

# The pegi3s Bioinformatics Docker Images Project

Hugo López-Fernández<sup>1,2,3,4\*</sup>, Pedro Ferreira<sup>1,2\*</sup>, Miguel Reboiro-Jato<sup>3,4</sup>, Cristina P. Vieira<sup>1,2</sup>, and Jorge Vieira<sup>1,2</sup>

<sup>1</sup>Instituto de Investigação e Inovação em Saúde (I3S), Universidade do Porto, Rua Alfredo Allen, 208, 4200-135 Porto, Portugal

<sup>2</sup>Instituto de Biologia Molecular e Celular (IBMC), Rua Alfredo Allen, 208, 4200-135 Porto, Portugal

<sup>3</sup>CINBIO, Universidade de Vigo, Department of Computer Science, ESEI – Escuela Superior de Ingeniería Informática, 32004 Ourense, Spain

<sup>4</sup>SING Research Group, Galicia Sur Health Research Institute (IIS Galicia Sur), SERGAS-UVIGO

\*contributed equally to this work

hugo.fernandez@i3s.up.pt, pedro.ferreira@i3s.up.pt,  
mrjato@uvigo.es, cgvieira@ibmc.up.pt, jbvieira@ibmc.up.pt

**Abstract.** Among the available Linux container technologies, Docker is one of the most popular ones. Docker images can be used to provide ready-to-use software packages, where all required dependencies are already installed, and they can be deployed in any operating system where Docker is installed. They are also a convenient way to store immutable working software packages, thus contributing to reproducibility. Moreover, the usage of Docker images greatly eases the development of complex pipelines, standalone software applications with graphical user interfaces that require external software, and even the development of databases. Therefore, not surprisingly, Docker images are now ubiquitously used in computational biology and bioinformatics. Here, we present the pegi3s Bioinformatics Docker Images Project (<https://pegi3s.github.io/docker-files/>), a collection of more than 70 Docker images for commonly used software in the fields of genomics, transcriptomics, proteomics, phylogenetics, and sequence handling, among others, that is constantly growing. Several features distinguish this project from much larger projects, namely: 1) by providing a list of tools that are classified into broad categories, it is easier to find the most adequate tool(s) for a given project; 2) by providing the hyperlinks to the software manuals, we facilitate the process of finding the parameter combinations that are best suited for a given processing step; 3) most importantly, we provide clear instructions on how to run the images, provide test data that can be used to quickly evaluate the Docker image, and give all details on how each Docker image was built. All images are routinely used by ourselves, in the context of our research and teaching activities, meaning that they have been extensively tested. Therefore, we believe that this project, which is offered as a service in the context of the European ELIXIR program, is of interest to many researchers, independently of having or not a background in informatics.

**Keywords:** docker, bioinformatics, reproducibility.

## 1. Introduction

Computational biology and bioinformatics rely heavily on the usage of different software tools and packages and, ultimately, on the successful connection of them to create workflows or automated pipelines that perform complete analyses without human intervention, greatly speeding up the creation of new knowledge [1]. Nevertheless, the installation of such software applications (including the management of their dependencies) can be a cumbersome task that negatively affects research efficiency and reproducibility. The main difficulty is that not all software versions will work in the context of a given pipeline. Moreover, older software versions as well as software libraries may become unavailable with time. Therefore, it may be impossible to re-analyze the data under the same conditions in the future. The usage of immutable, ready-to-use Linux containers solves these issues, and thus, it is not surprising that they are ubiquitously used. Since software applications that have been developed for operating systems (OS) other than Linux, require the usage of software applications and libraries that are protected by international laws, containers cannot be created for them, unless compatibility layers, such as Wine<sup>1</sup>, which allows running Windows applications in Linux OS, are used. Nevertheless, most scientific software has been developed for Linux OS.

