UNIVERSITY OF TARTU

FACULTY OF SCIENCE AND TECHNOLOGY

INSTITUTE OF MOLECULAR AND CELL BIOLOGY

INSTITUTE OF ECOLOGY AND EARTH SCIENCES

DEPARTMENT OF BOTANY, CHAIR OF MYCOLOGY

Development of a multi-platform metabarcoding bioinformatics software with an easy-to-use graphical user interface

Master's thesis

30 EAP

Martin Metsoja

Supervisors: PhD Sten Anslan, PhD Leho Tedersoo, PhD Maido Remm

TARTU 2024

Info page

Development of a multi-platform metabarcoding bioinformatics software with an easy-to-use graphical user interface

Metabarcoding, a widely adopted technique in molecular ecology, enables the simultaneous identification of organisms from environmental samples. However, the sheer volume of sequencing data generated by metabarcoding, diverse DNA markers and sequencing platforms introduces complexity during downstream bioinformatic processing. Existing bioinformatic tools for metabarcoding data analysis often operate via command line interfaces, involve intricate installation procedures and lack cross-platform compatibility. To address these challenges, we introduce PipeCraft2, a novel software package developed using state-of-the-art tools such as Docker, Electron, and Vue.js. Key features of PipeCraft2 include straightforward installation, cross-platform support, an intuitive graphical user interface, four distinct pipelines, and a range of individual modules.

Graafilise kasutajaliidesega metatriipkoodistamise bioinformaatika tarkvara arendus

DNA meta-triipkoodistamine on laialdaselt kasutatud meetod molekulaarses ökoloogias ning võimaldab keskkonnaproovidest üheaegselt tuvastada mitmete organismide DNA-d. DNA meta-triipkoodistamisel tekib suures koguses järjestus andmeid ning koos teiste teguritega, nagu DNA markerite ja sekveneerimis platvormide mitmekesisus võib tihti osutuda hilisem andmete bioinformaatiline töötluse üpriski keerukaks. Olemasolevad tööriistad meta-triipkoodistamis andmete analüüsiks on enamasti käsurea põhised, nende paigaldamine on tihti keerukas ning võimalik ainult Linux operatsioonisüsteemiga arvutitel. Kasutades kaasaegseid aredus tööriistu ja raamistikke nagu Docker, Vue.js ja Electron loodi uus bioinformaatika tööriist PipeCraft2, kus puuduvad eelnevalt mainitud kitsaskohad. PipeCraft2 on lihtsalt installeeritav, kasutatav Linux Mac ja Windows arvutitel, sisaldab intuitiivset graafilist kasutajaliidest nelja erinevat töövoogu ja mitmeid individuaalseid mooduleid.

CERCS codes: B110

Keywords: metabarcoding, bioinformatics, graphical user interface, pipecraft

Table of contents

Info page 1
Table of contents 3
Abbreviations 5
1. Introduction 7
2. Development tools and frameworks 10
    2.1. Docker 10
        2.1.1. Docker images 11
        2.1.2. Docker containers 12
    2.2. Vue.js 12
        2.2.1. Vuex 13
        2.2.3. Vue Router 14
        2.2.4. Vuetify 14
    2.3. Electron 14
    2.4. Git 15
    2.5. Github and Read the Docs 15
3. Design Principles and Features 16
    3.1. Layout 17
    3.2. Selecting data 18
    3.3. Custom front-end elements 18
    3.4. Workflow execution 20
    3.5. Save and Load 21
    3.6. Expert Mode 22
    3.7. Implemented third-party bioinformatic tools 23
    3.8. Quality check 25
    3.9. Built-in pre-defined pipelines and Quick Tools 25
        3.9.1. Demultiplexing 27
        3.9.2. Reorient 28
        3.9.3. Cut primers 28
        3.9.4. Quality filtering 29
        3.9.5. Paired-end assembly 29
        3.9.6. Chimera filtering 30
        3.9.7. ITS Extractor 30
        3.9.8. Clustering 31
        3.9.9. Post-clustering 31
        3.9.10. Taxonomy assignment 32
        3.9.11. Post-processing 32
4. Performance 33
5. PipeCraft2 in action 35

6. Discussion 35
7. Conclusion 38
8. Resümee 38
9. References 41
10. License for Reproduction and Public Accessibility of Thesis 48

Abbreviations

API – application programmatic interface

ASV – amplicon sequence variant

CLI – command line interface

CPU – central processing unit

CSS – cascading style sheets

DOM – document object module

DSOS – Docker sponsored open source program

eDNA – environmental DNA

GUI – graphical user interface

HPC – high performance computing

HTML – hypertext markup language

LINPACK – linear algebra package

MSFD – marine strategy framework directive

NAMD – molecular dynamics application

OS – operating system

OTU – operational taxonomic unit

PCA – principal component analysis

PCR – polymerase chain reaction

RAM – random-access memory

UI – user interface

WFD – water framework directive

WSL – Windows subsystem for Linux

1. Introduction

The decrease in biodiversity worldwide is a universally recognized reality (Thomsen and Willerslev, 2015) and tracking it using conventional methods like morpho-taxonomy is rather challenging due to multiple factors such as being time-intensive, having restricted temporal and spatial resolution, and being susceptible to errors stemming from variations in the taxonomic expertise of individual analysts. Metabarcoding, a simultaneous identification of organisms from environmental samples (Taberlet et al 2012), is an increasingly popular technique in the field of molecular ecology and offers a potential solution to these issues and has the capacity to enhance traditional bioassessment, biomonitoring and research strategies (Leese et al. 2016). It relies on the sequencing of DNA marker genes from the environmental DNA (eDNA) samples to assess the biodiversity and composition of organism groups of interest (Tedersoo et al. 2021). Environmental DNA for metabarcoding analyses may be extracted from a variety of substrates such as air, water, soil and sediments. Taxonomically informative DNA markers from the extracted eDNA are amplified via polymerase chain reaction (PCR) and thereafter sequenced on a high-throughput sequencer (Ruppert et al. 2019) with the ultimate aim of identifying the occurring species in the sampling units. eDNA metabarcoding exhibits enhanced species detectability, demands less effort, avoids ecosystem disruption, enables detection without prior knowledge of species, and can be applied in regions where conventional surveys are not feasible (Valentini et al., 2016). The developments in this molecular identification method have enhanced our understanding about the diversity, ecology as well as biogeography of many organism groups. For example, metabarcoding analysis of globally collected soil samples have shown that plant diversity does not causally impact the richness of most fungi. Instead, total fungal richness was most affected by distance from the equator and mean annual precipitation (Tedersoo et al. 2014). Fungi and bacteria have shown to have divergent biogeographic trends as fungal diversity seems to follow the conventional latitudinal diversity gradient but bacterial diversity peaks in temperate habitats this suggests niche segregation influenced by their disparate responses to environmental factors. Soil pH being the primary influence to the global distribution of bacteria, while precipitation emerges as a key determinant for the global distribution of fungi (Bahram et al. 2018). The Tara Ocean project, through the implementation of metabarcoding analyses, has established a pronounced correlation

between oceanic temperatures and microbial diversity, thereby yielding significant insights into the consequences of climate change (Sunagawa et al. 2015).

Furthermore, metabarcoding analyses exhibit promising prospects in the field of biomonitoring. In Europe there are already legally binding edicts such as the European Union's Water Framework Directive (WFD) and the Marine Strategy Framework Directive (MSFD) which focus on protecting, preserving and restoring aquatic ecosystems. These directives rely on conventional methods for biodiversity tracking until 2027, though there is potential for change as an Action in the European Cooperation in Science and Technology called DNAaqua-Net is working towards incorporating metabarcoding and eDNA techniques into fast and cost-effective biomonitoring protocols and, eventually, into existing legislation (Leese et al. 2016). The US Environmental Protection Agency (EPA) has also set up a strategic plan to incorporate DNA-based methods into water quality assessments (Stein et al. 2014). With the progression of technology and the refinement of methodologies, metabarcoding is poised to become a crucial instrument in the realm of biodiversity monitoring (Ruppert, Kline, and Rahman 2019) and might even become the golden standard supported by legislators and policymakers.

Metabarcoding analysis with high-throughput sequencing (HTS) machines generates a vast amount of sequencing data, generally hundreds of thousands of sequences per sample. This coupled with other factors such as the variety of DNA markers and sequencing platforms used in metabarcoding approaches leads to advanced complexity in downstream data processing. Several gigabytes of sequencing data require proper bioinformatic processing, which may vary depending on the sequenced gene fragment and sequencing platform itself. The ongoing rapid development of sequencing technologies means that constant improvements must be made to bioinformatics pipelines to keep pace with new technologies (Anslan et al. 2017; Bolyen et al. 2019). Popular tools for bioinformatic processing of metabarcoding data include mothur (Schloss et al. 2009), vsearch (Rognes et al. 2016), usearch (Edgar 2010) and OBITools (Boyer et al. 2016), which include algorithms for compiling complete sequence data analysis pipeline. By including some new algorithms in combination with a pre-selected functionalities of other third-party tools, many metabarcoding data analyzes pipelines, such as QIIME (Bolyen et al. 2019), LOTUS (Hildebrand et al. 2014), PIPITS (Gweon et al. 2015) are available for essential sequence data

analyses. Although each has its unique merits, they do tend to be exclusively operated from the command-line, which will require intermediate if not expert bioinformatic knowledge to use these tools. Many tools often lack cross-platform support and workflows can be tedious to repeat and reproduce. Therefore, releasing pipelines with a graphical user interface is a growing trend for example gDAT (Vasar et al. 2021) and SEED (Větrovský, Baldrian, and Morais 2018) are operated from a graphical user interface with the former also supporting multiple operating systems (Windows, Linux, Mac), yet their functionality is somewhat limited by including only a narrow set of sequence data processing tools. Many pipelines also rely on a multitude of external dependencies which makes installing them a tedious and time consuming task. To thwart all this complexity a new software package was developed called PipeCraft (Anslan et al. 2017). PipeCraft is a metabarcoding data analysis toolkit, first released in 2017, to provide flexible bioinformatic processing of high-throughput amplicon sequencing data. This graphical user interface toolkit for non-bioinformaticians implements and links many public tools for crafting custom bioinformatic pipelines to analyze sequencing data from various high-throughput sequencing platforms. However, the graphical user interface (GUI) of the first version of PipeCraft is not directly executable on Windows or Macintosh (Mac) machines, but requires executing through Virtual Machine (VM).

The aim of this thesis was to further develop PipeCraft by building a new easy-to-use GUI with a modern javascript framework, packaging and deploying all bioinformatics tools with virtualization technology, deploying the electron framework to make PipeCraft available for all platforms (Windows, Linux and Mac) and making it easier to share and reproduce bioinformatic workflows by save and load functionality. This also included the development of a Quality Check module to gain in-depth insight of the data before proceeding with further processing and a component for experienced bioinformaticians to allow for highly customized use of all of the integrated assets. As PipeCraft represents a wrapper for various popular and well-tested metabarcoding bioinformatics tools and pipelines, this thesis does not aim to benchmark bioinformatic workflows themselves.

2. Development tools and frameworks

To fulfill the objectives outlined during the initial stages of development, it was essential to employ modern and well-established tools and frameworks. The front-end, the user-visible and interactive aspect of the application, required a seamless and user-friendly experience. While there were simpler alternatives, such as Gambas that was used in the initial release, or utilizing a basic Python library like Tkinter, it became evident that opting for a modern web development framework such as Vue.js would provide PipeCraft with enhanced flexibility and adaptability for front-end design. Also, the back-end that is responsible for data handling and computations behind the scenes, underwent significant enhancements, but the focus of this thesis is on improving the graphical components, reproducibility and simplifying installation. This was achieved by utilizing the Electron framework (www.electronjs.org) and incorporating Docker (www.docker.com), a modern containerization application, to minimize dependencies and streamline the overall system. The following sections represent key tools and frameworks for front- and back-end development.

2.1. Docker

The implementation methods of bioinformatic software are diverse, leading to costly and skill-intensive installation processes. Docker, a lightweight containerization program, introduces pragmatic solutions, enhancing software deployment, analysis reproducibility, and easing the portability and maintainability of bioinformatics tools (Moreews et al. 2015). Docker leverages various capabilities of the Linux kernel to deliver its functionality. The Linux kernels feature, known as namespaces, is used by Docker to create isolated workspaces referred to as containers (Docker Incorporated 2024). As all Docker containers run on a single operating system kernel they tend to be much more lightweight than traditional virtual machines and more comparable to local environments performance wise (Kwon et al. 2018). PipeCraft2 utilizes Docker to package and deploy a great number of bioinformatic tools as containers, that include their mandatory libraries and dependencies, which releases the user from the burden of manually installing the required bioinformatic software. The Docker architecture consists of two main parts, the Docker client and the Docker daemon. The Docker client is used to relay user commands such as pull, run, build and others to the

Docker daemon, which takes on the substantial tasks of constructing, executing, and distributing Docker containers. Despite the advantages of Docker, communicating with the Docker client from the command line interface (CLI) might still pose considerable challenges for inexperienced users (Menegidio et al. 2019), therefore the configuration of the Docker daemon and communications with the Docker client are fully automated in PipeCraft2. Furthermore Docker can be effortlessly connected to cloud computing services as the Docker daemon and client are not constrained to the same system, a Docker client can be linked to a remote Docker daemon.

