Asc-Seurat: Analytical single-cell Seurat-based web application

Wendell J. Pereira and Felipe M. Almeida

Feb 01, 2022

GENERAL INFORMATION

Asc-Seurat (Analytical single-cell Seurat-based web application) is a web application based on Shiny[1]. Pronounced as "ask Seurat", it provides a click-based, easy-to-install, and easy-to-use interface that allows the execution of all steps necessary for scRNA-seq analysis (See *Asc-Seurat workflow*). It integrates many of the capabilities of the Seurat[2] and Dynverse[3] and also allows an instantaneous functional annotation of genes of interest using BioMart[4].

Asc_seurat relies on multiple R packages. Please, visit the *references* and check the complete list of packages and their references.

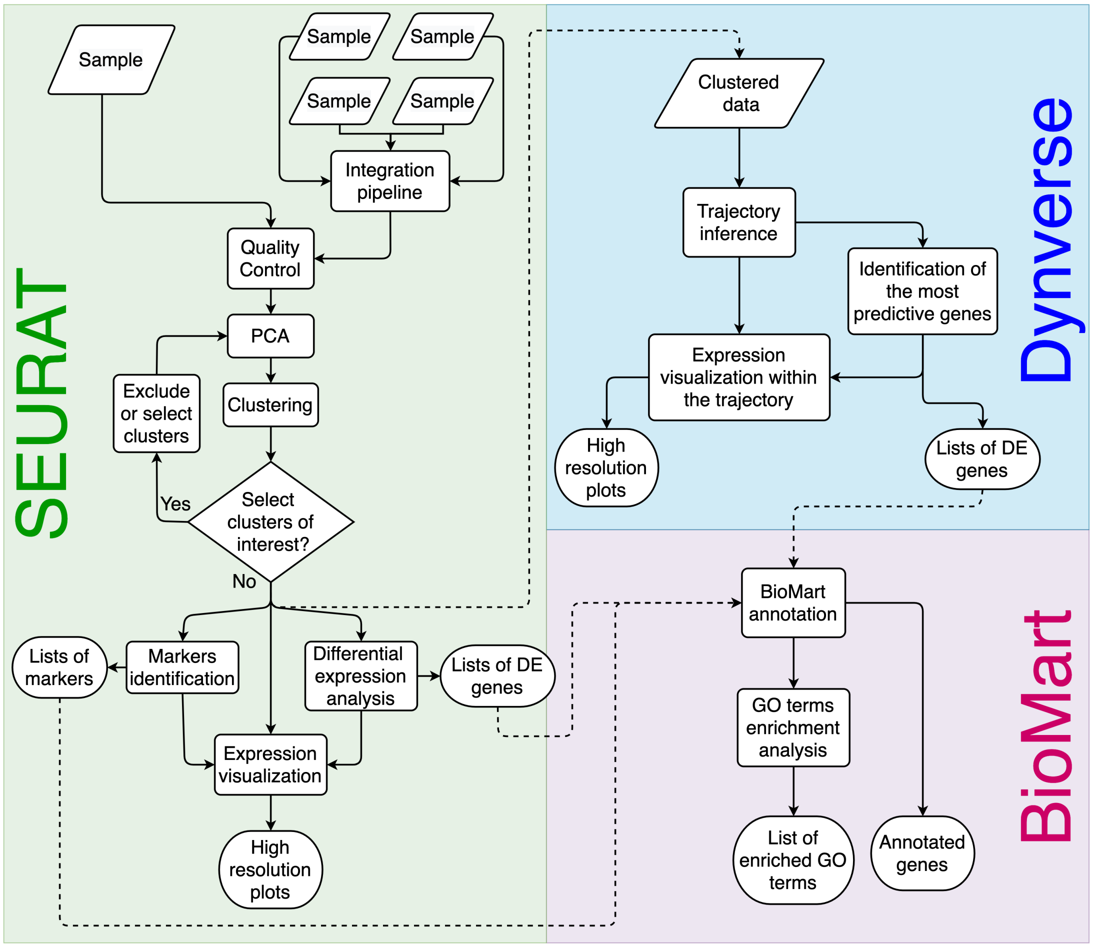


Fig. 1: **Asc-Seurat workflow overview.** Asc-Seurat is built on three analytical cores. Using Seurat, it is possible to explore scRNA-seq data of a population of cells to identify patterns that reflect the cell types of a sample(s) and identify markers and DEGs for each cell type/cluster. By incorporating Dynverse, Asc-Seurat allows the utilization of dozens of models to infer and visualize developmental trajectories (V and VI) and to identify genes differentially expressed on those trajectories (VII). Finally, using BioMart, Asc-Seurat allows immediate functional annotation and GO terms enrichment analysis for many species.

[1] shiny.rstudio.com/
[2] satijalab.org/seurat/
[3] dynverse.org
[4] www.biomart.org

CHAPTER

ONE

INSTALLATION

1.1 Dependencies

Asc-Seurat relies on multiple R packages and their dependencies (See *References*). However, we provide a Docker image that contains all necessary software and packages.

To install Asc-Seurat, it is necessary to have Docker installed on the machine. Docker needs to be correctly installed and configured in the user's machine. Check the installation instructions provided by Docker at https://docs.docker.com/engine/install.

> **Warning:** Single-cell RNA-seq data analysis can be resource-consuming. By default, Docker will use (allocate) only a fraction of your RAM memory. A minimum requirement of 8 Gb of RAM memory was necessary to analyze a dataset containing around eight thousand cells during our tests. Therefore, users need to adjust the amount of allocated memory according to their dataset. Please visit: https://docs.docker.com/docker-for-mac/space/ (MAC) or https://docs.docker.com/docker-for-windows/ (Windows) to learn how to make this adjustment.

1.1.1 Image download

After installing Docker, users can download the Docker image containing Asc-Seurat by executing the command below. The installation is quick and straightforward. After that, everything is set.

```
# Download the docker image:
docker pull kirstlab/asc_seurat
```

1.2 Starting Asc-Seurat

After downloading the image, users can start the app on their working directory. See below for the instructions on how to start the app in the different operational systems.

> **Note:** During the first execution, some folders will be created in the working directory. They include the folders `data/` and `RDS_files/` that users will use to store their datasets, allowing Asc-Seurat to read them.
>
> Always start the run inside the working directory to be able to use the data inside these folders.

1.2.1 For macOS and Linux

Tip: The code below will automatically update Asc-Seurat to the latest version. You can download and execute a specific version of Asc-Seurat by adding the version's tag to the image's name, i.e., replace `kirstlab/asc_seurat` by `kirstlab/asc_seurat:v.1.0` to use v1.0.

```
# Create the working directory
mkdir my_project
cd my_project

# Starts Asc-Seurat
docker pull kirstlab/asc_seurat && docker run -v $(pwd):/app/user_work -v /var/run/
↪docker.sock:/var/run/docker.sock -d --name Asc_Seurat --rm -p 3838:3838 kirstlab/
↪asc_seurat
```

Note: After executing the "docker run" command, open your preferred web browser and paste the address http://localhost:3838/. Asc-Seurat should be ready.

If users want to kill the Docker container, run the command below.

```
docker kill Asc_seurat
```

1.2.2 For Windows

To run Asc-Seurat on Windows via Docker, it is necessary to use Windows 10. Moreover, Windows Subsystem for Linux (WSL) needs to be installed. Before running Asc-Seurat, users must guarantee that Docker and its WSL 2 components are correctly installed and running. For that, check the two (sequential) tutorials below:

1. Docker installation info
2. Define windows WSL 2 as default (If you followed the link above correctly, you only need to execute step 5 of this tutorial).

The tutorials above contain all the necessary information to install Docker on Windows. However, it is also possible to find video tutorials on YouTube. Check the following link for an example: https://youtu.be/5nX8U8Fz5S0 .

After certifying that everything is working, Asc-Seurat can be started using the commands below:

Tip: The code below will automatically update Asc-Seurat to the latest version. You can download and execute a specific version of Asc-Seurat by adding the version tag to the image's name, i.e., replace `kirstlab/asc_seurat` by `kirstlab/asc_seurat:v.1.0` to use v1.0.

```
# Create the working directory
mkdir my_project
cd my_project

# If using Windows CMD
docker pull kirstlab/asc_seurat && docker run -v %cd%:/app/user_work -v /var/run/
↪docker.sock:/var/run/docker.sock -d --rm -p 3838:3838 kirstlab/asc_seurat
```

(continues on next page)

(continued from previous page)

```
# If using Windows Powershell
docker pull kirstlab/asc_seurat && docker run -v ${PWD}:/app/user_work -v /var/run/
↪docker.sock:/var/run/docker.sock -d --rm -p 3838:3838 kirstlab/asc_seurat
```

Note: After executing the "docker run" command, open your preferred web browser and paste the address http://localhost:3838/. Asc-Seurat should be ready.

If users want to kill the Docker container, run the command below.

```
docker kill Asc_seurat
```

CHAPTER

TWO

REFERENCES

Asc-Seurat is built on the work of many other people and relies on a diversity of R packages. These packages, in turn, have many dependencies. Here, we list all packages that Asc-Seurat directly calls.

2.1 Analytical core

Three packages are the analytical core of Asc-Seurat. Below is listed their information.

- Seurat
  - web page: https://satijalab.org/seurat/
  - Publications:
    - Satija, R. et al. (2015) Spatial reconstruction of single-cell gene expression data. Nat. Biotechnol., 33, 495–502.
    - Stuart, T. et al. (2019) Comprehensive Integration of Single-Cell Data. Cell, 177, 1888–1902.e21.
- Dynverse (dynplot, dynwrap, and dynfeature)
  - web page: https://dynverse.org/
  - Publications:
    - Saelens, W. et al. (2019) A comparison of single-cell trajectory inference methods. Nat. Biotechnol., 37, 547–554.
- biomaRt
  - web page: https://bioconductor.org/packages/release/bioc/html/biomaRt.html
  - Publications:
    - Durinck, S. et al. (2009) Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Nat. Protoc., 4, 1184–1191.

2.2 Additional packages

2.2.1 CRAN

- circlize: https://cran.r-project.org/web/packages/circlize/
- DT: https://cran.r-project.org/web/packages/DT/
- dplyr: https://mran.microsoft.com/web/packages/dplyr/index.html

- future: https://cran.r-project.org/web/packages/future/index.html
- ggplot2: https://cran.r-project.org/web/packages/ggplot2/index.html
- ggthemes: https://cran.r-project.org/web/packages/ggthemes/index.html
- hdf5r: https://cran.r-project.org/web/packages/hdf5r/
- metap: https://cran.r-project.org/web/packages/metap/index.html
- patchwork: https://cran.r-project.org/web/packages/patchwork/index.html
- rclipboard: https://cran.r-project.org/web/packages/rclipboard/index.html
- reactable: https://cran.r-project.org/web/packages/reactable/index.html
- scales: https://cran.r-project.org/web/packages/scales/index.html
- sctransform: https://cran.r-project.org/web/packages/sctransform/index.html
- SeuratObject: https://cran.r-project.org/web/packages/SeuratObject/index.html
- shiny: https://cran.r-project.org/web/packages/shiny/index.html
- shinycssloaders: https://cran.r-project.org/web/packages/shinycssloaders/index.html
- shinyFeedback: https://cran.r-project.org/web/packages/shinyFeedback/index.html
- shinyWidgets: https://cran.r-project.org/web/packages/shinyWidgets/index.html
- tidyverse: https://cran.r-project.org/web/packages/tidyverse/index.html
- utils: https://cran.r-project.org/web/packages/R.utils/index.html

2.2.2 Bioconductor

- ComplexHeatmap: https://bioconductor.org/packages/release/bioc/html/ComplexHeatmap.html
- glmGamPoi: https://bioconductor.org/packages/release/bioc/html/glmGamPoi.html
- multtest: https://bioconductor.org/packages/release/bioc/html/multtest.html
- SingleCellExperiment: https://bioconductor.org/packages/release/bioc/html/SingleCellExperiment.html
- slingshot: https://bioconductor.org/packages/release/bioc/html/slingshot.html
- topGO: https://bioconductor.org/packages/release/bioc/html/topGO.html
- tradeSeq: https://bioconductor.org/packages/release/bioc/html/tradeSeq.html

CHAPTER

THREE

LICENSE

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <https://www.gnu.org/licenses/why-not-lgpl.html>.

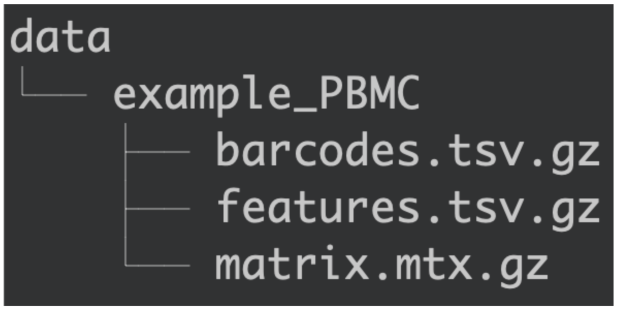
CHAPTER

FOUR

LOADING THE DATA OF AN INDIVIDUAL SAMPLE

4.1 Location of the dataset

For Asc-Seurat to read the datasets, they need to be located in a subdirectory inside the `data/` directory. The `data/` directory will be created during the installation and contains a subdirectory with an example dataset called `example_PBMC/`. This dataset is from the publicly available 10×'s Peripheral Blood Mononuclear Cells (PBMC) and contains 2700 cells.



```
data
└── example_PBMC
    ├── barcodes.tsv.gz
    ├── features.tsv.gz
    └── matrix.mtx.gz
```

Fig. 1: Organization of the `data/` directory.

Therefore, to add the dataset, create a subdirectory inside `data/` containing the counts' matrix (*matrix.mtx.gz*), cell barcodes (*barcodes.tsv.gz*), and gene names (*features.tsv.gz*).

Asc-Seurat provides separated environments (tabs) to analyze a single sample and the integrated analysis of multiple samples.

4.2 Format of the dataset

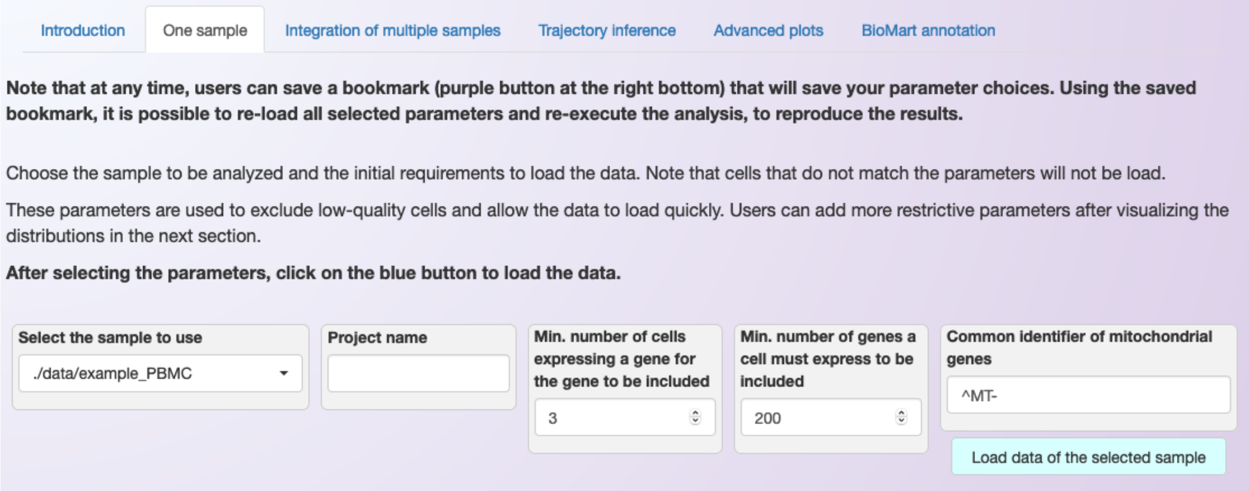
Asc-Seurat can only read the input files in the format generated by Cell Ranger (10x genomics). However, it is possible to convert your counts' matrix to the acceptable format. For example, the function write10xCounts(), from the DropletUtils package, is an easy option to make this conversion.

Tip: Using write10xCounts(), users can provide as output the path to the `data/` directory. In this way, Asc-Seurat can recognize the files automatically.

4.3 Loading the data

To analyze an individual sample, select the second tab in the web application, named `One sample`. Then, choose the sample to analyze and set the initial criteria to exclude cells that should not be load, as shown below.

> After inserting the datasets in the `data/` directory, the samples will be available to load in Asc-Seurat, as shown below.



Note that at any time, users can save a bookmark (purple button at the right bottom) that will save your parameter choices. Using the saved bookmark, it is possible to re-load all selected parameters and re-execute the analysis, to reproduce the results.

Choose the sample to be analyzed and the initial requirements to load the data. Note that cells that do not match the parameters will not be load.

These parameters are used to exclude low-quality cells and allow the data to load quickly. Users can add more restrictive parameters after visualizing the distributions in the next section.

After selecting the parameters, click on the blue button to load the data.

Select the sample to use: ./data/example_PBMC

Project name:

Min. number of cells expressing a gene for the gene to be included: 3

Min. number of genes a cell must express to be included: 200

Common identifier of mitochondrial genes: ^MT-

Fig. 2: Example of how to load an individual sample for analysis and of the requested initial parameters.

In the first box to the left, it is possible to select the sample to use. However, there are a few parameters that need to provide before loading the data. This step is based on Seurat's functions CreateSeuratObject and PercentageFeatureSet. Between parenthesis, we list the name of the parameter in the CreateSeuratObject function.

Below is a description of these parameters:

- **Project name**: Sets the name for the project. The name will appear in some of the plots, but it is not required (project).
- **Min. number of cells expressing a gene**: Include genes only if they are detected in at least this many cells (min.cells).
- **Min. number of genes a cell must express to be included**: Include cells only if they expressed at least this number of genes (min.features).
- **Regex to identify mitochondrial genes**: Here, the regular expression (Regex) is a sequence of characters that is used to determine the genes belonging to the mitochondrial genome (pattern). For example, when using the human genome, this sequence should be "^MT-".

After setting the parameters described above, click on the button *Load data of the selected sample* to start the analysis. A violin plot showing the distribution of cells will appear. This plot can then be used to set more restrictive parameters for *quality control*.