Any given bioinformatics software running on Linux OS can be containerized together with all its dependencies in such a way that it can be seamlessly executed regardless of the OS distribution used by the host system. This is achieved by using advanced features of modern Linux kernels that allow running a set of processes in a fully isolated environment [2]. By installing the required dependencies once only on a clean system (without undesired interactions with other installed packages), dependency management is greatly simplified. Unlike virtual machines, containers share the machine's OS kernel and therefore do not require an OS per application. Moreover, because containers are lightweight, usually, it does not take long to download any container from an online repository. Among the different container tools available, Docker is one of the most popular ones and many bioinformatics projects based on Docker have emerged in the last years. Some of these projects are presented in section 2. In addition, recommendations regarding the writing of *Dockerfiles* (descriptions of Docker images) and the creation of containerized bioinformatics software have been published [3, 4]. Despite the very appealing features described above, it should be noted that Docker is rarely used in the context of multi-user systems such as High Performance Computing (HPC) systems, grid infrastructures or Linux clusters. Indeed, processes within the Docker container are normally executed with root privileges under the Docker daemon process tree, thus escaping the resource usage policies, accounting controls, and process controls that are imposed to normal users. Moreover, users authorized to access the Docker remote API can easily gain privileged access to the host system [2]. Docker images that implement software with graphical interfaces cannot also be deployed within multi-user systems since these are usually headless systems. Therefore, Docker images are

---

<sup>1</sup> <https://www.winehq.org>

mostly appreciated in the context of institutional bioinformatics platforms that provide services to research groups, or even in the context of research groups that routinely develop and/or perform large-scale bioinformatics analyses. Although there is a tool that enables the user to execute Docker containers in user mode (udocker [2]), and is thus appropriate for multi-user systems, this solution only works for a fraction of all available Docker images.

Nearly three years ago, we started to use Docker at the Phenotypic Evolution Group at i3S (pegi3s) to manage the bioinformatics software needed. Instead of installing every software on every computer where it was required, we created one *Dockerfile* for each software and a GitHub project<sup>2</sup> to host the developed Dockerfiles and associated resources. These files were then used to create the Docker images that are hosted at Docker Hub<sup>3</sup>. Once this process is complete, all it takes is to perform a pull from Docker Hub to the computer where it is needed. This way, the pegi3s Bioinformatics Docker Images Project was started. Given the large number of Docker images that were developed, it seemed logical to create detailed instructions on how to run every Docker image, as well as providing hyperlinks to the software application manuals. This way, it is easier for new students and researchers to start using the required software tools. To date, 77 Docker images for commonly used software applications, pipelines, and software for the automated submission to web servers have been developed. As evidenced by the more than 70 000 pulls, at present, many researchers that do not belong to i3S are using this tool. The usefulness of the pegi3s Bioinformatics Docker Images Project has been recognized by ELIXIR, an intergovernmental organization that brings together life science resources from across Europe, where it is advertised as a service (search for pegi3s at <https://elixir-europe.org/services>). In this work, we present our project in detail as well as our own experience using Docker, for deployment of software commonly used in the genomics, transcriptomics, phylogenetics and protein modeling fields, as well as for development of pipelines and databases. New images are constantly being added. For users wishing to use containerized software with a graphical user interface (GUI) in Windows hosts, we also provide a VirtualBox image with Docker installed.

## 2. Related work

The first bioinformatics Docker projects to be published (2015) were BioBoxes [5] and BioShaDock [6]. BioBoxes<sup>4</sup> had the aim of defining a way to specify standardised bioinformatics containers. As authors state in their original publication: “A biobox is a software container with a standardised interface that describes what kind of input files and parameters are accepted and which output files are to be returned.” [5]. Along with a repository of Docker images (which currently has less than 10 images), the project had a command line interface to facilitate the usage of the images. BioShaDock was started as a bioinformatics-focused Docker registry, providing a lo-

---

<sup>2</sup> <https://pegi3s.github.io/dockerfiles>

<sup>3</sup> <https://hub.docker.com>

<sup>4</sup> <http://bioboxes.org>

cal and fully controlled environment to build and publish bioinformatics software as portable Docker images. The project was then merged into the BioContainers project created in 2017 [7]. BioContainers<sup>5</sup> provides the infrastructure and basic guidelines to create, manage, and distribute bioinformatics packages and containers. First, it provides a base specification and infrastructure to develop, build, and deploy new bioinformatics software. Second, it also has a repository with a series of containers ready to be used by the bioinformatics community. They also provide guidelines and help on how to create reproducible pipelines and workflows using bioinformatics containers [3]. Every container of the project is deployed and permanently deposited in a public registry (Docker Hub or Quay.io), and one of their main features is that the project builds automatically Docker containers for all BioConda packages.

