Powerful world of (meta-)genomics held back by lack of Standard Operating Procedures

A computer science-oriented analysis on automated metagenomic approaches and pipelines, their common practices, and technical shortcomings

Author Nima Salami, Computer Science and Engineering, EEMCS Faculty, TU Delft

TNW Supervisor: Prof. David Weissbrodt, TNW, BT/EBT/Weissbrodt Group, TU Delft
TNW Co-Supervisor: Ben Abbas, Head Molecular Biology Technician, TNW, BT/EBT, TU Delft
EEMCS Supervisor: Prof. Thomas Abeel, Bioinformatics Lab, EEMCS, TU Delft
EEMCS Examiner: Gosia Migut, Pattern Recognition Lab, EEMCS, TU Delft

Table of Contents

Abstract

Context: The study and analysis of (meta-)genomics have been providing scientists with valuable insights into the functioning and composition of microbial communities. Latest advancements in next-gen and high throughput sequencing technologies have resulted in significant growth in the data produced and made available for further research. These advancements can help scientists dive deeper into the analysis of uncultivated microbial populations that may have important roles in their environments.

Gap: However, analysis of such data requires multiple preprocessing and computational steps to interpret the microbial and genetic composition of samples. For most researchers, configuring these tools, linking them with advanced binning and annotation tools, and maintaining the provenance of the processing continues to be extremely challenging. Moreover, the most common issue with current practices of metagenomics is the reproducibility of the research due to the complexity of setup and configurations.

Aim: Our aim is to get a big-picture understanding of the common practices and approaches for metagenomic analyses and to find out which ones are more often used by researchers and why. Further, to compare some of the existing tools and look into possibilities of developing and/or using a reproducible pipeline and give some general recommendations for it.

Methods: For this purpose, three main methods were used. First, a literature survey was performed on metagenomic analysis approaches, methodologies, and tools. Next, researchers and scientists with different educational backgrounds active in this field were interviewed. Lastly, the process of pipeline construction and bottlenecks were evaluated through hands-on experience.

Findings: By conducting this research, several common pitfalls and shortcomings of metagenomic analysis practices were identified. Since the expertise of most researchers in this field is lacking a fundamental computer science and programming background, very few would attempt developing a pipeline from scratch. Therefore, if instead, they would opt for using “ready-made” General Purpose Pipelines (GPP), they would also face various difficulties in setting up and configuring them to their needs. Also, it has been observed that many of the existing metagenomic tools are not developed and maintained according to computer science code production standards. Therefore, even the more popular tools can suffer from detrimental bugs that can render them broken and consequently deprecated. However, with the emergence of the new “all-in-one” interface-based online platforms such as *Kbase.us* that enable simple point-and-click set-up and sharing of workflow, there is hope for entering a new era of reproducible metagenomic analysis.

1 Introduction

Metagenomics refers to the study of metagenomes (the collective genome of microorganisms) from a mixed community of organisms, usually microbial communities. [1] For this study, the DNA of all organisms in a particular sample is sequenced and the output is used to investigate the population structure and function of the microbial community existing in that sample. This investigation is conducted using sophisticated bioinformatics and metagenome analysis suite of tools to interpret the data. [2]

In the past decade, advancements in genome sequencing technologies such as High-Throughput Sequencing (HTS) [3], have revolutionized the study of metagenomes. These advancements combined with more competition from different producers of sequencing machines have consequently resulted in a reduction of costs for genome sequencing. Lower costs have given the possibility of more labs around the world having access to genome sequencing for their research questions. This means increasingly more people are seeking tools that allow them to analyze and study genomes. [4]

The constant growth in access to affordable genome sequencing and the rise of its popularity has given rise to the number of tools developed by researchers and labs to assess the output of sequencing machines. Moreover, based on the specific type of analysis, whether statistical or biological, different research groups have produced various tools and methods for analyzing the genomic data. Also important to note, the variety in sequencing technologies has additionally required new tools that adapt to the new formats and standards that come with these new technologies. [5]

The field of metagenomics, however, has not yet reached a maturity stage and the community of researchers active in this field is aware of the limitations and bottlenecks currently present in their field. [6] The lack of standard operating procedures (SOPs) and the numerous disjointed variety in the software tools used for analyses and their technical shortcomings have all as result spawned many causes for lack of reproducibility of the researches in this field. Mostly due to high levels of the discrepancy between the results produced by the original authors and the ones obtained by other research groups. [7]

Currently, labs or research groups worldwide resort to three approaches when it comes to processing and analyzing multi-omic microbial datasets: Develop a tool/pipeline if they have the expertise, install and use tools/pipelines developed by other research groups, or have their data analyzed by third parties. Each of these approaches, however, comes with its respective downsides and complications that each can lead to problems in the reproducibility of their outcomes. In the upcoming sections, we will look further into the implications of each approach and the consequences that come with it.

Many of the current metagenome analysis tools get published by bioinformaticians and developers who do not follow sustainable software development practices and protocols. The tools might perform the task that it is designed for correctly and accurately - at least by its original developers -, however, might lack many features that would enable users to take full advantage of the possibilities of the tool. Features such as help function, useful error messages, set up and installation guidelines, documentation, and more. [4]

Moreover, as with the nature of software technologies that usually depend on libraries and packages developed by external parties, it is most likely that many underlying structures of a metagenomic tool also rely on fundamental parts that are developed and maintained by other developers. This means many of these libraries and packages receive regular updates that might break the current tools that they depend on. Therefore, it is crucial that these metagenomic tools get developed and maintained using standard practices from software development protocols that try to mitigate such similar issues. [4]

In addition to not following software development protocols, another common issue that makes usage of metagenomic tools developed by other research groups a possible point of failure for researchers is the

lab-specificity of the tools. Many of these pipelines/tools are designed by a research group for the specific requirements and needs of their lab, from specific data types to the biological question they seek to find answers to. That means some of these tools are not developed to accommodate general users' needs and data types. [8]

One of the ways that some research groups have tried to address the issues mentioned above, is to develop some easy-to-use general-purpose pipelines (GPP) or web-based software suits that can process multiple datasets/inputs and help other researchers outside their labs to take advantage of their effort. In this way, by open-sourcing and providing the codebase and documentation for their developed pipeline, not only the chance of reproducing researches done by that specific pipeline increases but also more researchers can save time by not needing to "reinvent the wheel".

However, in reality, there are still many mismatches between the capabilities of the GPP and the exact expectation of research from these GPP based on their specific need for several intermediate steps that exist in these pipelines. The specific problems and mismatches will be explained further in the formal problem description section.

The aim of this paper is to investigate the current bottlenecks and technical shortcomings in the field of metagenomic analysis that limits the reproducibility of the research outputs, from a Computer Science point of view. For this purpose, three methodology approaches were selected with the goal to get an overview of the issue from various perspectives, from users to developers and the tools themselves. Lastly, Some possible solutions are delineated that could potentially improve the state of things for more reproducible research output.

Since this study is conducted by a student of Bachelor Computer Science and Engineering, and not from related fields to (meta-)genomics, the scope of this research is limited to a technical assessment of the tools and approaches, and not from a life science or bioinformatics point of view. However, it is important to mention that, nevertheless, the hope is to also take a more broad perspective into account. That is why this research is conducted with supervision and assistance from professors and supervisors from both the Bioinformatics Lab from the EEMCS Faculty and the Environmental Biotechnology Lab from the TNW faculty of the Delft University of Technology.

Research Question and Objectives:

Based on the identified gaps and clarifying the scope of the research, we formulated the following research question for this study:

What are the current technical shortcomings that limit the reproducibility of metagenomic analysis from a Computer Science perspective?

The following 3 working objectives were targeted to answer this research question:

- **Objective 1: Identify the bottlenecks of metagenomic tools at the <u>usage</u> stage.**
    - Computer Science perspective on technical challenges the users of metagenomic analysis tools face that eventually lead to lack of reproducibility of their research.

- **Objective 2: Identify the bottlenecks of metagenomic tools at the <u>development</u> stage.**
    - Computer Science perspective on technical shortcomings of metagenomic analysis tools that lead to their demise and eventually cause reproducibility issues for its users.

- **Objective 3: Derive potential Computer Science solutions for identified bottlenecks.**
    - Investigate standard protocols or guidelines from the field of Computer Science software development that can potentially address the bottlenecks.

2 Formal Problem Description

A (student) researcher who needs to analyze genomic datasets usually has three options, whether to develop a metagenomic pipeline on their own, or having their dataset being analyzed by a third party, or implementing and using a pre-built metagenomic pipeline by someone else. However, each of these options has its own advantages and disadvantages which will be addressed in the following paragraphs.

2.1 Developing a pipeline from scratch

Since most (student) researchers in the relevant fields of biology, microbiology, nanobiology, etc. do not usually have a computer science or programming background, developing a pipeline from the scratch is not a feasible option for most people. The process of learning how to code and build and develop such a pipeline can take months which is not enough time for many researchers. Moreover, whether the end result would be a suitable and well-designed pipeline can also vastly depend on many factors such as the experience in writing correctly engineered data processing systems.

More importantly, since the final result of the research should be reproducible, it requires the developer of the pipeline to make good documentation of the design of the pipeline and how to use it. This, however, is an arduous and time-consuming task that further requires the time availability of the researcher.

2.2 Analyze datasets by third-parties

Therefore, many student researchers who only have a few months for their whole research, resort to having their dataset analyzed by a third party. This third party can be another researcher from the lab in which they are conducting their own research, or another lab from a different university or institute that is working on the same type of datasets and has already built a pipeline. In some cases, they can also have their dataset tested and analyzed at the same organization that sequences their sample data, but this comes at a cost, and not every lab can afford it.

Nevertheless, for all the situations where you would have your dataset tested by a third party, some common issues remain the same. The main problem is not having control over the outcome of the pipeline. Making small changes and tweaking the pipeline until you get the final outcome is not really possible, since every time you require a change, you need to communicate it to the third party and hope they will do it for you the way you desire it and within the timeframe you need it.

2.3 Using pre-built pipelines

In the past few years, the number of General Purpose Pipelines (GPP) has been increasing. The codebase for these pipelines is made open-source and is available on some popular open-source project platforms such as GitHub and GitLab. They are usually offered with proper documentation on how to install and use them based on your needs. This makes it easier to follow step by step to have it set up on your own.

However, even using these pre-built pipelines comes with its own struggles. In order to read the documentation and follow the guide to have your GPP setup, it would still require some basic programming knowledge and learn how to use a bash environment or Linux terminal [ref or footnote]. Therefore, it still requires the researcher to be interested in spending some time learning some basics of Linux and some container platforms such as Docker or Conda.

The largest issue with using pre-built GPP however lies in the way they are designed and knowing which one suits your dataset better. Regarding their design, it is important to know that although these pipelines are developed with the intention to be general-purpose, they are still designed by a research group that normally has more experience with a certain dataset than all other possible datasets. So, it is better to

check whether their experience has had any influence on the sub-algorithms they have chosen for the genomic classification, assembly, and more. Regarding the dataset, it is important to know the technical information regarding your sequenced data such as whether it is short reads or long reads, etc. This information helps you with picking the right GPP for your specific needs.

Among other items to look out for when deciding to use pre-built GPP is whether the codebase is still maintained by the developers. This is important since if the tools and the packages used to build the GPP are outdated, it can cause multiple parts of your pipeline to not function properly. Moreover, if the codebase is still being maintained, that shows that this GPP is still being used by the original developers, so it has a higher likelihood that it is still relevant.

3 Methodology

In order to get started with researching the existing approaches for analyzing the metagenomic datasets, it is crucial to grasp a good understanding of the underlying science behind genomics. Therefore, first, a literature survey was conducted accompanied by reading articles that explain the basics of Microbial biology, DNA, genomics, and more. Later, a series of interviews were conducted to understand the (technical) issues and concerns in the field of metagenomics. Lastly, attempts were made to construct a metagenomic pipeline with two different approaches.

3.1 Literature Survey

As recommended by the supervisors, the research project was started by studying multiple papers from different sources with different goals. In total, more than twenty-five papers and published scientific articles were surveyed, where fifteen of them were read, and summaries were made.

The goal of conducting this literature survey was to first get familiarized with the field of genomics and the currently existing approaches for the analysis of metagenomics datasets. This was important since prior to starting with this research project, I did not have any official education or background in this field.

3.1.1 Metagenomic pipeline structures

The literature survey started by reading, analyzing, and understanding papers on common metagenomic approaches and bioinformatics tools required for creating a metagenomic pipeline for the analysis of microbial and bacterial datasets with a focus on activated sludge (wastewater).

As mentioned by Roumpeka et al. thanks to metagenomics, more researchers are finding novel genes encoded in metagenomes. However, conducting metagenomic analyses requires sophisticated bioinformatics tools for multiple steps such as assembly, binning, and annotation. [2]

As seen in Figure 1, a typical bioinformatics pipeline usually consists of the following steps in a big-picture overview: First, the extracted genomic material is sequenced, then the fragmented sequenced data are assembled in longer contigs. Next is to identify the potential genes by binning them into different categories. Lastly, is the gene annotation step in which the goal is to identify the domains, functions, and metabolic pathways in which the gene products are involved. In case, these steps have led to a meaningful output, then the findings of the research can be shared with the scientific community. [2]

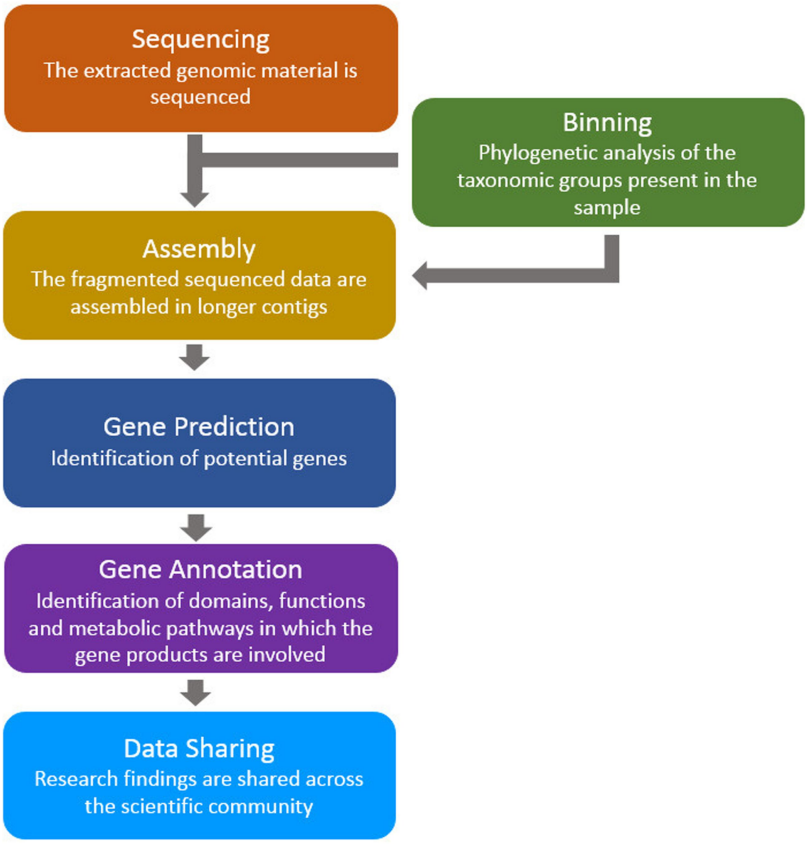


Figure 1. A common metagenomic pipeline. [2]