CHAPTER

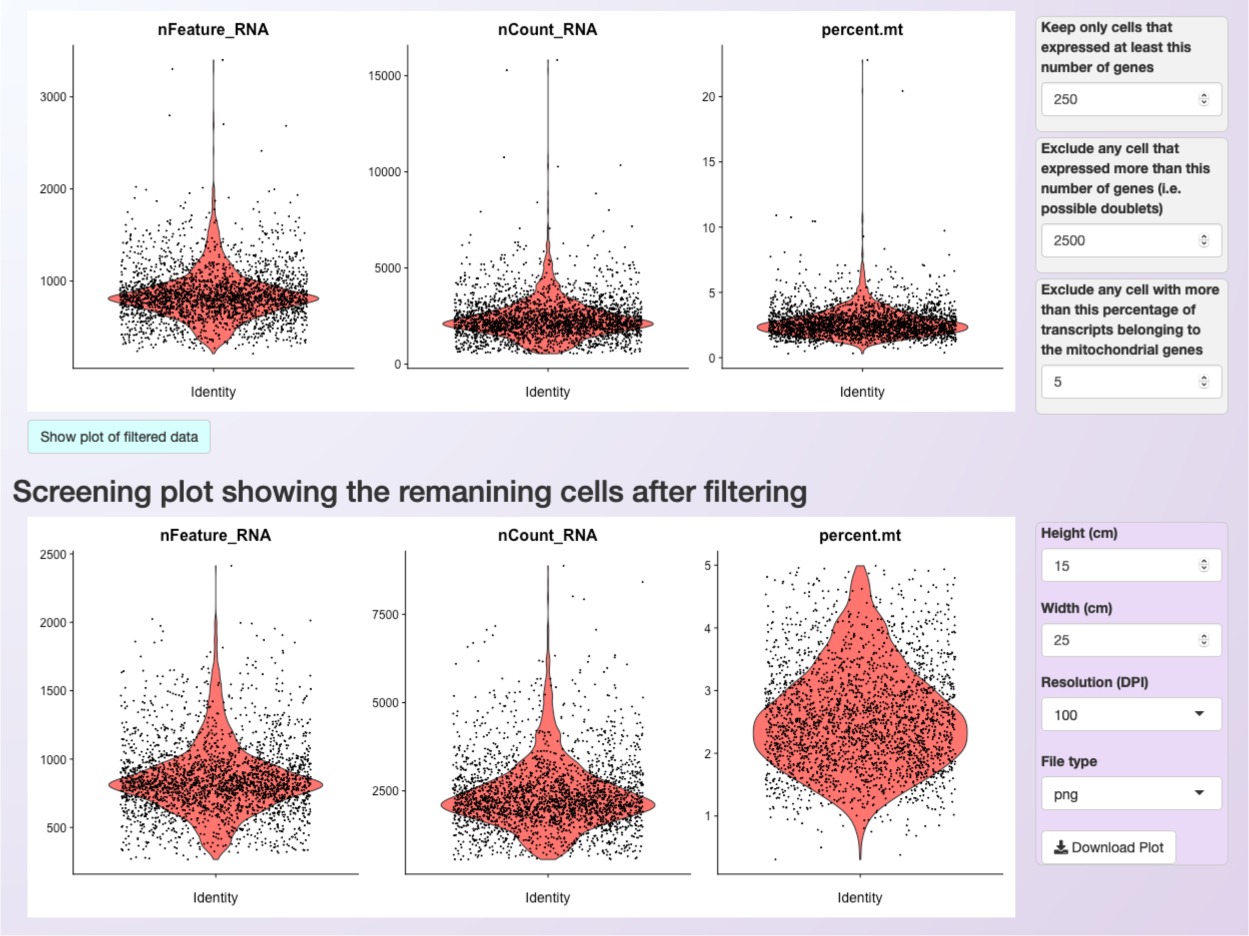
FIVE

QUALITY CONTROL

After loading the data, a violin plot will be generated showing the distribution of cells according to three parameters:

- nFeature_RNA: the number of genes detected in each cell
- nCount_RNA: the number of molecules detected per cell
- percent.mt: the percentage of transcripts that map to mitochondrial genes

After visualizing the distribution of cells, it is possible to set more restrictive parameters (on the right side of the plot) and filter cells based on the number of expressed genes per cell and the percentage of transcripts from mitochondrial genes. By clicking on *Show plot of filtered data*, users can see the distribution of cells after filtering and then readjust the parameters. The figure below shows the distribution of cells of the PBMC dataset before and after filtering.



Keep only cells that expressed at least this number of genes: 250

Exclude any cell that expressed more than this number of genes (i.e. possible doublets): 2500

Exclude any cell with more than this percentage of transcripts belonging to the mitochondrial genes: 5

Screening plot showing the remaining cells after filtering

Height (cm): 15

Width (cm): 25

Resolution (DPI): 100

File type: png

Asc-Seurat allows users to download each of the plots with high-resolution by clicking on the `Download plot` button.

CHAPTER

SIX

CLUSTERING

After filtering the data to remove low-quality cells, Asc-Seurat allows clustering the remaining cells according to their expression profiles. However, before clustering, a series of steps are executed, including normalization, scaling (if using LogNormalization), and dimensional reduction via PCA.

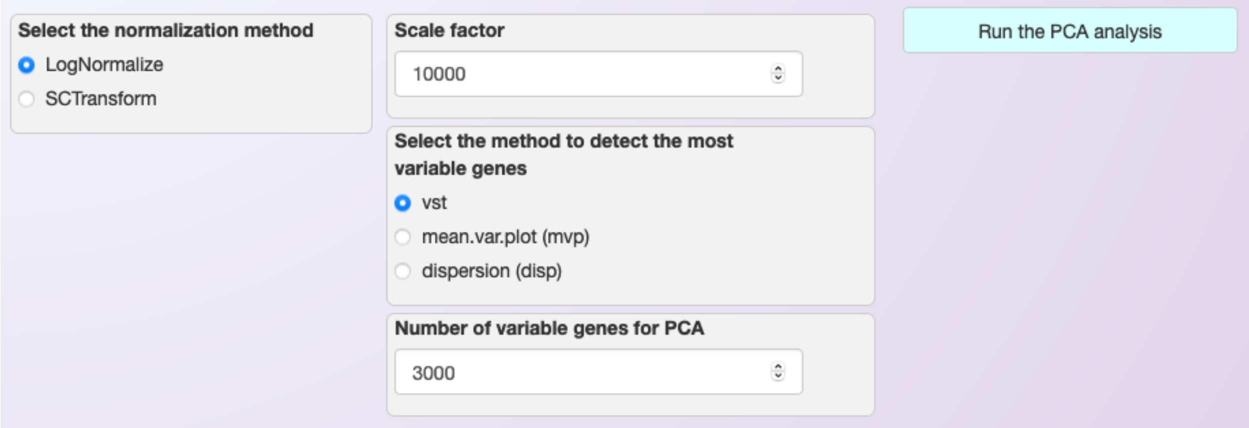
Moreover, users need to decide how many dimensions are to be used during the clustering after executing the PCA. Asc-Seurat provides an elbow plot to inform this decision. Below are instructions on how to perform the clustering depending on the normalization method of choice.

6.1 Normalization

6.1.1 LogNormalization

Asc-Seurat allows the normalization using Seurat's LogNormalize function. Users have the option to change the scaling factor if necessary, but it is typically not needed. In the same window (see the image below), users can select what method should be used to identify the most variable genes and how many of the most variable genes should be used during the dimension reduction (PCA).

The most variable genes exhibit high cell-to-cell variation in the dataset and therefore are more informative. We use Seurat's function FindVariableFeatures. The default setting should work well for the majority of cases.



Select the normalization method

● LogNormalize

○ SCTransform

Scale factor

10000

Select the method to detect the most variable genes

● vst

○ mean.var.plot (mvp)

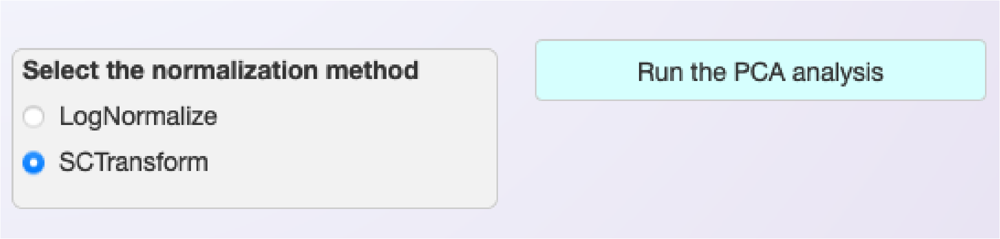
○ dispersion (disp)

Number of variable genes for PCA

3000

6.1.2 SCTransform

The second option of normalization provided by Asc-Seurat is Seurat's Seurat's SCTransform. When using this normalization, it is unnecessary to set the scale factor or identify the most variable genes (See image below).



Select the normalization method

○ LogNormalize

● SCTransform

Note: Currently, the recommendation of Seurat's team is to use the standard "RNA" assay when performing differential expression (D.E) analysis and for data visualization, even when using SCTransform (See here). Therefore, Asc-Seurat will use the SCTransformed data ("SCT" assay) until the clustering step only.

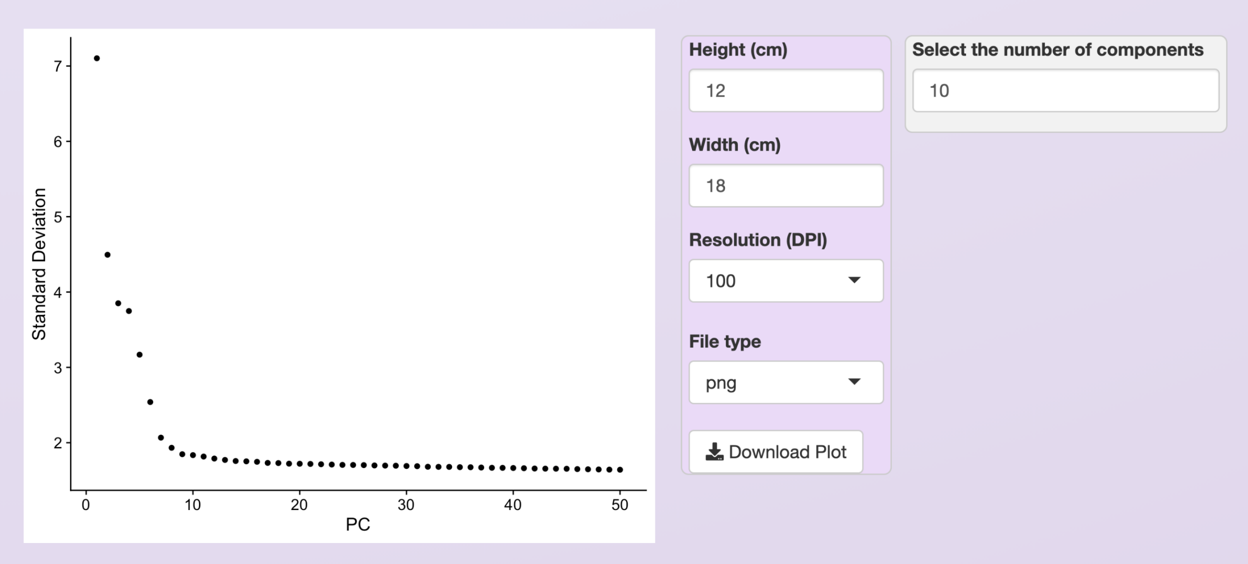
To use the "RNA" assay after SCTransform, Asc-Seurat will automatically perform the LogNormalization and scaling of the data in the RNA assay by applying the default parameters.

6.2 Dimensional reduction (PCA)

The PCA will be executed using Seurat's function RunPCA and, after its conclusion, an elbow plot is generated automatically, to help users to decide how many PCs should be included to inform the clustering step.

Users can use this plot to select the PCs with the highest standard deviation (more informative PCs). Also, users should set the number of PCs to include during clustering in the windows at the plot's right side.

In the example below, only the first 10 PCs are selected. Not that the resulting plot will be slightly different depending on the normalization method. Below we show the plot obtained using the LogNormalization.



Height (cm): 12

Width (cm): 18

Resolution (DPI): 100

File type: png

Select the number of components: 10

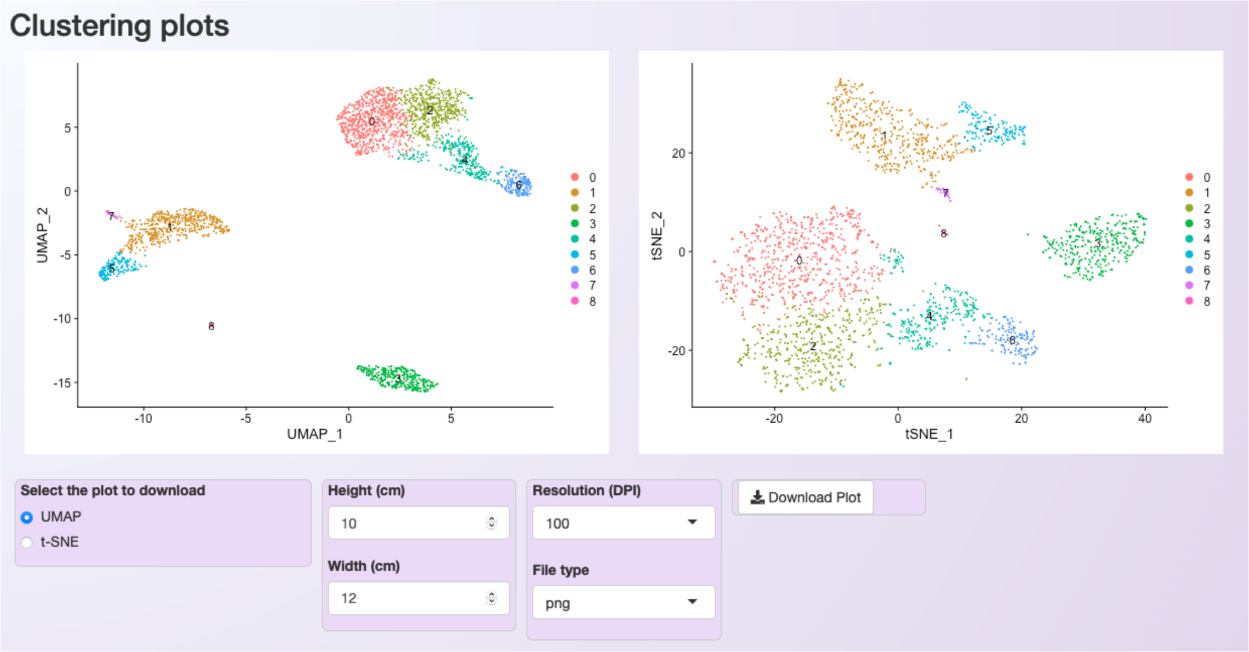
Fig. 1: Elbow plot provided to help to select the most informative PCs. For the PBMC dataset, and using the LogNormalization method, we chose the ten first PCs.

6.3 Clustering of cells

The next step is the clustering of the cells. For that, Asc-Seurat used both FindNeighbors and FindClusters functions of the Seurat package.

Before the execution, however, users need to set a value for the resolution parameter. The resolution is an important parameter to evaluate because it determines the profile and number of clusters identified for a dataset. Selecting larger values will favor splitting cells into more clusters while choosing a smaller value has the opposite effect. Quoting from Seurat's tutorial: "We find that setting this parameter between 0.6-1.2 typically returns good results for single-cell datasets of around 3K cells. Optimal resolution often increases for larger datasets".

Tip: There is no easy way to define an optimal value for the resolution parameter. Users need to try different values and evaluate the resulting clusters according to the expectation for their cells population. Visualizing the expression profile of cell-type-specific markers can provide a hint if the chosen value is too small or too large.



Select the plot to download
● UMAP
○ t-SNE

Height (cm)
10

Width (cm)
12

Resolution (DPI)
100

File type
png

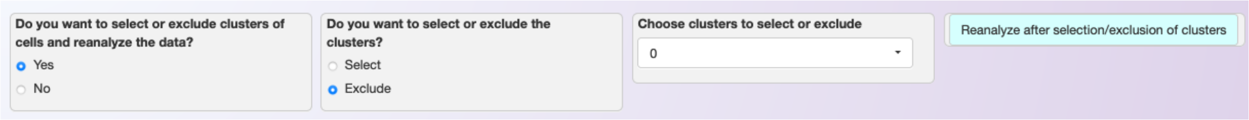
Fig. 2: Plot showing the clustering of the PBMC dataset after LogNormalization, using 10 PCs and a resolution value of 0.5.

After the execution of the clustering step, two plots are generated for cluster visualization. The first plot is generated using the Uniform Manifold Approximation and Projection (UMAP) technique (left). The second deploys the t-distributed Stochastic Neighbor Embedding (t-SNE) method (right).

6.3.1 Selecting clusters of interest

In some cases, it is interesting to select or exclude some clusters of cells from the dataset before executing the subsequent steps. This process is helpful, for example, when users desire to explore a developmental trajectory of a specific group of cell types.

Asc-Seurat makes this step simple. Users only need to select the cluster(s) to keep or exclude and start reanalysis of the remaining cells by clicking on *Reanalyze after selection/exclusion of clusters* (see below).



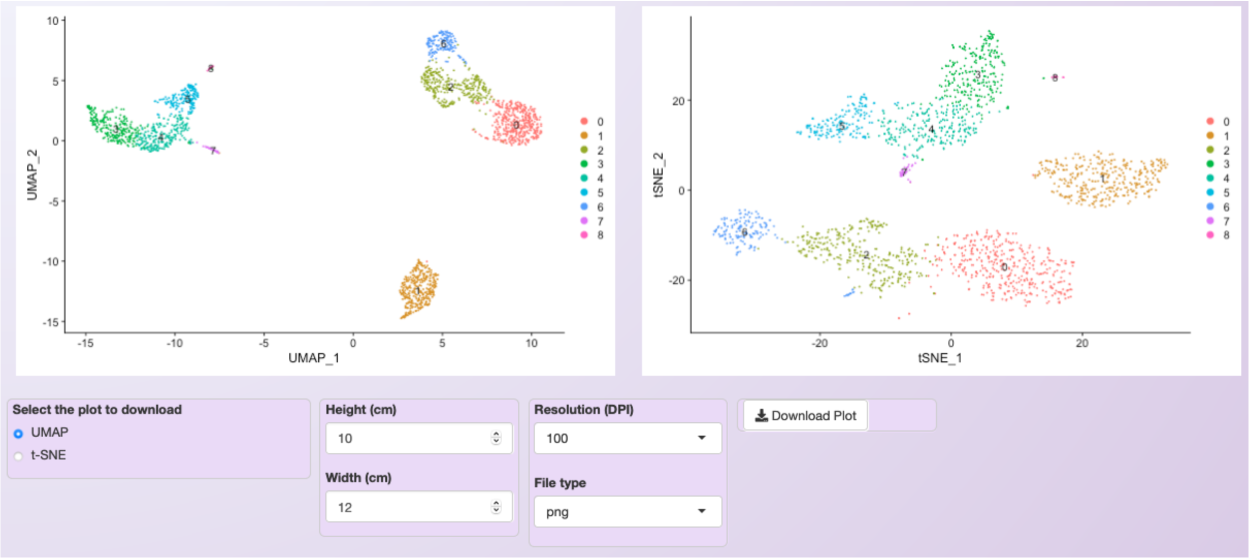
Do you want to select or exclude clusters of cells and reanalyze the data?
● Yes
○ No

Do you want to select or exclude the clusters?
○ Select
● Exclude

Choose clusters to select or exclude
0

Fig. 3: Asc-Seurat makes it easy to select or exclude a cluster (or clusters) of cells. In this example, we exclude all cells belonging to cluster 0.

Asc-Seurat will then execute the steps with the new set of cells up to the PCA. Then, **users need to evaluate the elbow plot and decide the number of PCs to cluster the new set of cells**. Users can either keep the same value for the resolution parameter or modify it before clicking on *Run the clustering analysis* to start the clustering once more.



Select the plot to download
● UMAP
○ t-SNE

Height (cm)
10

Width (cm)
12

Resolution (DPI)
100

File type
png

Fig. 4: Clustering of the PBMC dataset after excluding cells belonging to cluster 0 from the original dataset.

> **Warning:** The cluster's numbering will change every time that cluster(s) are selected or excluded.

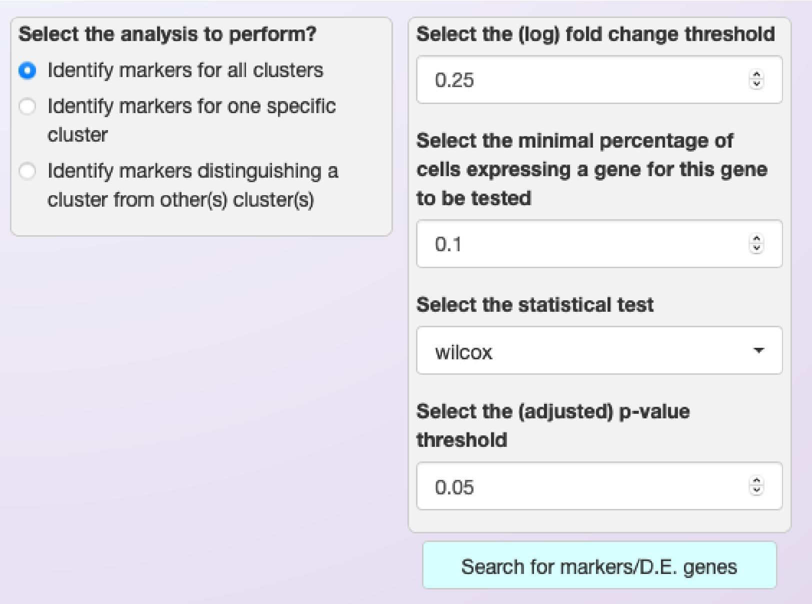
CHAPTER

SEVEN

MARKERS IDENTIFICATION AND DIFFERENTIAL EXPRESSION ANALYSIS

After clustering the cells, users may be interested in identifying genes specifically expressed in one cluster (markers) or in genes that are differentially expressed among clusters of interest. Asc-Seurat can apply multiple algorithms to identify gene markers for individual clusters or to identify differentially expressed genes (DEGs) among clusters, using Seurat's functions FindMarkers and FindAllMarkers.