Two other Docker-related projects with more specific goals were published in 2017: Dugong [8] and The Dockstore [9]. Dugong<sup>6</sup> is announced as a complete Docker desktop environment for bioinformatics and is based on Ubuntu 16.04, providing a complete GUI that facilitates the installation and use of different bioinformatics software obtained from LinuxBrew and BioConda. Dugong includes the Anaconda Navigator, a graphical package manager for Conda and the BioConda repository, and a ready-to-use Jupyter Notebook installation. The Dockstore<sup>7</sup> is an open platform used by the Global Alliance for Genomics and Health (GA4GH) for sharing Docker-based tools and workflows described with either the Common Workflow Language (CWL), the Workflow Description Language (WDL), or Nextflow.

In 2018, the Reproducible Bioinformatics Project<sup>8</sup> was published [8]. This project, focused on workflow reproducibility, aims to provide a schema and an infrastructure, based on Docker images and R packages, to guarantee reproducible results. Based on this project, authors provide five ready-to-use workflows for RNAseq, miRNA-seq, ChIPseq, single-cellRNAseq, and circular RNA.

To finish this brief overview, we make also a reference to ORCA<sup>9</sup>, a comprehensive bioinformatics container environment for education and research published in 2019 [10]. This environment includes hundreds of pre-compiled and configured bioinformatics tools and can be used to install a multitude of bioinformatics tools on a fresh Linux server easily, providing a private container to each individual user, or shared containers to a collaborative group of users.

The main difference between our project and much larger projects like BioContainers is the exhaustive documentation provided for each Docker image, allowing researchers without a background in informatics to use them easily. These includes: test cases, listing the most used commands and options, giving hyperlinks to the software's manual, and providing a "*docker run*" command where only data paths need to be adjusted to successfully execute the image. All images listed in our repository are curated, tested, and maintained by ourselves. A great effort is made to make sure that they work properly without posing a security risk to others. Images for new

---

<sup>5</sup> <https://biocontainers.pro>

<sup>6</sup> <https://dugongbioinformatics.github.io>

<sup>7</sup> <https://dockstore.org>

<sup>8</sup> <http://www.reproducible-bioinformatics.org>

<sup>9</sup> <https://hub.docker.com/r/bcgsc/orca>

software applications are created upon request of project users or as we need them to conduct our research and develop our own pipelines.

### 3. The pegi3s Bioinformatics Docker Images Project

#### 3.1 Docker images

The main aim of our project is to have a manageable, curated set of well-documented and tested Bioinformatics Docker images. By routinely using them, we can easily detect any unforeseen problems and timely correct them. For each Docker image in our project, we provide clear instructions on how to use them (main commands and most used options), give test cases, and provide the hyperlink to the software’s manual. Therefore, for every image, there is a “*docker run*” command where only data paths need to be adjusted to successfully execute the image.

The homepage of the project categorizes the available bioinformatics Docker images in the following categories:

- Programs: images associated with scientific software applications for genomics, proteomics, phylogenetics, and so on. This is the category with the largest number of images. While most of them are command-line applications, there are also several programs with GUIs. In the latter case, as mentioned in the project description, the user must first disable access control by typing “*xhost +*” in the command line, before running the Docker image.
- Automated submission to web servers: currently, this category contains four images that automate the submission process to web servers.
- Pipelines: images providing implementations of small pipelines. For instance, the “*splign-compart*” image includes a script that executes the Splign/Compart pipeline as implemented in our SEDA tool<sup>10</sup> [11].
- Compi pipelines: images of Compi-based pipelines [12] developed by us, as explained in more detail in subsection 3.2.
- External images: images that are only mirrors to external images, that is, images whose original *Dockerfiles* are not written by us. The purpose of having these images in the project is to add a basic documentation on the image usage to our repository, following the same format we have for the remaining images.
- Additional images: images for general-purpose software like R or Biopython.