2.1.1. Docker images

An increasingly adopted method is the distribution of bioinformatics tools as Docker images, bundling intricate configurations, libraries, and settings for a simplified and reproducible deployment experience (Kwon, et al. 2018). An image serves as a template that is read-only and contains instructions for the creation of a Docker container. PipeCraft2 mostly uses images that are built on top of a Ubuntu base image, though the exact version of the Ubuntu base image might differ based on the added bioinformatics software. Building custom images involves the use of a Dockerfile, a file that encompasses vital instructions for image creation. These directives detail the installation of bioinformatics tools and their corresponding dependencies. The Dockerfiles for all PipeCraft2 images are hosted on our GitHub repository (www.github.com/pipecraft2/pipecraft). Images themselves are hosted on Docker Hub, a container registry designed for developers and open source contributors, offering a platform where they can discover, use, and share their assets. PipeCraft2 is a part of the Docker-Sponsored Open Source Program (DSOS), images that are part of this program have a special badge on Docker Hub making it easier for users to identify projects that Docker has verified as trusted, secure, and active open-source projects. The Docker-Sponsored Open Source Program provides several features and benefits such as repository logo, verified Docker-Sponsored Open Source badge, additional insights and analytics, access to Docker Scout, removal of download rate limits and improved discoverability on Docker Hub (Docker Incorporated 2024). Currently PipeCraft2 maintains a repository of 22 images on Docker Hub. PipeCraft2's communication with the registry is automated, eliminating the need for manual image downloads. Users can also independently execute all images to use them as standalone tools via Docker CLI or PipeCraft2's Expert panel, allowing for the in-depth

deployment of the embedded bioinformatic software. PipeCraft2's images are widely utilized within the community amassing to about 2200 downloads from 1600 unique IPs during the 2023 calendar year.

2.1.2. Docker containers

A Docker container is like a living version of a Docker image. It's a practical and self-contained space where the application and all its dependencies run together. Containers work on their own yet share the underlying kernel of the host operating system, enabling consistent application execution experience across diverse environments. Containers can start, stop or be deleted easily via Docker CLI or Docker application programmatic interface (API); which, however, is done automatically when running the bioinformatics via PipeCraft2 platform. Each container has its own space for files, network, and processes, so it stays isolated from other containers and the main computer system (Docker Incorporated 2024). PipeCraft2 utilizes a Node.js library called Dockerode to programmatically interact with Docker's Remote API and manage containers and services. The Dockerode module is the primary means of communication between PipeCraft2 and Docker and enables any Javascript component effortless interactions with Dockers containers, images and registry (Dias 2024).

2.2. Vue.js

Vue.js is a library for building interactive web interfaces. Vue.js holds the third position in popularity among modern web development frameworks, trailing only behind React and Angular. This is evident in various metrics, including Google search statistics, Stack Overflow queries and surveys, Twitter and Reddit followers, and GitHub usage statistics (Krotoff 2023). Additional evidence of its widespread usage is showcased by its implementation in major corporations such as Alibaba, Xiaomi, Adobe, Nintendo, and other notable companies. At its core, Vue.js distinguishes itself through a robust reactivity system and a sophisticated data-binding mechanism. This architecture ensures efficient updates to the Document Object Model (DOM) when underlying data undergoes transformation. Such reactivity facilitates a responsive and dynamic user experience. Vue.js embraces a component-based architecture, mirroring the contemporary trend in web development. This modular structure not only enhances code maintainability but also facilitates the creation of reusable, self-contained

components. Vue also provides a set of built-in directives which are special tokens in the markup such as v-if, v-for, v-bind, v-model, v-on and others, these directives are powerful tools for manipulating the DOM and are extensively used in PipeCraft2's front-end components. Vue also features a diverse array of tools, libraries and resources, collectively known as the Vue ecosystem designed to complement the frameworks and enhance its capabilities. PipeCraft2 incorporates several notable tools within the Vue ecosystem, including Vuex, Vue Router, Vue CLI Plugin Electron Builder and Vuetify.

2.2.1. Vuex

Vuex is a state management library specifically designed for Vue.js applications. It serves as a centralized store for managing the state of a Vue application, making it easier to manage and maintain the state throughout the components of an application. Vuex follows the Flux architecture pattern (an architecture for creating data layers in JavaScript applications, it was designed by developers at Facebook) , providing a predictable and centralized approach to managing application state. The centralized state object that holds the entire state of the application. This state is reactive, meaning that any changes to it automatically trigger updates to components that rely on it. There are multiple ways to trigger changes in the state object, such as Getters, Mutations and Actions. Getters are functions in Vuex that allow you to compute derived state based on the current state. They are useful for encapsulating reusable logic and accessing computed properties in a consistent manner. PipeCraft2 employs a variety of Getters, with functions that involve overseeing pipelines based on input data and validating essential and interdependent inputs. Mutations are functions responsible for modifying the state. They ensure that changes to the state are explicit and trackable. Mutations must be synchronous to maintain predictability. Mutations handle the majority of state modifications in PipeCraft2. These span from basic adjustments, like initiating loading bars, to substantial transformations in the state, such as altering the entire workflow logic based on the attributes of the input data. Actions are similar to mutations, but they can be asynchronous. Actions are used to perform operations, such as fetching data, and then commit mutations to update the state. While the current version of PipeCraft2 does not utilize any Actions, their significance will become apparent in potential future developments, particularly in facilitating communication with high-performance computing (HPC) clusters.

2.2.3. Vue Router

Vue Router is the official routing library for Vue.js. It provides a way to manage the navigation in your Vue applications by allowing you to define routes and handle navigation between different views or components. While operating as a desktop application, PipeCraft2 doesn't fully exploit the capabilities of the router. Nevertheless, certain essential ideas like Dynamic Route Matching and Programmatic Navigation are effectively employed within the application. Given the absence of an address bar, all navigation within the app must be handled programmatically. This entails linking routing buttons and events to the router. The implementation of Dynamic Route Matching proves particularly beneficial, enabling active segments in routes. This routing approach is extensively applied to Quick Tools and pre-defined pipelines, contributing to enhanced code efficiency and readability.

2.2.4. Vuetify

Vuetify is a prominent user interface (UI) framework for Vue.js, recognized for its comprehensive set of material design components and seamless integration with Vue applications. Developed with a focus on both functionality and aesthetics, Vuetify facilitates the creation of visually appealing and responsive user interfaces. This framework adheres to the principles of material design, a design language pioneered by Google, emphasizing a consistent and intuitive user experience across diverse platforms. The framework additionally provides a grid system for a flexible and responsive layout structure, empowering developers to create adaptive designs suitable for various screen sizes. The majority of PipeCraft's front-end components rely on pre-built elements from Vuetify, albeit often subject to extensive customization.

2.3. Electron

Electron (www.electronjs.org) is a free, open-source software framework maintained and developed by Github (www.github.com). Electron framework incorporates the Chromium rendering engine and the Node.js runtime, which provides the means to build cross-platform desktop applications with web technologies such as hypertext markup language (HTML), cascading style sheets (CSS) and Javascript. A multitude of widely popular applications have

been developed using the electron framework, such as WhatsApp, Microsoft Teams, Visual Studio Code and others. The compatibility to run and build an application on Linux, Mac and Windows made the Electron framework an easy choice for the development of PipeCraft, henceforth PipeCraft2. PipeCraft2 makes use of Electrons features such as interacting with operating systems, native menus, file system and notifications. The active development branch of PipeCraft2 supports auto updating which is another wonderful Electron feature and will be certainly included in the next stable release. PipeCraft2 combines the Electron and Vue.js frameworks with the The Vue CLI Plugin Electron Builder. The plugin is designed to simplify the process of building and packaging Vue.js applications for desktop platforms via Electron. It offers a streamlined development process and robust build and packaging capabilities and is ready for use in a production environment (Klayman 2018). Currently PipeCraft2 is using Electron version 13.0.0 and Vue CLI Plugin Electron Builder version 2.1.1.

2.4. Git

Git is a distributed version control system, adept at monitoring alterations within a designated collection of computer files. It is commonly employed to facilitate collaborative efforts among programmers engaged in the joint development of source code during the software development process (Linus Torvalds 2007). Given the involvement of multiple developers in the PipeCraft2 project, the necessity for a version control tool to ensure seamless and efficient collaboration became apparent. The natural choice was Git, recognized as the foremost tool in its category. The incorporation of essential features, including branching, merging, and history tracking, has significantly streamlined the development process of PipeCraft2, leading to substantial time savings.

2.5. Github and Read the Docs

Github is the go-to platform for hosting Git repositories used by many individual developers, open-source contributors, and academic researchers alike. GitHub allows developers to host Git repositories remotely, providing a central location for code collaboration. Given PipeCraft2's open-source nature it was essential to provide public access to our source code through a web-based platform such as Github (github.com/pipecraft2/pipecraft). In addition to source code hosting, Github offers further useful features such as issue tracking,

dependency vulnerability alerts, automated code scanning and pull requests. These features establish means for our less experienced users to effortlessly seek guidance in case of issues, report bugs, and enable more proficient users to propose new features, recommend modifications to existing code, or actively contribute by writing and submitting code through pull requests.

Documentation is an integral part of the software development process that enhances user experience, facilitates collaboration, and ensures the long-term success and sustainability of the software. The user guide for PipeCraft2 is also hosted on the github repository (https://github.com/pipecraft2/user_guide) but distributed through Read the Docs (pipecraft2-manual.readthedocs.io). Read the Docs (www.readthedocs.com) is an open-source documentation hosting platform that has gained widespread popularity. Its emphasis on automation, collaboration, and user-friendly design made it a preferred choice for PipeCraft2.

3. Design Principles and Features

The development of PipeCraft2 involved critical design decisions to ensure usability, flexibility, and cross-platform compatibility. The front-end of PipeCraft2 employs modern JavaScript frameworks to achieve a high level of customization and reactivity in user interactions to ensure that users, regardless of their expertise, can navigate the interface seamlessly. The back-end architecture focuses on automation and accommodation of a diverse set of bioinformatic tools while at the same time minimizing dependencies for end-users; this is achieved via containerization with Docker. To ensure cross-platform compatibility, PipeCraft2 harnesses the capabilities of Docker and the Electron framework. Docker efficiently oversees the management of back-end assets, while Electron facilitates seamless portability across different operating systems (including Mac, Linux, and Windows). To encourage collaboration and to retain transparency, PipeCraft2 is released as an open source software package. These fundamental building blocks lay the groundwork for key functionalities within PipeCraft2, that currently comprises four predefined full pipelines, eleven Quick Tools panels offering a wide range of tools, an expert mode panel designed for

advanced users, a quality check module and the save and load capability. All of those features and design principles are outlined and described in the following subsections.

3.1. Layout

PipeCraft2 is a single-page application wherein the user interface undergoes dynamic rerendering within a single window (Figure 1). The user interface encompasses two persistent navigation panels flanking a central display window. This central display window maintains a connection to the Vue Router, facilitating real-time updates in response to user-initiated navigation inputs.

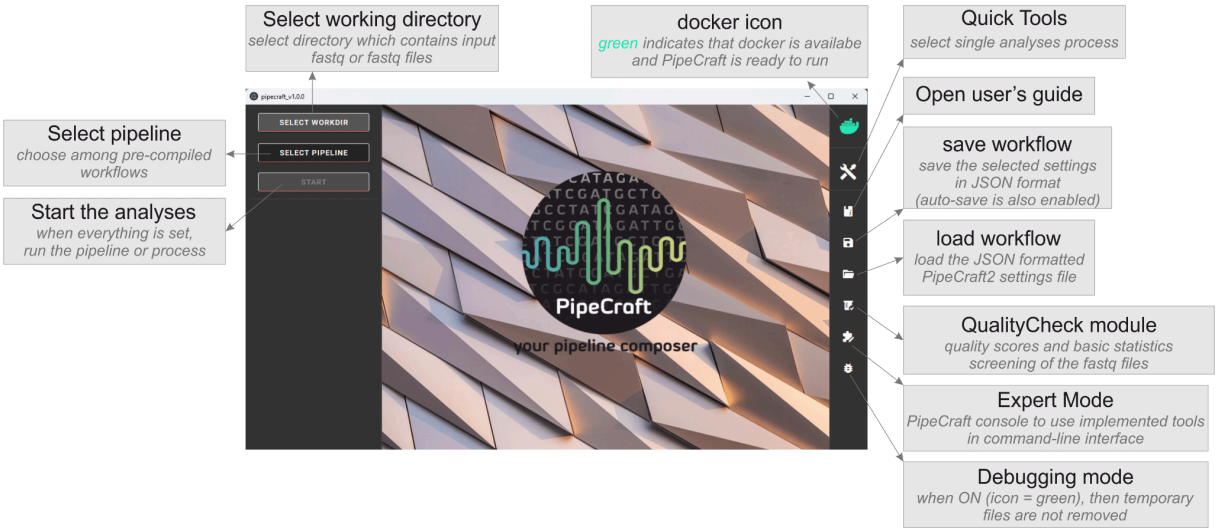

Figure 1. Layout of the PipeCraft2 (release v1.0.0) graphical user interface.