Asc-Seurat allows users to filter gene markers and DEGs by the fold change and minimal percentage of cells expressing a gene in the cluster(s). Moreover, users can define a significance level to exclude genes based on the adjusted p-value (see below).



Select the analysis to perform?

● Identify markers for all clusters

○ Identify markers for one specific cluster

○ Identify markers distinguishing a cluster from other(s) cluster(s)

Select the (log) fold change threshold

0.25

Select the minimal percentage of cells expressing a gene for this gene to be tested

0.1

Select the statistical test

wilcox

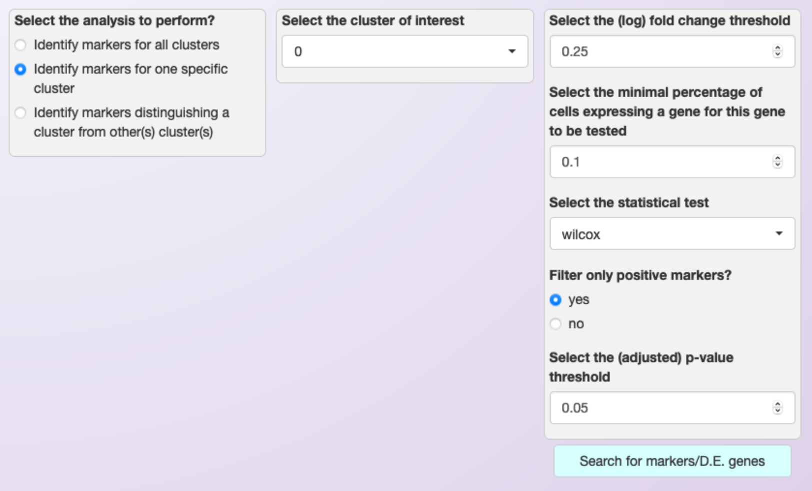
Select the (adjusted) p-value threshold

0.05

Fig. 1: Example of Asc-Seurat's interface showing the settings to the search for gene markers for each of the clusters using the Wilcox test.

An iterative table will be available after executing the search for marker or DEGs, showing the significant genes. Moreover, users can download the list of significant markers or DEGs as a csv file.

The list of genes in the csv can then be used to visualize their gene expression in a series of plots, as shown in the section *Expression visualization*.



Select the analysis to perform?

○ Identify markers for all clusters
● Identify markers for one specific cluster
○ Identify markers distinguishing a cluster from other(s) cluster(s)

Select the cluster of interest: 0

Select the (log) fold change threshold: 0.25

Select the minimal percentage of cells expressing a gene for this gene to be tested: 0.1

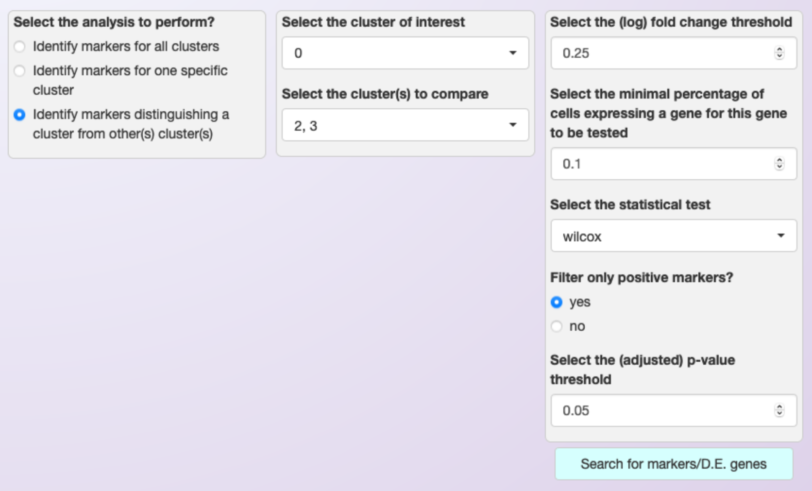
Select the statistical test: wilcox

Filter only positive markers?

● yes
○ no

Select the (adjusted) p-value threshold: 0.05

Fig. 2: Example of Asc-Seurat's interface showing the settings to the search for markers for a specific cluster (cluster 0).



Select the analysis to perform?

○ Identify markers for all clusters
○ Identify markers for one specific cluster
● Identify markers distinguishing a cluster from other(s) cluster(s)

Select the cluster of interest: 0

Select the cluster(s) to compare: 2, 3

Select the (log) fold change threshold: 0.25

Select the minimal percentage of cells expressing a gene for this gene to be tested: 0.1

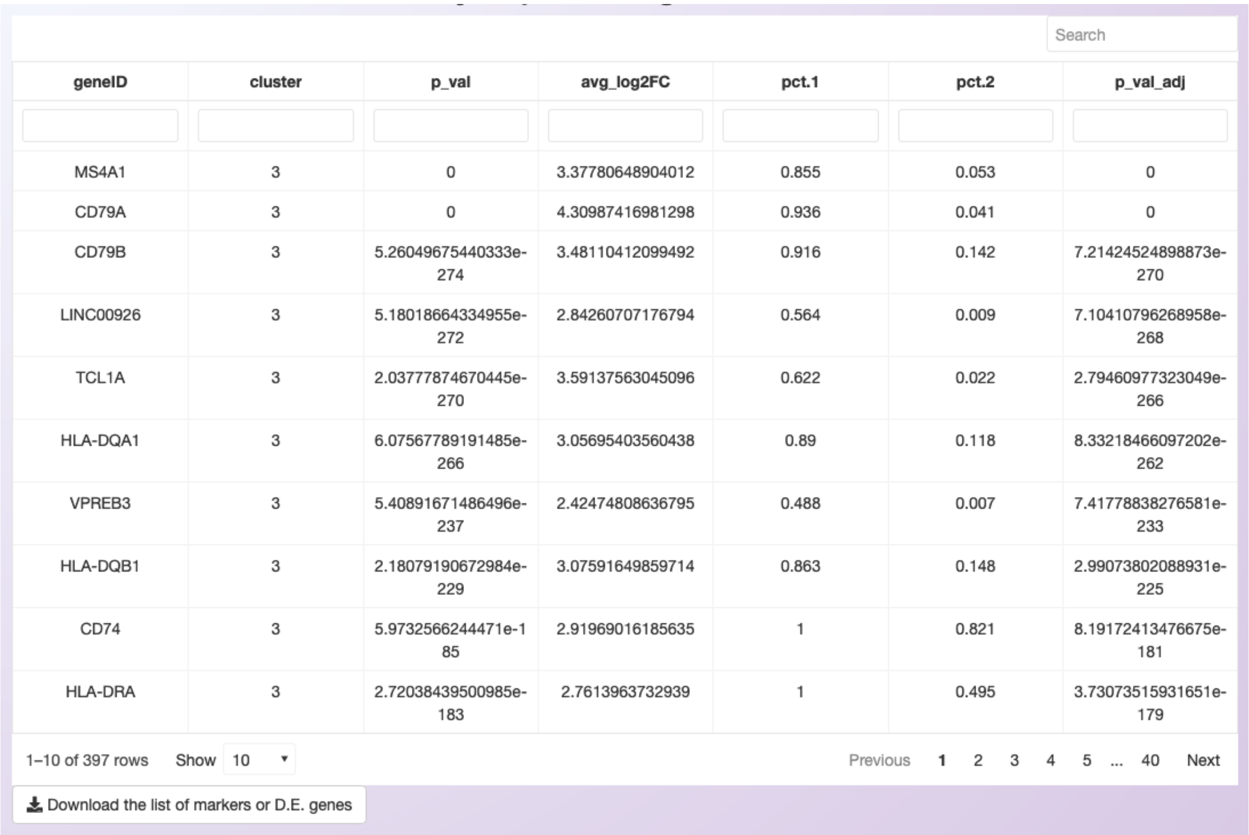
Select the statistical test: wilcox

Filter only positive markers?

● yes
○ no

Select the (adjusted) p-value threshold: 0.05

Fig. 3: Example of Asc-Seurat's interface showing the settings to search for DEGs genes among clusters 0, 2, and 3.


Search
1–10 of 397 rows
Show 10
Previous 1 2 3 4 5 ... 40 Next
Download the list of markers or D.E. genes

geneID	cluster	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
MS4A1	3	0	3.37780648904012	0.855	0.053	0
CD79A	3	0	4.30987416981298	0.936	0.041	0
CD79B	3	5.26049675440333e-274	3.48110412099492	0.916	0.142	7.21424524898873e-270
LINC00926	3	5.18018664334955e-272	2.84260707176794	0.564	0.009	7.10410796268958e-268
TCL1A	3	2.03777874670445e-270	3.59137563045096	0.622	0.022	2.79460977323049e-266
HLA-DQA1	3	6.07567789191485e-266	3.05695403560438	0.89	0.118	8.33218466097202e-262
VPREB3	3	5.40891671486496e-237	2.42474808636795	0.488	0.007	7.41778838276581e-233
HLA-DQB1	3	2.18079190672984e-229	3.07591649859714	0.863	0.148	2.99073802088931e-225
CD74	3	5.9732566244471e-185	2.91969016185635	1	0.821	8.19172413476675e-181
HLA-DRA	3	2.72038439500985e-183	2.7613963732939	1	0.495	3.73073515931651e-179

Fig. 4: List of the ten most significant markers identified for cluster 3 of the PBMC dataset (as defined in *Clustering*).

CHAPTER

EIGHT

EXPRESSION VISUALIZATION

Asc-Seurat provides a variety of plots for gene expression visualization. From a list of selected genes, it is possible to visualize the average of each gene expression in each cluster in a heatmap. It also provides plots for the visualization of gene expression at the cell level. Moreover, violin plots and dot plots allow the visualization of each cluster's expression, emphasizing the inter-cluster comparison.

8.1 Format of the input file containing genes for expression visualization

Asc-Seurat expects as input a csv (comma-separated value) or a tsv (tab-separated value) file containing at least two columns. The first column must contain the gene ID as present in the dataset, and the second column is a grouping variable. An optional third column can contain the common names of each gene. Any additional column will be ignored. The output files generated by the differential expression analysis are already in the correct format to be used as input for the visualization.

Below is shown an example of an input file used for expression visualization. It contains ten markers identified for clusters 2 and 3. In this case, the dataset uses the gene name as an identifier, and this is the information on the first column. The second column is used to group de marker according to their clusters.

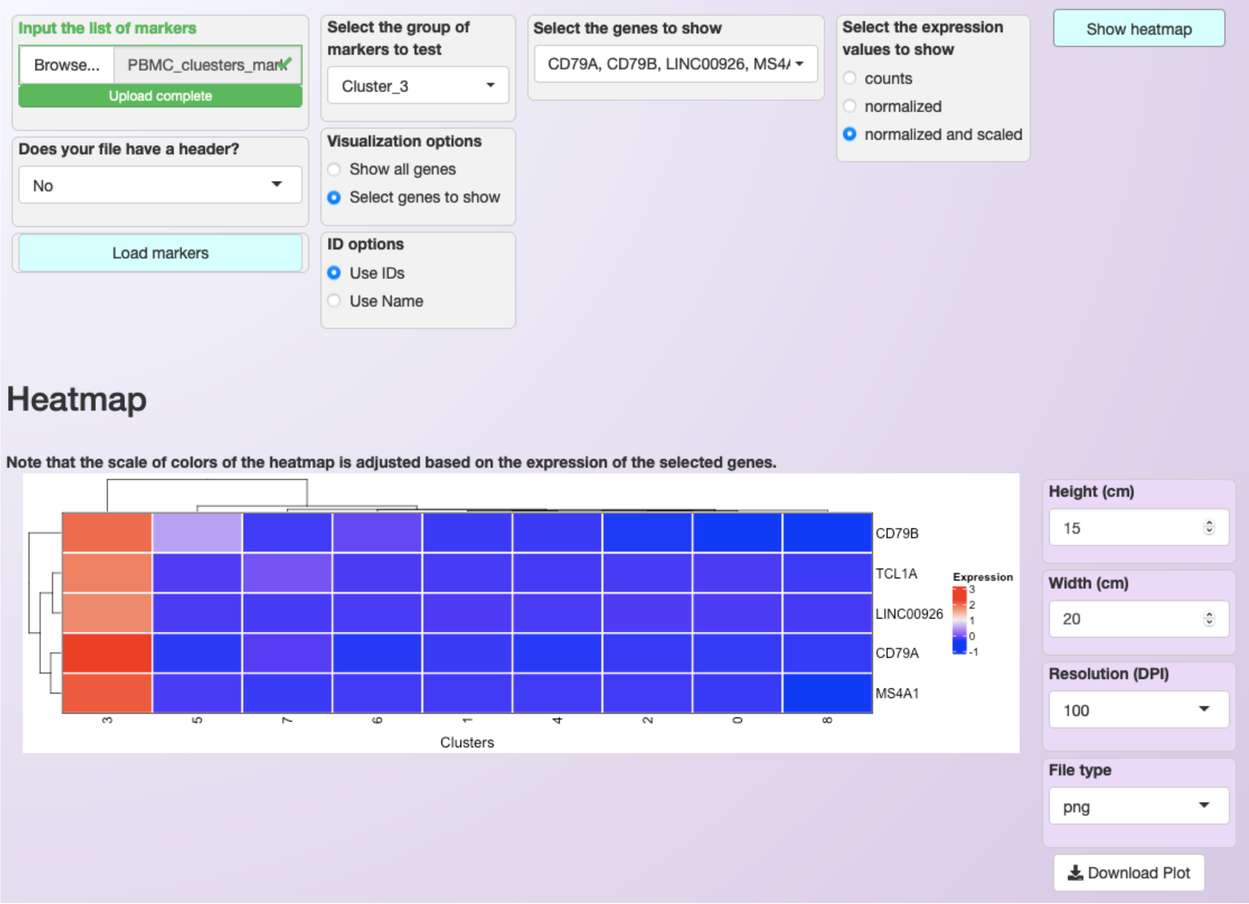
Table 1: Example of an input file for gene expression visualization showing the required columns.

IL32	Cluster_2	
LTB	Cluster_2	
LDHB	Cluster_2	
CD3D	Cluster_2	
IL7R	Cluster_2	
MS4A1	Cluster_3	
CD79A	Cluster_3	
CD79B	Cluster_3	
LINC00926	Cluster_3	
TCL1A	Cluster_3	

After loading the input file, users can select what group(s) of genes to explore and choose specific genes from each group. Moreover, if a third column is provided in the input file, users can use the genes' common name instead of the gene IDs to select the genes to be shown.

8.2 Heatmap

Once users selected their genes of interest, they can generate a heatmap of the average of each gene expression in each cluster by clicking on the button *Show heatmap with the average of expression per cluster*. The heatmap will adjust its height according to the number of selected genes. Moreover, rows and columns will be organized by a hierarchical clustering algorithm. A high-resolution copy of the heatmap plot can be download in a diversity of formats.



Input the list of markers: PBMC_cluesters_mar (Upload complete)

Does your file have a header? No

Select the group of markers to test: Cluster_3

Visualization options: ○ Show all genes ● Select genes to show

ID options: ● Use IDs ○ Use Name

Select the genes to show: CD79A, CD79B, LINC00926, MS4

Select the expression values to show: ○ counts ○ normalized ● normalized and scaled

Heatmap

Note that the scale of colors of the heatmap is adjusted based on the expression of the selected genes.

Height (cm): 15

Width (cm): 20

Resolution (DPI): 100

File type: png

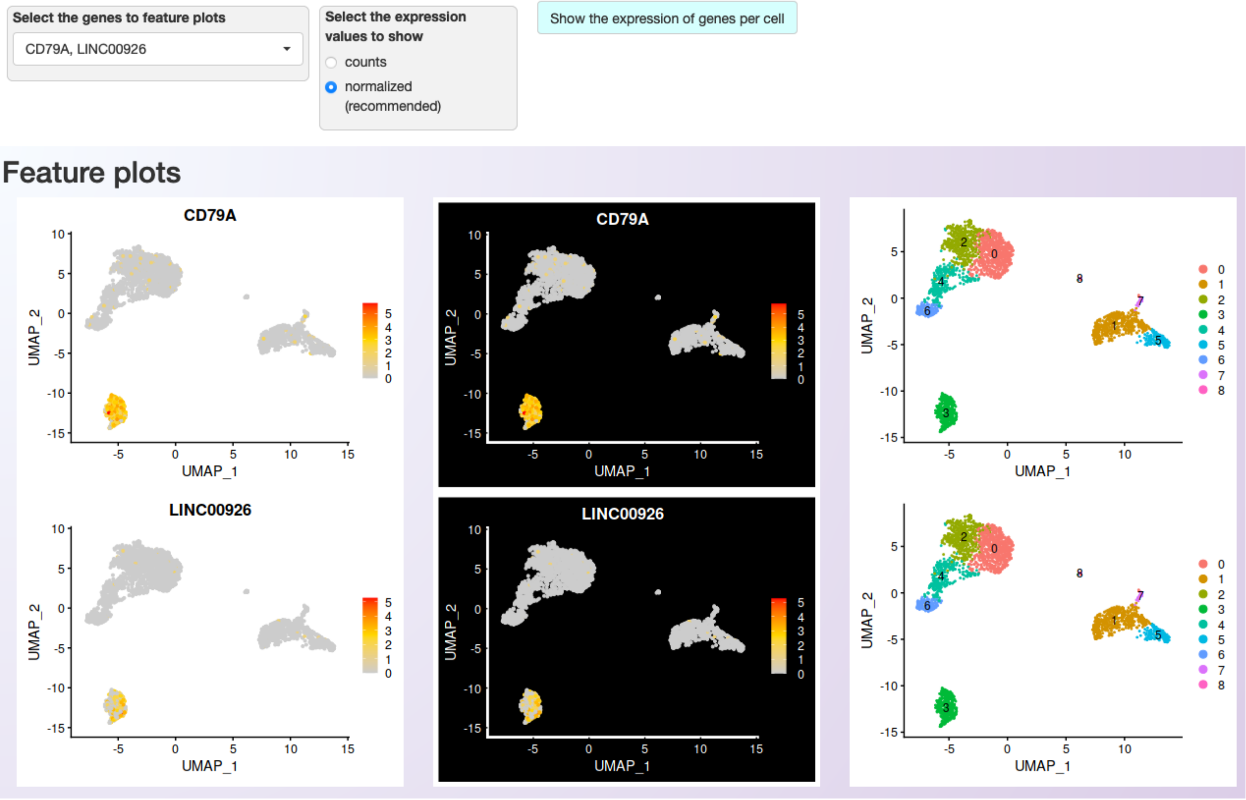
Fig. 1: Asc-Seurat's interface demonstrating the filtering options provided to select the genes for expression visualization. The heatmap shows the expression profile of the five most significant markers for cluster 3.

8.3 Gene expression at the cell level - Feature plots

From the list of genes on the heatmap, users can select genes to further explore by visualizing the expression at the cell level. For each selected gene, a couple of feature plots will be generated using Seurat's Feature plots function. The UMAP plot is shown side-by-side with the feature plots, so users can quickly compare the expression profile with the identified clusters.

8.4 Visualization of the expression among clusters

For each selected gene, Asc-Seurat will also generate plots to visualize the distribution of cells within each cluster according to the expression of the gene (violin plot) and the percentage of cells in each cluster expressing the gene



Select the genes to feature plots

CD79A, LINC00926

Select the expression values to show

○ counts

● normalized (recommended)

Fig. 2: Asc-Seurat's interface showing the filtering options provided to select the genes for expression visualization at the cell level. Two of the five genes shown on the heatmap were chosen for more detailed visualization.

(dot plot). Seurat's functions VlnPlot() and DotPlot() are deployed in this step.