As of April 16, 2021, our project has 77 Docker images. Only one of them (pegi3s/dnasp-v6) is for a software with a GUI developed for Windows OS. In order to develop this image, Wine was used as the translation layer. Although DnaSP is easy to install in Windows OS, this image allowed us to perform workflow analyses without having to use computers with different OS. Nevertheless, the most recent Windows OS (Windows 10 64-bit: Pro, Enterprise, or Education - Build 17134 or higher) allow running Docker as a native process<sup>11</sup>. Moreover, Linux applications can now be deployed as-is on the Windows Subsystem for Linux (WSL). Therefore, the

<sup>10</sup> <https://www.sing-group.org/seda/manual/operations.html#splign-compart-pipeline>

<sup>11</sup> <https://docs.docker.com/docker-for-windows/install>

need for Docker images for Windows software applications may diminish or even disappear in the near future, with the full and efficient integration of Windows and Linux OS. Table 1 shows the top 10 downloaded images, based on the pull counts reported by the Docker Hub API. Although these counts are not 100% accurate (e.g. pull count is increased in automated builds or when executing *docker pull* commands even when the image already exists in the host), they reflect the community usage of these images. In addition, all pegi3s Docker images can be run using Singularity, and we provide a guide with some examples<sup>12</sup>.

Table 1. Top 10 downloaded images of the project on April 16, 2021.

Docker Image	Number of Pulls
SAMtools-BCFtools	34453
FastQC	14710
HyPhy	2315
SRA Toolkit	1898
BWA	1633
SeqKit	1461
Utilities	855
Bedtools	852
T-Coffee	709
MrBayes	555

Following good programming practices, we have also created some images where we put common scripts that can be used in several scenarios. These images are “*utilities*”, “*blast\_utilities*”, and “*biopython\_utilities*”. The “*utilities*” mostly contains simple scripts to process FASTA files along with some other general scripts (e.g. remove the last line of a set of files or deinterleave FASTQ files). Similarly, the “*blast\_utilities*”, and “*biopython\_utilities*” contain reusable scripts that require BLAST or Biopython.

### 3.2 Pipeline development

We have been working on the development of Compi pipelines for large-scale detection of positively selected amino acid sites [13]. These pipelines, listed in Table 2, are part of the pegi3s Bioinformatics Docker images project. Such developments were leveraged by the pegi3s Docker images already available for commonly used Bioinformatics software, meaning that there is a Docker image for every pipeline step requiring a new software tool. In addition, scripts for common tasks were added to the utilities images presented before. This way, pipelines are defined in a way that they use “*docker run*” commands every time an external software is needed, and all used Docker images are from the pegi3s project. The only exception is FastScreen, which was the first pipeline to be developed and includes the dependencies in the same Docker image. As Table 2 shows, all the pipelines are publicly available at Compi Hub [14] and the corresponding Docker images are available at the project’s repository at Docker Hub.

Table 2. Docker images of Compi pipelines for Positively Selected Sites (PSS) identification.

<sup>12</sup> <https://github.com/pegi3s/dockerfiles/blob/master/tutorials/singularity.md>

Pipeline	Alias (GitHub and Docker Hub)	Compi Hub ID ( <a href="https://www.sing-group.org/compihub/explore/&lt;id&gt;">https://www.sing-group.org/compihub/explore/&lt;id&gt;</a> )
FastScreen	pss-fs	5d5bb64f6d9e31002f3ce30a
GenomeFastScreen [15]	pss-genome-fs	5e2eaacce1138700316488c1
IPSSA (Integrated Positively Selected Sites Analyses)	ipssa	5fa91806407682001ad3a1e9
Auto-PSS-Genome (Automatic Positively Selected Sites Genome)	auto-pss- genome	5faa52ccf05e940c9c2762e4

### 3.3 Containerization of applications

We also took advantage of Docker to containerize the software developed by us and make their installation easier to end users. This was the case of our desktop tools with GUIs, namely ADOPS [16], BDBM [17], and SEDA [11]. In the cases of ADOPS and SEDA, we simply created Docker images with the tools inside and then the GUIs can be accessed by simply sharing the Host's XServer with the container (creating a volume with `“-v $HOME/.Xauthority:/home/developer/.Xauthority”`) and the host's DISPLAY environment variable to the container (adding `“-e DISPLAY=$DISPLAY”`). This method only works for Linux OS. In the case of BDBM, the Docker image incorporates an XPRa server and, therefore, Linux, Windows, and OS X installers could be created using platform-specific XPRa clients<sup>13</sup>.