However, in reality, each of these steps is further broken down into smaller steps and for each of them, a separate tool is used for precise output based on the specific needs of the researchers. Since the aim of

this paper is not to analyze each substep and specific tool from a bioinformatician point of view, it is not necessary to look at all these tools in a detailed way. But one quick glance at Figure 2 can show an overview of some of these tools and their complexity for one of the common genomic approaches called Whole-Genome Shotgun (WGS). WGS is a DNA sequencing method that provides the possibility to characterize whole genomes and their genetic features. [5]

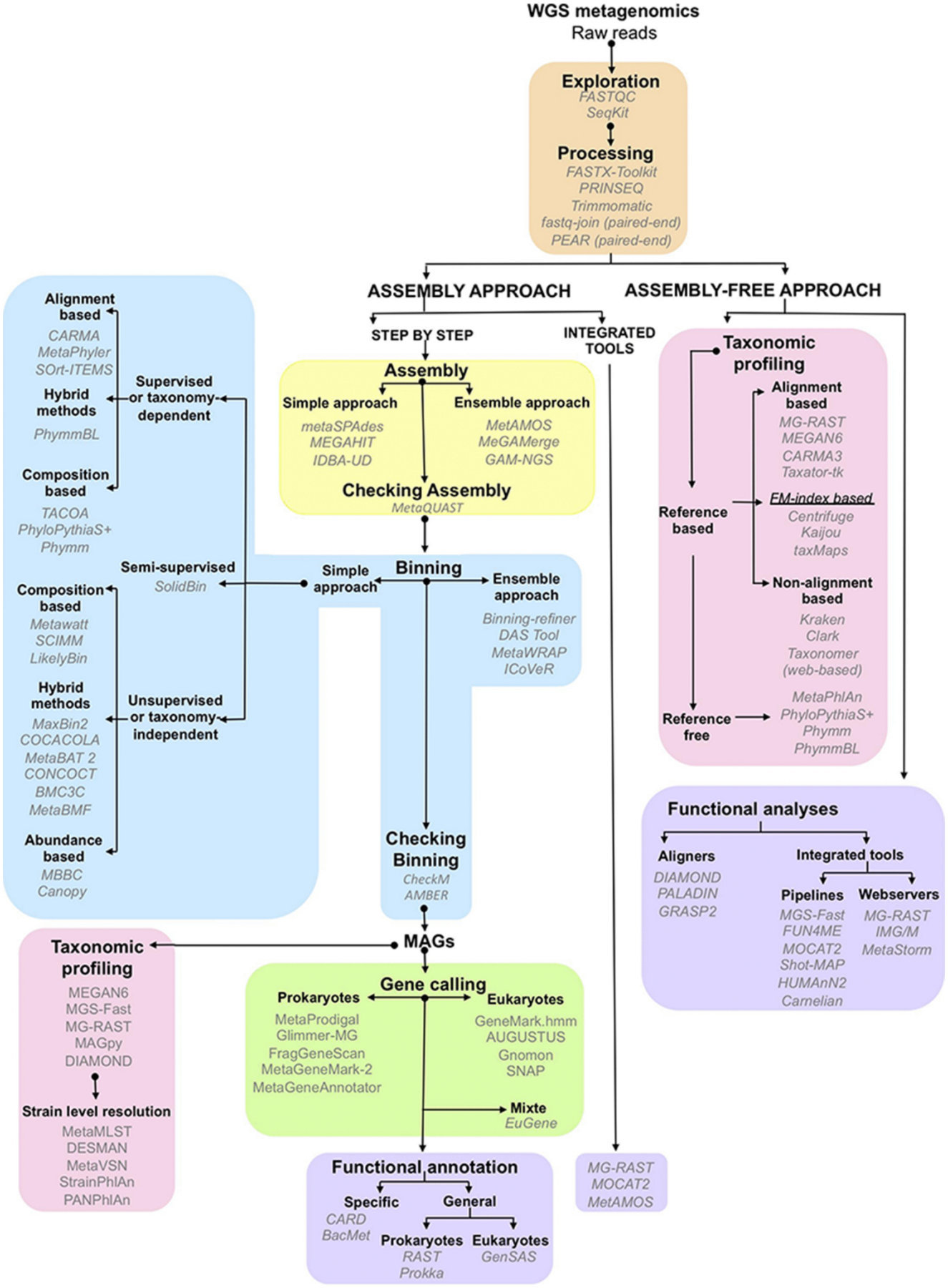


Figure 2. A flowchart of the main steps of a WGS metagenomics analysis. Software names are in italic. [5]

As it can also be seen from Figure [ref], having an overview of all the existing tools even for one single approach is already a challenging task for most. Moreover, new tools are also being developed and published every year which adapt to new sequencing technologies. This makes it nearly impossible to not get lost and always have a full overview of the best tools and methodologies that exist at every given moment. So, instead of choosing one tool for one given task, it is better to make use of several tools and compare the results. [5]

3.1.2 General Purpose Pipelines (GPP)

Among the papers that were researched and surveyed, there was also a focus on finding tools that are developed to be used as General Purpose Pipeline (GPP). In contrast to many of the pipelines are designed based on the specific needs of the lab, GPP is designed with the goal to be quickly and easily reproduced by other researchers all around the world. It is a relatively new concept and new GPP are being released every year. Considering the scope of the research and the time constraint, only four tools were selected for further research and comparison. The criteria for selecting these tools are based on how commonly they are used by other labs, the number of citations, and how often they have received updates. The five selected tools are ATLAS, IMP, MetaComp, MetaWRAP, and MetaPhlAn.

ATLAS

ATLAS [9] is a software package designed and developed in the Faculty of Medicine at Centre Medical Universitaire in Geneva, Switzerland (Swiss Institute of Bioinformatics). The official article for ATLAS was first published in June 2020 and so far has had 8 citations.

This is an open-sourced tool that can be found on GitHub[10] and is freely available, distributed under a BSD-3 license. It provides a customizable data processing environment for metagenomic data. As its input, it receives raw sequenced genomic reads and uses state-of-the-art sub tools to assemble, annotate, quantify, and bin metagenome data.

ATLAS is written in Python and operates in a Linux environment. Its only dependency is Conda [10,11] that installs all of the packages and databases required to run the whole pipeline from Quality Control to Annotation. Moreover, ATLAS uses Snakemake to allow for parallelization of the steps of the workflow when run on clusters. This tool is compatible with all versions of Python 3.5+ and Conda 3+. The pipeline of ATLAS can be seen in Figure 3.

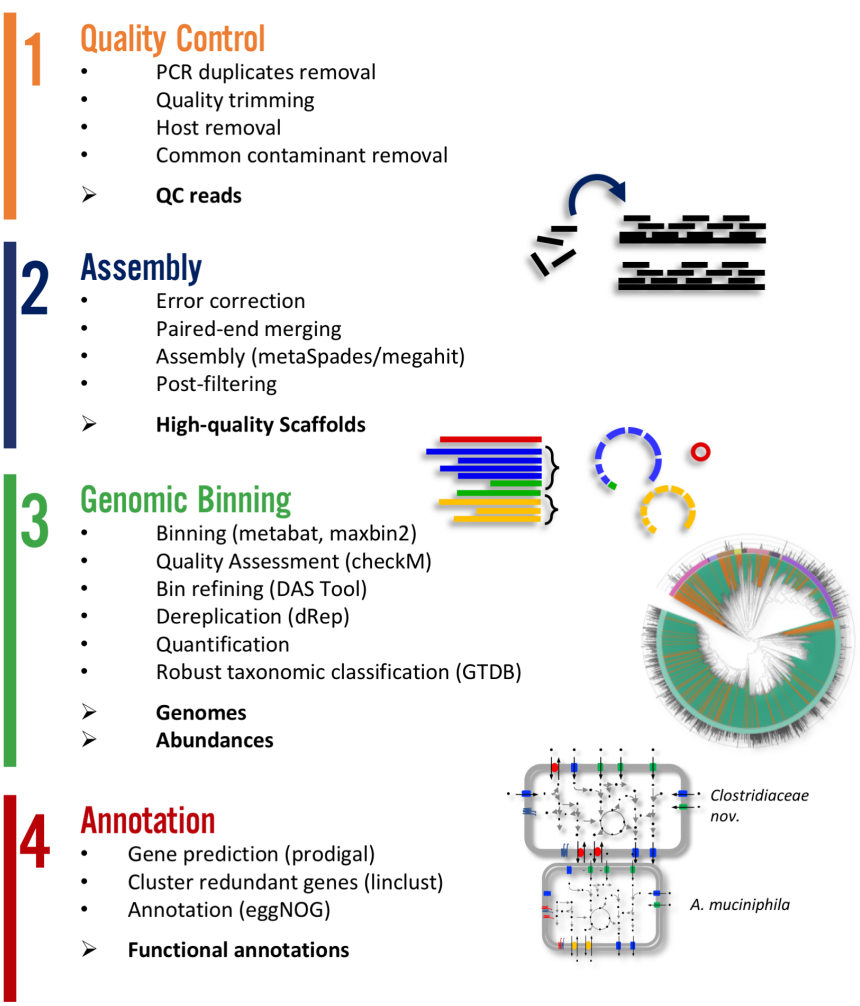


Figure 3. Overview of the steps in the ATLAS pipeline. [9]

IMP

Integrated Meta-omic Pipeline (IMP) [8] is another software package for metagenomic analyses. This pipeline is designed by the Paul Wilmes group at the Centre for Systems Biomedicine in Luxembourg. The scientific paper was published at Genome Biology of Biomedcentral in 2016 and has had 55 citations until the date of writing of this paper. Its codebase is available for free under MIT license on Gitlab which is hosted on the servers of the Luxembourg University.

IMP is developed with the goal to assist researchers to have a modular, reproducible, and reference-independent pipeline for the analyses of both metagenomics and metatranscriptomics microbiome datasets. Among its functionalities include read preprocessing, iterative co-assembly, automated binning, analyses of the structure and function of the microbial community, and visualization of signature-based genomics. An overview of steps in the IMP pipeline is depicted in Figure 4.

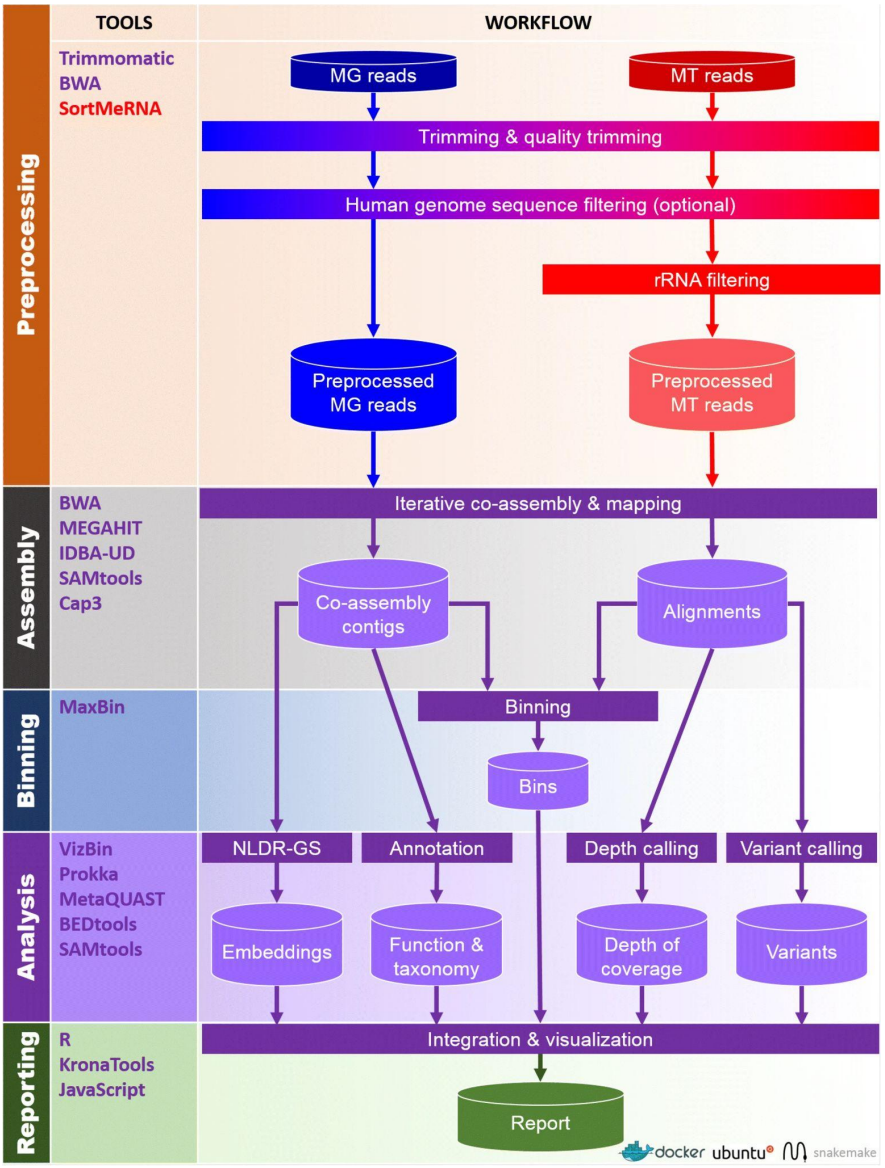


Figure 4. Overview of the steps in the IMP pipeline. [8]

The first iteration of this pipeline (IMP1) is implemented by Python and Docker [12]. Docker is similar to Conda that is used by ATLAS, however, in contrast to Conda, docker is not suitable for being installed and employed on clusters and servers by normal users, as it requires root access. More on this will be explained in a further section regarding the implementation of pipelines.

To address the shortcomings of docker in the first version, the developers of IMP added Conda to the second version and later upgraded it with more packages in the third version. IMP3 is the version that is still updated and maintained. In total, it uses more than 40 different sub tools and packages to analyze the dataset. Similar to ATLAS it also uses Snakemake workflow for parallelization of steps.

MetaComp

MetaComp [13] is by the Huaiqiu Zhu group in the Center for Quantitative Biology at the Peking University of Beijing in China. It was developed and published in 2017 and so far has had 8 citations. The codebase is open-sourced on Github, however, it has not been maintained or updated ever since. Therefore, it is not advisable to use this pipeline for future projects, as the chance of reproducibility is lowered.