The right navigation panel displays icons as shortcuts to features such as; Quick Tools, the manual, workflow saving and loading, quality check, expert mode, and a debugger (Figure 1). At its upper section, it displays a Docker icon, appearing either in red or green, symbolizing the operational status of Docker and whether it is prepared for deployment with PipeCraft2. Icons associated with navigational functions will be adorned in green hues to signify their activation. This also extends to the debugger icon, reflecting its on or off status. The left navigation panel hosts three fixed buttons: one for selecting input data and annotating its properties, a second button for choosing from preconfigured pipelines, and a third for initiating or halting workflows. The lower section of the panel dynamically presents any selected feature from the Quick Tools menu. Both panels incorporate tooltips for all buttons.

3.2. Selecting data

Prior to executing a workflow, users are required to specify the location (data path) of their sequencing data. This is accomplished by clicking on the "select workdir" button (Figure 1), which prompts two inquiries regarding the data's nature, such as whether it is paired-end (such as data from Illumina sequencing platforms) or single-end (such as data from Pacific Biosciences sequencing platforms) and the extension of your sequence files (Figure 2; supported extensions include: fastq, fasta, fq, fa, txt, fastq.gz, fasta.gz, fq.gz, fa.gz and txt.gz).

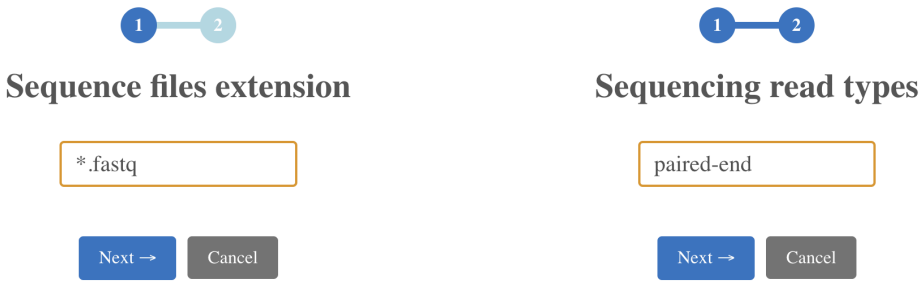

Figure 2. Prompts for selecting input data.

Based on the user-specified characteristics of the data, PipeCraft2 may automatically exclude or disable certain pipeline steps (e.g., removing and disabling assembling pairs if the user selects single-end data) and fine-tune certain back-end setups, as numerous processes deploy distinct scripts dependent on the attributes of the input data. Subsequent to completing the prompts, a file system window emerges, where users must specify the folder containing their samples. The working directory selection button will display a green indicator line upon completion, and hovering over the selection button reveals a tooltip containing all the previously provided information.

3.3. Custom front-end elements

As PipeCraft2 incorporates various bioinformatic tools, it necessitates a method to visually depict the diverse array of parameters associated with the functionality of these tools. Hence, PipeCraft2 includes ten distinct visual components for input parameters, encompassing input fields like numeric, boolean, select, file, link, chip, combo, slide, boolfile, and boolselect (Figure 3). Each of these interface components adheres to a consistent design

logic, presenting a compact card with a header, a tooltip, and an input field positioned at the center of the card. From a coding perspective, these input elements can be categorized into two parts: the Vue.js component responsible for rendering them and a minimal JavaScript object stored in the Vuex state. The template object used for rendering an input element includes fundamental attributes like name, value, tooltip, and input type. Additionally, it incorporates dynamic extra attributes such as disabled (based on the attributes of the input data), rules (for controlling the allowed range or quantity of inputs), and depends on (establishing interdependencies between inputs, where marking one may require the use of another).

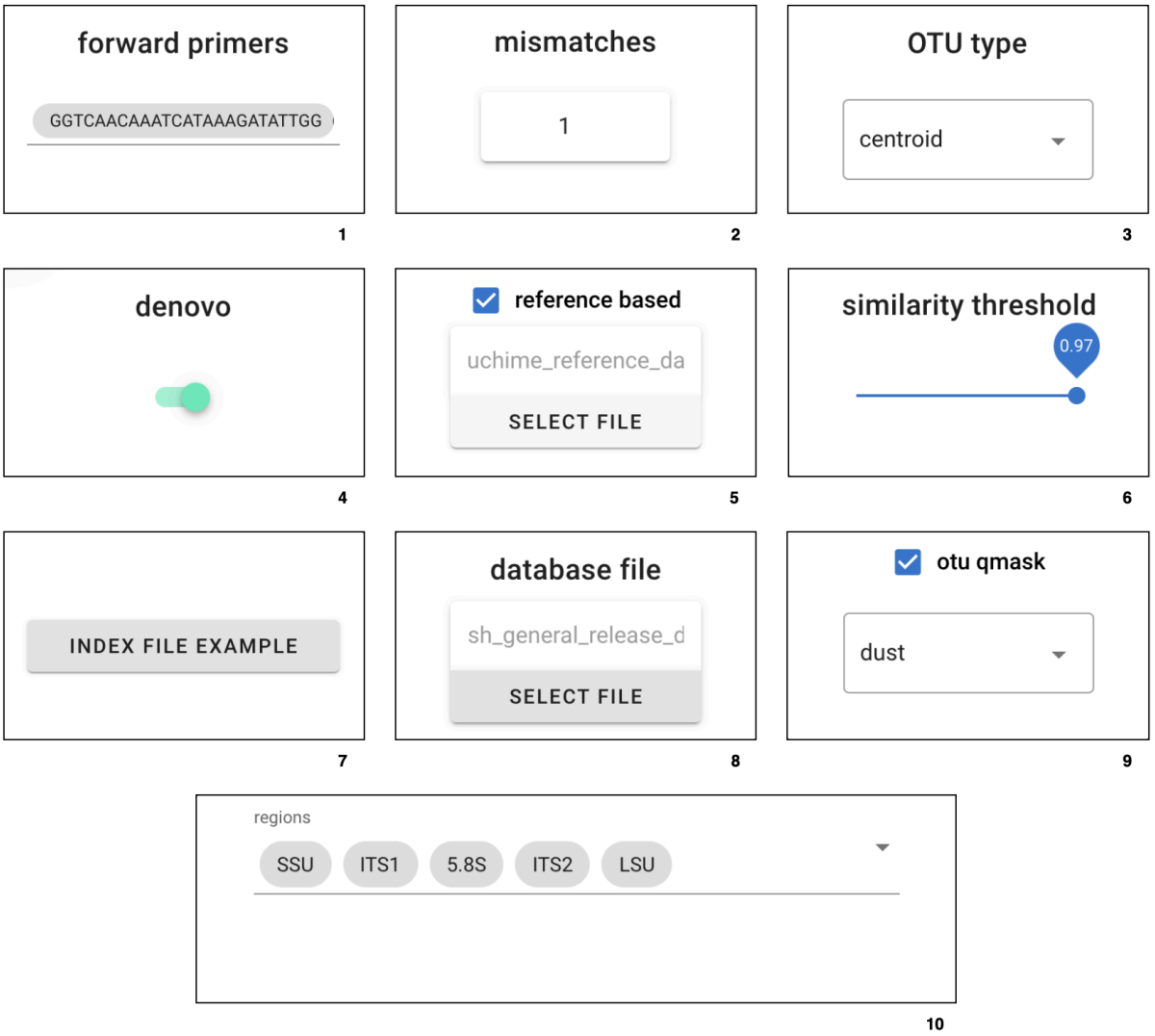

Figure 3. Examples of customized input fields. Starting from the upper left corner: (1) chip, (2) numeric, (3) select, (4) bool, (5) boolfile, (6) slide, (7) link, (8) file, (9) boolselect, (10) combo.

3.4. Workflow execution

To run any of the implemented processes in PipeCraft2, the user needs to ensure that Docker is running (this, however is also automatically checked by PipeCraft2), specify the location of their input data, select a tool or pipeline, or load a workflow, and fill out the mandatory parameter fields. Failure to meet these conditions will result in the *Start* button being disabled, accompanied by tooltips indicating which requirements are not fulfilled. Upon pressing the *Start* button, PipeCraft2 undergoes a comprehensive setup process (Figure 4). The current configuration is automatically saved for later access from the working directory. Simultaneously, PipeCraft2 starts tracking the execution time of the process. If the debugger is active, it establishes a stream to a text file to capture all logs during the run. A Docker properties file is generated, encompassing the folders and files to be passed on to the container, including the working directory, databases, primer or index files. User-defined parameters are formatted as environmental variables and a custom hostname is set for the container based on the process being executed. To avoid execution errors stemming from multiple containers having the same name PipeCraft2 automatically scans all existing containers and removes any with the same name hostname as marked in the current process. Subsequently, it verifies the presence of the required image for the process on the user's computer; if absent, the image is automatically pulled from DockerHub. The pulling process is displayed on the front-end through a dedicated loading bar. After this comprehensive setup, PipeCraft2 is prepared to run a Docker container, passing the Docker properties and environmental variables objects to the run command. The run command also encompasses a script to execute, each process is associated with a designated script. While the tool is executing, PipeCraft2 monitors for errors and different status codes on container exit. Additionally, PipeCraft2 scans container logs for predefined sections indicating changes in file format or read types. Upon container exit, PipeCraft2 removes the container and presents the user with a dialog containing either errors encountered during execution or a success message, marking the conclusion of the process. During the execution of a pipeline, PipeCraft2 iteratively undergoes this process in a loop for each step. After each step, it adjusts file attributes and the working directory. The success message is prompted only after the completion of all steps.

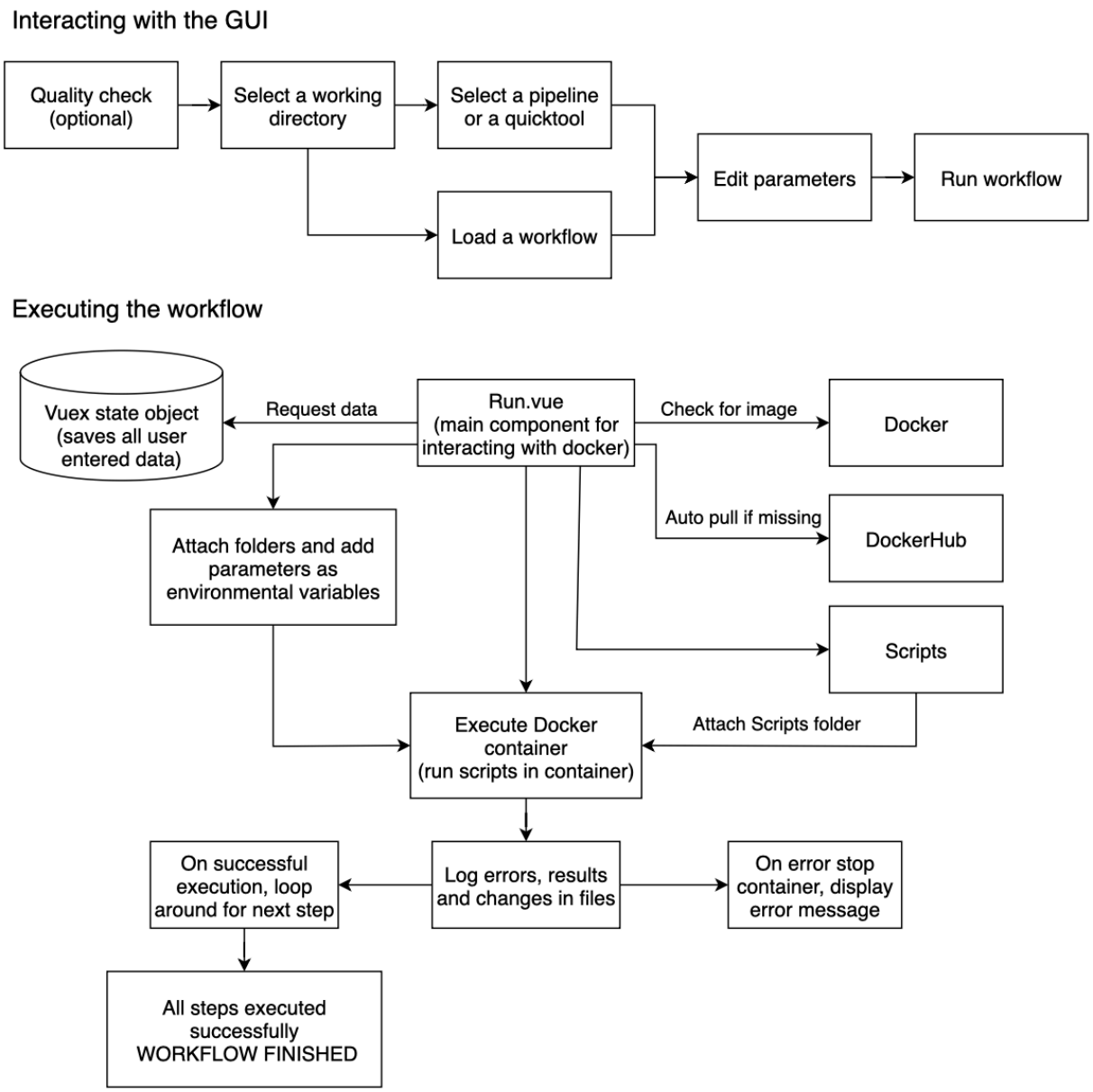

Figure 4. A flowchart describing the complex orchestration of execution of a process/pipeline.