ADOPS, BDBM, and SEDA, are desktop tools implemented in Java that run the external software dependencies (BLAST, T-Coffee, EMBOSS tools, Augustus, among others) as system processes. This means that their dependencies must be available in the system they are executed and, therefore, they must be included in the Docker image. In the case of SEDA, we included an additional execution mode of external dependencies through Docker images. This way, the user can provide an image name (SEDA is equipped with a set of default images) and SEDA uses it to execute the corresponding software tools. This also simplifies the containerization of SEDA itself, since only SEDA and Docker must be included in the Docker image. We also relied on Docker to develop and deploy EvoPPI [18].

EvoPPI is a web application for the comparison of multiple interactomes datasets from the same or different species. From an architectural point of view, EvoPPI is composed of a Single Page Application frontend implemented with Angular that communicates with a REST backend implemented with Java EE and that stores the information in a MySQL database. This means that frontend, backend and database can be seen as independent components. Based on this, a Docker container was defined for each component with the necessary dependencies for its execution. In addition, to orchestrate the execution of these containers, a Docker Compose configuration was defined, in which all the necessary services (e.g. network, persistent volumes, among others) are created. On the other hand, when comparing interactomes of different species in EvoPPI, it is necessary to make use of BLAST to establish orthologies. In this case, we also use Docker containers to run BLAST, but with the particularity that the backend application directly requests the execution of

<sup>13</sup> <https://xpra.org>

the container through the Docker Java API<sup>14</sup>. This EvoPPI deployment configuration is available in a GitHub repository<sup>15</sup>, so that any researcher can run EvoPPI in a simple way.

#### 4. Discussion

The usage of container technologies, especially Docker, is now ubiquitous in bioinformatics, as it offers a way to improve reproducibility of analyses and provides an easy way to migrate complex pipelines. The usage of Docker images provides many benefits to both biologists and bioinformaticians beyond reproducibility. It simplifies dependency management since programs must be installed only once (when the Docker image is created) and then they can be used in any system with Docker installed. This also allows having multiple versions of the same software available, something that can be difficult without containers. The maintenance of different software versions is important not only for reproducibility but also in those cases where the most recent software version do not include all functionalities of previous versions, usually due to licensing issues. This is the reason why in our home project page we list two different versions for the Genome Analysis Toolkit (*gatk-3* and *gatk-4*). In addition, since there are Docker clients for Windows and OS X systems, bioinformatics software that is only available in Linux systems can be used in such operating systems.

Our experience with Docker has been very positive, since all the software tools used at the pegi3s laboratory are now available as Docker images, and thus no time is wasted by new students/researchers with software installation and configuration when starting a new project. Having a list of tools that are classified in broad categories also eases the process of finding the most adequate tool(s) for a given project. For instance, for a project involving the *de novo* assembly of a genome there are four Docker images available, namely, *abyss*, *edena*, *soapdenovo2*, and *spades*. This is not an exhaustive list of the software available for this purpose, but it is a good start. Moreover, since these Docker images have been developed for our research and teaching needs, they are not a random collection of software tools. This means that the researcher visiting the pegi3S Bioinformatics Docker Images Project page will find there as well Docker images for read quality evaluation and to perform read trimming, for instance. The clear instructions given at the pegi3S Bioinformatics Docker Images Project, as well as the direct links to the software manuals, also greatly eases the process of finding the appropriate software parameters for a given processing step. The pipelines and desktop tools requiring external software applications that have been developed by us also greatly benefited from the available pegi3s Docker images. Moreover, they are a useful resource for teaching bioinformatics-related subjects at university. The philosophy of the pegi3S Bioinformatics Docker Images Project closely matches that of Canonical's Ubuntu open source collaborative project, the source image used in all our Docker images.