This graphical analysis software package can be installed on both Windows and Linux, and it is designed for comparative analyses of meta-omics including metagenomics, metatranscriptomics, metaproteomics,

and metabolomics data. It does that through a series of statistical analysis approaches such as multivariate statistics, hypothesis testing of two-sample, and more, and as an interesting bonus, it can provide visualized results. MetaComp is capable of receiving multiple types of data types as input and automatically choosing a two-group sample test that is suitable for the specific traits of the input abundance profile. The workflow of MetaComp is provided in Figure 5.

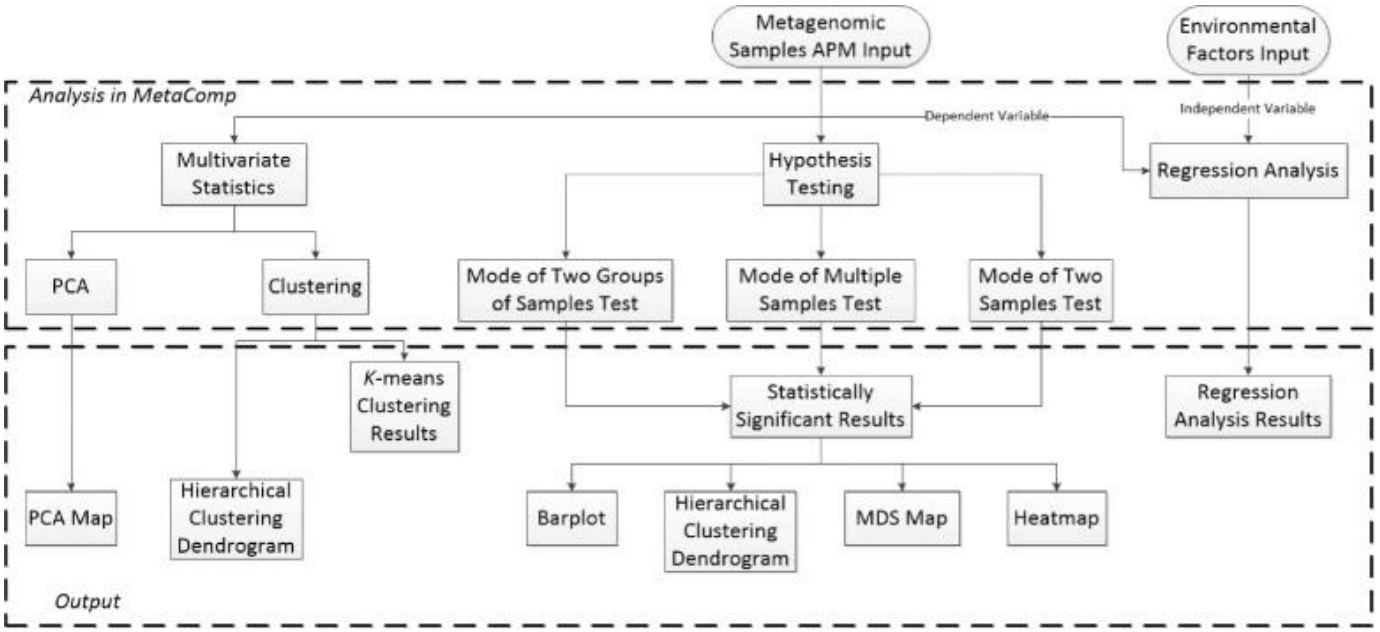


Figure 5. Overview of the workflow and steps in the MetaComp pipeline. [13]

MetaWRAP

MetaWRAP [14] developed at the Department of Biology at Johns Hopkins University in 2018 with 148 citations is the most cited tool among the ones researched in this paper. It is a modular pipeline for genome-resolved metagenomic data analysis with the claim that its bin refinement and reassembly components can outperform other binning approaches.

MetaWRAP is a command-line and Unix-based tool that uses a shell script to call on different modules for the metagenomic analysis subtools. It is built by Python and is distributed by Conda and thus can be installed both locally as well as on remote servers such as High-Performance Computing Clusters (HPC) [15] and the codebase is available for free on Github. An overview of the MetaWRAP workflow is provided in Figure 6.

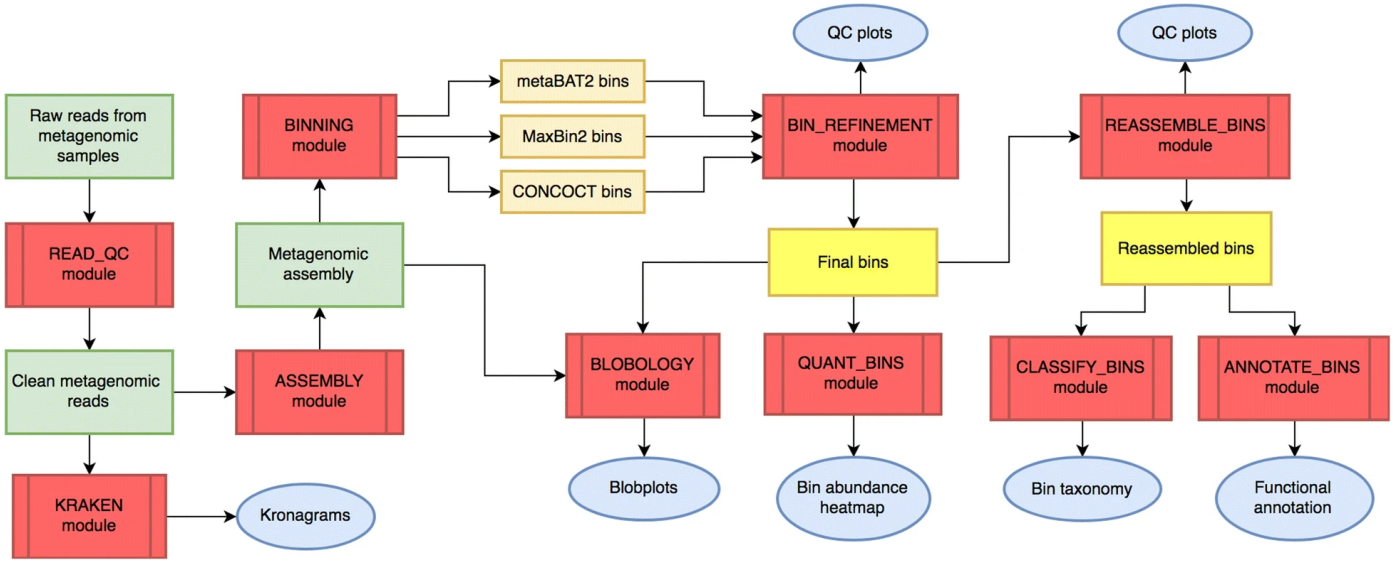


Figure 6. Overview of the workflow and steps in the metaWRAP pipeline. [14]

MetaPhlAN (BioBakery)

Metagenomic Phylogenetic Analysis or MetaPhlAn [16] is a computational metagenomic analysis tool developed and published by multiple renowned research institutes, among others The Broad Institute of MIT and Harvard, from the United States, European Institute of Oncology IRCCS, and University of Trento in Italy. The paper mentioning the early work MetaPhlAn under the name StrainPhlAn [17] for strain-level microbial profiling dates back to 2017. The latest edition of MetaPhlAn (version 3.0) was published in May 2021 with 14 citations on Google Scholar on July first, 2021, and both tools are available open-sourced under MIT License.

Both MetaPhlAn and StrainPhlAn run on BioBakery 3 [18] workflow systems. BioBakery is an open-source multi-omics meta-analysis environment and set of tools for processing raw shotgun sequencing data. It provides pre-configured analysis modules for profiling microbial communities such as quantitative taxonomic profiling or statistical analysis. Biobakery is actively being maintained by a support group [19] and new modules get added with new updates.

MetaPhlAn is also a GPP for profiling the composition of the microbial communities, Bacteria, Achaea, and Eukaryotes from metagenomic sequence data using the shotgun method. The latest stable release version of MetaPhlAn (3.0.10) was released 17 days ago, and now also includes other new features such as ChocoPhlAn for gene marking, virus profiling, calculation of metagenomic size for better estimation, Taxonomic profiling, and more. [20]

Similar to BioBakery, MetaPhlAn and its underlying apps are being maintained and further developed by a group of developers. Contrary to some other tools mentioned previously, such as ATLAS, developers of MetaPhlAn add new features, and bugs are being fixed by making official releases. This reduces the chance of making a change in the code that might break other parts of the software. Also, in case, something breaks in the new release, it is possible to revert it or use a previous stable version that still works correctly.

Summary of GPPs

The five pipeline reviews are just a small sample from numerous open-sourced GPP that are available to use for researchers, and new ones constantly emerging. So, it is important to note that our list and review are not exhaustive and just for the purpose of painting a general picture of the existing situation. Below, Table 1 compares these four tools based on some important criteria that we mentioned in this section. Lastly, the last column summarises our final verdict on whether we recommend this pipeline to the users based on the main criteria of reproducibility:

	First Release	**Language**	**Workflow**	**Container**	**Systems**	**Last Update**	**Stable release**	**Recom-mended**
ATLAS	2020	Python	Snakemake	Conda	Linux	May 2021	N/A	No
IMP	2016	Python	Snakemake	Conda	Linux	May 2021	N/A	No
Meta Comp	2017	C# & R	Custom	N/A	Linux & Win	Nov 2017	N/A	No
Meta WRAP	2018	Python	Custom	Conda	Linux	Nov 2020	v1.3.2 Aug 2020	Yes
Meta Phlan	2021	Python	BioBakery	Conda	Linux	June 2021	v3.0.10 Jun 2021	Yes

Table 1. Comparison of the four GPP reviewed.

3.1.3 Reproducible Scientific Software and Computational Research

This part of the literature survey was conducted at a later stage of the research after encountering several shortcomings of some metagenomic tools code base and software development procedures. After conducting the second and third methodology approaches, namely interviews and pipeline development, several common pitfalls and shortcomings of some metagenomic tools were brought to attention that required further investigation. The aim of this part of the literature survey is to find some general guidelines for the development of reproducible Scientific Software and Computational Research.

As mentioned in section 4.1.3, there are various reasons why many of the tools developed and published open-sourced suffer from a lack of users and often get buggy and eventually deprecated. Since many of these tools are developed by researchers with no official background in Computer Science and have not had experience in (large) software development projects, their tools do not adhere to industry standards of code development and maintenance. Further explanation on some of the issues identified is addressed in section 4.1.3, here below we look at some suggestions on how to alleviate some of these problems or avoid them.

Open Scientific Software Development

Many open-sourced scientific software and tools have had a significant impact in the field of genomics. However, not all of them adhere to software development guidelines and protocols which would keep them relevant after their publications. Some suggestions that can help the developers to make their software more sustainable are as follows: Learn to code according to some industry standards, one way to do so is to take part in and contribute to other successful projects in your field. Open your code to users and critics early and before publishing the first official release, to make important improvements before the whole world hears about it. Make installation and running procedures as simple as possible for your users who most likely do not have enough technical experience. As you create, nurture and grow a community around your tool by promoting to people and institutes that can be of your target users. Find sponsors and fundings that allow you to further develop and maintain and improve your software after its initial release. [21]

Another critically important software development and maintenance protocol and best practice that is often underemphasized is documentation. Having high-quality documentation can maximize the usability of software and make it impactful. Some useful tips according to Benjamin Lee are as follows: Write comments as you code, this is useful both for the developer and the user of the code to understand the logic and thought process behind each piece of code. In your documentation include examples to provide a starting point for the experimentation. To help users easily start playing with your tool and not lose interest over spending a lot of time figuring out how to use it, including a quick start guide. Make sure you have clearly written README file with useful basic information, this file is similar to a homepage to your project. For the command-line interfaces (CLIs) it is very important to include help commands to provide more info on how to use each command. Version-control your documentation with your code evolution. Using automated documentation tools can generate reports based on your comments and speed up the process. Last but certainly not least, make sure to write error messages that provide solutions or point to your documentation for further help to the users. [22]

Reproducible Computational Research

The replicability of cumulative research and the reproducibility of its results are the key components to scientific research, including scientific open-source software papers. However, due to the ever-evolving nature of many technologies, especially software tools, the reproducibility of researches that depends on such tools is considerably more complex. Therefore, it is useful to follow some protocols that can help with simplifying the process and improve the chances of reproducibility. Some tips that Sandve et al. mention in their article are as follows: For every result, keep track of how it was produced, interrelated steps, and

analysis of the workflow. Archive critical details such as name and the exact version of programs used and their parameters and inputs. Avoid manual data manipulation and instead, use standard commands and note them. Version-control all custom scripts to track the evolution of code by a version control system such as Git. Provide easy public access to scripts, runs, and results and submit and publish it together with your paper as supplementary material. [23]

While following the tips and rules mentioned above and adhering to the protocols and following guidelines can help increase the chance of making reproducible computational research and tools, they are not a guarantee that they would mitigate the problem completely. Since many of these tools depend on some other libraries and packages, some of the problems are completely unavoidable, such as the case with the left-pad fiasco in 2016, where a JavaScript node package was completely removed from NPM and thousands of projects that depend on it broke. [24]

3.2 Interviews

After surveying available literature online to find out about common practices in the field to assess and analyze metagenomic datasets, the next step was to survey some people who are doing research and need to do the same thing. This method was conducted through a series of semi-structured interviews. Although no specific interviewing system was used, for consistency and having the same goal all throughout the process, similar questions were asked to each person.

The goal of these interviews was to find out what approaches these researchers are taking to analyze their datasets and why they have chosen for that approach. The reasons varied from the type of datasets they're working with to the difficulty of implementing and using such tools.

In total, six people from the Environmental Biotechnology lab (ETB) in the faculty of TNW in TU Delft were interviewed. Further formal and informal communications were also made with other bioinformaticians and researchers in the relevant field from other groups and universities, in order to get a bigger picture of the current state of things. Among the people interviewed, there were students from Bachelor and Master of Life Science Technology, Nanobiology, Microbiology, Medicine, and more. Also, some PhDs, Post-docs, professors, and lab technicians were also interviewed.

3.2.1 Common issues

One of the most noticeable common points among everyone who was interviewed (except one person), is that no one had developed their own pipeline. That one person had developed her pipeline, had spent six months of her Ph.D. to learn how to do that. The main three reasons were: 1. not having programming expertise, difficulties with High-Performance Clusters (HPC), and not having enough time to learn how to do these. Therefore, all have opted for the option to have their samples' dataset being analyzed by third parties.

Lack of Programming background

Many of the students did not have sufficient courses in general bioinformatics or computer science courses. Hence, they have not yet required the skills to work with programming languages and computer algorithms. Starting to learn the paradigm of programming languages can be especially challenging if you have to combine that with other tasks you need to do for your research. Therefore, many students do not find enough time and space to start with during their research.

Difficulties with HPC

Besides programming languages, another computer-related issue is knowing how to work with shared clusters such as HPC. Since some of the steps in metagenomic analysis require large computer resources,

it is recommended to use big servers and clusters provided by the university or research institute where the scientist is conducting their research. However, working with such clusters requires either some training or at the very least, clear documentation and guides on how to make use of them for non-computer expert users.
For the case of people interviewed, they did not have access to either of these and found themselves struggling and consequently quitting it altogether.