3.5. Save and Load

PipeCraft2 enables the user to save and load customized processes configurations (Figure 5). This fundamental aspect of PipeCraft2's design is aimed towards enhancing the reproducibility of bioinformatic processes. The configurations for both, pre-defined pipelines and Quick Tools, exist as plain JavaScript objects within the Vuex state object. The system empowers users to capture a snapshot of these configurations, saving them as JSON files through the "*Save Workflow*" button. Subsequently, these configuration files can be reloaded through the "*Load Workflow*" button. Additionally, PipeCraft2 automatically saves a

configuration file in the working directory upon executing a process. However, it is imperative to note that compatibility issues may arise when attempting to use configuration files across different versions of PipeCraft2, which may have slight differences in the front-end components due to updates of the implemented bioinformatics tools.

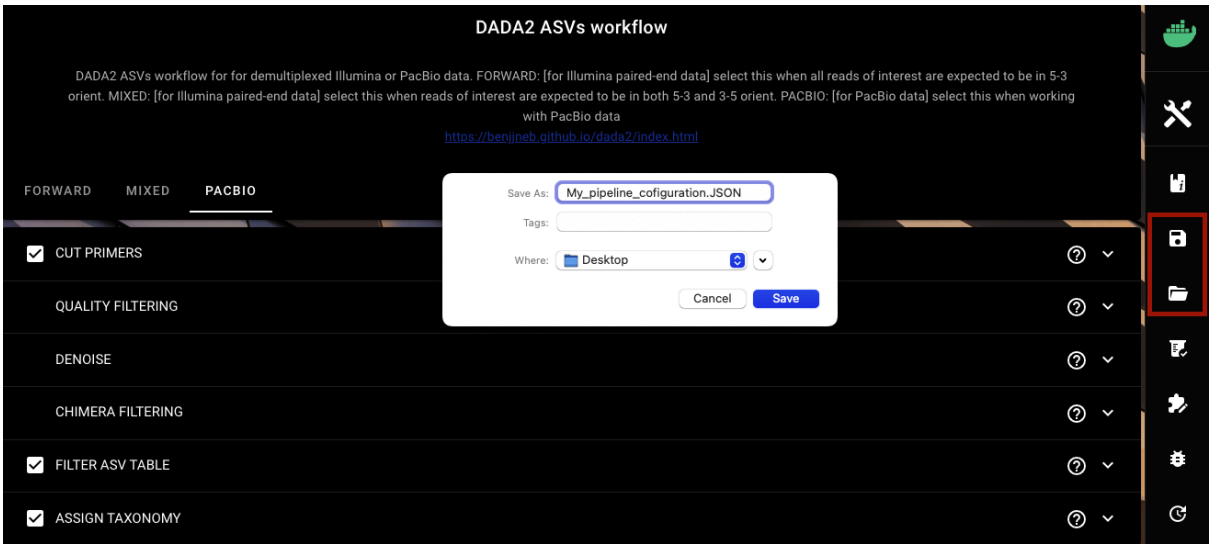

Figure 5. The save and load functionality demonstrated on the GUI.

3.6. Expert Mode

Docker images deployed by PipeCraft2 serve as standalone assets, representing containerized versions of the implemented bioinformatic tools without the need for complex installation procedures or additional dependencies. An Expert Mode panel was developed to facilitate the utilization of PipeCraft2 Docker images as independent resources. This panel comprises a live terminal window along with instructions and tooltips guiding users on how to interact with PipeCraft2's ecosystem of bioinformatic containers via the command line. The core idea involves running a Docker image in interactive mode within PipeCraft2's terminal to allow easy access to Linux-based bioinformatic tools on Windows computers..

Figure 6. The Expert Mode terminal in PipeCraft2.

3.7. Implemented third-party bioinformatic tools

The inclusion of a diverse array of bioinformatic tools (Table 1) stands out as a key feature in the design of PipeCraft2. Tools listed in Table 1 are accessible through the Quick Tools panel, but also integrated into the pre-defined pipelines (section 3.9). The utilization of Docker for containerization has facilitated the seamless integration of numerous third-party bioinformatic tools. A substantial aspect of PipeCraft2's back-end development can be conceptualized as constructing a framework to accommodate various third-party bioinformatic software. While it is acknowledged that PipeCraft2 would lack substance without these tools and their developers, it is crucial to recognize the considerable development effort invested in the automation of these tools. This involves tasks such as building and testing Docker images, automating their deployment, scripting to automate tool usage within containers, and simultaneous communication with the front end.

Table 1. Integrated third-party bioinformatic tools.

Analysis step	Program	Version	Webpage	Reference
Quality check	FastQC	0.11.9	github.com/s-andrews/FastQC	Andrews 2010
	MultiQC	1.1	multiqc.info	Ewels et al. 2016
Demultiplexing	cutadapt	4.4	cutadapt.readthedocs.io	Martin 2011
Cut adapters primers	cutadapt	4.4	cutadapt.readthedocs.io	Martin 2011
Assemble paired-end sequences	vsearch	2.23	github.com/torognes/vsearch	Rognes et al. 2016
	DADA2	1.28	benjjneb.github.io/dada2	Callahan et al. 2016
Quality filtering	vsearch	2.23	github.com/torognes/vsearch	Rognes et al. 2016
	Fastp	0.23.2	github.com/OpenGene/fastp	Chen et al. 2018
	trimmomatic	0.39	github.com/usadellab/Trimmomatic	Bolger, Lohse, and Usadel 2014
	DADA2	1.28	benjjneb.github.io/dada2	Callahan et al. 2016
Chimera filtering	vsearch	2.23	github.com/torognes/vsearch	Rognes et al. 2016
	DADA2	1.28	benjjneb.github.io/dada2	Callahan et al. 2016
Gene extraction	ITS extractor	1.1.3	microbiology.se/software/itsx	Bengtsson-Palme et al. 2013
OTU/ASV formation	vsearch	2.23	github.com/torognes/vsearch	Rognes et al. 2016
	swarm	3.1.3	github.com/torognes/swarm	Mahé et al. 2014
	DADA2	1.28	benjjneb.github.io/dada2	Callahan et al. 2016
Taxonomy assignment	BLAST+	2.14	blast.ncbi.nlm.nih.gov/Blast.cgi	Camacho et al. 2009
	RDP classifier	2.13	github.com/terrimporter/MetaWorks	Porter and Hajibabaei 2022
	DADA2 classifier	1.28	benjjneb.github.io/dada2	Callahan et al. 2016
Post-processing tools	lulu	8.3	github.com/tobiasgf/lulu	Frøslev et al. 2017
	DEICODE	0.2.4	github.com/biocore/DEICODE	Martino et al. 2019
	ORFfinder	0.4.3	ncbi.nlm.nih.gov/orffinder	Rombel et al. 2002
	HMMER	3.3.2	hmmer.org	Eddy 1998

3.8. Quality check

The assessment of the quality scores associated with the sequencing data can be examined through the *QualityCheck* panel (Figure 7). For that, the QualityCheck panel employs FastQC (Andrews 2010) and MultiQC (Ewels et al. 2016). FastQC produces individual quality reports for each sample, and MultiQC aggregates and summarizes these reports into an HTML-formatted document. The user is required to select the folder containing their input data, which must be in fastq format (compressed input data is also supported). Subsequently, the user initiates the report generation process by clicking the "*create report*" button. Once this process is completed, the user can access the HTML-formatted report by clicking the "view report" button, which opens the report in their default web browser.

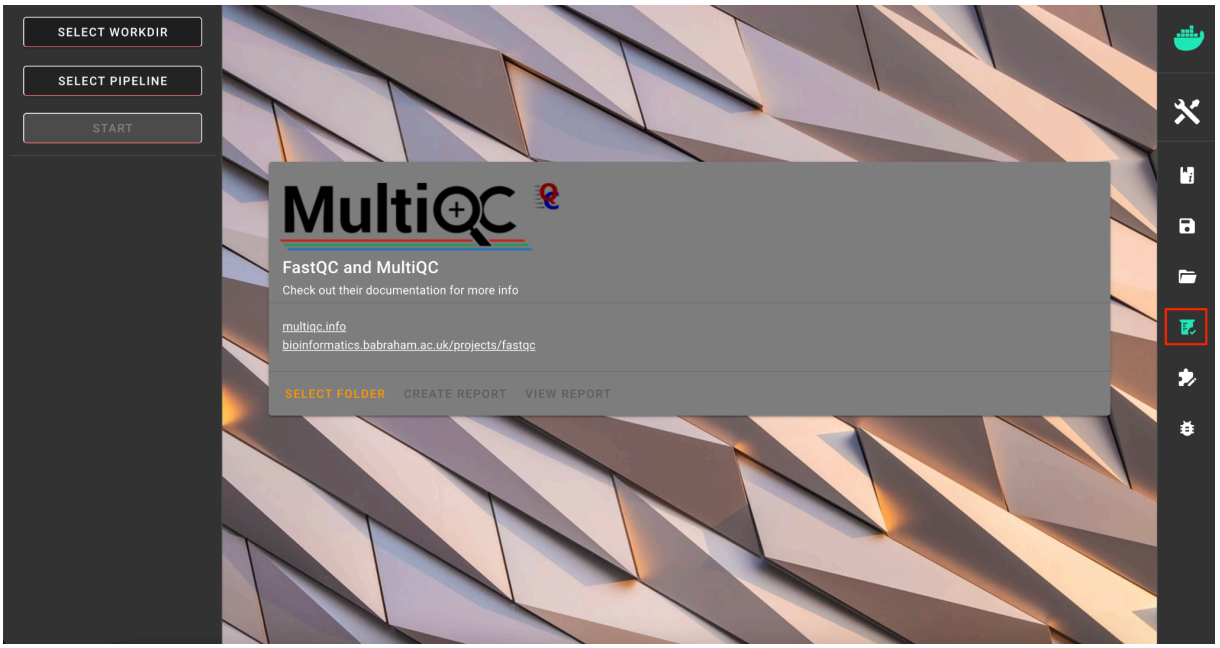

Figure 7. The quality check panel, accessible from the right navigation panel.

3.9. Built-in pre-defined pipelines and Quick Tools

Preceding taxonomic identification, the data acquired from metabarcoding experiments undergoes multiple processing stages (Figure 8). A sequence analysis pipeline is constructed through the sequential application of diverse software tools and algorithms, aiming to

generate a precise features table accompanied by taxonomic annotations per sample (Hakimzadeh et al. 2023). PipeCraft2 currently implements four predefined pipelines which produce operational taxonomic units (OTUs) or amplicon sequence variants (ASVs) that are subjected to taxonomy assignment (Figure 8). However, they exhibit variations in the sequence of operations and diverge based on the specific algorithms employed. These pipelines are entirely automated and, once configured by the user, will execute all steps consecutively, ceasing solely upon encountering errors during execution.

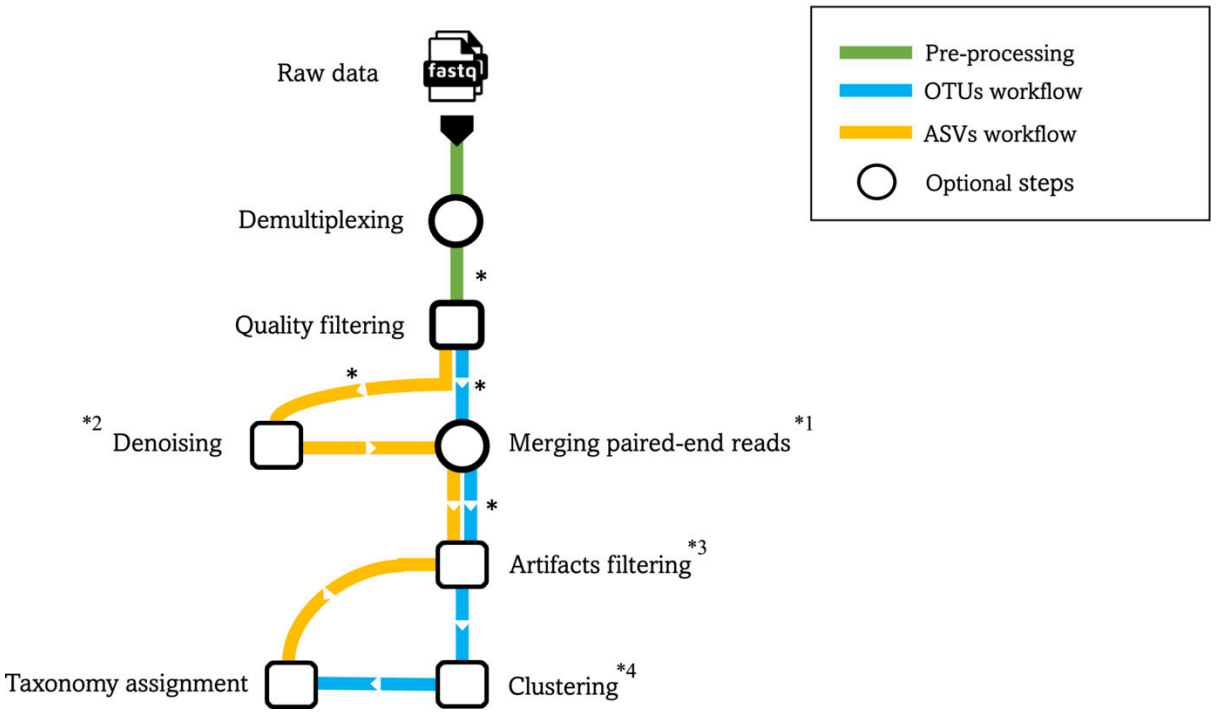

Figure 8. Examples of basic bioinformatics workflows for metabarcoding data (figure from Hakimzadeh et al. 2023).