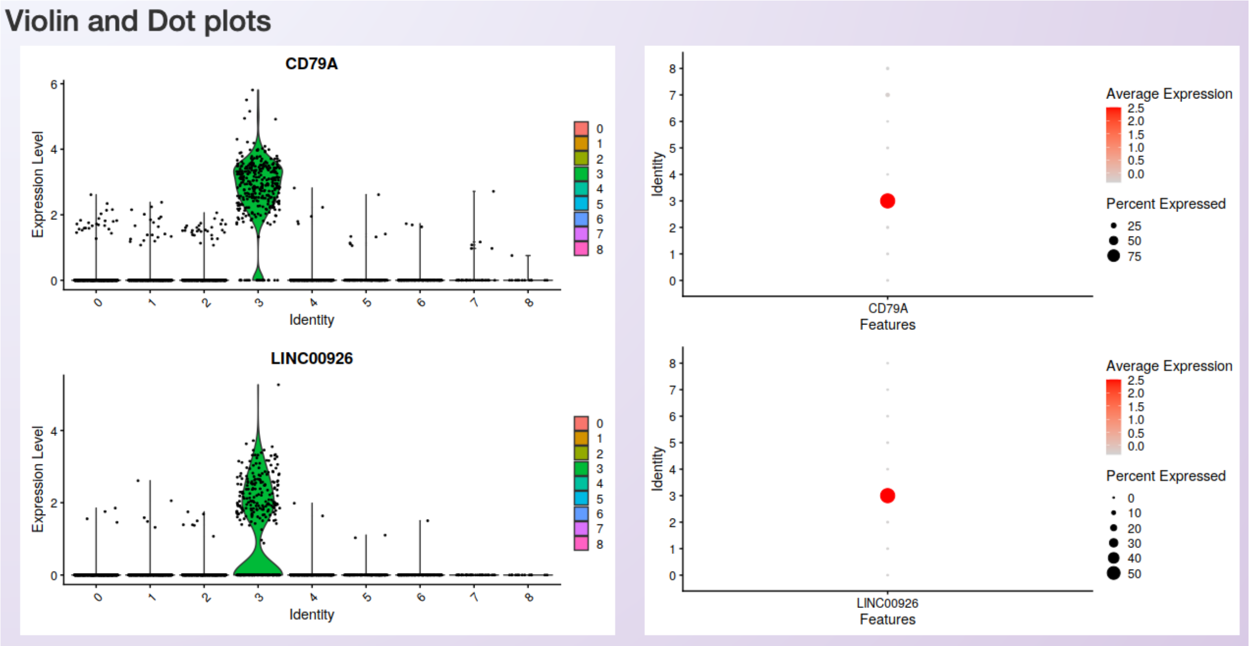


Fig. 3: Visualization of the distribution of cells within each cluster according to the gene expression (violin plot; left) and the percentage of cells in each cluster expressing the gene (dot plot; right).

Tip: Sometimes, it is necessary to make fine adjustments to an image before publication. Saving the plots as a Scalable Vector Graphic (svg), allows the edition of all aspects of the plot by image edition software as Inkscape.

CHAPTER

NINE

LOADING THE DATA AND INTEGRATION OF MULTIPLE SAMPLES

To analyze multiple samples, select the third tab in the web application, named `Integration of multiple samples`.

Note: The integration is based on Seurat's functions FindIntegrationAnchors and IntegrateData. For more information, see Seurat's integration tutorial and Stuart, T. et al. (2019).

9.1 Format of the dataset

Asc-Seurat can only read the input files in the format generated by Cell Ranger (10x genomics). However, it is possible to convert your counts' matrix to the acceptable format. For example, the function write10xCounts(), from the DropletUtils package, is an easy option to make this conversion.

Tip: Using write10xCounts(), users can provide as output the path to the `data/` directory. In this way, Asc-Seurat can recognize the files automatically.

9.2 Location of the dataset

For the integration of multiple samples, the process is a little different. Users still need to add their datasets in the `data/` directory, creating a subdirectory for each sample. However, users also need to provide a configuration file containing the parameter values for each sample. During the installation, an example file named *configuration_file_for_integration_analysis.csv* will be created in the directory and can be used as a model.

Note: The integration of samples can be biased if the parameters are not chosen appropriately. Therefore, it is recommended to explore each sample separately in the tab *One sample*, defining adequate parameters to remove deficient quality cells before the integration.

The user's configuration file must have six columns and a header (the column names are not restricted). They specify what cells should be kept for each sample while loading the data before the integration.

Also, the columns need to be in a specific order, as listed below.

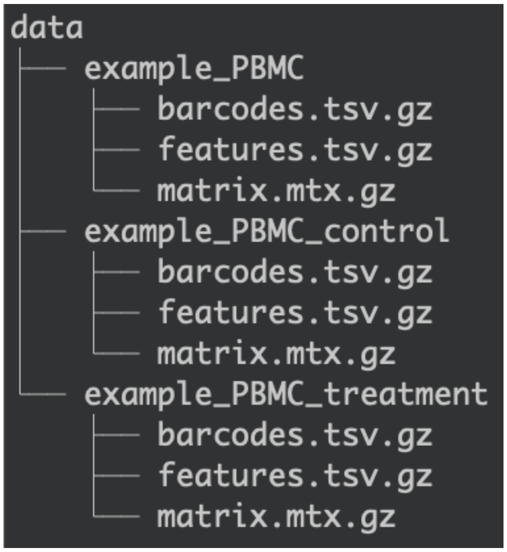
1. **Subdirectory name**: The name of the subdirectories containing the datasets. Each sample must have a unique name for its subdirectory, even if they are replicates.

2. **Sample name (any name you prefer)**: Your choice of name for each sample. If you have replicates and want them to be considered as one in the plots and analysis, use the same name for all replicates.
3. **Min. number of cells expressing a gene**: Include genes only if they are detected in at least this many cells.
4. **Min. number of genes a cell must express to be included**: Include cells only if they expressed at least this number of genes.
5. **Maximum number of genes a cell can express and still be included**: Remove cells that express more than this number of genes. Useful to remove cells that are suspected to be doublets.
6. **Maximum percentage of genes belonging to the mitochondrial genome**: Here, the regular expression (Regex) is a sequence of characters that is used to identify the genes belonging to the mitochondrial genome. For example, when using the human genome, this sequence should be "^MT-".

9.3 Loading the data and performing integration

To demonstrate the necessary steps to load and integrate multiple datasets using Asc-Seurat, we used two groups of cells from Kang et al., 2017, that are also used in Seurat's tutorial demonstrating the comparison of multiple samples. Two datasets are used, both containing peripheral blood mononuclear cells (PBMCs). However, the first dataset contains the cells of the control group (Control), while the second dataset contains cells treated with interferon-beta (Treatment).

The first step is two create two folders inside the `data/` folder. The folders were named `example_PBMC_control` and `example_PBMC_treatment`, each containing the three necessary input files (shown in the image below).



```
data
├── example_PBMC
│   ├── barcodes.tsv.gz
│   ├── features.tsv.gz
│   └── matrix.mtx.gz
├── example_PBMC_control
│   ├── barcodes.tsv.gz
│   ├── features.tsv.gz
│   └── matrix.mtx.gz
└── example_PBMC_treatment
    ├── barcodes.tsv.gz
    ├── features.tsv.gz
    └── matrix.mtx.gz
```

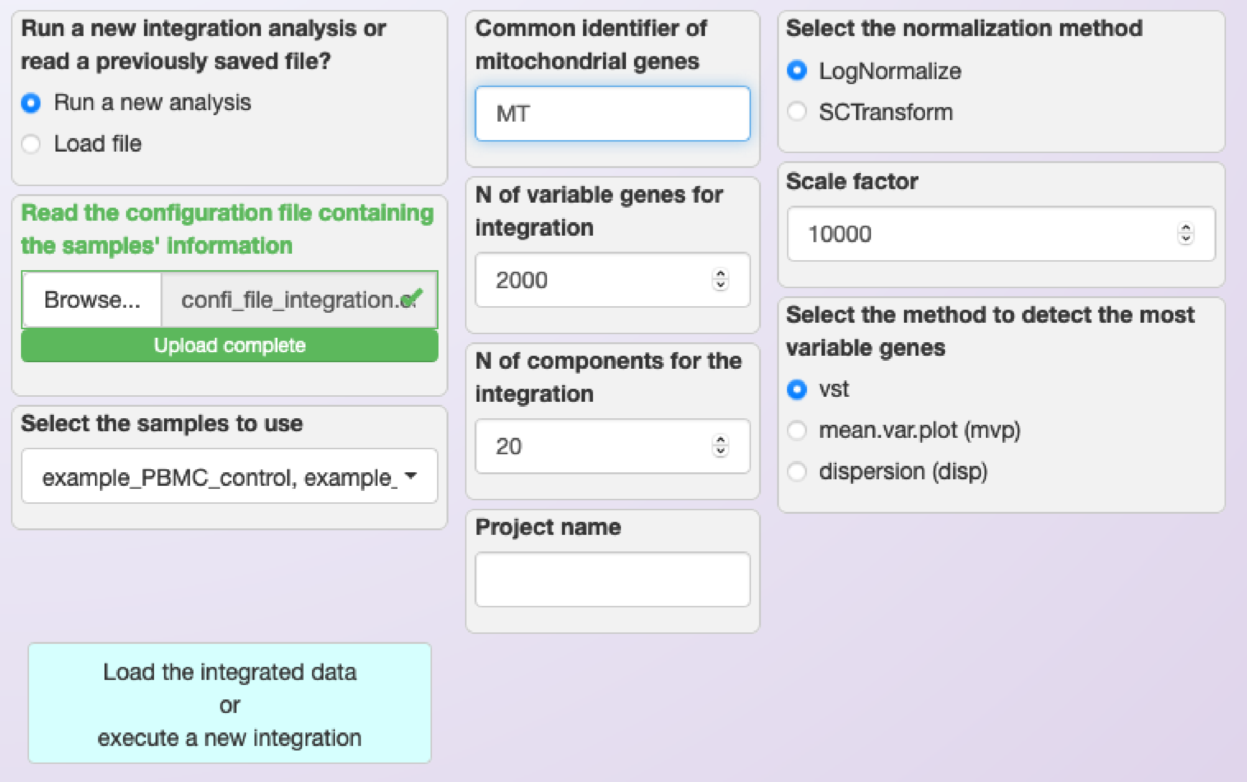
Fig. 1: Organization of the `data/` folder the different datasets.

After that, it is necessary to create a configuration file in the csv format. During the installation, an example file named *configuration_file_for_integration_analysis.csv* is created in the directory. It can then be used as a model. For this example, the configuration file contains the information shown below.

Table 1: Example of a configuration file for the integration of multiple samples.

Subdirectory name (must be inside data/)	Sample name (any name you prefer)	Min. number of cells expressing a gene	Min. number of genes a cell must express to be included	Max. number of genes a cell can express and still be included	Max. percentage of transcripts belonging to mitochondrial genome
example_PBMC_control	Control	3	250	2500	5
example_PBMC_treatment	Treatment	3	250	2500	5

Once the configuration file is ready, users only need to load it in the app and select the samples they want to integrate (see image below). Also, it is necessary to choose the normalization method, the regex string to detect mitochondrial transcripts, the number of Principal Components to be used during the integration (see below). Note that while default values are provided, users need to set these parameters based on their evaluation of the individual samples that are being integrated.



Run a new integration analysis or read a previously saved file?

● Run a new analysis

○ Load file

Read the configuration file containing the samples' information

Browse... confi_file_integration.c ✓

Upload complete

Select the samples to use

example_PBMC_control, example_

Common identifier of mitochondrial genes

MT

N of variable genes for integration

2000

N of components for the integration

20

Project name

Select the normalization method

● LogNormalize

○ SCTransform

Scale factor

10000

Select the method to detect the most variable genes

● vst

○ mean.var.plot (mvp)

○ dispersion (disp)

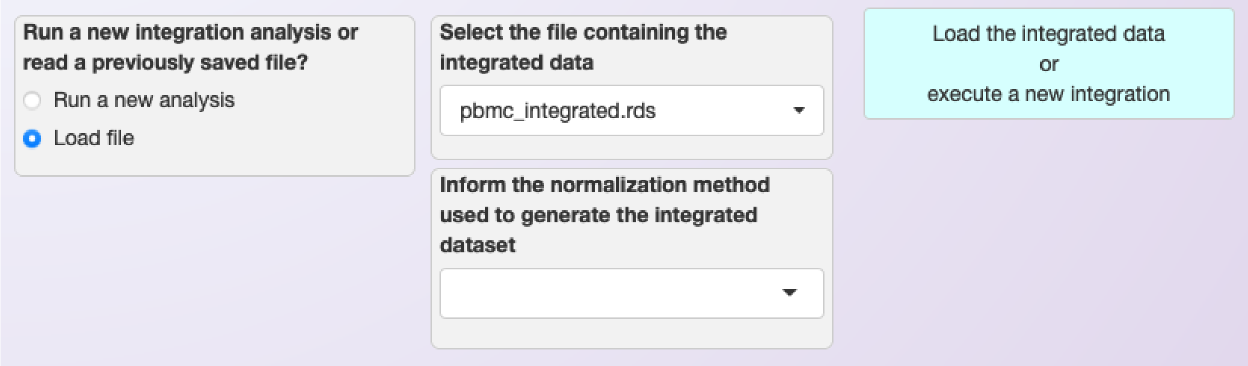
Fig. 2: Loading configuration file and defining parameters for the integration of multiple samples using LogNormalization.

9.4 Saving integrated data for reanalysis

The integration of multiple samples is a timing-consuming step of the analysis. The amount of time necessary to execute this step depends on the number of datasets and the number of cells in each dataset, and it can take several minutes to be concluded.

Therefore, Asc-Seurat allows users to save the integrated data and skip the integration step the next time users need to use the same dataset. To save the data, users can click on the button `Download RDS object containing the integrated data.` and save the rds file inside the `RDS_files/` folder.

Next time this data is necessary, users can select the option "Load file" and skip the integration step, as shown below.



Run a new integration analysis or read a previously saved file?

○ Run a new analysis

● Load file

Select the file containing the integrated data: pbmc_integrated.rds

Inform the normalization method used to generate the integrated dataset:

CHAPTER

TEN

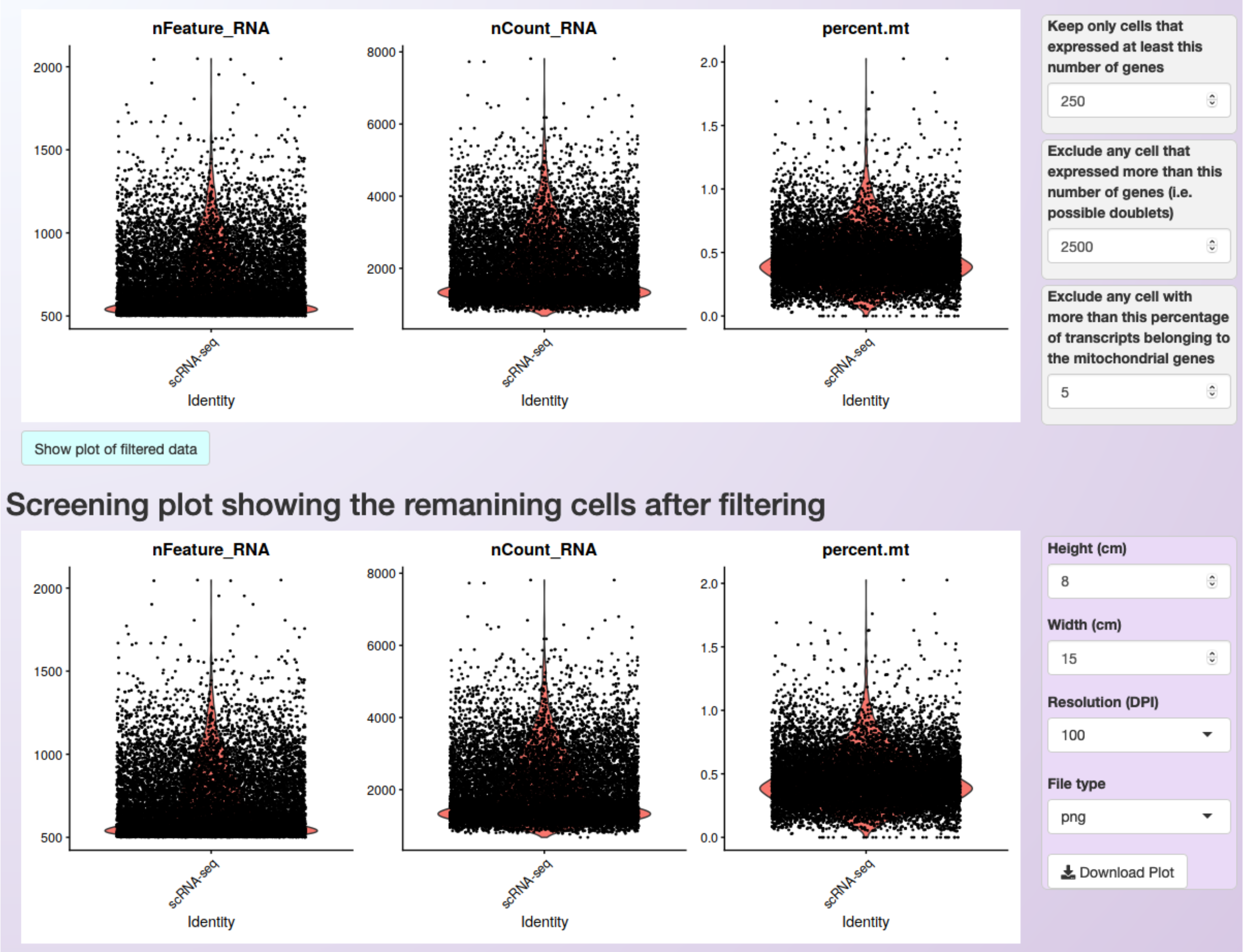
QUALITY CONTROL

After integrating the datasets, a violin plot will be generated showing the distribution of cells according to three parameters:

- nFeature_RNA: the number of genes detected in each cell
- nCount_RNA: the number of molecules detected per cell
- percent.mt: the percentage of transcripts that map to mitochondrial genes

After visualizing the distribution of cells, it is possible to set more restrictive parameters (on the right side of the plot) and filter cells based on the number of expressed genes per cell and the percentage of transcripts from mitochondrial genes. By clicking on *Show plot of filtered data*, users can see the distribution of cells after filtering and then readjust the parameters. The figure below shows the distribution of cells of the PBMC integrated (containing the Control and Treatment datasets, see *Loading the data and integration of multiple samples*) dataset before and after filtering.

Asc-Seurat allows users to download each of the plots with high-resolution by clicking on the `Download plot` button.



Keep only cells that expressed at least this number of genes: 250

Exclude any cell that expressed more than this number of genes (i.e. possible doublets): 2500

Exclude any cell with more than this percentage of transcripts belonging to the mitochondrial genes: 5

Screening plot showing the remaining cells after filtering

Height (cm): 8

Width (cm): 15

Resolution (DPI): 100

File type: png

CHAPTER

ELEVEN

CLUSTERING

11.1 Normalization

When integrating multiple samples, the normalization is executing during the integration.

Note: Currently, the recommendation of Seurat's team is to use the standard "RNA" assay when performing differential expression (D.E) analysis and for data visualization, even when using SCTransform (See here). Therefore, Asc-Seurat will use the SCTransformed data ("SCT" assay) until the clustering step only.

To use the "RNA" assay after SCTransform, Asc-Seurat will automatically perform the LogNormalization and scaling of the data in the RNA assay by applying the default parameters.

11.2 Dimensional reduction (PCA)

The PCA will be executed using Seurat's function RunPCA and, after its conclusion, an elbow plot is generated automatically, to help users to decide how many PCs should be included to inform the clustering step.

Users can use this plot to select the PCs with the highest standard deviation (more informative PCs). Also, users should set the number of PCs to include during clustering in the windows at the plot's right side.