Lack of time

Most of these researchers had metagenomic analysis as part of their project, and not the only work they had to do. Therefore, they did not have enough time to focus on only one task. Since setting up and configuring these pipelines are not straightforward and usually come with several technical challenges, they needed either technical guidance or had to learn everything on their own. But there's no technical guidance provided to them, and they do not have enough time to start learning to program and working with HPC from scratch.

3.2.2 Common wishes

Arising from the common issues stated above, these researchers had hoped for a more ideal scenario they had wished was possible in their cases. These points can be summarised as follows:

Easy to navigate Interface-based web apps with no installation required

In an ideal scenario, if there could exist a simple-to-use software that would easily be installed on their PC, Mac or Linux personal computers or better yet, a web-based app that requires no installation is ready to use is the most common wish. In this way, no time and effort need to be spent on the basic installation of a tool before learning how to use it. Such tools do exist for other purposes for bioinformaticians, such as Cytoscape which is used for visualizing molecular interaction networks. [25] Up until recently, not many up-to-date tools existed for general metagenomic analysis due to technical challenges, and the fact that some analysis steps require large amounts of CPU power or memory that is not available on personal computers. But some new web-app tools are emerging which will be addressed in sections 3.3 and 4.2.

Better technical guidance for installation and usage of metagenomic tools

If an ideal scenario for user-friendly software on a personal computer is not yet feasible, then there is no choice by needing to install them on large servers. This however requires programming and Linux knowledge. Learning these on one's own, while conducting research can be excruciating. Therefore, researchers wish they could outsource this task to someone with more experience in such matters that can take care of this step for them. However, at least in the case of the people interviewed, such a person does not exist in most research groups outside the Computer Science departments.

Support and guidance for using HPC

When it comes down to using the HPC and similar cluster systems, the overall structure can be confusing for those who are not familiar with it. Such server systems are set up and run differently than personal computers where most users are used to. Therefore, researchers often struggle to get started with using them and taking full advantage of their great capabilities. Guides and documentation provided for using HPC are usually written in general and high-level ways that can be hard to read and apply for users looking to do a simple task. Therefore, it has been a common wish for everyone interviewed to be able to have access to a support system for their specific needs or channels to get answers to their questions in a more responsive and quick way.

3.3 Pipeline Development

In order to get a more hands-on experience with the difficulties researchers to encounter on their journey to conduct metagenomic analysis, we decided to test two of the common approaches to metagenomic analysis pipeline development, CLI-based and GUI-based tools. CLI stands for Command-Line interface and GUI stands for Graphical User Interface. Each of these approaches is explained further in their respective sections below.

The reason for selecting these two tools was to take similar approaches to the ones many of the researchers in this field also take, and find out what common issues they encounter. Additionally, the goal was to see if there exist some practices to solve these issues and report about them.

3.3.1 Command-Line Interface (CLI)-based tools

Command-Line Interface or in short, CLI, refers to the process of writing commands and scripts (lines of text) to a computer in order to interact with it. [26] The process usually takes place on a black screen with white texts on it commonly called the Shell terminal and involves only typing with a keyboard, since there are no clickable items on the screen similar to modern Operating Systems such as Windows or macOS. One of the commonly used CLIs used by servers, databases, macOS, Linux, and recently even Windows, is the Unix terminal. CLI-based tools pipeline such as ATLAS which we address in the next section, make use of such terminals in order to install and run them.

ATLAS

As mentioned in section 3.1.2 GPP tools, ATLAS is one of the recent CLI tools developed and maintained at the Swiss Institute of Bioinformatics in Geneva. ATLAS is also installed at Utrecht University clusters and is used by researchers there. One of the students who was interviewed for this research also has her dataset run through ATLAS at Utrecht University. She is a student at TU Delft but since this tool did not exist at her university, she had reached out to researchers at Utrecht University to help her.

We selected ATLAS as the CLI pipeline we are going to research and attempt to install and run on TU Delft clusters based on a few criteria. First, it is considered as a complete pipeline in the sense that it handles all steps from QC, Assembly, Binning, to Annotation for microbial/bacterial dataset. So, in theory, it would be an all-in-one tool that can deliver to the most needs of researchers in this field. Secondly, it is configurable to run either all steps at the same time or just one specific (intermediate) step. This can help reduce processing time, in case a researcher only seeks after seeing different results by tweaking only one step at a time. Thirdly, it uses Python and Snakemake which are amongst the programming languages and environments that are considered familiar for bioinformaticians and are relatively easier to learn in case of no extensive background with programming.

Implementation

For the implementation of ATLAS, a Linux environment is required. It can be installed both locally on a Linux distribution of your choice on a personal computer or on a (high-computing) cluster Linux environment. However, since several steps of data processing such as assembly require more processing power and large storage, it is most likely required to run it on HPC.

At first attempt, by following the instructions provided in the documentation, we installed ATLAS on a local PC in a Virtual Ubuntu Linux environment. If all the steps are followed correctly, and the environment is new, the installation procedure shouldn’t encounter any issues and usually takes a couple of hours. However, in case a different Python or Conda version is already installed in your environment, multiple difficulties can be encountered to adjust it to ATLAS required versions of Python and Conda.

After installing Python and Conda and setting up the environment, you can use the commands provided to install the additional libraries and dependencies needed to run ATLAS through the Conda package manager. This process can take a while, as some of these packages require the installation of large databases. So, the speed of downloading and installing the packages also depends on your internet speed and your hard drive writing speed.

ATLAS suggests you also install the Mamba [27] package through Conda, with the command "`conda install mamba`". Mamba is a cross-platform package manager similar to Conda that is reimplemented in C++ for maximum efficiency. Its advantage in comparison to Conda is the speed by which it installs the packages thanks to its parallel downloading and multi-threading capabilities.

We tried installing ATLAS once through Mamba and once without it. From our experience, Mamba proved to perform significantly faster. Therefore, we can recommend running all the consequent commands to run various steps of ATLAS with Mamba instead of Conda. This means when at each step new packages need to be downloaded and installed, Mamba will do that faster and save significant time overall.

After completing all the setup and installation procedures, we can now run ATLAS with either an example dataset provided or with another dataset. However, trying with both datasets, various errors were printed to the terminal that did not clearly explain what is exactly going wrong. Upon consulting with other researchers in the EBT lab or Computer Science experts, no further solutions were found.

HPC

The next step was to try implementing ATLAS on TU Delft HPC. In order to get started with HPC, first, an account was required which had to be requested to the department which has the responsibility of running and maintaining HPC. Bioinformatics lab from the EEMCS faculty provided a student account that had some limitations on the resources that can be used to run metagenomic pipelines. Nevertheless, it was sufficient to get started with the installation of ATLAS and attempt running it with a small test dataset.

To get started with HPC, it is necessary and strongly recommended to first read the provided documentation. Besides learning how to do things correctly and more efficiently, in this way, any unnecessary burden to the servers can also be avoided. Since installing and running software packages on HPC differs from directly implementing into a simple Linux environment. Moreover, to run tasks that require large processing power, it is important to learn how to use HPC sufficiently by considering matters such as how to dedicate the correct amount of resources to a submitted task and for how long to run the algorithm, and more.

As also mentioned unanimously by the people interviewed, this step to get one familiarized with HPC is not a straightforward task, especially if not coming from a computer science background. Although the documentation provided on the HPC login website gives a good general explanation and some small examples, it does not seem to have been prepared in a way that someone with no experience can read, understand and follow step by step to get a task done.

For the context, as attested by everyone interviewed, a simple step-by-step guide is something that most researchers are hoping to receive. Since they are not necessarily interested in the complex inner workings of such a large structure, but rather just prefer to know how to use it to their advantage and focus on conducting their metagenomic research. This argument, however, can be countered with the point that knowing how this system works can help them better optimize their pipeline and get the best out of it.

After learning the basics of HPC, the same procedures were followed to install ATLAS. However, this time it did not go as direct as installing it locally. The reason is a version of Python and Conda is already installed on HPC and they are not compatible with the versions required by ATLAS. So, some workarounds needed to be learned and implemented to create a separate environment to adjust the version of Python and

Conda. Some guides were provided on a different source from the INSY department [ref], however, this was sufficient to solve the issue. So, further research online and support from researchers at the Bioinformatics lab helped resolve the issue.

Lastly, after resolving the Python and Conda environment and installing ATLAS on HPC successfully, we attempted to run the pipeline on the example dataset. This however returned multiple errors. So, we tested to just run the first step of the pipeline which is the Quality Control.

To run the Quality Control step ATLAS requires you to provide the left and the right raw reads of your dataset in a common zip format such as fq.gz format. It first unzips the file and then runs several underlying steps such as PCR duplicate removal, Quality Trimming, Host removal, and common contaminant removal and at the end returns an HTML file that contains the FastQC report. An example screenshot is shown below.

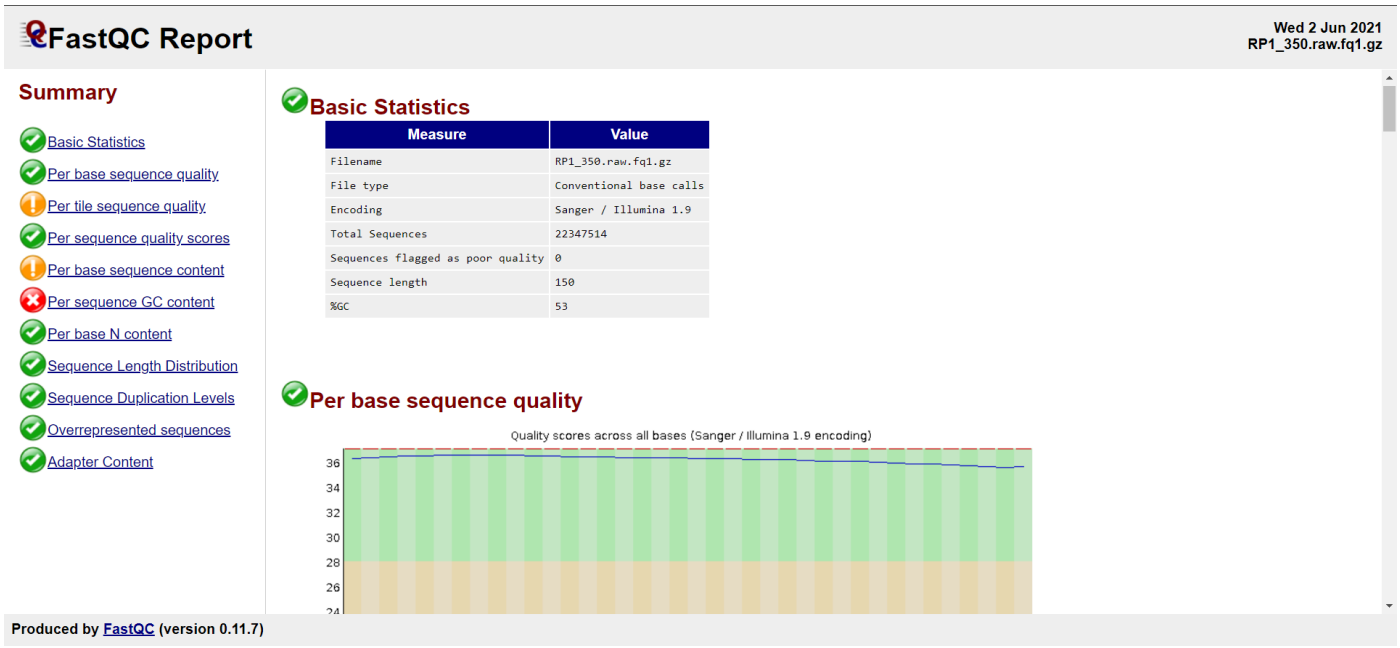


Figure 7. FastQC Report: Screenshot from the output produced after running the first step in the ATLAS pipeline, a clickable HTML file.

Runtime errors

However, upon trying to run the consequent steps after Quality Control, some errors were encountered which we naturally tried to resolve through multiple approaches. We tried reinstalling everything on HPC and again on a local PC by paying more attention to every step we took to make sure we didn't make any mistakes. That however did not solve the problem. So, we contacted people from Utrecht University to see if they had the same problem. They mentioned they have not faced the same issue when installing ATLAS following the same exact steps.

Upon further investigation, it was made clear that the cause for the error is not from the client side with for example wrong environment parameters, but the problem is with the installer itself. It turns out that it is caused due to a broken library among the dependency packages of Snakemake which is the installer package for ATLAS.

When trying to reach out to the developers of ATLAS on their Github page, and creating an issue/bug report, we found out that the developer team was informed about this bug already in May and he has not yet figured it out how to solve it. [28] Although some attempts were made by the developers to mitigate the issue temporarily, still the problem remained persistent. It seems to be a problem with the current version of

Snakemake and the developer is not able to do anything about it at the moment but to wait until a fixed update is released by the developers of Snakemake.

Therefore, unfortunately, our experiment with installation and performance testing of ATLAS had to be halted at that stage. Although it was frustrating to face such issues, nevertheless, it taught us valuable lessons and insights into such similar common problems metagenomic researchers would also encounter. In our case, we were able to follow up and find out the problem is not from the user side but rather the developer's side, however, we can imagine not every user would be able to find this out very quickly and might stay stuck at this stage for a long period.

3.3.2 Graphical User Interface (GUI)-based tools

Besides the numerous CLI pipelines available, a new trend for metagenomic pipelines are tools that also have a Graphical User Interface (GUI). GUIs are a form of abstraction from the underlying computer (software and hardware) programming configurations and provide a user interface that allows users to interact with computers with no programming knowledge required. [29] Thanks to GUI pipelines, more biologists and researchers would have access to metagenomic tools since it will not require them to learn programming and shell-based environments. Two of the tools we review in this section are Kbase, a free open-source online platform, and Qiagen CLC, a commercial OS-dependable software product.

Kbase

Kbase is a knowledge creation and discovery environment designed for biologists and bioinformaticians. It is an open-source software and data platform, developed by the department of bioengineering and the University of California, the USA for the Department of Energy with (indirect) funding and collaborations with large multinational enterprises such as Google, Microsoft, and Tata Consultancy service. [30]

Kbase is an answer to the common wish of many biologists and genomic researchers, an accessible web-based interface that provides a platform for analysis of microbes, plants, and their communities and allows for sharing of the data and workflows with other users. It maintains a reference database that aggregates data from multiple external sources and makes it publicly available for the users for their analysis.

The main competitive advantages of Kbase with similar tools such as Galaxy [31], BaseSpace [32], and more, according to its authors, can be summarized in some key points: support for data provenance and reproducibility, sharing data and workflows, integrated database of genomes, point and click interface for analysis tasks and storing the results, the possibility of using custom code as well, and a software development kit which allows external developers to add applications to KBase. [30]

To use Kbase, users can sign up using a personal account or with a single sign-on of their organization. In our case, we used ORCID to log in through the TU Delft institution. This would allow for easier sharing of data and workflow with other users of our organization, in this case, the Environmental Biotechnology Lab.