The individual components of the full pipeline (as listed in Table 1) are accessible via Quick Tools (Figure 9). The Quick Tools menu provides eleven distinct panels, each dedicated to executing a specific bioinformatic process, such as quality filtering, assembling paired-end reads, clustering, and more. Several panels offer a variety of tools tailored to accomplish specific bioinformatic tasks, allowing users to select the tool most suitable for their data and research objectives. This feature enables users to further customize their workflows compared to predefined pipelines, albeit requiring manual execution of processes one by one.

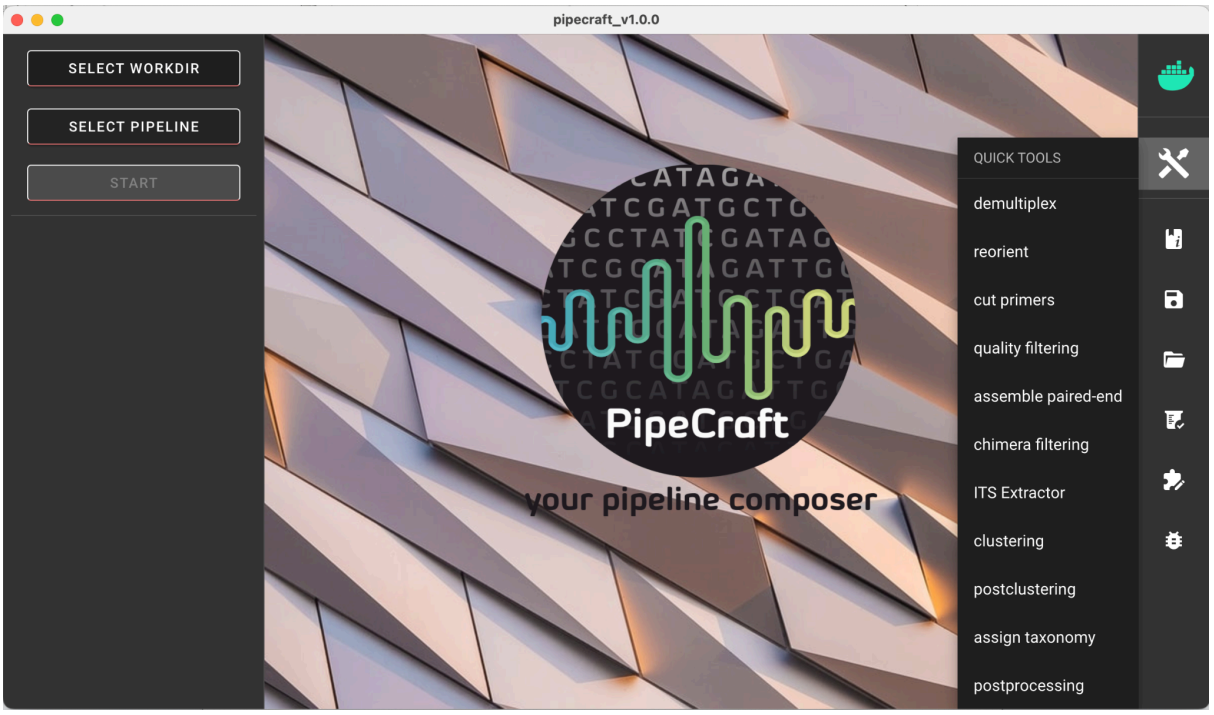

Figure 9. The Quick Tools menu, accessible on the right menu ribbon.

3.9.1. Demultiplexing

Sample multiplexing is a prevalent technique in high-throughput sequencing experiments, involving marking sample libraries with unique index sequences and pooling them together for increased sequencing throughput and cost-effectiveness. Subsequently, the pooled sequences undergo a downstream bioinformatic process known as demultiplexing, which allocates the sequences to their respective samples. Although sequencing centers often demultiplex data, this is not always the case. Consequently, PipeCraft2 offers a dedicated Quick Tool for the demultiplexing process. The demultiplexing process is carried out by cutadapt and relies on a user specified indexes file, which includes molecular identifier sequences (so called indexes/tags/barcodes) per sample. The demultiplexing procedure also searches for reverse complementary matches. Supported data formats include fastq/fasta for both paired-end and single-end reads. The output directory demultiplexed_out is going to include fastq/fasta files per sample and an unknown.fastq file containing sequences where the specified index combinations were not found. The sequences have their indexes truncated, and paired-end samples are assigned .R1 and .R2 read identifiers.

3.9.2. Reorient

Depending on the amplicon library preparation steps, the raw sequencing datasets may contain both 5’-3’ and 3’-5’ oriented sequences. If the data includes the PCR primers, the reorient Quick Tool can perform sequence reorientation based on user-specified primers. For paired-end data, R1 files are reoriented to 5’-3’, while R2 reads are reoriented to 3’-5’ for merging. Sequences with multiple forward or reverse primers are discarded as potential chimeric sequences (or so called multi-primer artifacts). During reorientation, primers are not truncated from the reoriented sequences. For single-end inputs, both fastq and fasta formats are supported, however, for paired-end reads, the only supported format is fastq. The resulting output files are in either fastq or fasta format and are stored in a directory named "reoriented_out".

3.9.3. Cut primers

For optimal OTU or ASV generation, it is recommended to truncate primers from the reads. The Quick Tools menu offers this feature where PCR primers (and/or adapters) can be eliminated from the input data through a process largely dependent on cutadapt. By default, sequences lacking specified PCR primer strings are discarded (but stored in the 'untrimmed' directory). A reverse complementary search of the primers in the sequences is also conducted, resulting in the clipping of primers from both 5’-3’ and 3’-5’ oriented reads. However, for paired-end reads, reorientation to 5’-3’ does not occur during this process; only single-end reads are reoriented to 5’-3’. For paired-end data, the ‘seqs_to_keep’ option is recommended to be left as default ('keep_all'), outputting sequences where at least one primer has been clipped. Alternatively, the 'keep_only_linked' option outputs sequences where both the forward and reverse primers are found, and it may be employed for single-end data to retain only full-length amplicons. Supported formats include fastq/fasta for both paired-end and single-end data, and the outputs are stored in the ‘primersCut_out’ directory with primers truncated from the sequences.

3.9.4. Quality filtering

As erroneous reads lead to an overestimation of microbial diversity (Bokulich et al. 2013) the quality filtering process is a pivotal process in metabarcoding data analysis; it aims to remove reads containing sequencing errors, thereby enhancing the reliability of downstream analyses. These errors are quantified through sequencing quality scores, which measure the probability of base-calling inaccuracies. Common approaches include: exclusion of reads with scores below a predefined threshold, trimming of read ends based on quality, and exclusion based on the number or rate of expected errors (Creedy, Vogler, and Penlington 2020). The Quick Tools panel for quality filtering offers four different tools for quality filtering; vsearch, trimmomatic, fastp and DADA2. The core filtering concept of Trimmomatic and fastp utilize a sliding windows approach, wherein a user-defined window size is evaluated, and the read is truncated once the mean quality falls below the user-specified threshold. In contrast, vsearch and DADA2 measure the maximum number of expected errors per sequence or per base to discard reads. Additionally, these tools offer extended configuration options including filters for read length, the presence of ambiguous bases and various other parameters. Notably, Trimmomatic offers distinctive functionalities for trimming read beginnings and ends, while fastp facilitates poly tails trimming. Selecting the appropriate tool will depend on the characteristics of the input data and the objectives of the experiment. Supported input data includes single-end and paired-end reads in fastq format. Output fastq files are written into a directory called ‘qualFiltered_out’.

3.9.5. Paired-end assembly

Paired-end sequencing approach in metabarcoding experiments offers extended read lengths in comparison to single-end sequencing in second generation sequencing platforms (such as from Illumina and MGI-Tech). This enhances taxonomic resolution, providing a more comprehensive coverage of the targeted genomic region and, consequently, enables improved species identification. However paired-end data necessitates assembly in a subsequent bioinformatic process. The Quick Tools menu currently offers two tools for assembling paired-end data: vsearch and DADA2. Both tools feature crucial input parameters such as minimum overlap between the reads and the allowed number of mismatches in the overlap. The vsearch protocol incorporates an additional integrated module, denoted by the

"include_only_R1" option, that incorporates all R1 reads from unassembled pairs into the final output. This option proves significant when dealing with for example ITS2 amplicons, as in certain taxa, the region's length hinders proper assembly, and without this option, the reads would be discarded. DADA2 conducts denoising before assembly, necessitating the input data to be devoid of ambiguous nucleotides. Both assembly tools accommodate paired-end fastq formatted data, and the resulting outputs are stored in the ‘assembled_out’ directory.

3.9.6. Chimera filtering

Chimeras are hybrid artifacts resulting from the fusion of multiple parental sequences, they pose a risk of being misinterpreted as distinct biological organisms, thereby artificially inflating diversity estimates (Haas et al. 2011). These artifacts arise when segments of one sequence are erroneously combined with segments from different organisms during PCR amplification (Edgar et al. 2011; Schloss et al. 2011). Studies employing markers within rRNA genes and internal transcribed spacer (ITS) regions commonly report chimeric read frequencies ranging from 1% to 15% (García de León et al. 2018; Nilsson et al. 2010; Schloss et al. 2011). The Quick Tools menu offers two algorithms for removing chimeric sequences ‘uchime denovo’ and ‘uchime3 denovo’ both provided by the vsearch tool. Both panels offer an input to also apply reference based chimera filtering with the uchime_ref algorithm that will be executed after denovo filtering. Supported input data formats include fastq and fasta. The resulting fasta files are stored in a directory labeled as "chimera_Filtered_out".

3.9.7. ITS Extractor

The nuclear ribosomal internal transcribed spacer (ITS) region serves as the primary marker gene for molecular identification of fungi; it consists of two highly variable spacers (ITS1 and ITS2) predominantly species-specific, intervened by a conserved 5.8S gene. Using either of the variable spacers enhances sequence clustering and taxonomy annotation (Bengtsson-Palme et al., 2013). PipeCraft2 incorporates the ITSx software tool, streamlining the identification and extraction of ITS1 and ITS2 from sequencing data across fungi and nineteen other eukaryotic groups. ITSx facilitates targeted extraction of the variable segments of the ITS region while distinguishing sequences not originating from the ITS

regions. This functionality holds particular relevance in high-throughput sequencing applications, where amplicon-based runs often contain non-target sequences that are difficult to distinguish (Quince et al. 2011). Misinterpreting non-relevant sequences as target sequences may lead to inflated diversity assessments (Dickie, 2010; Tedersoo et al. 2010). The ITSx panel, available within the Quick Tools menu, enables the users to specify the organism groups for searching, determine the regions to be included in the output results, and configure other relevant settings. Supported input data formats include single-end fastq and fasta. The resulting fasta files are stored in a directory labeled as "ITSx_out".

3.9.8. Clustering

Clustering sequences into OTUs is a widely adopted approach in microbial ecology for analyzing gene sequence datasets, providing functional estimates for potential microbial species, though modern approaches are shifting towards the preference for ASVs (Xia and Sun, 2023). The clustering panel in the Quick Tools menu offers users the choice between employing the unoise3 algorithm for generating ASVs (or zOTUs as referenced by the developer of the algorithm (Edgar 2016)) or conducting sequence clustering into OTUs, with both methods powered by the vsearch tool. Both panels offer a wide range of user customizable parameters. The supported file format for input data is fasta. Output files include OTUs.fasta, OTU_table.txt, and OTUs.uc, all located within the ‘clustering_out’ directory.