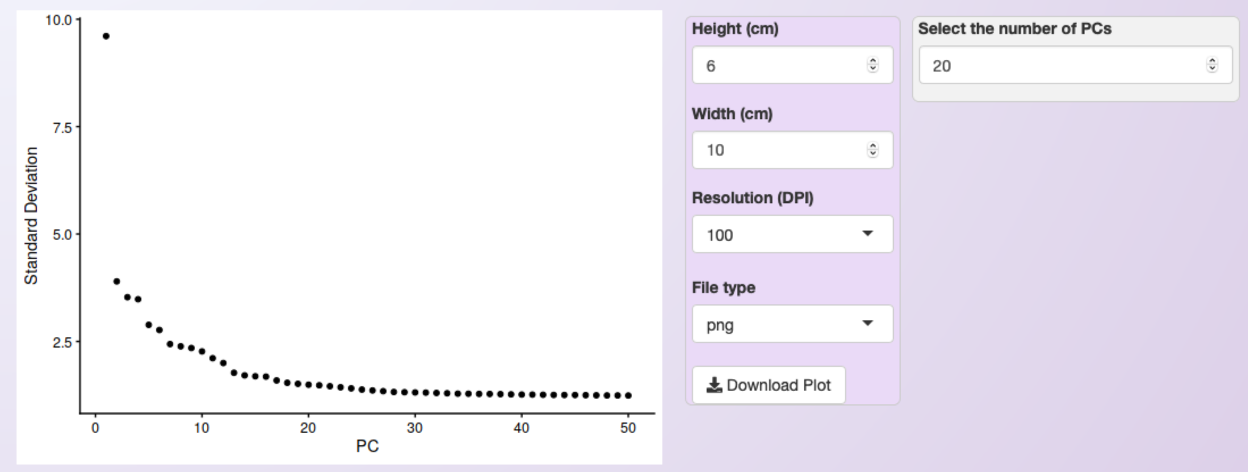
In the example below, the first 20 PCs are selected. Not that the resulting plot will be slightly different depending on the normalization method. Below we show the result obtained using LogNormalization.

11.3 Clustering of cells

The next step is the clustering of the cells. For that, Asc-Seurat used both FindNeighbors and FindClusters functions of the Seurat package.

Before the execution, however, users need to set a value for the resolution parameter. The resolution is an important parameter to evaluate because it determines the profile and number of clusters identified for a dataset. Selecting larger values will favor splitting cells into more clusters while choosing a smaller value has the opposite effect. Quoting from Seurat's tutorial: "We find that setting this parameter between 0.6-1.2 typically returns good results for single-cell datasets of around 3K cells. Optimal resolution often increases for larger datasets".

Tip: There is no easy way to define an optimal value for the resolution parameter. Users need to try different values and evaluate the resulting clusters according to the expectation for their cells population. Visualizing the expression profile of cell-type-specific markers can provide a hint if the chosen value is too small or too large.



Height (cm): 6

Width (cm): 10

Resolution (DPI): 100

File type: png

Select the number of PCs: 20

Fig. 1: Elbow plot provided to help to select the most informative PCs. For the PBMC integrated dataset, and using the LogNormalization method, we chose the 20 first PCs.

After the clustering step's execution, three plots are generated for cluster visualization, all of them using the Uniform Manifold Approximation and Projection (UMAP) technique. The first plot shows the clustering of the whole dataset colored by cluster. The second plot shows the same plot, but cells are colored by sample. The third plot shows the clustering of the cells of each sample, with one subplot per sample.

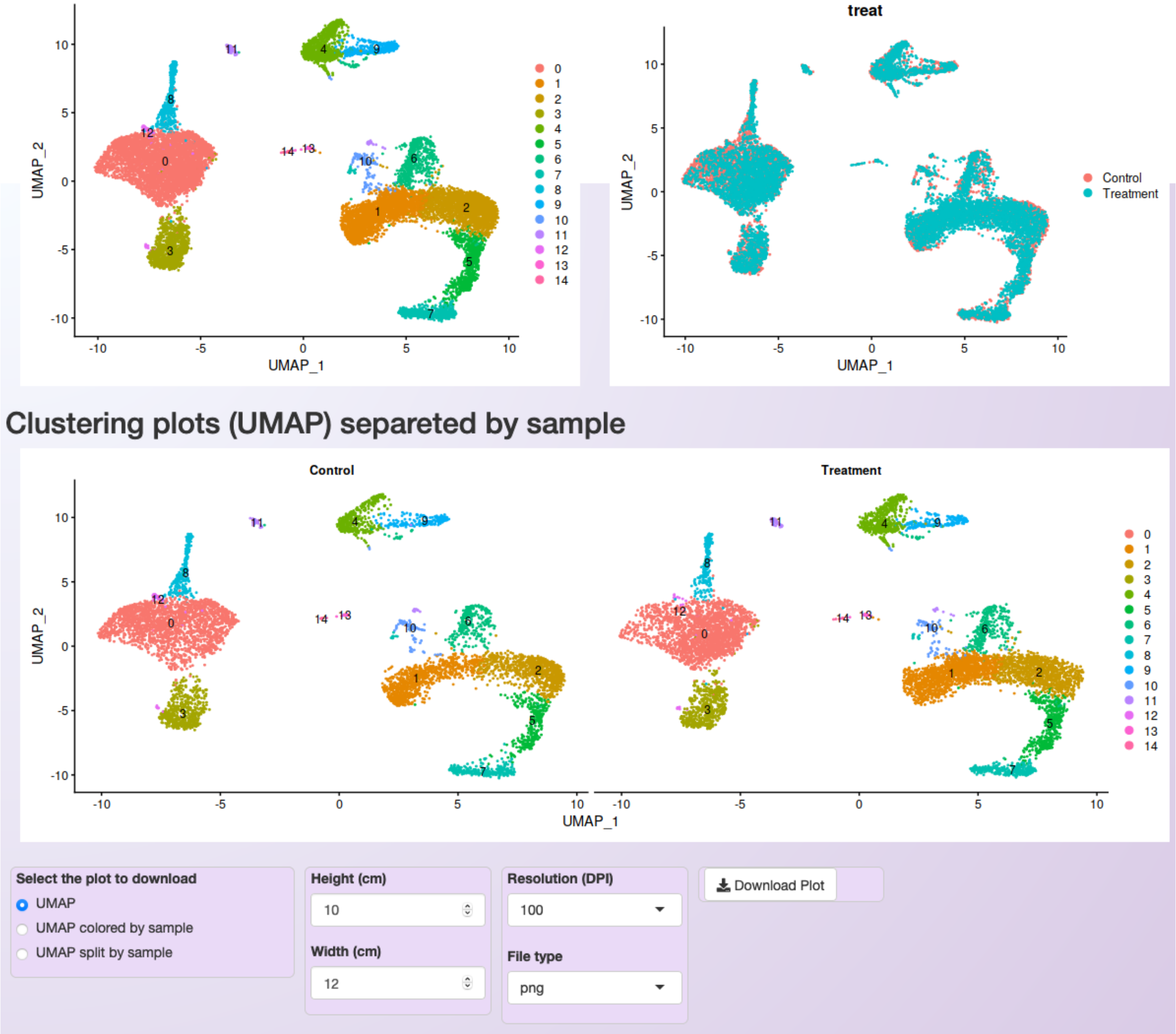
11.3.1 Selecting clusters of interest

In some cases, it is interesting to select or exclude some clusters of cells from the dataset before executing the subsequent steps. This process is helpful, for example, when users desire to explore a developmental trajectory of a specific group of cell types.

Asc-Seurat makes this step simple. Users only need to select the cluster(s) to keep or exclude and start reanalysis of the remaining cells by clicking on *Reanalyze after selection/exclusion of clusters* (see below).

Asc-Seurat will then execute the steps with the new set of cells up to the PCA. Then, **users need to evaluate the elbow plot and decide the number of PCs to cluster the new set of cells**. Users can either keep the same value for the resolution parameter or modify it before clicking on *Run the clustering analysis* to start the clustering once more.

Warning: The cluster's numbering will change every time that clusters are selected or excluded.



Clustering plots (UMAP) separeted by sample

Select the plot to download

● UMAP

○ UMAP colored by sample

○ UMAP split by sample

Height (cm): 10

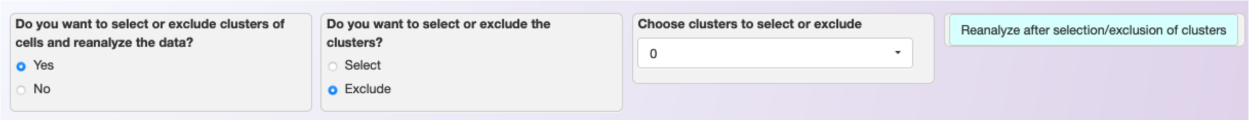
Width (cm): 12

Resolution (DPI): 100

File type: png

Download Plot

Fig. 2: Plot showing the PBMC integrated dataset clustering using 20 PCs, LogNormalization, and a resolution value of 0.5.



Do you want to select or exclude clusters of cells and reanalyze the data?

● Yes

○ No

Do you want to select or exclude the clusters?

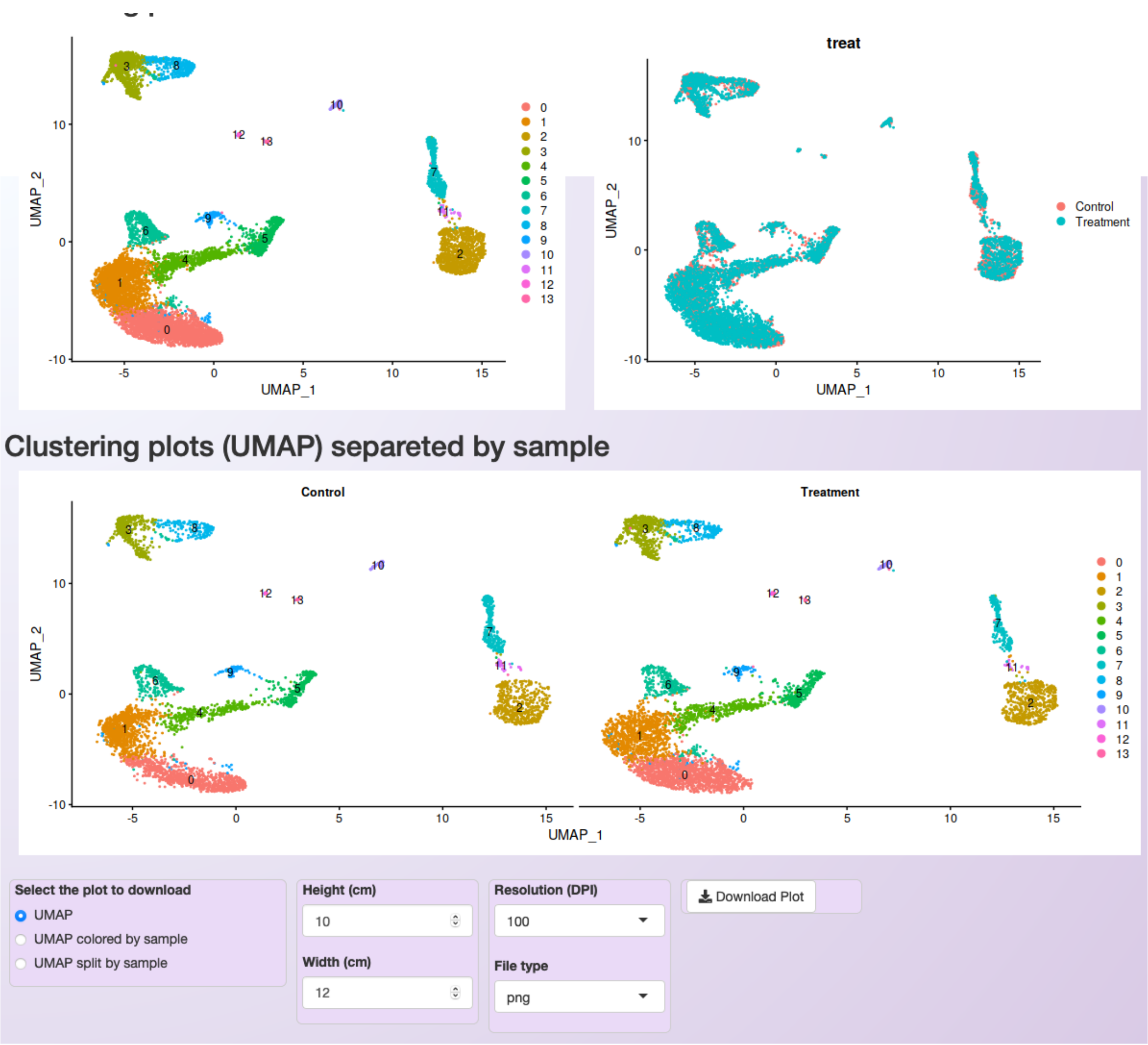
○ Select

● Exclude

Choose clusters to select or exclude: 0

Reanalyze after selection/exclusion of clusters

Fig. 3: Asc-Seurat makes it easy to select or exclude a cluster (or clusters) of cells. In this example, we exclude all cells belonging to cluster 0.



Clustering plots (UMAP) separeted by sample

Select the plot to download

● UMAP

○ UMAP colored by sample

○ UMAP split by sample

Height (cm): 10

Width (cm): 12

Resolution (DPI): 100

File type: png

Fig. 4: Clustering of the PBMC integrated dataset after excluding cells belonging to cluster 0 from the original dataset.

CHAPTER

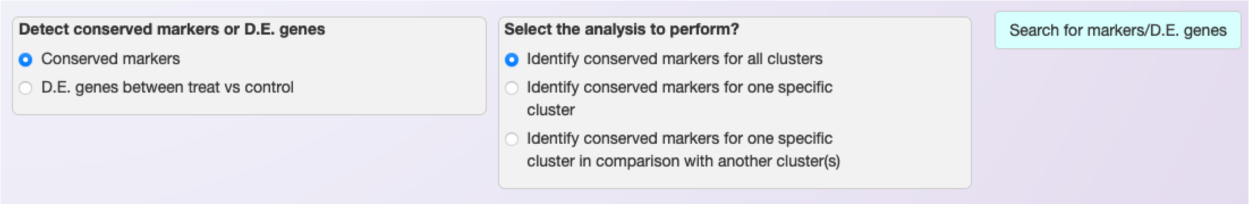
TWELVE

MARKERS IDENTIFICATION AND DIFFERENTIAL EXPRESSION ANALYSIS

After clustering the cells, users may be interested in identifying genes specifically expressed in one cluster (markers) or in genes that are differentially expressed among clusters of interest. Asc-Seurat can apply multiple algorithms to identify gene markers for individual clusters or identify differentially expressed genes (DEGs) among clusters. **Moreover, when using an integrated dataset containing multiple samples, it is possible to identify DEGs among samples for each cluster.**

Note: When searching for markers of a cluster or DEGs among clusters using an integrated dataset, the search will attempt to find markers or DEGs conserved among samples.

Asc-Seurat allows users to filter gene markers and DEGs by the fold change and minimal percentage of cells expressing a gene in the cluster(s). Moreover, users can define a significance level to exclude genes based on the adjusted p-value (see below).



Detect conserved markers or D.E. genes

● Conserved markers

○ D.E. genes between treat vs control

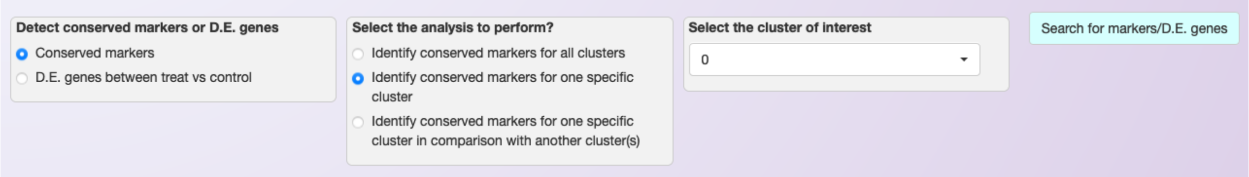
Select the analysis to perform?

● Identify conserved markers for all clusters

○ Identify conserved markers for one specific cluster

○ Identify conserved markers for one specific cluster in comparison with another cluster(s)

Fig. 1: Example of Asc-Seurat's interface showing the settings to the search for gene markers for each of the clusters and conserved among samples.



Detect conserved markers or D.E. genes

● Conserved markers

○ D.E. genes between treat vs control

Select the analysis to perform?

○ Identify conserved markers for all clusters

● Identify conserved markers for one specific cluster

○ Identify conserved markers for one specific cluster in comparison with another cluster(s)

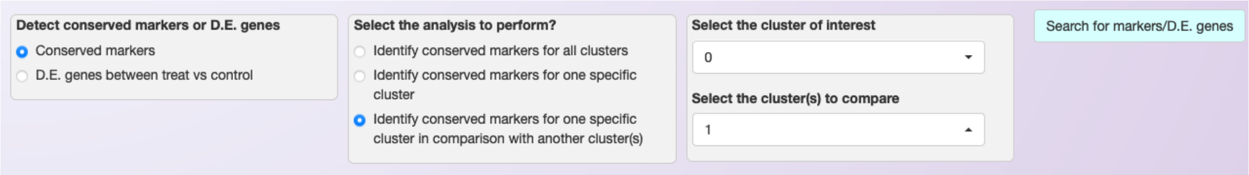
Select the cluster of interest

0

Fig. 2: Example of Asc-Seurat's interface showing the settings to the search for gene markers for a specific cluster and conserved among samples.

An iterative table will be available after executing the search for marker or DEGs, showing the significant genes. Moreover, users can download the list of significant markers or DEGs as a csv file.

The list of genes in the csv can then be used to visualize their gene expression in a series of plots, as shown in the section *Expression visualization*.



Detect conserved markers or D.E. genes

● Conserved markers

○ D.E. genes between treat vs control

Select the analysis to perform?

○ Identify conserved markers for all clusters

○ Identify conserved markers for one specific cluster

● Identify conserved markers for one specific cluster in comparison with another cluster(s)

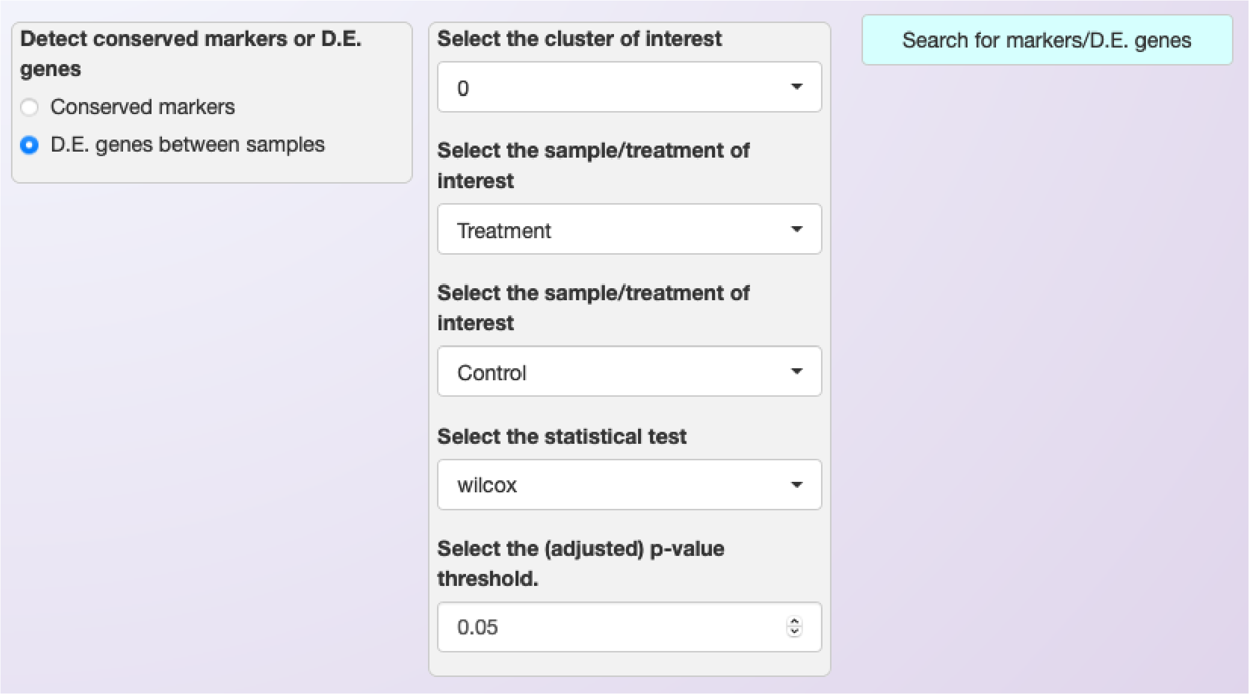
Select the cluster of interest

0

Select the cluster(s) to compare

1

Fig. 3: Example of Asc-Seurat's interface showing the settings to search for DEGs genes among clusters 0 and 1.