To build metagenomic pipelines, you need to work with "Narratives". Narratives are where you can upload your datasets, and apply different applications to them. As soon as entering the Narratives pages, users get prompted to follow a tutorial or read the documentation. We also took advantage of the information provided by Kbase for developing our pipeline, and in case of having questions, the FAQ page also provides useful information.

The dataset used for testing with Kbase was extracted from a sequenced bacterial dataset from the REPARES [33] workshop conducted by David Weissbrodt et. al. in 2020 on antimicrobial resistances in the wastewater environment. [34] The file sizes for each left and right read were around 1.7 Gigabytes, which is a suitable size for Kbase considering its processing speed for various steps.

In order to build a metagenomic pipeline, it is useful to know what is biological question the researchers want to answer, then the pipeline can be developed with an end goal in mind. There are more than 100 different apps available to manipulate and analyze the data, from Quality Check with FastQC to assembly, binning, taxonomy classification, annotation, and more.

Since the goal of using Kbase was to test the general usability of the software and its performance, no specific biological question was considered during the development of the pipeline. However, in order to make sure the pipeline follows the logical steps of a typical metagenomic analysis workflow, it is important to use some guidelines. One guideline was an official tutorial on genome extraction from metagenome sequence data, from the Kbase website [35], and the other was a workflow developed by one of the master researchers at the EBT lab.

The pipeline developed on Kbase successfully runs all the major steps in a typical metagenome analysis. First, the FastQ data is imported, then a read quality is assessed through FastQC. Using metaSPAdes, the metagenomic reads are assembled, and through KAIJU we perform a taxonomic classification. Later, the contigs are categorized into lineages (bins) using depth-of-coverage, nucleotide composition, and marker genes by the MaxBin2 app. Next, the bins are extracted as an assembly from the BinnedContig dataset using BinUtil. Then QUAST was run on a set of assemblies to assess their quality and using GTDB-Tk objective taxonomic assignments for bacterial genomes were obtained based on the Genome Taxonomy Database (GTDB). At the last step, Prokka was used to annotate some of the assemblies.

The report and the results of all the steps of the pipeline are available in the workflow narrative [36] and some screenshots are provided in the appendix section. To run all the steps of the pipeline one after each other would take approximately 7 hours for the REPARES dataset. However, this time can vastly differ based on the size of the output, and the complexity of the community of microorganisms on the sample.

To optimize the pipeline and make it run faster, It is also possible to run some steps simultaneously, as long as their input or outputs do not depend on each other. Moreover, it is also possible to import multiple datasets in the same narrative and have them go through the same pipeline and workflow. In this way, you can compare the result of applying the same pipeline on different extracted samples or sequencing methods. This of course can work the other way around: have the same dataset go through different steps, workflows, or narratives.

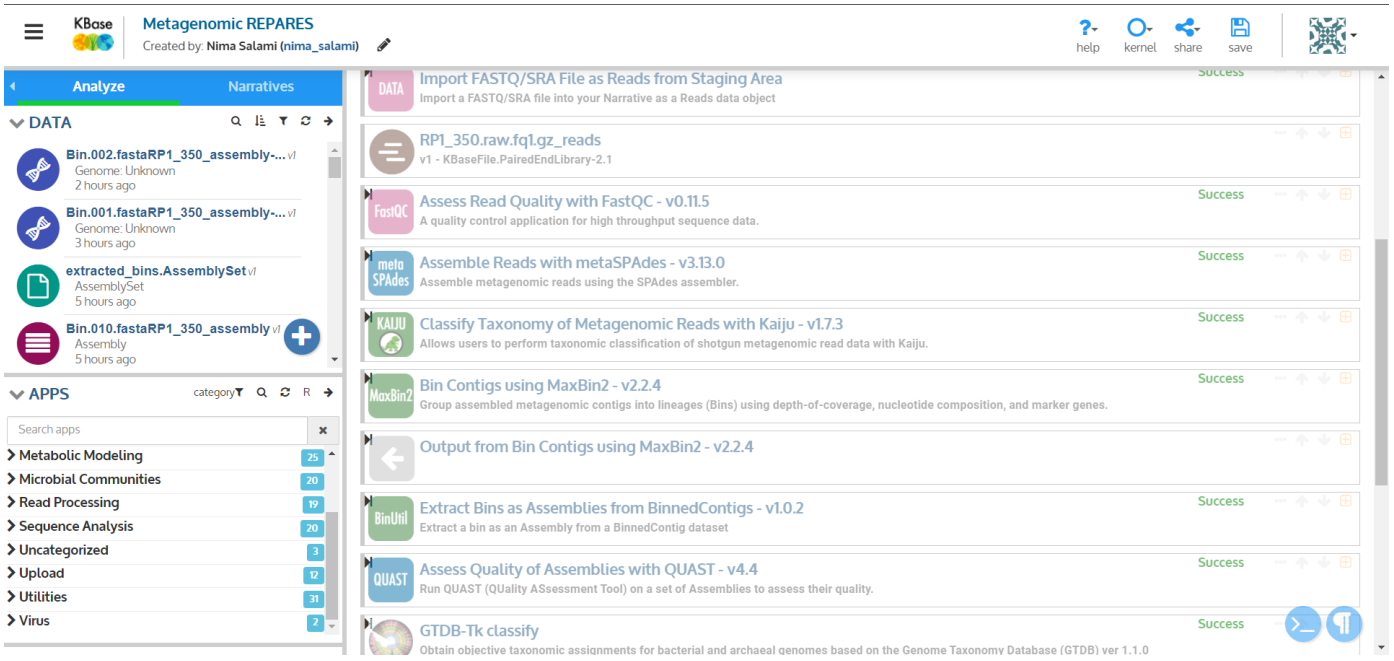


Figure 8. Overview of the pipeline/workflow developed on Kbase web-app interface.

QIAGEN CLC

QIAGEN CLC Main Workbench [37] is a GUI-based metagenomic analysis tool for researchers of the DNA, RNA, and protein sequencing world. It is one of the many software products made by the German biotechnology company QIAGEN, and first released in 2006. [38] This software is maintained and receives frequent updates and improvements. The last update was made in May 2021, in version 21.0.4 to fix several bugs. [39]

CLC Main Workbench is offered on three major operating systems, namely, Windows, macOS, and Linux, and offers many functionalities. Some of the main features include Sanger sequencing analysis, molecular cloning, phylogenetic analyses, and sequence data management, RNA structure prediction and editing, gene expression analysis, integrated 3D molecule view, sharing of data among researchers, and many more.

QIAGEN CLC is a commercial product with an annual license for individual users, however, it does not mention the pricing on the website. In order to see the pricing, you need to create a user account and log in. However, in our case, even after creating an account and logging in, the website did not show any pricing and is not able to add the product to the cart. Therefore, it is not possible to know how much it costs. One review website mentions that the price can vary per user and use case, and each user needs to request a quote. [40] This lack of transparency in pricing is not a welcoming sign to potential new users.

Another approach for testing this tool is to use the 14-day trial version. However, after downloading and installing the software package on our Windows 10 machine, we faced the error that mentions our trial period has expired on this machine. We have never installed CLC Main Workbench or any other software product from QIAGEN on our Windows machine. This shows that, besides the (front-end) of their website, their other servers that handle such trial requests also have technical issues.

Although QIAGEN CLC has been around for around 15 years now and is relatively mature, due to its alleged high pricing, it is not a very affordable and accessible tool for many researchers, according to some of our interviewees.

4 Results and Discussion

The aim of this section is to review and categorize the findings from the three selected methodology approaches. The main identified problems in the field of metagenomic analysis are grouped in three categories: Abundance of tools and lack of overview, lack of expertise in tool setup, and missing in tool development. We believe these problems - whether from the user or developer side of metagenomic tools or other reasons - are among the main causes that lead to the issue of lack of reproducibility of research efforts in this field. In the next subsection, we look at possible solutions or practices that can help with alleviating or altogether avoiding some of these issues and look towards a future where many of these problems would not need to occur for most entry-level researchers.

All statements that are not cited in the following sections are based on the issues brought up by the researchers we interviewed. Common points made by the interviewees are collected and paraphrased into the statements mentioned in the 4.1 section and its subsections. The summary of the interviews is provided in the appendix section for further reading for those interested.

4.1 **Problems** outlined from *development* and *usage* of tools

In order to better understand the problem of reproducibility of researchers that depend on metagenomic analysis tools, it is not sufficient to only look at the situation from the users' (researchers') point of view. Therefore, by looking at the tools themselves and the developers who published these tools and the way they may or may not maintain them, we can find more root causes. Here below, we categorize our findings in three main points:

4.1.1 Abundance of metagenomic tools and lack of overview for researchers

In the past decade, advancements in genome sequencing technologies such as HTS have revolutionized the study of metagenomes. These advancements combined with more competition from different producers of sequencing machines have consequently resulted in a reduction of costs for genome sequencing. Lower costs have given the possibility of more labs around the world having access to genome sequencing for their research questions. This means increasingly more people are seeking tools that allow them to analyze and study genomes. [4]

The constant growth in access to affordable genome sequencing and the rise of its popularity has given rise to the number of tools developed by researchers and labs to assess the output of sequencing machines. Moreover, based on the specific type of analysis, whether statistical or biological, different research groups have produced various tools and methods for analyzing the genomic data. Also important to note, the variety in sequencing technologies has additionally required new tools that adapt to the new formats and standards that come with these new technologies. [5]

Besides the current abundance in the availability of tools for metagenomic analysis, new tools are continuously being developed and published every year. This renders even the most recent large review papers outdated in terms of having a current overview of all available tools for any given context specificity.

For individual researchers, looking for the right tool for the specific needs of their research is a challenging task. Tool review papers are not complete or up to date, and finding the right tool takes a considerable long time, and there is no guarantee the tool they found is actually suitable to their needs. The common advice is to analyze the datasets with multiple tools, test and compare the results before deciding which one to eventually use. All this adds to the lack of confidence in the tool selected and the time spent to find it.

4.1.2 Lack of computer science expertise in setup and usage of metagenomic tools

One common concern raised by everyone who was interviewed for this paper is the lack of expertise in setting up these tools on clusters such as HPC. This process requires programming knowledge, and experience in working with distributed cluster systems and their structure. [4] As mentioned by our interviewees, for this purpose, the researchers need to spend time learning these new skills in the limited time of their research period and this has caused unexpected stress and delay and kept them from having full focus on their research goals. [see interviewee notes in appendix]

For this reason, the majority had opted for having their dataset being analyzed by a third party. This, however, takes the full control away from them since they would not be able to make small tweaks and see different results and further investigate the (biological) questions they are trying to find an answer to. They are then limited to the pipeline that some other researcher, research group, or company has.

Another approach that some others have taken is to learn to program, working with Linux and HPC, and trying to set up a pipeline based on their needs. However, a similar challenge is a lack of time to learn these at an expert level and getting stuck with many technical issues along the way. Due to a lack of official technical guidance such as a resident bioinformatician or Computer Science expert at many labs, getting stuck at a step and not having someone to assist you can be frustrating and demotivating. Eventually, many would stop developing their pipeline because of all the frustrations and would rather outsource this step.

4.1.3 Lack of computer science expertise in development and maintenance of tools

By getting hands-on experience from constructing a pipeline, many (technical) shortcomings of the existing tools were made more clear. Even amongst the most cited GPP tools, basic Computer Science standards can be missing which would over time render them broken and not useful anymore. No efficient and optimized code, lack of proper packaging of libraries and sub tools, no version control and more, can be commonly found among these tools.

Looking at when a GPP has been last updated is one good indication to see if they are still maintained and relevant for users. The reason is, many of these tools are built using several programming languages, software packages, and libraries that constantly get updated and improved. These new versions of software packages and libraries often migrate to new technologies that make them incompatible with previous versions. If the GPP doesn't update their underlying software foundation with the new versions, they soon will have broken parts and get deprecated. [4,10]

When a GPP tool is released, it is safe to assume that it was released in a stable state, meaning that all the promised features would work. That is why it is important to make a “stable release" of the tool in Github. However, many of the tools that were looked at did not have a stable release. So, even those that were getting regular updates from the developers, could have an update that can break parts of the tool at that time. Therefore, new users will not be able to install the tool until from a previous stable version and should wait until the next update that fixes everything. In some cases, this can take months or never happen. [41]

Another common element that was noticed among many GPP that is published open-source, is the fact that many never get updated in the future. This phenomenon can be explained by looking at the incentives for maintaining these tools. Keeping software up to date is an arduous task and requires a lot of time and effort. If the developers are finished with their study and research and are not working at a company or institute with different research goals, it means, possibly they will not use their own tool anymore and other users of their code will not pay them to maintain it. So there remains little incentive to continue working on it. [2]

These are our observations on some of the most popular tools, and they cannot be generalized to all the existing tools. However, it is also safe to assume that many of the less popular tools would suffer from

similar issues and maybe even more severely, and using them leads to not reproducible research. These issues and similar ones can be avoided with better Computer Science software development and maintenance guidelines and practices that will be discussed in the next recommendation section.

4.2 **Solutions** delineated for *reproducible* and *user-friendly* pipeline

Now that problems are identified, it is a good-will logical step to also look at some possible solutions. Because we believe it is not a good approach to only point out the flaws and shortcomings without considering points for improvement. Therefore, we would like to also address some of these issues and make some recommendations on how to improve the situation in some possible ways. For this purpose and in order to make recommendations and propose possible solutions that are researched and grounded, some extra articles were read as mentioned in section 3.1.3 and some experts from the field of Computer Science and bioinformatics were involved as part of the discussion. Here below are the Solutions and recommendations grouped in four points:

4.2.1 Curated and crowdsourced **Directory/Wiki** of metagenomic analysis tools

Since the number of all existing tools from all around the world is almost uncountable, and many new tools are emerging all the time, no one person or small group can keep count of them, let alone have an overview of all of them. Also, the review papers often have to limit themselves to only a few tools and they get outdated as soon as new tools are released.

Therefore, a crowdsourced and curated directory or wiki of these tools that get regular updates from the users and experts could be a useful solution. In this way, this directory can always receive new additions and its current entries can get up to date with new updates and releases and remain relevant. By adding forums, users can also share their opinions, questions, and answers and create communities around similar interests or struggles with certain tools, and even offer suggestions to the developers.

4.2.2 Integrate programming and application usage training as part of the curriculum

Interviews with researchers showed that almost all of them lacked proper or any training with the usage of metagenomic tools during their studies. Even at TU Delft, as a Technical University with an emphasis on learning technology as part of any study, still, some programs do not have programming knowledge as part of their regular curriculum. This leaves students with many potential hurdles down the road when having to set up and use metagenomic tools for their future research purposes.

Therefore, including some basic Computer Science and programming knowledge can be extremely useful, especially if it is provided with practical assignments and hands-on experience. Another useful addition could be inter-faculty and multidisciplinary courses and projects where students from different backgrounds, such as Computer Science and Life Science can collaborate together and prepare for a possible journey towards becoming bioinformaticians.

