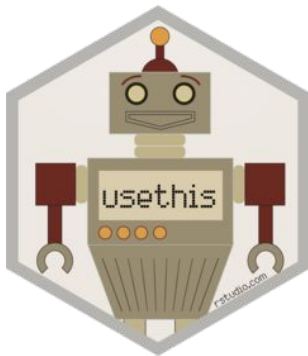
STEP 1 >

Create LICENSE.md



STEP 2 >

Use git (version control)



STEP 3 >

Use Zen4R package and add CITATION.md

A screenshot of the Zenodo website. The page title is "zen4R: R Interface to Zenodo REST API". The page shows the date "September 4, 2020" and the authors "Blondel, Emmanuel; Barde, Julien". The page also shows the number of views (338) and downloads (127), which are circled in red. The page is indexed in OpenAIRE. The Zenodo logo is in the top left corner, and the OpenAIRE logo is in the bottom right corner of the screenshot.

More details can be found in my blog:

<https://batool-blabber.netlify.app/posts/2021-06-23-make-your-computational-analysis-citable/>

## CITABLE CODE:

- It tracks research output and changes overtime!
- It make it useable and impactful.

**More details can be found in my blog:**

<https://batool-blabber.netlify.app/posts/2021-06-23-make-your-computational-analysis-citable/>