Detect conserved markers or D.E. genes

○ Conserved markers

● D.E. genes between samples

Select the cluster of interest

0

Select the sample/treatment of interest

Treatment

Select the sample/treatment of interest

Control

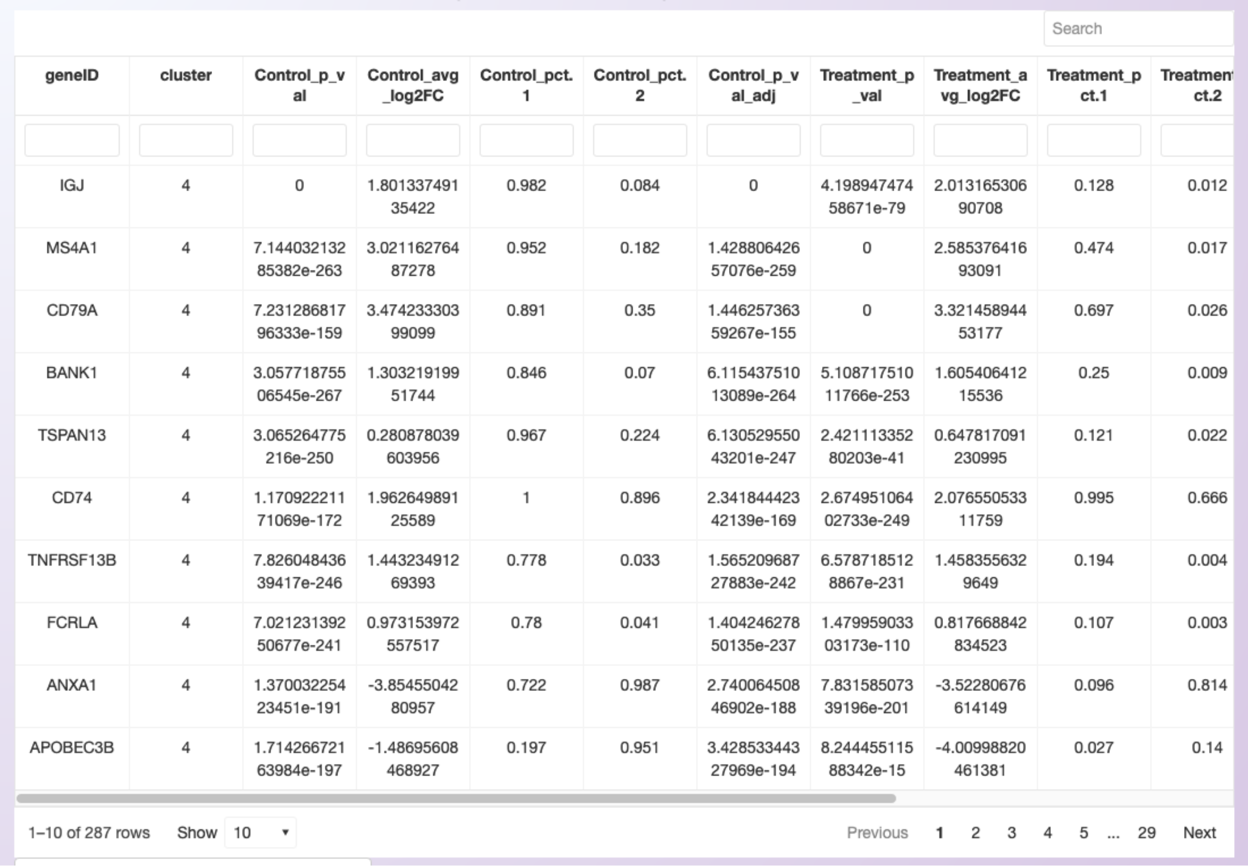
Select the statistical test

wilcox

Select the (adjusted) p-value threshold.

0.05

Fig. 4: Example of Asc-Seurat's interface showing the settings to search for DEGs among samples for a specific cluster (cluster 0).



geneID	cluster	Control_p_val	Control_avg_log2FC	Control_pct.1	Control_pct.2	Control_p_val_adj	Treatment_p_val	Treatment_avg_log2FC	Treatment_pct.1	Treatmen ct.2
IGJ	4	0	1.80133749135422	0.982	0.084	0	4.19894747458671e-79	2.01316530690708	0.128	0.012
MS4A1	4	7.14403213285382e-263	3.02116276487278	0.952	0.182	1.42880642657076e-259	0	2.58537641693091	0.474	0.017
CD79A	4	7.23128681796333e-159	3.47423330399099	0.891	0.35	1.44625736359267e-155	0	3.32145894453177	0.697	0.026
BANK1	4	3.05771875506545e-267	1.30321919951744	0.846	0.07	6.11543751013089e-264	5.10871751011766e-253	1.60540641215536	0.25	0.009
TSPAN13	4	3.065264775216e-250	0.280878039603956	0.967	0.224	6.13052955043201e-247	2.42111335280203e-41	0.647817091230995	0.121	0.022
CD74	4	1.17092221171069e-172	1.96264989125589	1	0.896	2.34184442342139e-169	2.67495106402733e-249	2.07655053311759	0.995	0.666
TNFRSF13B	4	7.82604843639417e-246	1.44323491269393	0.778	0.033	1.56520968727883e-242	6.5787185128867e-231	1.4583556329649	0.194	0.004
FCRLA	4	7.02123139250677e-241	0.973153972557517	0.78	0.041	1.40424627850135e-237	1.47995903303173e-110	0.817668842834523	0.107	0.003
ANXA1	4	1.37003225423451e-191	-3.8545504280957	0.722	0.987	2.74006450846902e-188	7.83158507339196e-201	-3.52280676614149	0.096	0.814
APOBEC3B	4	1.71426672163984e-197	-1.48695608468927	0.197	0.951	3.42853344327969e-194	8.24445511588342e-15	-4.00998820461381	0.027	0.14

1–10 of 287 rows Show 10 ▾

Fig. 5: The ten most significant markers identified for cluster 4 of the PBMC integrated dataset (the clustering is shown in *Clustering*).

CHAPTER

THIRTEEN

EXPRESSION VISUALIZATION

Asc-Seurat provides a variety of plots for gene expression visualization of the integrated data. From a list of selected genes, it is possible to visualize the average of each gene expression in each cluster in a heatmap. It also provides plots for the visualization of gene expression at the cell level. Moreover, violin plots and dot plots allow the visualization of each cluster's expression, emphasizing the inter-cluster comparison.

For the integrated dataset, besides identifying markers for each cluster and DEGs among clusters, it is also possible to identify DEGs among samples (See *Markers identification and differential expression analysis*). Below are shown examples of plots that Asc-Seurat generates to allow the expression visualization in all these cases.

13.1 Expression visualization of genes identified as markers

13.1.1 Format of the input file containing genes for expression visualization

Asc-Seurat expects as input a csv (comma-separated value) or a tsv (tab-separated value) file containing at least two columns. The first column must contain the gene ID as present in the dataset, and the second column is a grouping variable. An optional third column can contain the common names of each gene. Any additional column will be ignored. The output files generated by the differential expression analysis are already in the correct format to be used as input for the visualization.

Below is shown an example of an input file used for expression visualization. It contains ten markers identified for cluster 4 of the PBMC integrated dataset (Control and Treatment). In this case, the dataset uses the gene name as an identifier, and this is the information contained in the first column. The second column is used to group de marker according to their clusters.

Table 1: Example of an input file for gene expression visualization showing the required columns.

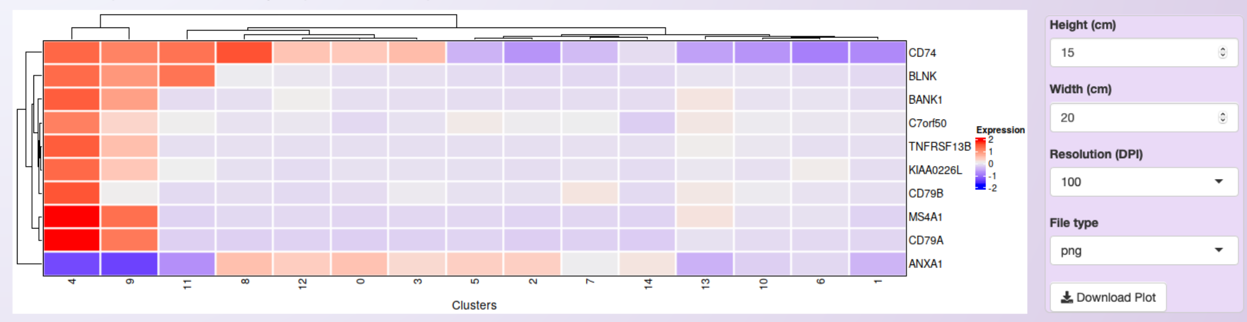
MS4A1	Cluster_4	
CD79B	Cluster_4	
CD79A	Cluster_4	
BANK1	Cluster_4	
CD74	Cluster_4	
TNFRSF13B	Cluster_4	
ANXA1	Cluster_4	
KIAA0226L	Cluster_4	
BLNK	Cluster_4	
C7orf50	Cluster_4	

After loading the input file, users can select what group(s) of genes to explore and choose specific genes from each group. Moreover, if a third column is provided in the input file, users can use the genes' common name instead of the gene IDs to select the genes to be shown.

13.1.2 Heatmap

Once users selected their genes of interest, they can generate a heatmap of the average of each gene expression in each cluster by clicking on the button *Show heatmap with the average of expression per cluster*. The heatmap will adjust its height according to the number of selected genes. Moreover, rows and columns will be organized by a hierarchical clustering algorithm. A high-resolution copy of the heatmap plot can be download in a diversity of formats.

> **Warning:** For the integrated dataset, the heatmap shows the average expression of all samples together. It is only helpful to identify if the cell types' markers make sense with the number of generated clusters.



Height (cm) 15

Width (cm) 20

Resolution (DPI) 100

File type png

Fig. 1: Heatmap showing the expression profile of the then most significant markers for cluster 4 of the integrated datasets.

13.1.3 Gene expression at the cell level - Feature plots

From the list of genes on the heatmap, users can select genes to further explore by visualizing the expression at the cell level. For each selected gene, a feature plot showing each sample's profile will be generated using Seurat's Feature plots function. The UMAP plot is shown side-by-side with the feature plots, so users can quickly compare the expression profile with the identified clusters.

13.1.4 Visualization of the expression among clusters

For each selected gene, Asc-Seurat will also generate plots to visualize the distribution of cells within each cluster according to the expression of the gene (violin plot) and the percentage of cells in each cluster expressing the gene (dot plot) in each sample. Seurat's functions VlnPlot() and DotPlot() are deployed in this step.

13.2 Expression visualization of differentially expressed genes

13.2.1 Format of the input file containing genes for expression visualization

As before, a csv or tsv file is necessary as input for the expression visualization of DEGs. In this case, it contains ten genes identified as DEGs between the PBMC Treatment and PBMC Control datasets in cluster 4.

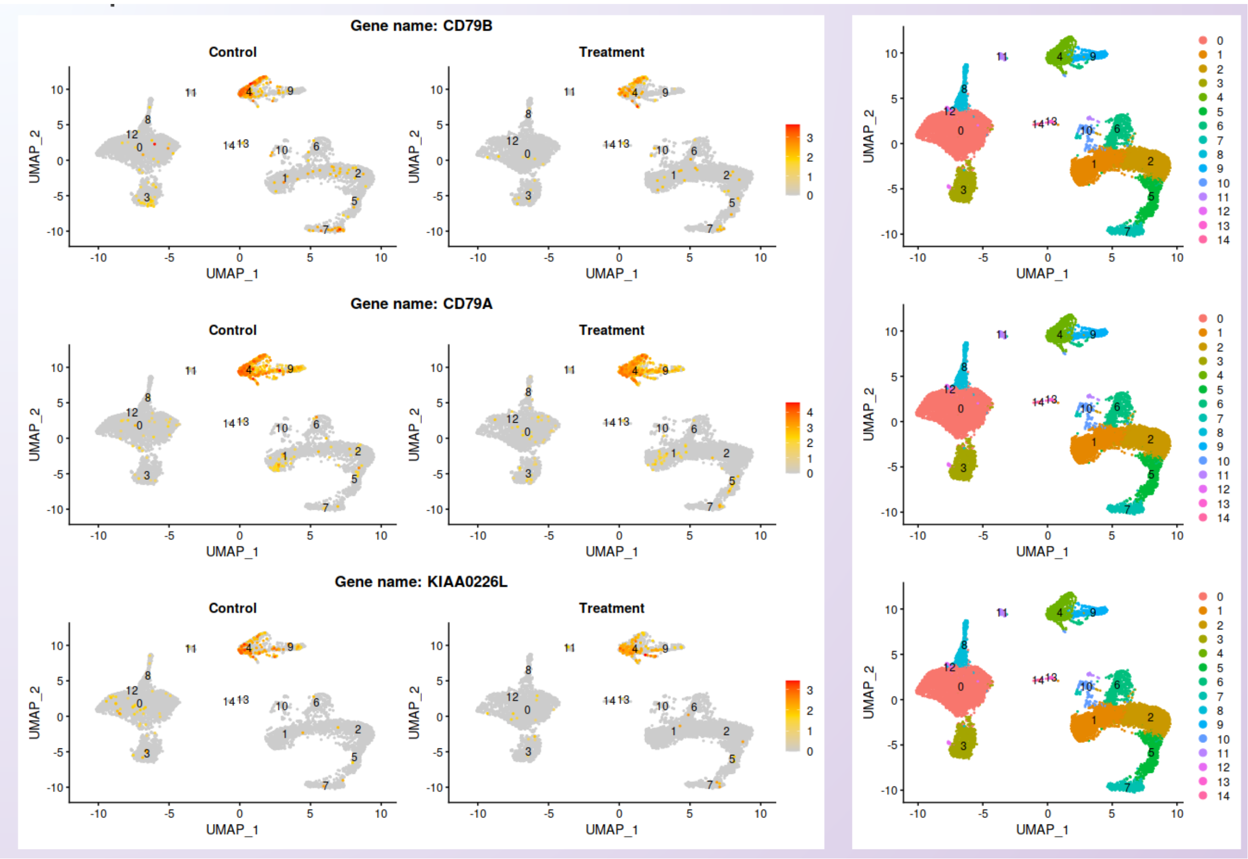


Fig. 2: Visualization of the expression profile of three of the genes shown on the heatmap in each sample.

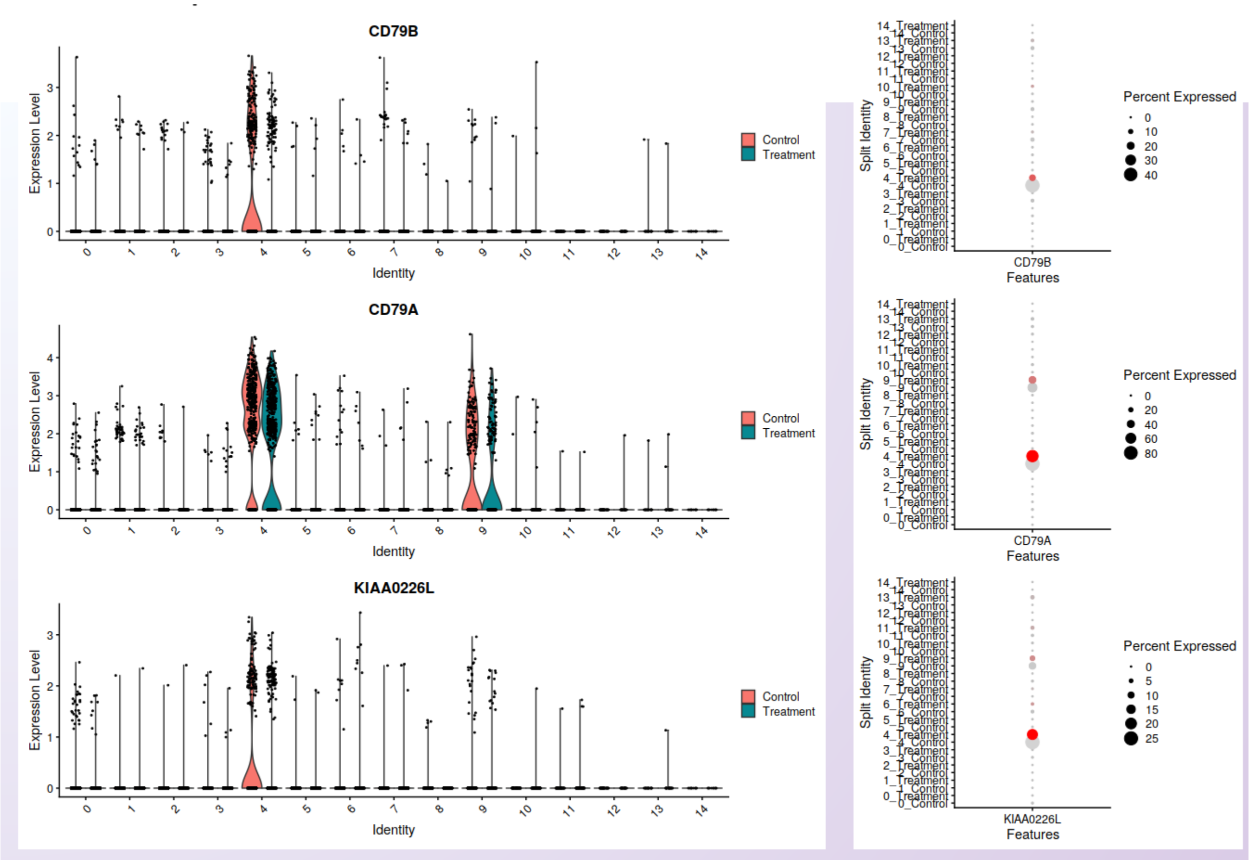


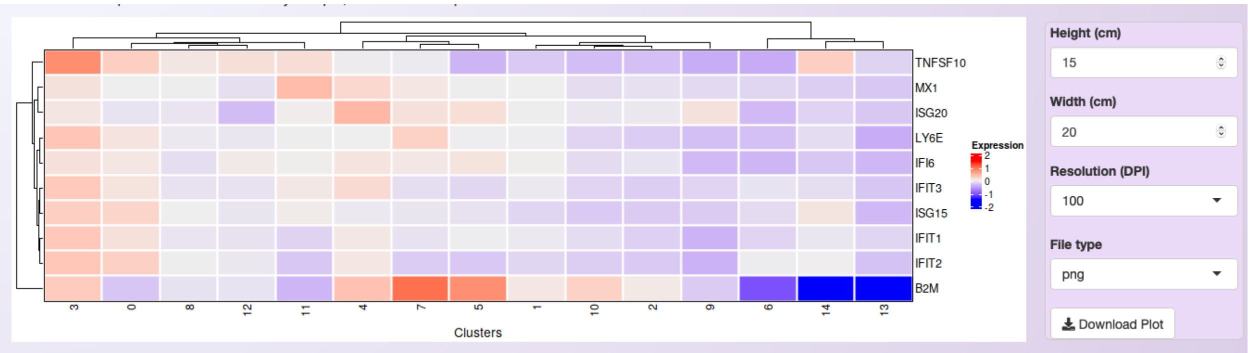
Fig. 3: Visualization of cells' distribution within each cluster according to the gene expression (violin plot; left) and the percentage of cells in each cluster expressing the gene (dot plot; right) in each sample. The three genes shown are the same used for the feature plots.

Table 2: Example of an input file for gene expression visualization of DEGs.

ISG15	DEGs Cluster_4	
IFIT3	DEGs Cluster_4	
IFI6	DEGs Cluster_4	
ISG20	DEGs Cluster_4	
IFIT1	DEGs Cluster_4	
MX1	DEGs Cluster_4	
LY6E	DEGs Cluster_4	
TNFSF10	DEGs Cluster_4	
IFIT2	DEGs Cluster_4	
B2M	DEGs Cluster_4	

13.2.2 Heatmap

All ten genes were selected for visualization in the heatmap. Once more, it is important to mention that the heatmap shows the average expression among all samples. However, by investigating the heatmap below, it is possible to notice that while these genes are the most significant DEGs between samples in cluster 4, they are widely expressed in other clusters too.