4.2.3 Software development training and protocols for developers of bioinformatics tools

Many of the current metagenome analysis tools get published by bioinformaticians and developers who do not follow sustainable software development practices and protocols. The tools might perform the task that it is designed for correctly and accurately - at least by its original developers -, however, might lack many features that would enable users to take full advantage of the possibilities of the tool. Features such as help function, useful error messages, set up and installation guidelines, documentation, and more.

Moreover, as with the nature of software technologies that usually depend on libraries and packages developed by external parties, it is most like that many underlying structures of a metagenomic tool also

rely on fundamental parts that are developed and maintained by other developers. This means many of these libraries and packages receive regular updates that might break the current tools that they depend on.

Therefore, it is crucial that these metagenomic tools get developed and maintained using standard practices from software development protocols that try to mitigate such similar issues. In an ideal scenario, it would be best to teach and train metagenomic tool developers with these guidelines before they publish their tools or get assistance from experts who have experience in the development of professional software applications.

4.2.3 Self-contained mashup interfaces like KBase for entry-level users

Using interface-based software applications that abstract away the underlying complex programming structure, is a common practice in many fields of science. For example, many (entry-level) students or researchers use Excel or Matlab for their calculations, Adobe Illustrator or AutoCAD for design, and more, instead of writing scripts or programming codes to get their job done. Such popular and comprehensive software applications, however, do not yet exist in the field of metagenomic analysis, at least up until recently.

Kbase is a great example of such self-contained mashup [42] software applications that ease up many basic and even advanced processes for users. As explained in section 3.3.2, Kbase is a web application that contains numerous metagenomic tools and databases in location and provides a graphical user interface for developing a pipeline. It also allows for sharing of the output or the whole workflow with other users. Similar all-in-one tools exist but none provides as many features and apps as Kbase. Such comprehensive and interface tools are quite new in this field and still not yet well known by many, but they could provide a great insight into a future where most (entry-level) users do not need to worry about underlying programming detail and can instead focus on the scientific task they want to perform or biological question they want to find an answer to.

SOPs: a potential long term solution

Although the potential solutions mentioned above can each have a positive influence on the current state of the metagenomics field, they do not address the main issue of lack of Standard Operating Procedures. But in order to develop such SOPs, more than just a single suggestion is required. Creating SOPs usually requires a team, such as a board or a committee of experts in that field, as well as experienced people from outside the field that can have a different perspective. [43]

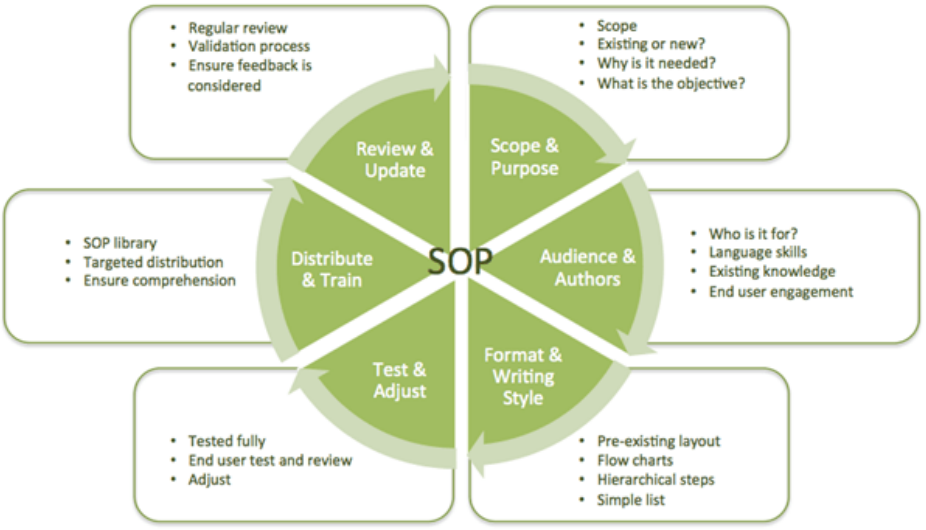


Figure 9. Overview of the pipeline/workflow developed on Kbase web-app interface. [44]

Standard Operating Procedure usually refers to a set of step-by-step instructions with the aim to help a group of people such as an organization to execute some routine operations. The goal is to reach more efficient procedures and find ways to improve the quality of and unify the output and the results. Moreover, it is important to consider ways to improve communication between the parties involved and to the outside organizations. [45]

The process of making SOPs requires attention to detail and therefore is not a quick process. As seen in Figure 9, It requires several fundamental steps and needs to go through multiple iterations. It usually starts with creating a list of processes that need SOP, then making a plan for developing and managing the processes, including determining the format and making a template, and Identifying experts and collecting relevant information, and discussing it within the board. [43]

These are some common steps in creating general SOPs, however, to devise such steps for the metagenomic field would involve its own specific procedures that are out of the scope of this research. Luckily, there have already been some attempts in coming up with such SOPs, such as mentioned in the paper published by Garrity et al. in 2008, from Vrije Universiteit in Amsterdam, [46] and van Gelder et al. in 2017. [47] The Dutch Techcenter for Life Sciences (DTL) and ELIXIR Netherlands [48] and BioSB [49] are three good examples of such efforts in making such nation-wide SOPs in the country of the Netherlands

Further recommendations to developers of metagenomic analysis tools

In order to make reproducible and sustainable scientific software and tools, there are many general recommendations and guidelines. It is difficult to measure how applicable all these recommendations are to metagenomic tools specifically, however, they are considered good common practices for (open source) software development.

These guidelines and good practices are outlined to more extent in the Methodology section 3.1.2 and articles can be found in the references section, but to summarize, code quality checks, unit, and integration tests, build and pipeline tests, thorough documentation, and publishing stable releases of the software, are among important tips for software development.

Improving the inner quality of the software developed is not the only thing that can improve a software experience. For many users, the interface of the software is of even higher importance. If a software product is not designed in a way that its targeted users can easily navigate through the features and functionalities, they get frustrated and will not be able to make use of it in an optimal way.

Therefore, having a “user-friendly” interface for metagenomic tools can make a big difference for its users who do not typically attain coding background or have experience in working in a shell and terminal environment with only texts and a black background. This point was specifically brought up as a common wish by everyone who was interviewed, as well.

To conclude, in order to have better metagenomic tools, it is recommended that the developers would receive training and learn general scientific software development guidelines. This can hopefully avoid many possible ways that their software gets broken and deprecated due to broken libraries or other causes. Moreover, to improve usability experience, look into possibilities of developing an interface or in case of fund availability, collaborate with other software development groups.

Disclaimer on identified problems

It is important to mention that, throughout this research, we strived for unbiasedness in how the problems are found and investigated. The common problems outlined in the previous section are a summary of a small collection of papers, interviews, and tools reviewed. The intention is not to paint every tool, its

developers and users with the same brush and claim they all have the same issues. However, many of the shortcomings that were identified and mentioned in this paper have been brought to light multiple times by various papers, researchers interviewed, and the pipelines that were looked at.

5 Responsible Research

Conducting ethical and responsible research is of utmost importance, that is why this matter has been a key component of this research paper throughout the whole process. We started this research with the goal to find the possible root cause of why lack of reproducibility is a common issue with many types of research conducted using metagenomic analysis tools. Throughout this quest, we strived to remain in an unbiased position and look at the issue at hand from more than one angle and present them with providing evidence that can accompany the claims.

For the three methodology approaches selected, the process and journey have been first documented in a separate logbook document and later carefully translated into the text provided in each respective section. The process of finding relevant literature has been done by making sure multiple points of view are taken into account and reliable sources are used. The literature that has been surveyed has been referenced in the reference section, to provide further reading opportunities to the readers. Tools that are selected are among the more cited and more commonly used tools worldwide, and also more obscure ones are surveyed.

The interview process has been conducted in a semi-formal structure, similar questions are asked to each interview and the answers are noted down and collected in separate documents. The interviewees are selected from a diverse group, from several aspects. Different levels of education, from Bachelor and Master to Ph.D., Post Docs, and professors. Different backgrounds in science, from Life Sciences, Nano Biology, Biology, Computer Science, Bioinformatics, Ecology and more. They have given consent to be interviewed and their answers being used for the purposes of this research anonymously.

The tools selected to get hands-on experience are also selected from two different worlds: Installable on personal computers and High-Performance Computing clusters, as well as web-based interface software suits. In this way, multiple user experiences have been taken into account. The issues encountered at each step have been taken note of and documented presented in the paper and verified by trying to reproduce it on one different machine, from Mac to PC and Linux, and HPC.

Since the goal of this paper was to give a general overview of the tools and their differences, the technical output of the tools has not been reviewed. Therefore, no data has been produced with the intention of measuring and comparison, and no data point has been calculated or manipulated. All the source code and scripts that have been used for the installation and running of the pipelines have been documented and referenced. So, transparency of the outcome has been taken into account and we endeavored to make sure the problems encountered are reproducible by trying it ourselves on other environments and computers.

6 Conclusions

Acquiring (meta-)genomics data on microbial systems has become a standard procedure for many researchers in the past decade. However, the approaches for the analysis of such data still lack Standard Operating Procedures. New metagenomic analysis tools are emerging increasingly more in the past few years and there is no database or directory for keeping an overview of them. This has led to many confusions for researchers when it comes to selecting a tool that suits their needs. Moreover, setting up and installation of these tools often require programming and computer science knowledge which is still mostly lacking in the typical curriculum of most studies. This induces technical difficulties which can cause a significant delays in a researcher's work and distract them from focusing on their main (biological) question. Additionally, all the technical configurations for installing the tools do not stop when the research is finished, but rather get propagated to the future for anyone who would want to falsify the research and reproduce the same pipeline.

One way to address these issues would be a software or online platform where the majority of such metagenomic analysis tools are available to use in a user-friendly interface for users with no technical background. One such promising platform is Kbase which enables researchers to upload their data, build pipelines using numerous tools, and download or share their workflow and results with others all in one web-based interface. We believe that as the field of metagenomics becomes more mature and enters a new era, more similar platforms such as Kbase can remove the technical difficulties of tool setup for non-technically interested users.

Summary of Findings

The findings of this research regarding the technical shortcomings that limit the reproducibility of metagenomic analysis and potential solutions can be summarized as follows:

Identified problems

1. Too many existing metagenomic tools and the constant emergence of new ones results in confusion and a lack of overview for (beginner) researchers to find a suitable tool for their need
2. Minimal or lack of computer science or programming background is a major reason many researchers have difficulty setting up and using metagenomic analysis tools and pipelines.
3. Minimal or lack of software development expertise of their developers is a major reason why many metagenomic tools have detrimental issues and bugs that makes them deprecated.

Potential solutions

1. A curated and crowdsourced Directory/Wiki of metagenomic analysis tools for a better overview and comparison of the existing and new tools
2. More integration of *programming* and *application usage* training as part of the curriculum to help prepare students for their future research careers.
3. Software development training and protocols for developers of bioinformatics tools to help them create and maintain sustainable tools and software applications.
4. Self-contained mashup interfaces like KBase for entry-level users who would want to only focus on conducting the research and not have to deal with technical difficulties.

We hope that this Computer Science-oriented perspective would shed some light on common issues that are hindering researchers, and hopefully, the solutions recommended can help with reducing the issue of reproducibility. However, it is important to note that this is not extensive research and our findings are not an exhaustive list for all problems and solutions possible. So, certainly, further extensive research in more tools and interviewing more researchers can possibly bring other important points to attention that were not covered in this research. Lastly, we hope more "all-in-one" tools such as Kbase gets funded and eventually provided to researchers for a more streamlined approach to metagenomic analysis.

Appendix

In this section, we append the extra documents that are part of the paper but due to their length, we keep them in a separate section at the end. Here below, we will address the interviews, datasets, and pipelines used in the methodology section.

A1. Interviews

As mentioned in the Interview section in the paper, the interviews took place in a semi-structured manner. This means that the setup of the interview was made fluid and dynamic to allow the interviewees to share their experiences freely without being influenced by the way the interviewer asked them the questions.

However, in order to make sure similar goals are reached with all the interviews, there were some similar questions that were asked to each interviewee. The goal of these interviews was for it to be exploratory and complementary to the literature survey that was conducted, and not to have a quantitative methodology and outcome.

First, several problems were identified by reading the papers, and then, to verify those points, we wanted to add human perspective experiences to it. We did not inform the interviewees of the outcome of the literature survey or what other interviewees mentioned. We wanted to make sure our findings would make the interviewees biased.

Questions:

Some of these questions that were asked to interviewees can be found below here:

What is your current study program?

What is/are your previous study programs?

How much programming experience do you have? (as part of the curriculum or self-learned)

What is your goal for conducting metagenomic analysis?

What is your experience in conducting metagenomic analysis?

In case you are conducting metagenomic analysis as part of a research, how much of your research depends on the result of this analysis?

How much have you spent so far on conducting such an analysis?

Where did most of your time have been spent? (On conducting the analysis itself or setting up technical problems?)

How many technical problems did you encounter in the setup of your pipeline?

What type of dataset are you using? Prokaryotes, Eukaryotes, etc.

What is the file format of your dataset? (Fastq etc) which zipping format?

What metagenomic tools are you using for your datasets?

If you have outsourced your dataset, where did you outsource it to? What was the process and how was your experience? What are the difficulties you faced in the process?

What has been your experience in using clusters such as HPC? What are the main difficulties you faced?

Here below, you can find the notes that were made during these interviews:

Interviewee #1:

She has been using ATLAS
ATLAS is very useful for the Bacteria dataset
Very high quality, small contamination because of CAT

She doesn't have access to the pipeline on TU Delft
Some professor has a Ph.D. student in Utrecht University

Because she doesn't have a Ph.D. she doesn't have access to the clusters

Nina built a pipeline herself, Ph.D. student, she built her own very curated based on her samples, took her 6 months she has written it down (documented), she is also running it herself on TU Delft servers

Nina Roothans (EBT GROUP, ENVIRONMENTAL BIOTECHNOLOGY)
David is one of the investigators in the group)

For atlas is important what samples do you have and what output do you want

Send samples for sequencing
Then she gets back reads f 150 base pairs
Atcg format
Atlas does QC session

Then Assembly to HQ scaffolds
They are bigger fragments. They combine the reads
They make a bigger fragment of the piece
These are called Contigs

Binning
Certain bacteria have a different ratio for atcg
Atlas uses GC ratios to do the binning
Also does classification

Each bin is one organism
From metagenome you go to MAGS

Then taxonomic classification

For puck samples, she was looking for a complete community of fungi, yeast, etc
However, for her samples, this pipeline is not suitable
Because the Binning methods of atlas (MetaBat, MaxBin2) are biased and don't work well for her. They are more suitable for bacteria

It can only bin (group) bacteria

For her type of samples, ATLAS is not suitable because of its binning approach

It's important to mention in my paper tat which methods are suitable for which data types

Pipelines for bacteria are quite established

Puck works with eukaryotic genomic data, like fungi, etc

There's another pipeline that she uses,
She gets the scaffolds from the atlas to another pipeline, CAT
https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1817-x

Each of these pipelines is mostly based ON THE CHARACTERISTICS OF ONE SPECIFIC ORGANISM

CAT:
Doesn't do binning
She wants to see in her sample what organisms are there
She got contigs from atlas
Within the sequences, cheeks which genes are expressed are proteins using ORF PREDICTION
Then checks it with the NCBI database

CAT blasts" open reading frames to the database from NCBI

Some contigs are big some are small

It checks for every ORF how each relates to what organism, and you can use different parameter

She has a bioreactor
Certain columns of liquids
Her reactor runs on a chemostat, certain flow in and flows out
Sterile system. Closed. No organism from outside should enter the system

She put human feces as a start culture
The ecosystems different than inside the gut

It's a mixed culture. Diff organism

She did antibiotic
And she took samples all the time

Cat is discussed in the CAMI benchmark tool and in cat paper, to see if it's good enough

Now same Ph.D. guy is running CAT also for her

Takes 2 days to ruin CAT

The file size of the sequenced data is HUUUGE. She can't even download it to her computer
Initial samples: 2.7 GB and 2.8 GB and add up. And that's one sample
Depends on how you sequence
Big or short reads and what depth.