3.9.9. Post-clustering

PipeCraft2 includes a post-clustering panel that enables users to perform two distinct operations. The first operation involves utilizing LULU, a method designed to eliminate erroneous molecular OTUs. This is achieved by analyzing the co-occurrence patterns of OTUs across samples, identifying those that consistently meet user-defined criteria for being errors of more abundant OTUs, and subsequently merging them (Frøslev et al., 2017). The LULU curation algorithm requires a tab-delimited OTU table and OTU sequences in fasta format as input data, and the resulting outputs will be stored in a directory labeled "lulu_out". The

second operation entails utilizing the DADA2 collapseNoMismatch function to collapse a DADA2 ASV table. This function identifies sequences that exhibit no mismatches or internal indels upon alignment and condenses them into a single representative sequence, with the most abundant sequence serving as the representative (Callahan et al. 2016). As input the user needs to supply an ASV table in RDS format, which is produced during a DADA2 pipeline in PipeCraft2. The resulting files will be saved in a directory named 'filtered_table'.

3.9.10. Taxonomy assignment

Metabarcoding analyses rely on accurate taxonomic assignment to contextualize sequencing data within biological and ecological frameworks (Mugnai et al., 2023). The Quick Tools panel for taxonomic assignment presents three approaches: BLAST, which leverages local alignment to identify similarity regions between query sequences and database entries, and the RDP or DADA2 classifier, both of which employ the Naive Bayesian algorithm to allocate sequences to taxonomic groups based on k-mer frequencies. All of the classifiers require the user to specify the location of their reference database. Input data for all classifiers must be in fasta format, with resulting output files saved to the 'taxonomy_out' directory.

3.9.11. Post-processing

The post-processing panel from the Quick Tools menu offers a choice of four different operations; ASV to OTU, filter tag-jumps, filter NUMTs and DEICODE. ASV to OTU panel allows users to use vsearch clustering to cluster ASVs into OTUs and make an OTU table. It requires an ASVs fasta file and an ASV table as input. Tag-jumps represent significant concerns in high-throughput sequencing (HTS) data (Tedersoo et al. 2022). They have the potential to induce technical cross-contamination between samples, leading to potential distortions in the estimation of microbial community composition. The tag-jump filtering panel offers a solution to assess index-switches utilizing the UNCROSS2 algorithm (Edgar, 2018), which assigns a customized score to quantify the probability of tag-jump occurrences. The DEICODE software is designed for the execution of Robust Aitchison Principal Component Analysis (PCA) on sparse compositional omics datasets (Martino et al. in 2019). This tool establishes connections between specific features and beta-diversity ordination. The preprocessing steps involve applying the rCLR transformation (centered log-ratio

transformation on non-zero values without introducing pseudo counts). Subsequently, DEICODE performs dimensionality reduction using robust PCA, effectively handling sparse data through matrix completion. NUMTs (nuclear mitochondrial pseudogenes) are non-functional gene regions that result from the transposition of mitochondrial DNA (mtDNA) into the nuclear genome, however, their inclusion in DNA metabarcoding analyses may yield misleading outcomes (Porter and Hajibabaei 2021). To mitigate this, the Filtering NUMTs panel can be employed for open reading frame length filtering, either independently or in conjunction with hidden Markov model profile analysis. This approach effectively identifies and excludes spurious NUMTs from extensive datasets (Porter and Hajibabaei 2021).

4. Performance

Performance testing pipelines and tools implemented in PipeCraft2 is somewhat counter-intuitive as most algorithms and tools are imported from already well established software packages and have been tested by their authors. Although GUI and Docker have immense merits regarding user experience and development, they also represent some overhead costs in terms of performance compared to native execution. The GUI itself will use a couple of hundred megabytes of RAM, thus has a negligible effect in terms of the performance of modern computers. The Docker software will also consume approximately two hundred megabytes of RAM and there are some additional overhead costs in performance stemming from container management. In the comparative evaluations of virtual machines (VM), Docker and native execution conducted by Felter and colleagues (Felter et al. 2015) across file compression, linear algebra, and file Input/Output (I/O) tasks, Docker consistently demonstrates performance levels comparable with or exceeding those of VM. Moreover, Docker incurs minimal overhead in terms of the central processing unit (CPU) and memory utilization in contrast to native execution (Felter et al. 2015). Another evaluation of performance of Docker versus native execution was conducted by Di Tomasso and colleagues (Di Tommaso et al. 2015) where they benchmarked both with three different genomic pipelines; RNA-Seq data analysis, assembly-based variant calling, and detection and mapping of long non-coding RNAs. Results indicated minimal overhead for the first two pipelines when executed with Docker, registering 0.1% and 2.4% overhead, respectively.

However, the lncRNA detection and mapping pipeline exhibited a considerable slowdown of 65%. This decrement was attributed to the pipeline's characteristic of comprising numerous short-lived tasks, with a median execution time of 5.5 seconds, where Docker's overhead in bootstrapping the environment and mounting the host file system became notable (Di Tommaso et al. 2015). Another crucial aspect of performance in PipeCraft2 is the use of Windows Subsystems for Linux (WSL) on the Windows operating system (OS) and its performance compared to native Linux. A comparative performance analysis between native Linux and WSL was conducted by Dr Donald Kinghorn (Kinghorn 2022) using popular benchmarking tools such as Linear Algebra PACKage (LINPACK) and molecular dynamics application (NAMD). NAMD structural simulation of biomolecules was almost identical performance wise while LINPACK benchmarking showed a drop in performance by WSL as the number of cores was increased. This primarily arises from the overhead introduced by the Windows OS (Kinghorn 2022).

To further investigate the overhead performance costs of Docker, an experiment was conducted by comparing the execution of the DADA2 pipeline in a native environment and in PipeCraft2. A test sequencing data set consisted of 98 samples, with approximately 14.5 million ITS sequences. The RAM and CPU usage were homogenous, but the execution time differed by 16 minutes. Surprisingly, the Docker environment (via PipeCraft2 execution of the pipeline) beat the native execution, 26 minutes in PipeCraft2 and 42 minutes in a native environment. The native environment of the test system was Linux Mint 20.1, whereas the Docker environment was running a newer version of Linux (Ubuntu 22.04). This newer version of Linux in Docker container has performance upgrades to the kernel compared with the native environment, which boosted the speed of the analyses. While this speed test failed to deliver an accurate one-to-one comparison, it highlights the advantages of keeping the execution environment up to date, which is rather convenient with Docker but can be cumbersome and time-consuming on a personal computer. Therefore, depending on the native environment of the personal computer or computer cluster, the containers' updated environments may boost the speed of the analyses.

5. PipeCraft2 in action

The first release of PipeCraft2 (release version 0.1.0) was published on github on 14th of December 2021. It has served as a useful bioinformatic platform in many ongoing research projects for which several scientific publications are expected to be published within the next few years. For example, the bioinformatic workflows implemented in PipeCraft2 are used to analyze sequencing data from 16S, ITS2 and COI amplicons in the framework of Silva Nova project (Gundersen et al. 2023) that aims towards successful restoration and landscape-scale afforestation strategies that optimize productivity and biodiversity. Each of those amplicon genes are targeting different groups of organisms – 16S for prokaryotes, ITS2 for fungi and COI for arthropods – for which PipeCraft2 has suitable tools implemented. Moreover, PipeCraft2 has been used for the analysis of rbcL gene for identifying the diatom communities from the sediment samples (Anslan et al. 2022) and mitochondrial large subunit rRNA gene (mt16S) to study the diet of Galápagos sea lions (Urquía et al. 2024).

The easy-to-use graphical user interface makes bioinformatic processing of the metabarcoding data operable for researchers with limited bioinformatic skills. However, the cost of having GUI is the lack of compatibility with HPC clusters that are more time-effective in processing large amounts of data. Nevertheless, via PipeCraft2's GUI, reasonably large datasets can be processed on a laptop within a day. Just for the timing demonstration, a set of 252 samples (Illumina paired-end data) with an average of 344,796 sequences per sample (total number of input sequences was 173,777,174) flowed through the implemented DADA2 pipeline (including taxonomy assignment with RDP Classifier) within 7 hours and 36 minutes. The most time-consuming process was DADA2 denoising, which took 3 hours and 40 minutes, almost half of the total run time. This dataset was processed on a laptop with 30 GB of random access memory (RAM) (maximum usage throughout the process was 19 GB) and AMD Ryzen 7 PRO 4750U processor with 8 cores.

6. Discussion

While the early days of metabarcoding evolved around the 16S rRNA amplicons and just a few bioinformatic tools such as mothur (Schloss et al., 2009), USEARCH (Edgar, 2010), and

QIIME 1 (Caporaso et al., 2010) it has since gone through an expansion in its applications scope, encompassing a diverse array of taxa and amplicons sourced from a variety of environmental samples. This expansion in turn has led to a surge in the development of metabarcoding pipelines (Hakimzadeh et al. 2023). Choosing one from several dozen options can be quite challenging and it ultimately depends on factors such as the nature of the input data, the expertise of the researcher, and the characteristics of the computational resources available. When considering operating systems, Windows emerges as the least preferred among developers, with only a limited number of pipelines accommodating all three major platforms (Windows, Linux, Mac). In terms of interfaces, the majority are accessible via a CLI, with only a few offering a GUI. Regarding supported input data and DNA barcodes, there is notable versatility, as several pipelines support multiple sequencing platforms and DNA markers. In comparison to other widely used tools in the field, PipeCraft2 stands out as one of the most versatile due to its GUI coupled with a built-in CLI, cross-platform compatibility, a variety of supported input data formats and DNA markers. A key difference between PipeCraft2 and many other software for metabarcoding data bioinformatics processing is reliance on only a single dependency; a containerization software called Docker which is utilized on the back-end, to package and integrate third-party bioinformatic tools. Docker enables PipeCraft2 to execute bioinformatic processes in a Linux environment on any operating system and thus is one of the integral features for cross-platform support. PipeCraft2's GUI is designed using some of the most modern web development tools and therefore is rather more modern, sleek and intuitive in design compared to the handful of other tools that offer a GUI. Finally, PipeCraft2 is packaged with the Electron framework, which essentially converts it into a desktop app allowing for easy communication with the operating system and also enables it to easily build installers for all operating systems. As an open-source program, PipeCraft2 maintains a public repository on github, fostering collaboration between developers and the community. Beyond tooltips and robust error logging, PipeCraft2 also offers comprehensive documentation, particularly beneficial for users with limited experience in the field. Additionally, its support for workflow saving and loading enhances reproducibility — a rarity among similar software packages. Many of these design principles facilitate the ease of future developments as the customizability of PipeCraft2's front-end elements and the containerization of the back-end assets enables it to theoretically import any bioinformatic tool.

As a desktop application, PipeCraft2 faces certain limitations, such as reduced computational capacity when compared to pipelines that can be executed on high-performance computing clusters. Nevertheless, the increasing computational power of modern personal computers enables users to process a moderate-sized dataset within a working day. Currently, PipeCraft2 also lacks features like workflow pausing and resuming, which are supported by popular workflow managers such as Snakemake (Köster and Rahmann 2012) and Nextflow (Di Tommaso et al. 2017). However, a partially completed pipeline (due to some error) may be resumed by applying the following steps individually via the Quick Tools panel. While the addition of a graphical user interface (GUI) and Docker improves the user experience, it introduces slight performance overheads. These overheads are primarily due to the RAM usage by both PipeCraft2 and Docker, as well as the overhead from containerization, which is more notable in short-lived tasks where environment initialization and file system mounting can become significant in terms of time consumption.

The development of PipeCraft2 is ongoing, with exciting future advancements in progress that will be gradually rolled out. An important forthcoming feature is the implementation of auto-update functionality within PipeCraft2, eliminating the need for manual installation of updated versions (when available) and simplifying the process of delivering new features and bug fixes by the developers. The next release will also include a post-processing tool called metaMATE (metabarcoding Multiple Abundance Threshold Evaluator) to determine putative NUMTs and other erroneous sequences (Andújar et al. 2021). When working with large amounts of sequencing data, pooling the data might not always be the correct way as some bioinformatic processes such as filtering tag-jumps and DADA2 error modeling should be done per sequencing run, thus a feature is being developed to allow users to select and process multiple directories (i.e. sequencing runs) at once without pooling data for the aforementioned processes. Other upcoming future developments include optimizing the code for the management of Docker containers or integrating a modern workflow manager, granting user more detailed control over their computing resources such as RAM and CPU cores, enabling HPC cluster compatibility with alternative containerization methods (such as Apptainer), integrating and automating a R package for visualizing metabarcoding data, and staying up-to-date by importing new tools and pipelines.

7. Conclusion

In the context of this thesis, the primary objective was to enhance the functionality of the existing bioinformatic analysis toolkit, PipeCraft2. The key improvements focused on several critical aspects. First, user experience was prioritized by enhancing the user-friendliness and intuitiveness of the graphical user interface. Second, overall flexibility was improved by ensuring that PipeCraft2 is cross-platform, minimizing dependencies, and simplifying the process of importing bioinformatic tools. Notably, PipeCraft2 introduces a diverse array of new tools and modules. These additions enhance reproducibility, provide users with deeper insights into their data, and allow for highly customized utilization of PipeCraft2's assets.