Height (cm)

15

Width (cm)

20

Resolution (DPI)

100

File type

png

Fig. 4: Heatmap showing the expression profile of the ten most significant DEGs between Treatment and Control in cluster 4 of the integrated datasets.

13.2.3 Gene expression at the cell level - Feature plots

To compare the expression profile among samples, the visualization at the cell level is more relevant, as shown below.

From the list of genes contained on the heatmap, three genes were selected. While the expression is not localized in cluster 4, it is clear the increment of the expression in the Treatment dataset. The UMAP plot is shown side-by-side with the feature plots, allowing comparing the expression profile with the identified clusters.

13.2.4 Visualization of the expression among clusters

As in the feature plot, the violin and dot plots clearly show the increased level of expression in the cells of the PBMC Treatment sample compared to the PBMC control.

Tip: Sometimes, it is necessary to make fine adjustments to an image before publication. Saving the plots as a

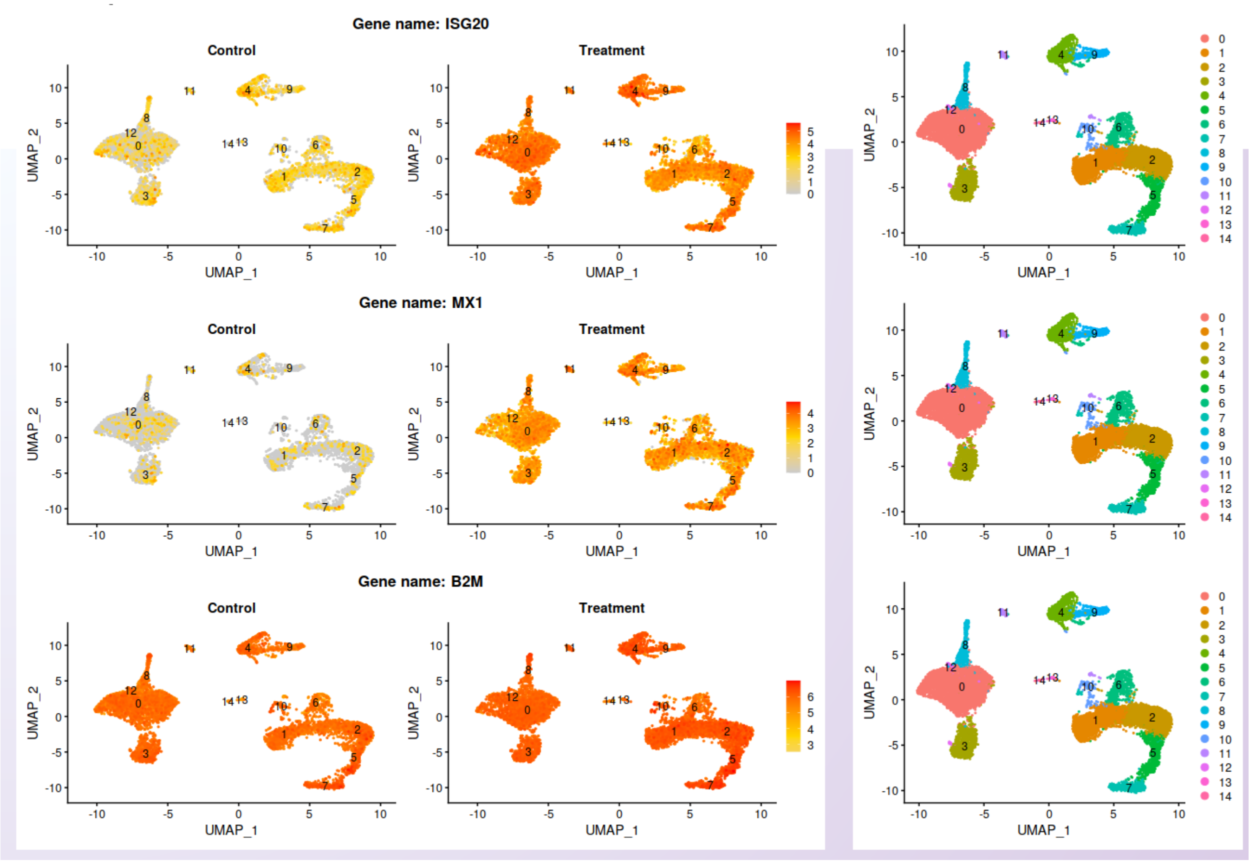


Fig. 5: Visualization of the expression profile of three of the genes shown on the heatmap in each sample.

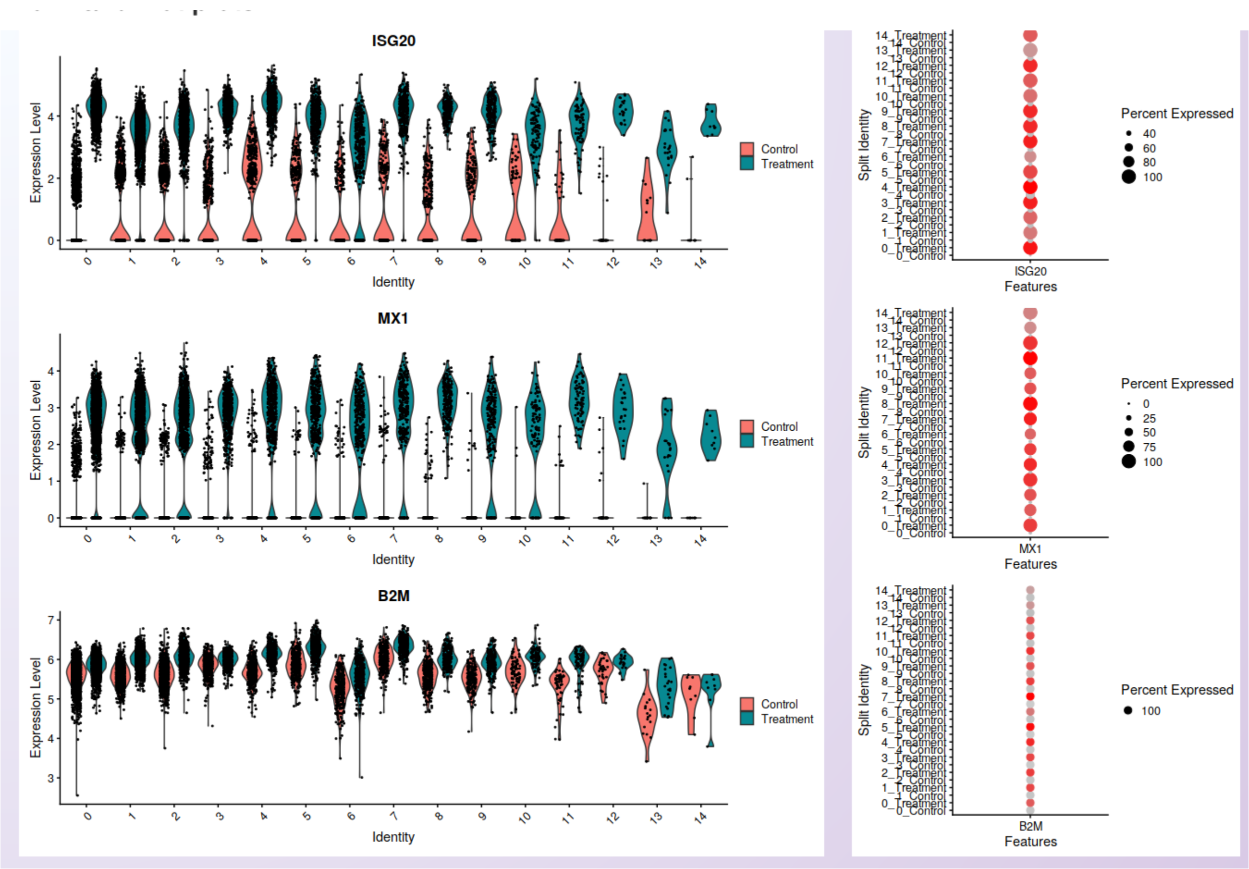


Fig. 6: Visualization of cells' distribution within each cluster according to the gene expression (violin plot; left) and the percentage of cells in each cluster expressing the gene (dot plot; right) in each sample. The three genes shown are the same used for the feature plots.

Scalable Vector Graphic (svg), allows the edition of all aspects of the plot by image edition software as Inkscape (free).

CHAPTER

FOURTEEN

TRAJECTORY INFERENCE

For the trajectory inference analysis, users can either execute it through capabilities of the embedded slingshot (Bioconductor) package or select another model contained in dynverse, executed using a docker image provided by dynverse. In both options, users only need to choose the model and initial parameters (see below). However, the direct execution of slingshot is faster than executing models via dynverse's docker image.

To inform the model's choice, it is recommended the reading of Saelens et al., 2019 paper, that benchmarked a diversity of models. Different models will perform better or worst depending on the topology of the developmental trajectory of the dataset (that is unknown a priory). Therefore, users need to consider what topology they may expect for their dataset. For example, slingshot performs well for bifurcated, multifurcated, or "tree" like topologies, but not for cyclic or more complex disconnected trajectories.

Warning: Some of the models included in dynverse are computationally intensive. It is strongly recommended to check the requirements for a model before executing it on Asc-Seurat. You can use dynguidelines web application to investigate the necessary resources to analyze your dataset. The amount of resources also depends on the number of cells and the complexity of the dataset.

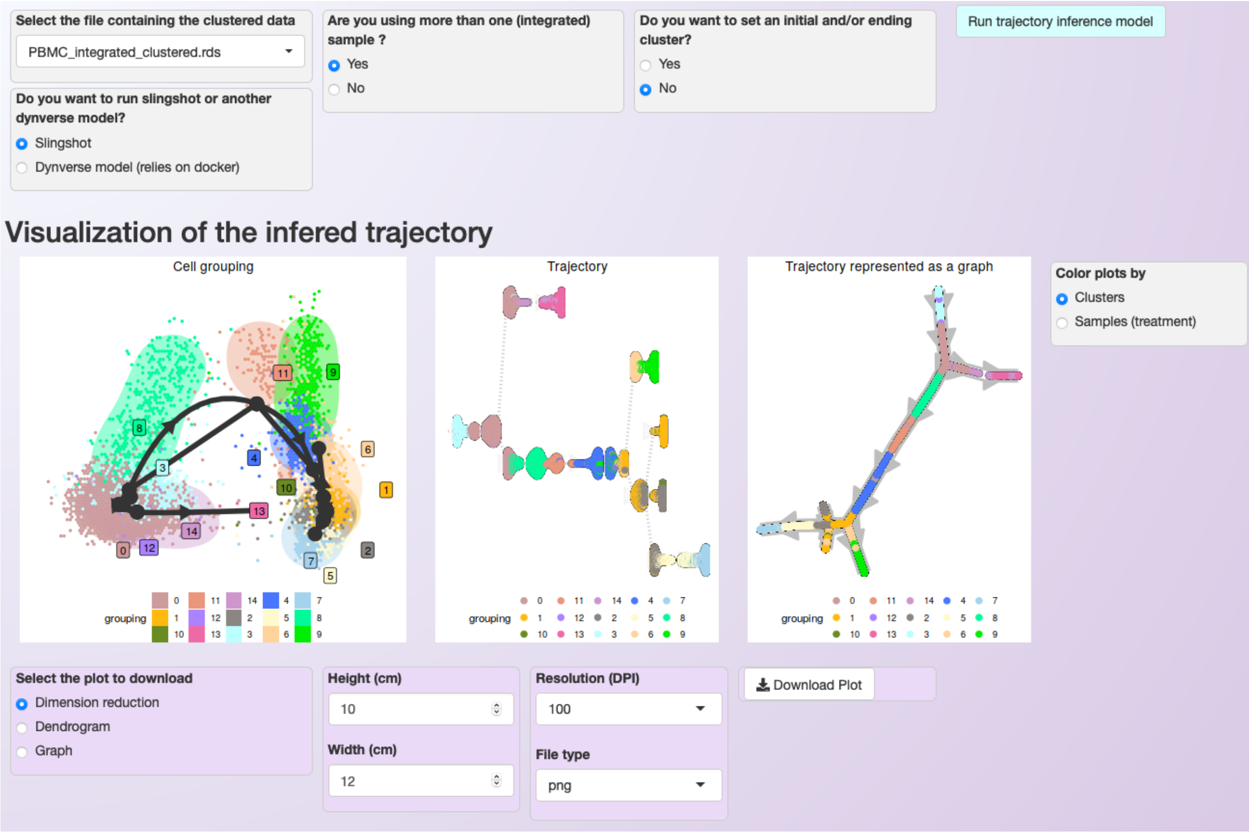
14.1 Executing the trajectory inference and trajectories visualization

To start the trajectory inference analysis, users need to save the clustered data in a specific folder automatically created during the installation (`RDS_files/`). Asc-Seurat recognizes the data automatically, and users can select the sample to be used. Next, users need to select the model to be used, inform if the data is composed of one or multiple integrated samples, and, optionally, inform the cluster(s) expected to be at the beginning and/or end of the inferred trajectory. After executing the analysis, three plots showing different inferred trajectory representations are generated. Moreover, when using an integrated dataset, users can also color the cells according to the sample of origin. To demonstrate these capabilities, we used the PBMC integrated dataset (containing two samples: Control and Treatment).

Note: The time to execute the trajectory varies from minutes to hours, depending on the complexity of the dataset and the chosen model. Visit dynguidelines web application for an estimative.

For the PBMC integrated dataset, slingshot was used to infer the developmental trajectory. Note that no cluster was select as the start or end of the trajectory, so slingshot makes this decision. If you know what cluster (or cell type) is expected at the beginning or end of the trajectory, providing this information will allow a better interpretation of the generated trajectory.

When users inform that multiple samples are used, Asc-Seurat offers coloring the cells by cluster identify or by samples. Both options are demonstrated below for the PBMC integrated dataset.



Select the file containing the clustered data

PBMC_integrated_clustered.rds

Are you using more than one (integrated) sample ?

● Yes

○ No

Do you want to set an initial and/or ending cluster?

○ Yes

● No

Run trajectory inference model

Do you want to run slingshot or another dynverse model?

● Slingshot

○ Dynverse model (relies on docker)

Visualization of the infered trajectory

Color plots by

● Clusters

○ Samples (treatment)

Select the plot to download

● Dimension reduction

○ Dendrogram

○ Graph

Height (cm)

10

Width (cm)

12

Resolution (DPI)

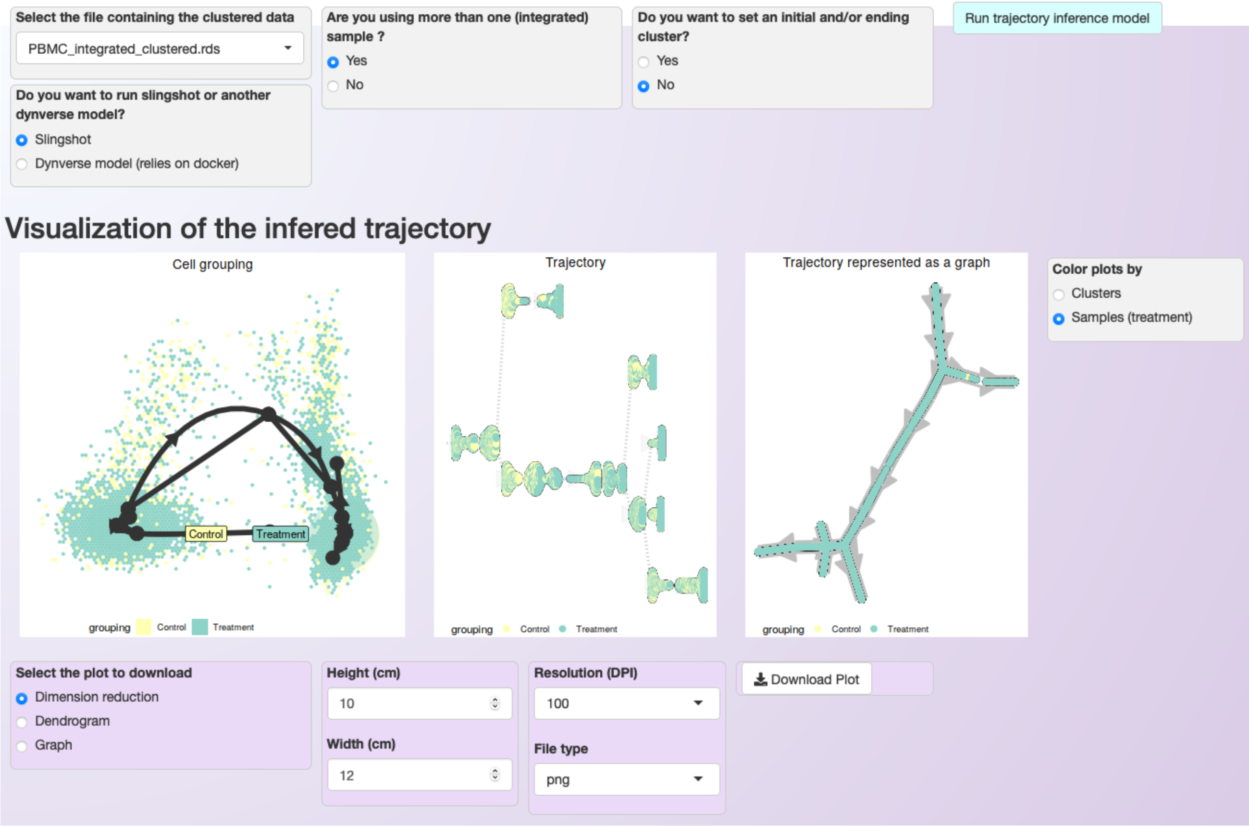
100

File type

png

Download Plot

Fig. 1: **Asc-Seurat provides multiple models for trajectory inference analysis and three options for trajectory visualization**. In this case, cells are colored by clusters.



Select the file containing the clustered data

PBMC_integrated_clustered.rds

Are you using more than one (integrated) sample ?

● Yes
○ No

Do you want to set an initial and/or ending cluster?

○ Yes
● No

Run trajectory inference model

Do you want to run slingshot or another dynverse model?

● Slingshot
○ Dynverse model (relies on docker)

Visualization of the infered trajectory

Color plots by

○ Clusters
● Samples (treatment)

Select the plot to download

● Dimension reduction
○ Dendrogram
○ Graph

Height (cm)

10

Width (cm)

12

Resolution (DPI)

100

File type

png

Download Plot

Cell grouping
Control
Treatment
grouping 🟨 Control 🟩 Treatment
Trajectory
grouping ● Control ● Treatment
Trajectory represented as a graph
grouping ● Control ● Treatment

Fig. 2: **Asc-Seurat provides multiple models for trajectory inference analysis and three options for trajectory visualization**. In this case, cells are colored by sample.

Tip: Suppose you are interested in studying the developmental trajectory of a subgroup of clusters only. In that case, it is better to exclude the other clusters than to try to infer the trajectory using the whole dataset. The model will execute quicker and provide a better resolution of the trajectory since the complexity of the dataset is reduced. Asc-Seurat allows the exclusion of clusters from your dataset, see *Selecting clusters of interest* (one sample) or *Selecting clusters of interest* (integrated dataset).