Novogene
https://en.novogene.com/services/research-services/genome-sequencing/whole-genome-sequencing/?gclid=Cj0KCQjw4cOEBhDMARIsAA3XDRjtsVk0TBsJj_h7t59yreBvL-aqfh1V9xPGXDjRKp19b0syaVOpTJ8aAn3GEALw_wcB

She spin the liquids sample, discards liquid and the solid remains (palette), the she does DNA extraction
Cell is encapsulated
You wanna inly take out the dna

Then she sends a liquid with her extracted dna

You "dissolve" in a fluid solution
Every month with her group she sends it to NOVOGENE to sequencing

TU Delft doesn't have sequencing machine
They are expensive and require many steps

It's expensive with high throughput machine
350 EURO

Interviewee #2:

BT: Enviromental BioTechnology . The section that David's group belongs to, which is part of TNW faculty

Nanobioology fgot bsc. Did bioinformatics with Thomas Abeel

She's doig graduatrion project
Nano part at the end of june, and ecology later

Grow e-coli at difff speeds.
Do comparative analysis for genome and protemoe

She had contamination in her samples

Eventually sequences were 3-4 species. And had to bin those genomic sequences to find what she wanted

She only has a small mix culturer witha predominant specie

Metagenomics:
She uses StrainGE, produced by Lucas at Thomas Abeel lab
Needs Linux

K-base (online) and Galaxy Europe (online platform)
And RRRRR to do the plotting and stuff

DNA eextraction she sent to a company to sequence it for her. Reads. Raw data
(It's possible to order them to do metagenomics, taxnomy analysis etc for her)

K-base:
Narrative: FASTQ format of reads
Forward and reverse read files and she added both FASQ to Narrative
K-base generates a read library.
Quality Control with FastQC: she ran it on K-base.

In total she has 6 samples. These read files are not massive. She has limited number of species.
If you have many speciie, file is too big, hard to work with, can't upload somewhere like k-base

MetaSPADES (also on K-base)to assemble the reads to create contigs. To then align and compare to the reference genome
KAIJU (on k-base), checks the taxonomy, it tells you e.g. Your sample s 50% e-coli, 50% etc. It's a detailed taxonomy information. You have to use a reference database. She use REFSEQ genome database.

Then she Binned (on k-base), MAXBIN2 for binning, it uses a ref gen , it puts all contigs in one bin. E.g. E-coli in this bin
If the specie doesn't reach a threshold, it will discard them.

Then you can extract as Assemblies./ you can take all the reads from Bin 1 as assembly.
She checked the quality of the assemblies: with QUAST . It's QC for your asssemblies. The longer the contig the better (rule of thumb)

She uses a taxonomic classifier tool GTDBTK to check which bin belongs to which organism

Now she has all the bin assembly and now which bin belongs to which organism

PROKKA (k-base) it annotates your genome: \
Annotation software: (also e.g. RAST): Stella uses RAST (probably on j-base or Linux bioconda)
Prokka uses a database, looks at your genome, and identifies features of interests and labels them.
Protein codings will be labeled.
Prokka outputs a GenBank files.

Galaxy Europe platform
You can use the GenBank files to extract information such a s Metaproteomics information at Galaxy

So fart this is all metagenomics sequencing

If you only wanna 16S rRNA sequencing is for determ,ining what bacterial species do you have
It only gives a n OTU tables that gives you the abundance for the bacterial specie.
It's smalll info. Less info, less interesting. Just a simple tool for determining bacterial composition that is in your sample.
Not as exciting as metyagenomic sequencing

This alll took Aisha WEEEKS to understand what she needs to for metagenomics part of her researech
She thinks such tool at TU Delft would be useful

Lucas van Dijk, PhD at Thomas Abeel lab
https://github.com/broadinstitute/StrainGE

StrainGE: Strain-level Genome Exploration
StrainGE is a set of tools to analyse the within-species strain diversity in bacterial populations. It consists of two main components: 1) StrainGST: Strain Genome Search tool, a tool to find close reference genomes for strains present in a sample and 2) StrainGR: Strain Genome Recovery, a tool to perform strain-aware variant calling at low coverages.

He has a tool that has 2 functionalities

Aisha had contamination in her samples. There's also a lot of different strains of an specie

She wanted to do a variant calling: align your genome against refence genome to see how many mutations there are
Buit if you haver a strain that is not a good match, then you gonna see a lot more false positive rates of differences with ref genbome

StrainGE tool:
Linux env

Download teh tool and activate the environment

First part 1:
StrainGST: Database creation
Make a directory and download all the NCBI genomes for a certain specie gene.
Splits chromoses and plasmites
Then you do kmer clustering (find the similar ones) and compare with each other

When you download all diff strains of your specie e.g. E-voli, thety are very similiar
KMER clustering, removes the redundant info and create a database file
You also KMERize your sample genome.

She did it on her raw data. And it looks at the KMER cluster for your ssample and references genome and shows the difference.

It can be useful if you look at wastewater samples, and see how one organism in your sample overtime changes.
To extract one organism

PART 2
StrainGR: looks at all the straisn that you have in your sample
For gher it identifed 4 strains. StrainGR puts all the refernece genomes with each other to create Scaffolds
Copncatenated reference genomes contains (e.g. 4) scaffolds
Every scaffold is 1 reference genome

Then you index it with StrainGR with BWA (integrated to StrainGR)
Then you create a .SAM file and .BAM file to calll variants
You aligne your genome vs ref genome. To know where your sequence starts and ends.

You can do variant calling with StrainGR

Then .vcf is the output from StrainGR with command:
straingr call outputs .vcf
It's a file that has the genomic positions and mutations

Last thing it can run straingr compare to compare the sambles. Like Jaccard similarity, etc.
It outputs a table that has all that information

VCF compare tool (in a linux env.) PIP install
You input all vcf file and it gives you amount of share variants
E.g. It shares 2000 mutations and it shares 4000 that didn't mutate

You can also install a Virtual Machines instead of dual booting

---

SnpEff in Galaxy Europe environment

She has 4 ref genomes, then she merges those (they are FASTA files.)
She merges them in Galaxy , FASTA Merge tool

Then she annotated it with Prokka in Galaxy. In Galaxy itakes a bit longer but has different output formats

Using SnpEff tool, it takes a vcf file and aligns with the annotated gene. , then it can tell you where your mutations are positioned.
It can also do impact prediction: if you use or gain sopmething in protein. Z(sttill not a trustworthy tool)

You can get a percentage of the silent mutations.

ABRicate: Anti biotic resistance check

Goes through your coding sequencers from GenBank files. You can extract coding sequences
You can run ABRicate, it will tell you if there's antibiotic resistance genes.

People in EBT do a lot of research into antibiotic resistant genes.

Interviewee #3:

Full conrik
She wanted tio it her own ,
wastewater treatment
So many micro organisms. dataset will be very big,,
Usually pre made pipelines are not suitable for big datasets

For deciding which tools at very step, she read paper

metaWRAP had many tools she needed
She wants to know what happens to her data at every step,
That's what makes it difficult to make a general pipeline
Depends on the input and the output you're looking for

If you have a simple sample, maybe it would work

Her datasetL:
From a treatment plant.
To clean the water they have this microorganisms
Activated sludge (thousands of microorganisms which are present in very low amount)
Bacterias (procvkaruyets)

She want to look with details what microorganisms are there and what genes each micro organism has

She wants to make MAGs , Metagenomic Assembled Genomes from Activated Sludge

The order of doing the steps in a pipelines also depends on what you want to do

Her biggest problems:
Are in assembly step

1. She loses a lot of data there. So many puzzle pieces that she can't put them together

Assembly/scaffolding: combining small dna pieces to make longer pieces.
There are many small pieces that the pipeline can't put together with any other pieces.
she was doing short read sequencing, higher quality,, less errors but when you want to do assembly you might have a problem like in gher case

solution: combine short read abnd long read sequencing.
short read to fix the errors and long read sequences to combine the pieces
then you merge this two
some pipelines already do that.

2. Binning step: complicated because if any of the bacteria have similar dna, software has a problem separating them properly/
there exist some software that are specific for different samples
In Denmark, they are making MAGs from wastewater.
Aarlborg University, Mads Albertsen
she has read their papers, but haven't contacted them
she is waiting to understand her problems first
they have a good approach for fixing the binning method

3. Annotation is gonna be difficult

4. Depending existing databases from NCBI for example

In Denmark they use MiDAS databases so other people can use them as well.
problem with using big general databases such as NCBI instead of more targeted ones such MiDAS for specifically wastewater bacteria, is that annotation becomes a lot more difficult
NCBI has some certain subsets such as gut bacteria, but not waste water yet.

Technical:
First defined pipeline on paper by reading papers etc.
And she chooses the best software for each step. Then she had it theoretically on papert

She started with Linux
And started building one by one step ,
She had it directly on the server
It was a big step to learn how to use it.

You don't have admin rights, so you need to follow the documentation from HPC
And she gets all these errors in lnux she doesn';t understand

Most of the tools need linux and need a lot of memory or cpu

If a lot of the dependencies were already installed , that would save so much time.

Iiirc, FastQC was already installed.
If there';s a general pipeline that everyone uses,

WALKTHROUGH:
She prepares scripts from before hand that she wants to run

Interviewee #4:

To contact:
Roel Sarelse: developed a metagenomic pipeline

I don't think General purpose would work

She didn't know how to program
Se studied biology

They were collaborating with Warsaw in Poland

She helped a lot but the programming part she didn't do

Sh eknows another person who took 8 months to develop something

She did a research on assessing tool and databases

We analyze metahominc data from wastewater treatment plants
Antibiotic resistance
Synthetic gene
Synthetic sequence is built from scratch in biology. This requires patents and hard to retrieve

Plasmit: a piece of DNA, vectors, they carry genes

Mobile genetic element
They also search for taxonomy, what are the microorganisms that are there

PROCEDURE:

RAW DATA,
Illumina sequencing, short reads
FastQC to check the quality the data
They used MetaHeat for Assembly (put short reads together to build Contigs)
1001 base pairs for the tools they use
5-10% of the total info they had
It's not good,m but probably it's because of short reads of ilumina, maybe nanopore long redas would have helped but they are expensive

Then they had a database for the antibiotic resistance
In her thesis objective to assess the combination of more
She created a dictionary for Unifying all the data

Roel Sarelse also worked on similar thing with a different pipeline (master student)
He used the same sample

Taxonomy analysis,
He used Kraken and even more tools.
The end goal is to see how many plasmid and antibiotic resistance ones
He did two rounds of
He used another tools with raw data, some preprocessing for

She used CAT BAT that works with Contigs for Taxonomy Analysis with NCBI

Universal adapters for the sequencing.they need to be trimmed

In wastewater treatment plant, detection of biocide is a chemical compound in detergent/ disinfectant for plants,
They are correlated with antibiotic resistance,
BACMED database

MiDAS David will like this database more

She provided short reads from illumina sequencing to Warsaw
She gave them the like a list of all the tools she had found, and compared the strength and weaknesses

For assembly she got the congis and their length,, and summary of the distribution of the samples
How many base pairs each area, e.g. Above 1000 or 2000

Warsaw also run protein based analysis

They would choose the database that would return the best results

Identity above 90% coverage

They were using the ABRicate tool for metagenomic analysis.

He used PROKKA for protein analysis

For plasmids for classifications for chromosome okasit
They used : PlasFlow and PlasClass
They used both and combined the data. At the end chose for PlasClass

They took chromosomal sequenced above x%
Then Phage identification tools

The tools for Phagues:
VirSorter2
DeepVirFinder

He gave her CSV files and then they had contigs that are plasmids/phages

Interviewee #5:

You need longer reads
For Binning specifically

You would only 5-10 bing instead of 100

3 datasets
Short reads 150 base pairs from repares

300 base pairs (illumina)

Long reads (nanopore etc)

Antibiotic resistant in wastewater
Experience with metagenomics:
User level
QC, trimming, assembly binning

Tools:
Pipelines cant be developed universally

Different needs , diff quality of files, diff databases

Universal Pipelines don't exist

I don't think your tools will be useful for everybody

FastQC
They want something visual
Then it's easy to see

If you can export

MetaHeat
MetaSpades

Marcos Cuesta

(biologist, but has experience in HPC, metagenomic datasets)

Trimming
QC
Taxonomy

KMERS is
Kraken2 - RAW reads, KMERS based
Centrifuge, Metaphlan, Kaiju

Contigs instead of raw reads

For wastewater databases samples:
MIDAS genome database
For comparison of diff methods

Different ways to align if there's specific genes (e.g. antibiotic resistance)

BWA (Burrow-Wheeler Alignment)- mem
BLASTn
(you don't need HPC)

For assembly and binning you need clusters

Database David cal uses for antibiotic resistant bacteria
ResFinder or CARD

Command line

HPC
https://datasainslab.com/high-performance-computing-hpc-tu-delft/

https://www.tudelft.nl/dhpc/

https://login.hpc.tudelft.nl

PuTTy
On Slack:
David worked on his own laptop for
BWA (Burrow-Wheeler Alignment)- mem
BLASTn
(you don't need HPC)