---

<sup>14</sup> <https://github.com/docker-java/docker-java>

<sup>15</sup> <https://github.com/sing-group/evoppi-docker>



According to Canonical<sup>16</sup>, Ubuntu is an ancient African word meaning ‘humanity to others’ that remind us that ‘I am what I am because of who we all are’.

### Acknowledgments

This work was financed by the National Funds through FCT—Fundação para a Ciência e a Tecnologia, I.P., under the project UIDB/04293/2020 and through the individual scientific employment program-contract with Hugo López-Fernández (2020.00515.CEECIND), and also by BioData.pt (project 22231/01/SAICT/2016). This work was also partially supported by the Consellería de Educación, Universidades e Formación Profesional (Xunta de Galicia) under the scope of the strategic funding ED431C2018/55-GRC Competitive Reference Group.

### References

1. Perkel, J.M.: Workflow systems turn raw data into scientific knowledge. *Nature*. 573, 149–150 (2019). <https://doi.org/10.1038/d41586-019-02619-z>.
2. Gomes, J., Bagnaschi, E., Campos, I., David, M., Alves, L., Martins, J., Pina, J., López-García, A., Orviz, P.: Enabling rootless Linux Containers in multi-user environments: The udocker tool. *Computer Physics Communications*. 232, 84–97 (2018). <https://doi.org/10.1016/j.cpc.2018.05.021>.
3. Gruening, B., Sallou, O., Moreno, P., da Veiga Leprevost, F., Ménager, H., Søndergaard, D., Röst, H., Sachsenberg, T., O’Connor, B., Madeira, F., Dominguez Del Angel, V., Crusoe, M.R., Varma, S., Blankenberg, D., Jimenez, R.C., BioContainers Community, Perez-Riverol, Y.: Recommendations for the packaging and containerizing of bioinformatics software. *F1000Res*. 7, 742 (2019). <https://doi.org/10.12688/f1000research.15140.2>.
4. Nüst, D., Sochat, V., Marwick, B., Eglen, S.J., Head, T., Hirst, T., Evans, B.D.: Ten simple rules for writing Dockerfiles for reproducible data science. *PLOS Computational Biology*. 16, e1008316 (2020). <https://doi.org/10.1371/journal.pcbi.1008316>.
5. Belmann, P., Dröge, J., Bremges, A., McHardy, A.C., Sczyrba, A., Barton, M.D.: Bioboxes: standardised containers for interchangeable bioinformatics software. *GigaScience*. 4, (2015). <https://doi.org/10.1186/s13742-015-0087-0>.
6. Moreews, F., Sallou, O., Ménager, H., Le bras, Y., Monjeaud, C., Blanchet, C., Collin, O.: BioShaDock: a community driven bioinformatics shared Docker-based tools registry. *F1000Res*. 4, 1443 (2015). <https://doi.org/10.12688/f1000research.7536.1>.
7. da Veiga Leprevost, F., Grüning, B.A., Alves Aflitos, S., Röst, H.L., Uszkoreit, J., Barsnes, H., Vaudel, M., Moreno, P., Gatto, L., Weber, J., Bai, M., Jimenez, R.C., Sachsenberg, T., Pfeuffer, J., Vera Alvarez, R., Griss, J., Nesvizhskii, A.I., Perez-Riverol, Y.: BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*. 33, 2580–2582 (2017). <https://doi.org/10.1093/bioinformatics/btx192>.
8. Menegidio, F.B., Jabes, D.L., Costa de Oliveira, R., Nunes, L.R.: Dugong: a Docker image, based on Ubuntu Linux, focused on reproducibility and replicability for bioinformatics analyses. *Bioinformatics*. 34, 514–515 (2018). <https://doi.org/10.1093/bioinformatics/btx554>.