The development process for these improvements involved rebuilding PipeCraft2 from the ground up, utilizing some of the most modern and widely adopted frameworks and tools available. While most development objectives were successfully achieved, there remains room for further enhancement. Our future efforts will primarily focus on two key areas: ensuring compatibility with HPC clusters and implementing robust version control through the introduction of an auto-update feature.

Importantly, the principles and tools employed during development have transformed PipeCraft2 into more than just a bioinformatic analysis toolkit. Instead, it can be regarded as a comprehensive software suite with a modern interface, capable of accommodating any bioinformatic tool. Its user-friendly interface and installation process positions PipeCraft2 as an ideal entry point for users who may not possess extensive bioinformatic expertise.

8. Resümee

Bioinformaatilise tarkvara arendus DNA meta-triipkoodistamis andmete analüüsiks

Martin Metsoja

DNA meta-triipkoodistamine on liikide identifitseerimise meetod, mis võimaldab keskkonnaproovidest üheaegselt tuvastada mitmeid organismirühmasid. Meetod on muutunud üha populaarsemaks molekulaarse ökoloogia valdkonnas ning on suuteline võimendama traditsioonilisi bioseire ja teadustöö strateegiaid. DNA meta-triipkoodistamisel tekib suures koguses DNA-järjestuste andmeid ning koos teiste teguritega, nagu DNA markergeenide ja sekveneerimisplatvormide mitmekesisus, võib hilisem andmete töötluse osutuda üpriski keerukaks. Meta-triipkoodistamise andmete analüüsiks on olemas mitmeid erinevaid tarkvarapakette, kuid tihti on nende installeerimine keerukas ning võimalik ainut arvutitel millel on Linux-i operatsioonisüsteem, kasutajaliidesed on sageli käsurea põhised ning analüüside kordamine on enamasti keerukas ning ajakulukas protsess. Eelnimetatud tegurid on tihti takistuseks kasutajatele kellel puuduvad varasemad kogemused bioinformaatika valdkonnas. Antud magistritöö eesmärgiks oli täiustada tarkvara paketti PipeCraft, mille tulemusena sündis PipeCraft2, millel puuduvad eelnimetatud kitsaskohad. Tarkvaraloome protsessi käigus kasutati kaasaegseid ja populaarseid aredus tööriistu ja raamistikke nagu Docker, Vue.js ja Electron. Bioinformaatiliste tööriistade korrektne töötamine sõltub tihti mitmetest lisamoodulitest või programmidest, kuid PipeCraft2 on sõltuv ainult ühest lisa programmist – Dockerist, mis võimaldab konteineriseerida bioinformaatilisi tööriistu ning neid importida, hoiustada ja käitada nii Windows, Linux kui ka Mac operatsioonisüsteemidega arvutitel. Electron raamistiks vastutab kasutajaliidese ning failisüsteemide vahelise suhtluse eest ning on vajalik ka tarkvara paigalduspakettide loomiseks. Docker ning Electron koostöös võimaldavad PipeCraft2-te kasutada kõigi operatsiooni süsteemidega. Kui paljusid populaarseid bioinformaatika tööriistu tuleb opereerida käsurealt, siis PipeCraft2 sisaldab endas intuitiivset ning kasutajasõbralikku graafilist kasutajaliidest, mille arendamisel on kasutatud üht populaarsemat ning kaasaegsemat JavaScript-i raamistikku, Vue.js. Põhimõtted, tööriistad ning raamistikud mida rakendati arendustegevuse käigus on teinud PipeCraft2-st midagi rohkemat kui lihtsa bioinfomaatilise tööriista, PipeCraft2-te võiks vaadelda kui karkassi mis võimaldab kergelt lõimida mitmeid erinevaid bioinfromaatilisi tööriistu ning lihtsustab nende kasutamist intuitiivse graafilise kasutajaliidesega. PipeCraft2-e kasutajasõbralik disain ja lihtne paigalduse protseduur muudavad selle ideaalseks valikuks kasutajatele, kellel puuduvad eelnevad bioinformaatika alased teadmised ja kogemused.

9. References

Andújar, Carmelo, Thomas J. Creedy, Paula Arribas, Heriberto López, Antonia Salces-Castellano, Antonio José Pérez-Delgado, Alfried P. Vogler, and Brent C. Emerson. 2021. "Validated Removal of Nuclear Pseudogenes and Sequencing Artefacts from Mitochondrial Metabarcode Data." *Molecular Ecology Resources* 21 (6): 1772–87. https://doi.org/10.1111/1755-0998.13337.

Anslan, Sten, Mohammad Bahram, Indrek Hiiesalu, and Leho Tedersoo. 2017. "PipeCraft: Flexible Open-Source Toolkit for Bioinformatics Analysis of Custom High-Throughput Amplicon Sequencing Data." *Molecular Ecology Resources* 17 (6): e234–40. https://doi.org/10.1111/1755-0998.12692.

Anslan, Sten, Wengang Kang, Katharina Dulias, Bernd Wünnemann, Paula Echeverría-Galindo, Nicole Börner, Anja Schwarz, et al. 2022. "Compatibility of Diatom Valve Records With Sedimentary Ancient DNA Amplicon Data: A Case Study in a Brackish, Alkaline Tibetan Lake." *Frontiers in Earth Science* 10. https://www.frontiersin.org/articles/10.3389/feart.2022.824656.

Bahram, Mohammad, Falk Hildebrand, Sofia K. Forslund, Jennifer L. Anderson, Nadejda A. Soudzilovskaia, Peter M. Bodegom, Johan Bengtsson-Palme, et al. 2018. "Structure and Function of the Global Topsoil Microbiome." *Nature*560 (7717): 233–37. https://doi.org/10.1038/s41586-018-0386-6.

Bengtsson-Palme, Johan, Martin Ryberg, Martin Hartmann, Sara Branco, Zheng Wang, Anna Godhe, Pierre De Wit, et al. 2013. "Improved Software Detection and Extraction of ITS1 and ITS2 from Ribosomal ITS Sequences of Fungi and Other Eukaryotes for Analysis of Environmental Sequencing Data." *Methods in Ecology and Evolution* 4 (10): 914–19. https://doi.org/10.1111/2041-210X.12073.

Bokulich, Nicholas A., Sathish Subramanian, Jeremiah J. Faith, Dirk Gevers, Jeffrey I. Gordon, Rob Knight, David A. Mills, and J. Gregory Caporaso. 2013. "Quality-Filtering Vastly Improves Diversity Estimates from Illumina Amplicon Sequencing." *Nature Methods* 10 (1): 57–59. https://doi.org/10.1038/nmeth.2276.

Bolger, Anthony M., Marc Lohse, and Bjoern Usadel. 2014. "Trimmomatic: A Flexible Trimmer for Illumina Sequence Data." *Bioinformatics* 30 (15): 2114–20. https://doi.org/10.1093/bioinformatics/btu170.

Bolyen, Evan, Jai Ram Rideout, Matthew R. Dillon, Nicholas A. Bokulich, Christian C. Abnet, Gabriel A. Al-Ghalith, Harriet Alexander, et al. 2019. "Reproducible, Interactive, Scalable and Extensible Microbiome Data Science Using QIIME 2."

Nature Biotechnology 37 (8): 852–57. https://doi.org/10.1038/s41587-019-0209-9.

Boyer, Frédéric, Céline Mercier, Aurélie Bonin, Yvan Le Bras, Pierre Taberlet, and Eric Coissac. 2016. "Obitools: A Unix-Inspired Software Package for DNA Metabarcoding." *Molecular Ecology Resources* 16 (1): 176–82. https://doi.org/10.1111/1755-0998.12428.

Callahan, Benjamin J., Paul J. McMurdie, Michael J. Rosen, Andrew W. Han, Amy Jo A. Johnson, and Susan P. Holmes. 2016. "DADA2: High Resolution Sample Inference from Illumina Amplicon Data." *Nature Methods* 13 (7): 581. https://doi.org/10.1038/nmeth.3869.

Camacho, Christiam, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L. Madden. 2009. "BLAST+: Architecture and Applications." *BMC Bioinformatics* 10 (1): 421. https://doi.org/10.1186/1471-2105-10-421.

Chen, Shifu, Yanqing Zhou, Yaru Chen, and Jia Gu. 2018. "Fastp: An Ultra-Fast All-in-One FASTQ Preprocessor." *Bioinformatics* 34 (17): i884–90. https://doi.org/10.1093/bioinformatics/bty560.

Di Tommaso, Paolo, Maria Chatzou, Evan W. Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. "Nextflow Enables Reproducible Computational Workflows." *Nature Biotechnology* 35 (4): 316–19. https://doi.org/10.1038/nbt.3820.

Di Tommaso, Paolo, Emilio Palumbo, Maria Chatzou, Pablo Prieto, Michael L. Heuer, and Cedric Notredame. 2015. "The Impact of Docker Containers on the Performance of Genomic Pipelines." *PeerJ* 3 (September):e1273. https://doi.org/10.7717/peerj.1273.

Dickie, Ian A., Nicola Bolstridge, Jerry A. Cooper, and Duane A. Peltzer. 2010. "Co-Invasion by Pinus and Its Mycorrhizal Fungi." *New Phytologist* 187 (2): 475–84. https://doi.org/10.1111/j.1469-8137.2010.03277.x.

Eddy, S R. 1998. "Profile Hidden Markov Models." *Bioinformatics* 14 (9): 755–63. https://doi.org/10.1093/bioinformatics/14.9.755.

Edgar, Robert C. 2010. "Search and Clustering Orders of Magnitude Faster than BLAST." *Bioinformatics* 26 (19): 2460–61. https://doi.org/10.1093/bioinformatics/btq461.

Edgar, Robert C. 2016. "UNOISE2: Improved Error-Correction for Illumina 16S and ITS Amplicon Sequencing." bioRxiv. https://doi.org/10.1101/081257.

Edgar, Robert C., Brian J. Haas, Jose C. Clemente, Christopher Quince, and Rob Knight. 2011. "UCHIME Improves Sensitivity and Speed of Chimera Detection." *Bioinformatics* 27 (16): 2194–2200. https://doi.org/10.1093/bioinformatics/btr381.

Edgar, Robert C. 2018. “UNCROSS2: Identification of Cross-Talk in 16S rRNA OTU Tables.” bioRxiv. https://doi.org/10.1101/400762.

Ewels, Philip, Måns Magnusson, Sverker Lundin, and Max Käller. 2016. “MultiQC: Summarize Analysis Results for Multiple Tools and Samples in a Single Report.” *Bioinformatics* 32 (19): 3047–48. https://doi.org/10.1093/bioinformatics/btw354.

Felter, Wes, Alexandre Ferreira, Ram Rajamony, and Juan Rubio. 2015. “An Updated Performance Comparison of Virtual Machines and Linux Containers.” In *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 171–72. https://doi.org/10.1109/ISPASS.2015.7095802.

Frøslev, Tobias Guldberg, Rasmus Kjøller, Hans Henrik Bruun, Rasmus Ejrnæs, Ane Kirstine Brunbjerg, Carlotta Pietroni, and Anders Johannes Hansen. 2017. “Algorithm for Post-Clustering Curation of DNA Amplicon Data Yields Reliable Biodiversity Estimates.” *Nature Communications* 8 (1): 1188. https://doi.org/10.1038/s41467-017-01312-x.

García de León, David, John Davison, Mari Moora, Maarja Öpik, Huyuan Feng, Inga Hiiesalu, Teele Jairus, et al. 2018. “Anthropogenic Disturbance Equalizes Diversity Levels in Arbuscular Mycorrhizal Fungal Communities.” *Global Change Biology* 24 (6): 2649–59. https://doi.org/10.1111/gcb.14131.

Gundersen, Per, T. Martijn Bezemer, Sebastian Rojas, Leho Tedersoo, Lars Vesterdal, and Inger Schmidt. 2023. “Silva Nova – Restoring Soil Biology and Soil Functions to Gain Multiple Benefits in New Forests.” *Research Ideas and Outcomes* 9 (February):e101455. https://doi.org/10.3897/rio.9.e101455.

Gweon, Hyun S., Anna Oliver, Joanne Taylor, Tim Booth, Melanie Gibbs, Daniel S. Read, Robert I. Griffiths, and Karsten Schonrogge. 2015. “PIPITS: An Automated Pipeline for Analyses of Fungal Internal Transcribed Spacer Sequences from the Illumina Sequencing Platform.” *Methods in Ecology and Evolution* 6 (8): 973–80. https://doi.org/10.1111/2041-210X.12399.

Haas, Brian J., Dirk Gevers, Ashlee M. Earl, Mike Feldgarden, Doyle V. Ward, Georgia Giannoukos, Dawn Ciulla, et al. 2011. “Chimeric 16S rRNA Sequence Formation and Detection in Sanger and 454-Pyrosequenced PCR Amplicons.” *Genome Research* 21 (3): 494–504. https://doi.org/10.1101/gr.112730.110.

Hakimzadeh, Ali, Alejandro Abdala Asbun, Davide Albanese, Maria Bernard, Dominik Buchner, Benjamin Callahan, J. Gregory Caporaso, et al. 2023. “A Pile of Pipelines: An Overview of the Bioinformatics Software for Metabarcoding Data Analyses.” *Molecular Ecology Resources* n/a (n/a). https://doi.org/10.1111/1755-0998.13847.