A2. Datasets

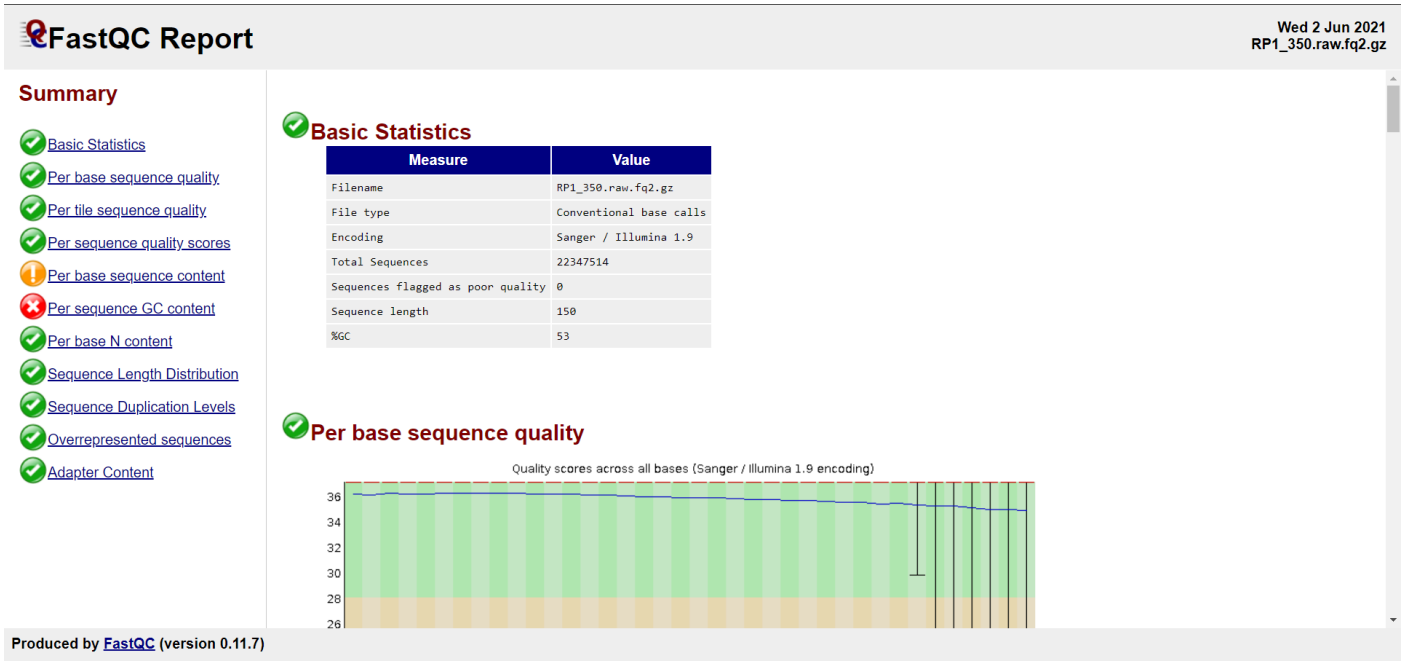
The dataset that has been used for the pipeline development methodology was provided by the EBT group of TNW faculty. These datasets are sequenced metagenomic data from bacterial communities found in wastewater treatment plants. The sequencing has taken place by different groups that attended a workshop conducted by David Weissbrodt and David Calderon for REPARES. [34]
Here is the link to download the datasets:
https://surfdrive.surf.nl/files/index.php/s/cFidbU34EElZpHz

A3. Pipeline Development

As mentioned in the paper, two approaches were taken for the pipeline development section. First was installing a CLI GPP tool called ATLAS on a personal computer and HPC. Then, trying an all-in-one online GUI tool called Kbase. Here below you can find the report that was made through these pipelines and the workflow.

ATLAS FastQC report

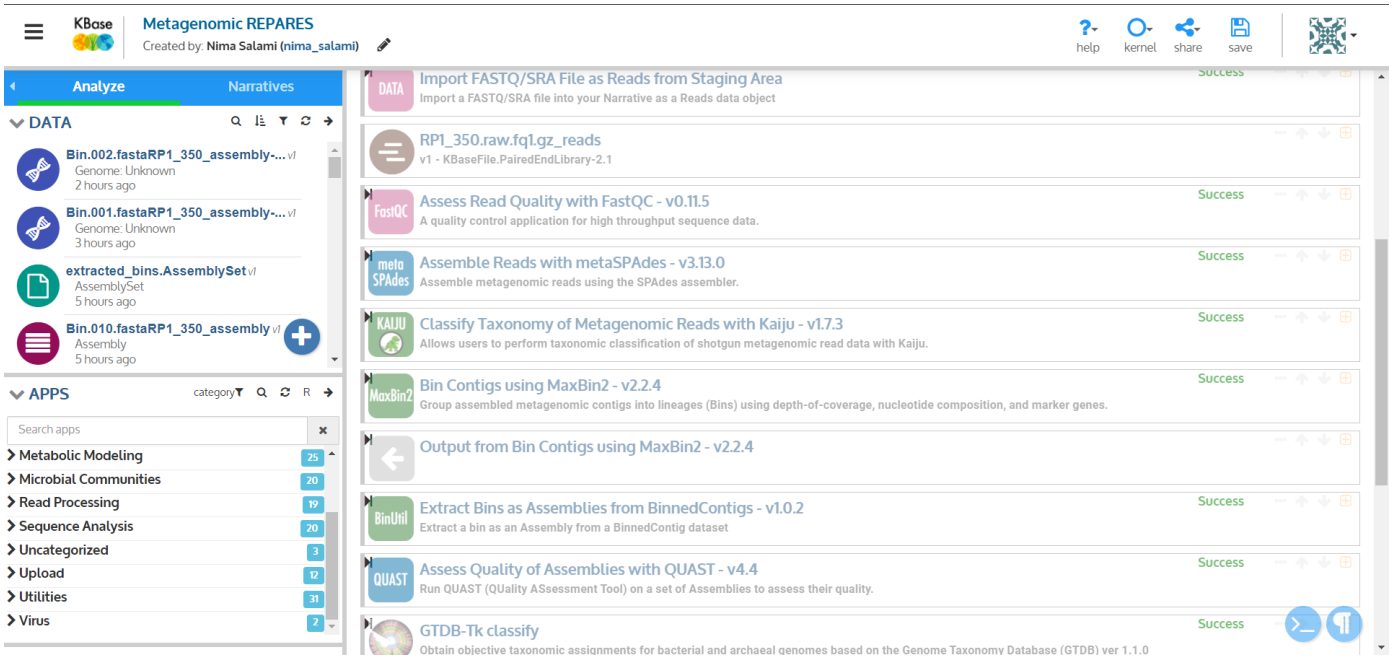
The FastQC file produced after running repares dataset through the first step of ATLAS. The link to download is below:
https://drive.google.com/drive/folders/1PBGTbNG7AFhElxPhIk86y-T3uNSIDX0b?usp=sharing



Kbase Workflow

The workflow/pipeline created through Kbase is available with the link below. In case, the link did not work, you can email your user ID and email address to my email address so I can share the workflow with you.

https://narrative.kbase.us/narrative/93375



References

1. Metagenomics. [cited 25 Jun 2021]. Available: https://www.genome.gov/genetics-glossary/Metagenomics
2. Roumpeka DD, Wallace RJ, Escalettes F, Fotheringham I, Watson M. A Review of Bioinformatics Tools for Bio-Prospecting from Metagenomic Sequence Data. Front Genet. 2017;8. doi:10.3389/fgene.2017.00023
3. Taguchi Y-H. Comparative Transcriptomics Analysis. Encyclopedia of Bioinformatics and Computational Biology. 2019. pp. 814–818. doi:10.1016/b978-0-12-809633-8.20163-5
4. Lightbody G, Haberland V, Browne F, Taggart L, Zheng H, Parkes E, et al. Review of applications of high-throughput sequencing in personalized medicine: barriers and facilitators of future progress in research and clinical application. Brief Bioinform. 2019;20: 1795–1811.
5. Pérez-Cobas AE, Gomez-Valero L, Buchrieser C. Metagenomic approaches in microbial ecology: an update on whole-genome and marker gene sequencing analyses. Microb Genom. 2020;6. doi:10.1099/mgen.0.000409
6. Poussin C, Sierro N, Boué S, Battey J, Scotti E, Belcastro V, et al. Interrogating the microbiome: experimental and computational considerations in support of study reproducibility. Drug Discov Today. 2018;23: 1644–1657.
7. Shkoporov AN, Ryan FJ, Draper LA, Forde A, Stockdale SR, Daly KM, et al. Reproducible protocols for metagenomic analysis of human faecal phageomes. Microbiome. 2018;6: 68.
8. Narayanasamy S, Jarosz Y, Muller EEL, Heintz-Buschart A, Herold M, Kaysen A, et al. IMP: a pipeline for reproducible reference-independent integrated metagenomic and metatranscriptomic analyses. Genome Biol. 2016;17: 1–21.
9. Kieser S, Brown J, Zdobnov EM, Trajkovski M, McCue LA. ATLAS: a Snakemake workflow for assembly, annotation, and genomic binning of metagenome sequence data. BMC Bioinformatics. 2020;21: 1–8.
10. metagenome-atlas. metagenome-atlas/atlas. [cited 27 Jun 2021]. Available: https://github.com/metagenome-atlas/atlas
11. Conda — Conda documentation. [cited 27 Jun 2021]. Available: https://docs.conda.io/en/latest/
12. Empowering app development for developers. [cited 27 Jun 2021]. Available: https://www.docker.com/
13. Zhai P, Yang L, Guo X, Wang Z, Guo J, Wang X, et al. MetaComp: comprehensive analysis software for comparative meta-omics including comparative metagenomics. BMC Bioinformatics. 2017;18: 1–16.
14. Uritskiy GV, DiRuggiero J, Taylor J. MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. Microbiome. 2018;6: 1–13.
15. Supercomputer. [cited 27 Jun 2021]. Available: https://en.wikipedia.org/wiki/Supercomputer
16. Beghini F, McIver LJ, Blanco-Míguez A, Dubois L, Asnicar F, Maharjan S, et al. Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3. Elife. 2021;10. doi:10.7554/eLife.65088
17. Truong DT, Tett A, Pasolli E, Huttenhower C, Segata N. Microbial strain-level population structure and genetic diversity from metagenomes. Genome Res. 2017;27: 626–638.
18. McIver LJ, Abu-Ali G, Franzosa EA, Schwager R, Morgan XC, Waldron L, et al. bioBakery: a meta'omic analysis environment. Bioinformatics. 2018;34: 1235–1237.
19. Segata Lab - Computational Metagenomics. [cited 1 Jul 2021]. Available: http://segatalab.cibio.unitn.it/tools/index.html
20. MetaPhlAn. Github; Available: https://github.com/biobakery/MetaPhlAn
21. Prlić A, Procter JB. Ten Simple Rules for the Open Development of Scientific Software. PLoS Comput Biol. 2012;8: e1002802.
22. Lee BD. Ten simple rules for documenting scientific software. PLoS Comput Biol. 2018;14: e1006561.
23. Sandve GK, Nekrutenko A, Taylor J, Hovig E. Ten Simple Rules for Reproducible Computational Research. PLoS Comput Biol. 2013;9: e1003285.
24. Williams C. How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript. In: The Register [Internet]. 23 Mar 2016 [cited 23 Jun 2021]. Available: https://www.theregister.com/2016/03/23/npm_left_pad_chaos/
25. Ono K. Cytoscape. [cited 27 Jun 2021]. Available: https://cytoscape.org/
26. Wikipedia contributors. Command-line interface. In: Wikipedia, The Free Encyclopedia [Internet]. 16 Jun 2021 [cited 1 Jul 2021]. Available: https://en.wikipedia.org/w/index.php?title=Command-line_interface&oldid=1028938688
27. Welcome to Mamba's documentation! [cited 27 Jun 2021]. Available: https://mamba.readthedocs.io/en/latest/index.html
28. Website. [cited 27 Jun 2021]. Available: https://github.com/metagenome-atlas/atlas/issues/390
29. Wikipedia contributors. Graphical user interface. In: Wikipedia, The Free Encyclopedia [Internet]. 1 Jul 2021 [cited 1 Jul 2021]. Available: https://en.wikipedia.org/w/index.php?title=Graphical_user_interface&oldid=1031411284
30. Arkin AP, Cottingham RW, Henry CS, Harris NL, Stevens RL, Maslov S, et al. KBase: The United States Department of Energy Systems Biology Knowledgebase. Nat Biotechnol. 2018;36: 566–569.
31. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018;46: W537–W544.
32. BaseSpace Sequence Hub. [cited 22 Jun 2021]. Available: https://www.illumina.com/products/by-type/informatics-products/basespace-sequence-hub.html

33. REPARES - REPARES. [cited 22 Jun 2021]. Available: https://repares.vscht.cz/

34. Seminar EU project REPARES on antimicrobial resistances in the wastewater environment. [cited 22 Jun 2021]. Available: https://www.tudelft.nl/evenementen/2020/tnw/tnw-corporate/seminar-eu-project-repares-on-antimicrobial-resistances-in-the-wastewater-environment

35. KBase. [cited 22 Jun 2021]. Available: https://narrative.kbase.us/narrative/33233

36. KBase. [cited 22 Jun 2021]. Available: https://narrative.kbase.us/narrative/93375

37. QIAGEN CLC Main Workbench. 21 Nov 2019 [cited 1 Jul 2021]. Available: https://digitalinsights.qiagen.com/products-overview/discovery-insights-portfolio/analysis-and-visualization/qiagen-clc-main-workbench/

38. Latest improvements. 16 Dec 2019 [cited 1 Jul 2021]. Available: https://digitalinsights.qiagen.com/products/qiagen-clc-main-workbench/latest-improvements/archive/

39. Latest improvements. 16 Dec 2019 [cited 1 Jul 2021]. Available: https://digitalinsights.qiagen.com/products/qiagen-clc-main-workbench/latest-improvements/current-line/

40. QIAGEN CLC Main Workbench Review. 13 Oct 2020 [cited 1 Jul 2021]. Available: https://www.softwareradius.com/clc-main-workbench-review/

41. atlas. Github; Available: https://github.com/metagenome-atlas/atlas

42. Mashup (web application hybrid). [cited 25 Jun 2021]. Available: https://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)

43. Pearce O. 5 fundamental steps to creating powerful standard operating procedures. [cited 3 Jul 2021]. Available: https://blog.montrium.com/experts/5-fundamental-steps-to-creating-powerful-standard-operating-procedures

44. How to implement standard operating procedures (SOPs)? 28 Dec 2017 [cited 3 Jul 2021]. Available: https://blog.v-comply.com/implement-standard-operating-proceduressops/

45. Wikipedia contributors. Standard operating procedure. In: Wikipedia, The Free Encyclopedia [Internet]. 23 Jun 2021 [cited 3 Jul 2021]. Available: https://en.wikipedia.org/w/index.php?title=Standard_operating_procedure&oldid=1030077168

46. Garrity GM, Field D, Kyrpides N, Hirschman L, Sansone S-A, Angiuoli S, et al. Toward a standards-compliant genomic and metagenomic publication record. OMICS. 2008;12: 157–160.

47. van Gelder CWG, Hooft RWW, van Rijswijk MN, van den Berg L, Kok RG, Reinders M, et al. Bioinformatics in the Netherlands: the value of a nationwide community. Brief Bioinform. 2017;20: 375–383.

48. ELIXIR Netherlands. [cited 3 Jul 2021]. Available: https://elixir-europe.org/about-us/who-we-are/nodes/netherlands

49. BioSB research school. [cited 3 Jul 2021]. Available: https://www.biosb.nl/