---

<sup>16</sup> <https://ubuntu.com/about>

9. O'Connor, B.D., Yuen, D., Chung, V., Duncan, A.G., Liu, X.K., Patricia, J., Paten, B., Stein, L., Ferretti, V.: The Dockstore: enabling modular, community-focused sharing of Docker-based genomics tools and workflows. *F1000Res.* 6, 52 (2017). <https://doi.org/10.12688/f1000research.10137.1>.
10. Jackman, S.D., Mozgacheva, T., Chen, S., O'Huiginn, B., Bailey, L., Birol, I., Jones, S.J.M.: ORCA: a comprehensive bioinformatics container environment for education and research. *Bioinformatics.* 35, 4448–4450 (2019). <https://doi.org/10.1093/bioinformatics/btz278>.
11. López-Fernández, H., Duque, P., Vázquez, N., Fdez-Riverola, F., Reboiro-Jato, M., Vieira, C.P., Vieira, J.: SEDA: a Desktop Tool Suite for FASTA Files Processing. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.* 1–1 (2020). <https://doi.org/10.1109/TCBB.2020.3040383>.
12. López-Fernández, H., Graña-Castro, O., Nogueira-Rodríguez, A., Reboiro-Jato, M., Glez-Peña, D.: Compi: a framework for portable and reproducible pipelines. *PeerJ Computer Science.* 7, e593 (2021). <https://doi.org/10.7717/peerj-cs.593>.
13. López-Fernández, H., Duque, P., Vázquez, N., Fdez-Riverola, F., Reboiro-Jato, M., Vieira, C.P., Vieira, J.: Inferring Positive Selection in Large Viral Datasets. In: Fdez-Riverola, F., Rocha, M., Mohamad, M.S., Zaki, N., and Castellanos-Garzón, J.A. (eds.) *Practical Applications of Computational Biology and Bioinformatics, 13th International Conference.* pp. 61–69. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-23873-5\\_8](https://doi.org/10.1007/978-3-030-23873-5_8).
14. Nogueira-Rodríguez, A., López-Fernández, H., Graña-Castro, O., Reboiro-Jato, M., Glez-Peña, D.: Compi Hub: A Public Repository for Sharing and Discovering Compi Pipelines. In: Panuccio, G., Rocha, M., Fdez-Riverola, F., Mohamad, M.S., and Casado-Vara, R. (eds.) *Practical Applications of Computational Biology & Bioinformatics, 14th International Conference (PACBB 2020).* pp. 51–59. Springer International Publishing, Cham (2021). [https://doi.org/10.1007/978-3-030-54568-0\\_6](https://doi.org/10.1007/978-3-030-54568-0_6).
15. López-Fernández, H., Vieira, C.P., Fdez-Riverola, F., Reboiro-Jato, M., Vieira, J.: Inferences on Mycobacterium Leprae Host Immune Response Escape and Antibiotic Resistance Using Genomic Data and GenomeFastScreen. In *Practical Applications of Computational Biology & Bioinformatics, 14th International Conference (PACBB 2020).* pp. 42–50. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-54568-0\\_5](https://doi.org/10.1007/978-3-030-54568-0_5).
16. Reboiro-Jato, D., Reboiro-Jato, M., Fdez-Riverola, F., Vieira, C.P., Fonseca, N.A., Vieira, J.: ADOPS--Automatic Detection Of Positively Selected Sites. *J Integr Bioinform.* 9, 200 (2012). <https://doi.org/10.2390/biecoll-jib-2012-200>.
17. Vázquez, N., López-Fernández, H., Vieira, C.P., Fdez-Riverola, F., Vieira, J., Reboiro-Jato, M.: BDBM 1.0: A Desktop Application for Efficient Retrieval and Processing of High-Quality Sequence Data and Application to the Identification of the Putative Coffea S-Locus. *Interdiscip Sci Comput Life Sci.* 11, 57–67 (2019). <https://doi.org/10.1007/s12539-019-00320-3>.
18. Vázquez, N., Rocha, S., López-Fernández, H., Torres, A., Camacho, R., Fdez-Riverola, F., Vieira, J., Vieira, C.P., Reboiro-Jato, M.: EvoPPI 1.0: a Web Platform for Within- and Between-Species Multiple Interactome Comparisons and Application to Nine PolyQ Proteins Determining Neurodegenerative Diseases. *Interdiscip Sci Comput Life Sci.* 11, 45–56 (2019). <https://doi.org/10.1007/s12539-019-00317-y>.