Hildebrand, Falk, Raul Tito Tadeo, Anita Voigt, Peer Bork, and Jeroen Raes. 2014. “LotuS:

An Efficient and User-Friendly OTU Processing Pipeline." *Microbiome* 2 (September):30. https://doi.org/10.1186/2049-2618-2-30.

Köster, Johannes, and Sven Rahmann. 2012. "Snakemake—a Scalable Bioinformatics Workflow Engine." *Bioinformatics*28 (19): 2520–22. https://doi.org/10.1093/bioinformatics/bts480.

Kwon, ChangHyuk, Jason Kim, and Jaegyoon Ahn. 2018. "DockerBIO: Web Application for Efficient Use of Bioinformatics Docker Images." *PeerJ* 6 (November):e5954. https://doi.org/10.7717/peerj.5954.

Leese, Florian, Florian Altermatt, Agnès Bouchez, Torbjørn Ekrem, Daniel Hering, Kristian Meissner, Patricia Mergen, et al. 2016. "DNAqua-Net: Developing New Genetic Tools for Bioassessment and Monitoring of Aquatic Ecosystems in Europe." *Research Ideas and Outcomes* 2 (November):e11321. https://doi.org/10.3897/rio.2.e11321.

Mahé, Frédéric, Torbjørn Rognes, Christopher Quince, Colomban de Vargas, and Micah Dunthorn. 2014. "Swarm: Robust and Fast Clustering Method for Amplicon-Based Studies." *PeerJ* 2 (September):e593. https://doi.org/10.7717/peerj.593.

Martin, Marcel. 2011. "Cutadapt Removes Adapter Sequences from High-Throughput Sequencing Reads." *EMBnet.Journal* 17 (1): 10–12. https://doi.org/10.14806/ej.17.1.200.

Martino, Cameron, James T. Morton, Clarisse A. Marotz, Luke R. Thompson, Anupriya Tripathi, Rob Knight, and Karsten Zengler. 2019. "A Novel Sparse Compositional Technique Reveals Microbial Perturbations." *mSystems* 4 (1): 10.1128/msystems.00016-19. https://doi.org/10.1128/msystems.00016-19.

Menegidio, Fabiano B, David Aciole Barbosa, Rafael dos S Gonçalves, Marcio M Nishime, Daniela L Jabes, Regina Costa de Oliveira, and Luiz R Nunes. 2019. "Bioportainer Workbench: A Versatile and User-Friendly System That Integrates Implementation, Management, and Use of Bioinformatics Resources in Docker Environments." *GigaScience* 8 (4): giz041. https://doi.org/10.1093/gigascience/giz041.

Moreews, François, Olivier Sallou, Hervé Ménager, Yvan Le bras, Cyril Monjeaud, Christophe Blanchet, and Olivier Collin. 2015. "BioShaDock: A Community Driven Bioinformatics Shared Docker-Based Tools Registry." *F1000Research* 4 (December):1443. https://doi.org/10.12688/f1000research.7536.1.

Mugnai, Francesco, Federica Costantini, Anne Chenuil, Michèle Leduc, José Miguel Gutiérrez Ortega, and Emese Meglécz. 2023. "Be Positive: Customized Reference Databases and New, Local Barcodes Balance False Taxonomic Assignments in Metabarcoding Studies." *PeerJ* 11 (January):e14616. https://doi.org/10.7717/peerj.14616.

Porter, T. M., and M. Hajibabaei. 2021. “Profile Hidden Markov Model Sequence Analysis Can Help Remove Putative Pseudogenes from DNA Barcoding and Metabarcoding Datasets.” *BMC Bioinformatics* 22 (May):256. https://doi.org/10.1186/s12859-021-04180-x.

Porter, T. M., and M. Hajibabaei. 2022. “MetaWorks: A Flexible, Scalable Bioinformatic Pipeline for High-Throughput Multi-Marker Biodiversity Assessments.” *PLOS ONE* 17 (9): e0274260. https://doi.org/10.1371/journal.pone.0274260.

Quince, Christopher, Anders Lanzen, Russell J. Davenport, and Peter J. Turnbaugh. 2011. “Removing Noise From Pyrosequenced Amplicons.” *BMC Bioinformatics* 12 (1): 38. https://doi.org/10.1186/1471-2105-12-38.

Rognes, Torbjørn, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. 2016. “VSEARCH: A Versatile Open Source Tool for Metagenomics.” *PeerJ* 4 (October):e2584. https://doi.org/10.7717/peerj.2584.

Rombel, Irene T, Kathryn F Sykes, Simon Rayner, and Stephen Albert Johnston. 2002. “ORF-FINDER: A Vector for High-Throughput Gene Identification.” *Gene* 282 (1): 33–41. https://doi.org/10.1016/S0378-1119(01)00819-8.

Ruppert, Krista M., Richard J. Kline, and Md Saydur Rahman. 2019. “Past, Present, and Future Perspectives of Environmental DNA (eDNA) Metabarcoding: A Systematic Review in Methods, Monitoring, and Applications of Global eDNA.” *Global Ecology and Conservation* 17 (January):e00547. https://doi.org/10.1016/j.gecco.2019.e00547.

Schloss, Patrick D., Dirk Gevers, and Sarah L. Westcott. 2011. “Reducing the Effects of PCR Amplification and Sequencing Artifacts on 16S rRNA-Based Studies.” *PLoS ONE* 6 (12): e27310. https://doi.org/10.1371/journal.pone.0027310.

Schloss, Patrick D., Sarah L. Westcott, Thomas Ryabin, Justine R. Hall, Martin Hartmann, Emily B. Hollister, Ryan A. Lesniewski, et al. 2009. “Introducing Mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities.” *Applied and Environmental Microbiology* 75 (23): 7537–41. https://doi.org/10.1128/AEM.01541-09.

Stein, Eric D., Maria C. Martinez, Sara Stiles, Peter E. Miller, and Evgeny V. Zakharov. 2014. “Is DNA Barcoding Actually Cheaper and Faster than Traditional Morphological Methods: Results from a Survey of Freshwater Bioassessment Efforts in the United States?” *PLOS ONE* 9 (4): e95525. https://doi.org/10.1371/journal.pone.0095525.

Sunagawa, Shinichi, Luis Pedro Coelho, Samuel Chaffron, Jens Roat Kultima, Karine Labadie, Guillem Salazar, Bardya Djahanschiri, et al. 2015. “Structure and Function

of the Global Ocean Microbiome." *Science* 348 (6237): 1261359. https://doi.org/10.1126/science.1261359.

Taberlet, Pierre, Eric Coissac, François Pompanon, Christian Brochmann, and Eske Willerslev. 2012. "Towards Next-Generation Biodiversity Assessment Using DNA Metabarcoding." *Molecular Ecology* 21 (8): 2045–50. https://doi.org/10.1111/j.1365-294X.2012.05470.x.

Tedersoo, Leho, Mohammad Bahram, Sergei Põlme, Urmas Kõljalg, Nourou S. Yorou, Ravi Wijesundera, Luis Villarreal Ruiz, et al. 2014. "Global Diversity and Geography of Soil Fungi." *Science* 346 (6213): 1256688. https://doi.org/10.1126/science.1256688.

Tedersoo, Leho, Mohammad Bahram, Lucie Zinger, R. Henrik Nilsson, Peter G. Kennedy, Teng Yang, Sten Anslan, and Vladimir Mikryukov. 2022. "Best Practices in Metabarcoding of Fungi: From Experimental Design to Results." *Molecular Ecology* 31 (10): 2769–95. https://doi.org/10.1111/mec.16460.

Tedersoo, Leho, R. Henrik Nilsson, Kessy Abarenkov, Teele Jairus, Ave Sadam, Irja Saar, Mohammad Bahram, Eneke Bechem, George Chuyong, and Urmas Kõljalg. 2010. "454 Pyrosequencing and Sanger Sequencing of Tropical Mycorrhizal Fungi Provide Similar Results but Reveal Substantial Methodological Biases." *New Phytologist* 188 (1): 291–301. https://doi.org/10.1111/j.1469-8137.2010.03373.x.

Tedersoo, Leho, Vladimir Mikryukov, Sten Anslan, Mohammad Bahram, Abdul Nasir Khalid, Adriana Corrales, Ahto Agan, et al. 2021. "The Global Soil Mycobiome Consortium Dataset for Boosting Fungal Diversity Research." *Fungal Diversity* 111 (1): 573–88. https://doi.org/10.1007/s13225-021-00493-7.

Thomsen, Philip, and Eske Willerslev. 2014. "Environmental DNA – An Emerging Tool in Conservation for Monitoring Past and Present Biodiversity." *Biological Conservation* 183 (December). https://doi.org/10.1016/j.biocon.2014.11.019.

Urquía, Diego O., Sten Anslan, Pacarina Asadobay, Andrés Moreira-Mendieta, Miguel Vences, Jaime A. Chaves, and Diego Páez-Rosas. 2024. "DNA-Metabarcoding Supports Trophic Flexibility and Reveals New Prey Species for the Galapagos Sea Lion." *Ecology and Evolution* 14 (3): e10921. https://doi.org/10.1002/ece3.10921.

Valentini, Alice, Pierre Taberlet, Claude Miaud, Raphaël Civade, Jelger Herder, Philip Francis Thomsen, Eva Bellemain, et al. 2016. "Next-Generation Monitoring of Aquatic Biodiversity Using Environmental DNA Metabarcoding." *Molecular Ecology* 25 (4): 929–42. https://doi.org/10.1111/mec.13428.

Vasar, Martti, John Davison, Lena Neuenkamp, Siim-Kaarel Sepp, J. Peter W. Young, Mari Moora, and Maarja Öpik. 2021. "User-Friendly Bioinformatics Pipeline gDAT (Graphical Downstream Analysis Tool) for Analysing rDNA Sequences." *Molecular*

Ecology Resources 21 (4): 1380–92. https://doi.org/10.1111/1755-0998.13340.

Větrovský, Tomáš, Petr Baldrian, and Daniel Morais. 2018. "SEED 2: A User-Friendly Platform for Amplicon High-Throughput Sequencing Data Analyses." *Bioinformatics* 34 (13): 2292–94. https://doi.org/10.1093/bioinformatics/bty071.

Andrews, Simon. 2010. "Babraham Bioinformatics - FastQC A Quality Control Tool for High Throughput Sequence Data." 2010. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

Creedy, Thomas J., Alfried P. Vogler, and Liam Penlington. 2020. "Bioinformatic Methods for Biodiversity Metabarcoding — Bioinformatic Methods for Biodiversity Metabarcoding Documentation." 2020. https://learnmetabarcoding.github.io/LearnMetabarcoding/index.html.

Docker Incorporated. 2024. "Docker Overview." Docker Documentation. 2024. https://docs.docker.com/get-started/overview/.

Docker Incorporated. 2024. "Docker-Sponsored Open Source Program." Docker Documentation. 2024. https://docs.docker.com/trusted-content/dsos-program/.

Kinghorn, Dr Donald. 2022. "WSL2 vs Linux (HPL HPCG NAMD)." Puget Systems. August 31, 2022. https://www.pugetsystems.com/labs/hpc/wsl2-vs-linux-hpl-hpcg-namd-2354/.

Klayman, Noah. 2018. "Vue CLI Plugin Electron Builder." 2018. https://nklayman.github.io/vue-cli-plugin-electron-builder/.

Krotoff, Tanguy. 2023. "Front-End Frameworks Popularity (React, Vue, Angular and Svelte)." Gist. 2023. https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190.

Dias, Pedro. (2013) 2024. "Apocas/Dockerode." JavaScript. https://github.com/apocas/dockerode.

Xia, Yinglin, and Jun Sun. 2023. "Clustering Sequences into OTUs." In *Bioinformatic and Statistical Analysis of Microbiome Data: From Raw Sequences to Advanced Modeling with QIIME 2 and R*, edited by Yinglin Xia and Jun Sun, 147–59. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-21391-5_6.

"OpenAI. 'The text in this thesis was edited with the assistance of an AI language model provided by OpenAI.' 2024."

10. License for Reproduction and Public Accessibility of Thesis

I, Martin Metsoja, hereby grant Tartu University a free license for my created work:

Development of a multi-platform metabarcoding bioinformatics software with an easy-to-use graphical user interface

which was supervised by Leho Tedersoo and Sten Anslan.

The purpose of this license is to allow reproduction, including adding it to the digital archive DSpace, until the expiration of copyright.

I authorize Tartu University to make the aforementioned work accessible to the public through Tartu University's web environment, including the digital archive DSpace, under the Creative Commons license CC BY NC ND 4.0. This license permits reproduction, distribution, and public dissemination of the work with proper attribution to the author, while prohibiting derivative works and commercial use, until the expiration of copyright.

I am aware that the aforementioned rights are also retained by the author.

I confirm that by granting this license, I do not violate the intellectual property or personal data protection rights of other individuals.

Martin Metsoja 15.02.2024