14.2 Expression visualization within the trajectory and identification of DEGs in the trajectory

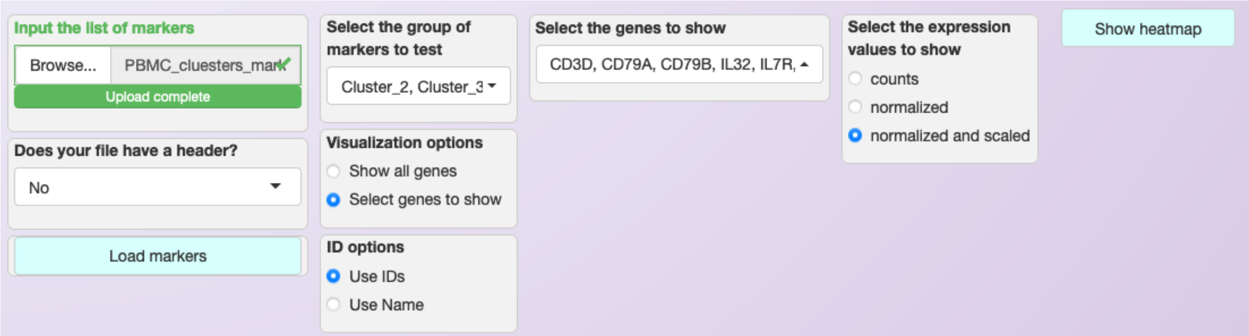
After inferring the developmental trajectory, it is possible to visualize the expression of genes of interest in the cells within the trajectory. Asc-Seurat provides two options for this visualization, 1) a heatmap displaying the expression of genes in each cell, ordered by the cell position within the trajectory, and 2) the visualization of the same three trajectory's representation shown above but colored by the gene expression.

Users can either load their list of genes of interest or identify DEGs within the trajectory for the visualization.

14.2.1 Visualizing the expression of a list of selected genes

To visualize the expression of specific genes, the process is similar to the described on *Expression visualization*. Asc-Seurat expects as input a csv (comma-separated value) file containing at least two columns. The first column must contain the gene ID as present in your dataset, and the second column is a grouping variable. An optional third column can contain the common names of each gene. Any additional column will be ignored. **No header is allowed for this file**.

After loading the input file, users can then select what group(s) of genes to explore, as well as select specific genes from each group. Moreover, if a third column is provided in the input file, users can use the common name of the genes instead of the gene IDs to select the genes to be shown.



Input the list of markers
Browse... PBMC_cluesters_mar ✓
Upload complete

Does your file have a header?
No

Load markers

Select the group of markers to test
Cluster_2, Cluster_3

Visualization options
○ Show all genes
● Select genes to show

ID options
● Use IDs
○ Use Name

Select the genes to show
CD3D, CD79A, CD79B, IL32, IL7R,

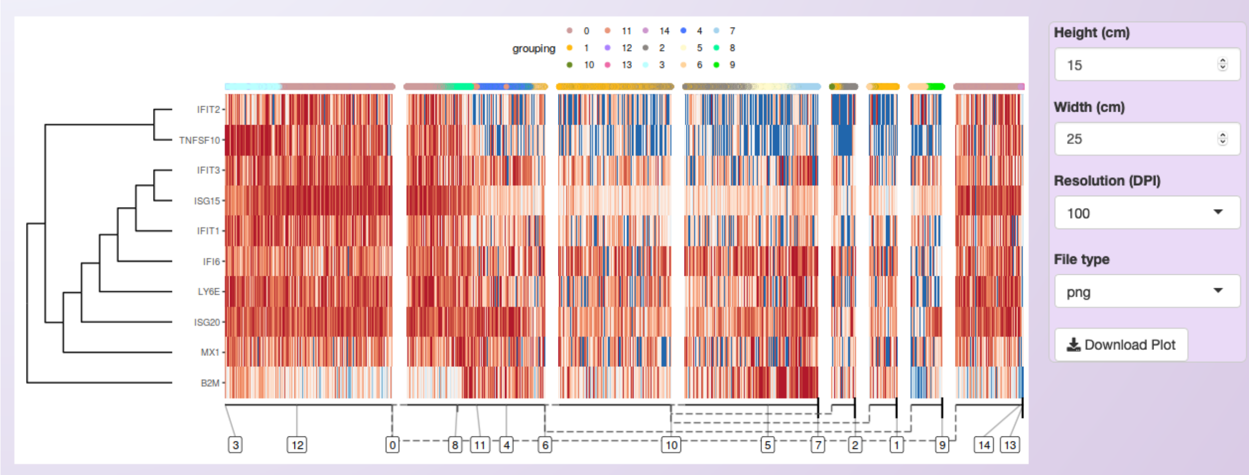
Select the expression values to show
○ counts
○ normalized
● normalized and scaled

Show heatmap

After choosing the genes, a heatmap showing the expression in the cells sorted by their position in the inferred trajectory is shown. Then, users can select genes for individual visualization.

As an example, it is shown the expression of the same ten DEGs identified for cluster 4 in the comparison of Control and Treatment for the PBMC integrated dataset (see *Markers identification and differential expression analysis*).

Next, three of those genes were selected to show the expression on the cells in the inferred trajectory.

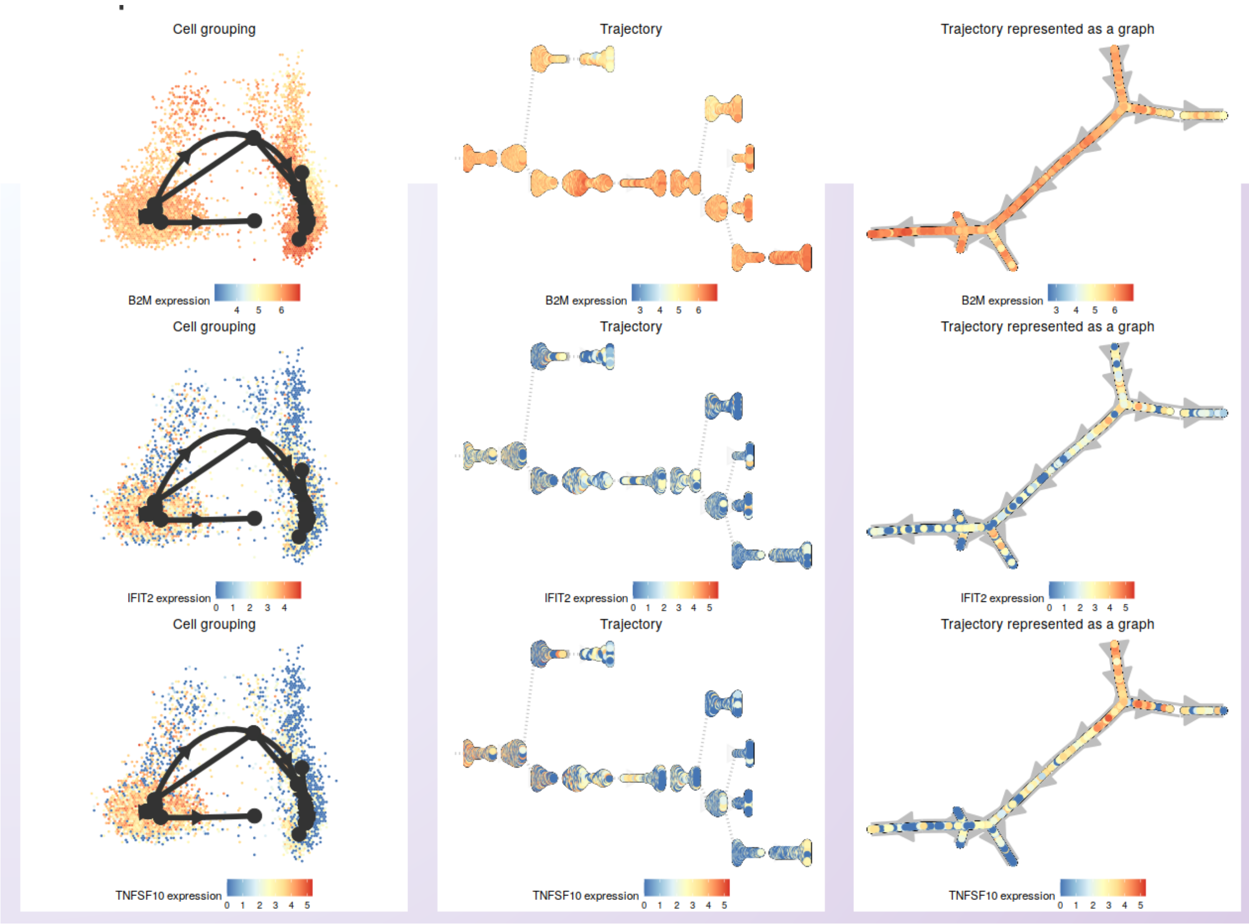


Height (cm)
15

Width (cm)
25

Resolution (DPI)
100

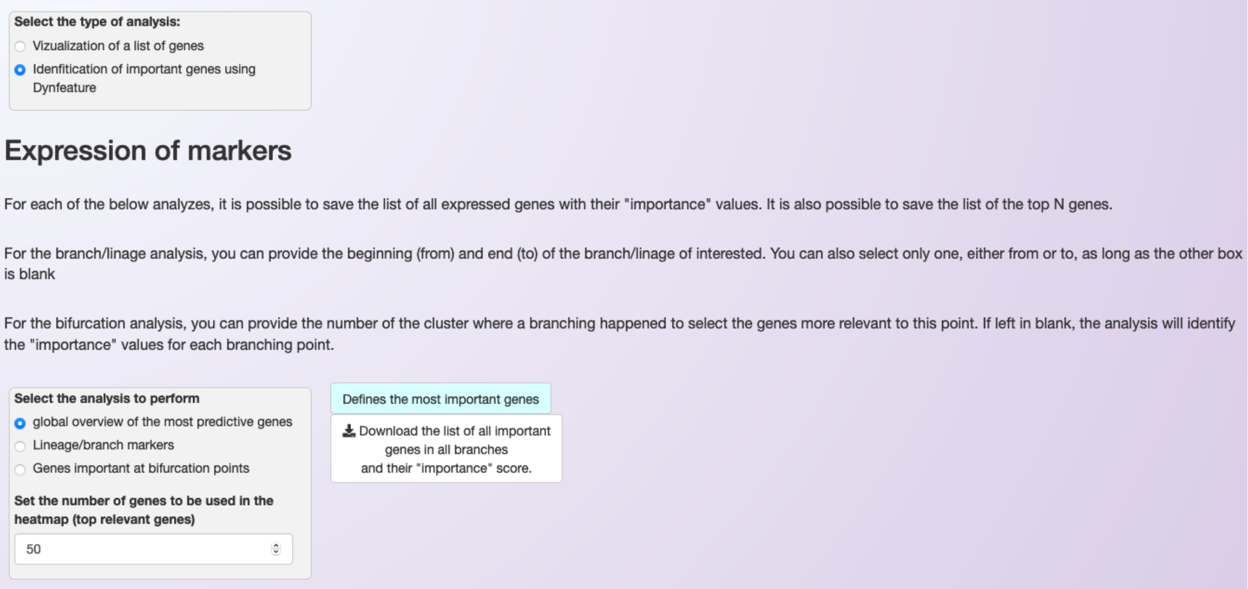
File type
png



14.2.2 Identification of DEGs in the trajectory

To identify differentially expressed genes, Asc-Seurat deploys the dynfeature, part of dynverse's collection of packages. Here we provide a short introduction to these methods. Please, visit dynverse's Trajectory differentially expression page for a demonstration of each method.

Asc-Seurat allows the search for DEGs within the whole trajectory, in a branch of the trajectory between two clusters or in a branching point. Each of these methods will rank all genes of the dataset. Therefore, users need to select the number of genes (ranked by the most important genes) to visualize in the heatmap. Also, users can download the list of all genes and their "importance values".



Select the type of analysis:

○ Vizualization of a list of genes

● Idenfitication of important genes using Dynfeature

Expression of markers

For each of the below analyzes, it is possible to save the list of all expressed genes with their "importance" values. It is also possible to save the list of the top N genes.

For the branch/linage analysis, you can provide the beginning (from) and end (to) of the branch/linage of interested. You can also select only one, either from or to, as long as the other box is blank

For the bifurcation analysis, you can provide the number of the cluster where a branching happened to select the genes more relevant to this point. If left in blank, the analysis will identify the "importance" values for each branching point.

Select the analysis to perform

● global overview of the most predictive genes

○ Lineage/branch markers

○ Genes important at bifurcation points

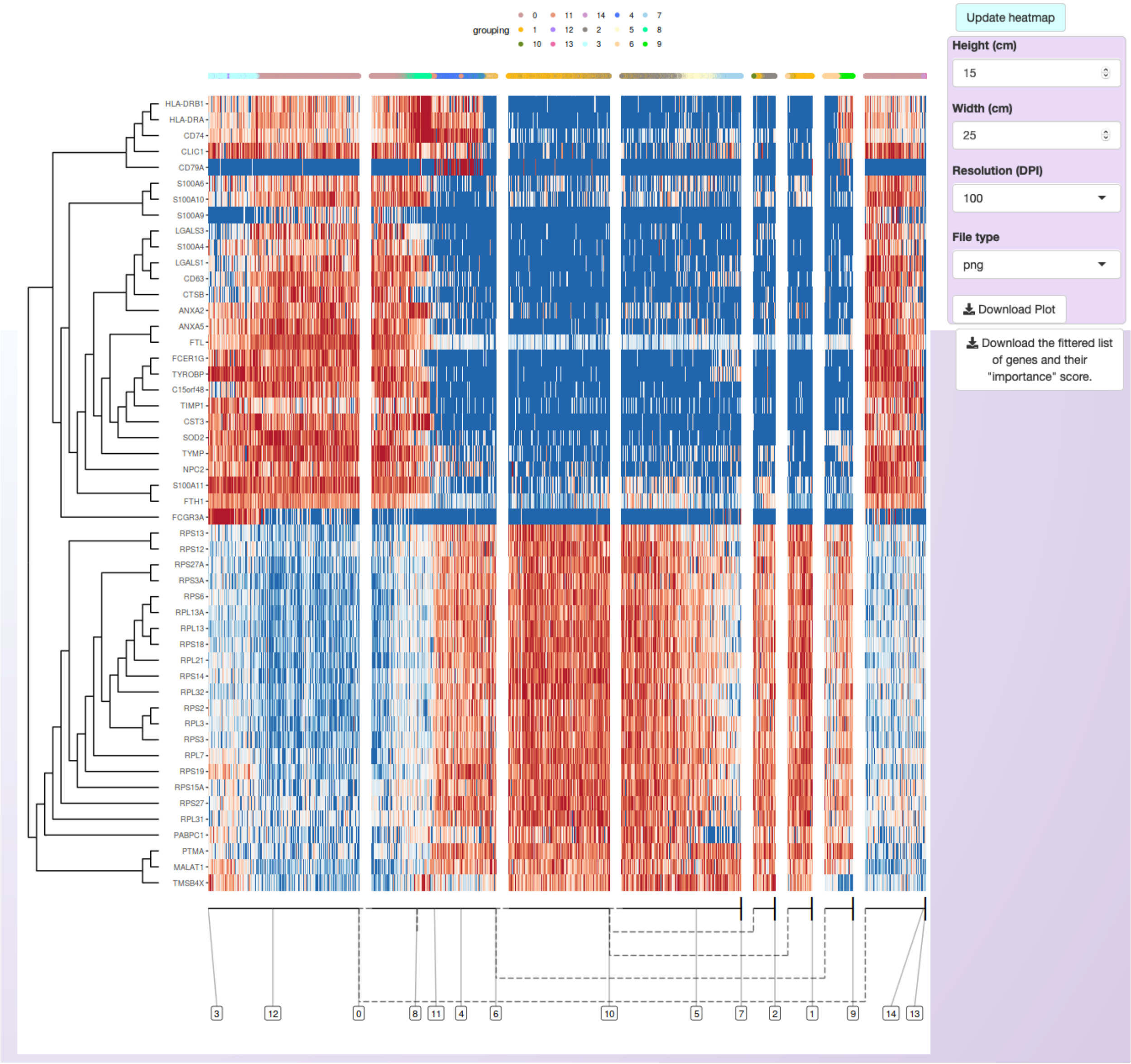
Set the number of genes to be used in the heatmap (top relevant genes)

50

Defines the most important genes
Download the list of all important genes in all branches and their "importance" score.

As an example, for the PBMC integrated dataset, we opted to show the 50 most significant DEGs within the trajectory, as ranked by their "importance" value on explaining the inferred trajectory.

From those, a few genes were selected for expression visualization on the trajectory.

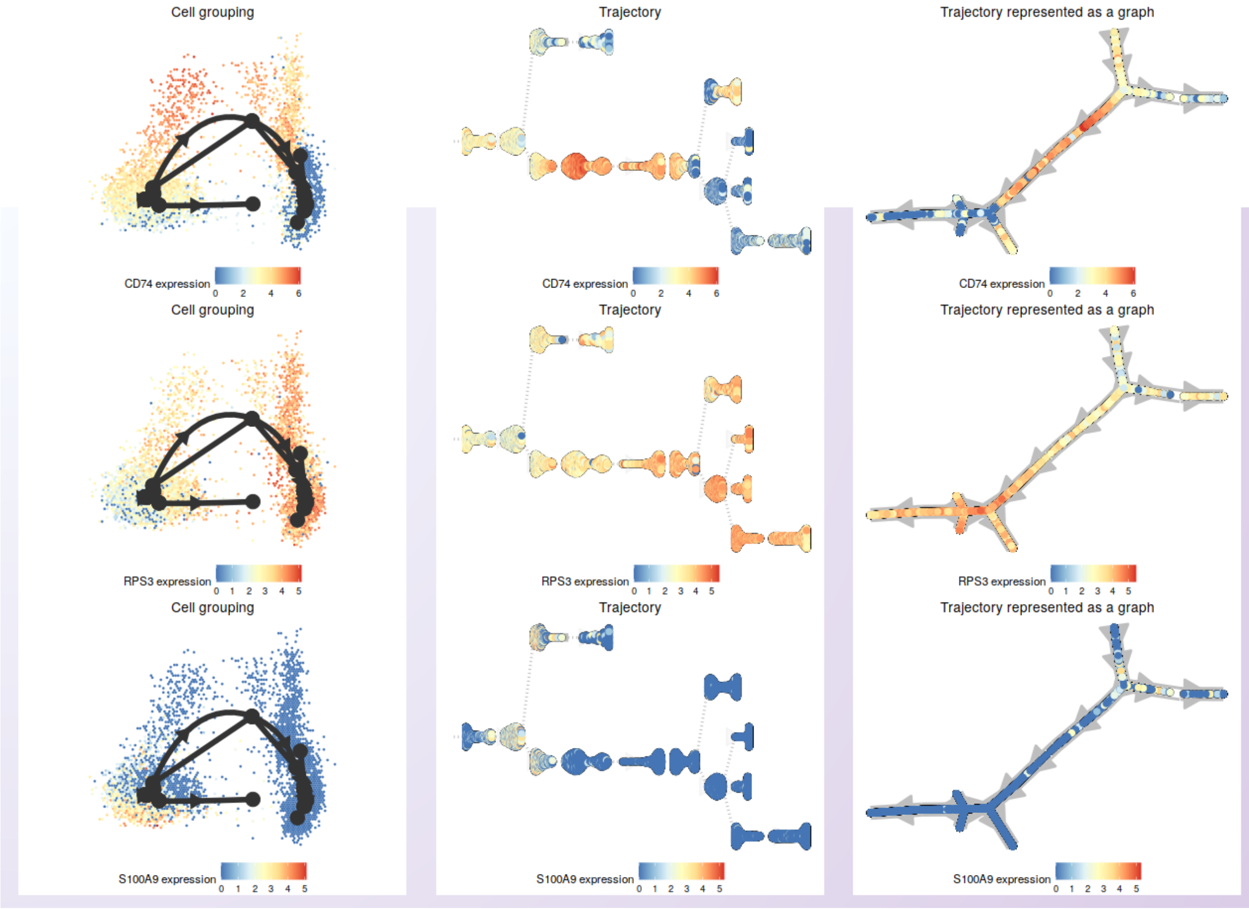


Height (cm): 15

Width (cm): 25

Resolution (DPI): 100

File type: png


Cell grouping
Trajectory
Trajectory represented as a graph
CD74 expression
0 2 4 6
CD74 expression
0 2 4 6
CD74 expression
0 2 4 6
Cell grouping
Trajectory
Trajectory represented as a graph
RPS3 expression
0 1 2 3 4 5
RPS3 expression
0 1 2 3 4 5
RPS3 expression
0 1 2 3 4 5
Cell grouping
Trajectory
Trajectory represented as a graph
S100A9 expression
0 1 2 3 4 5
S100A9 expression
0 1 2 3 4 5
S100A9 expression
0 1 2 3 4 5

CHAPTER

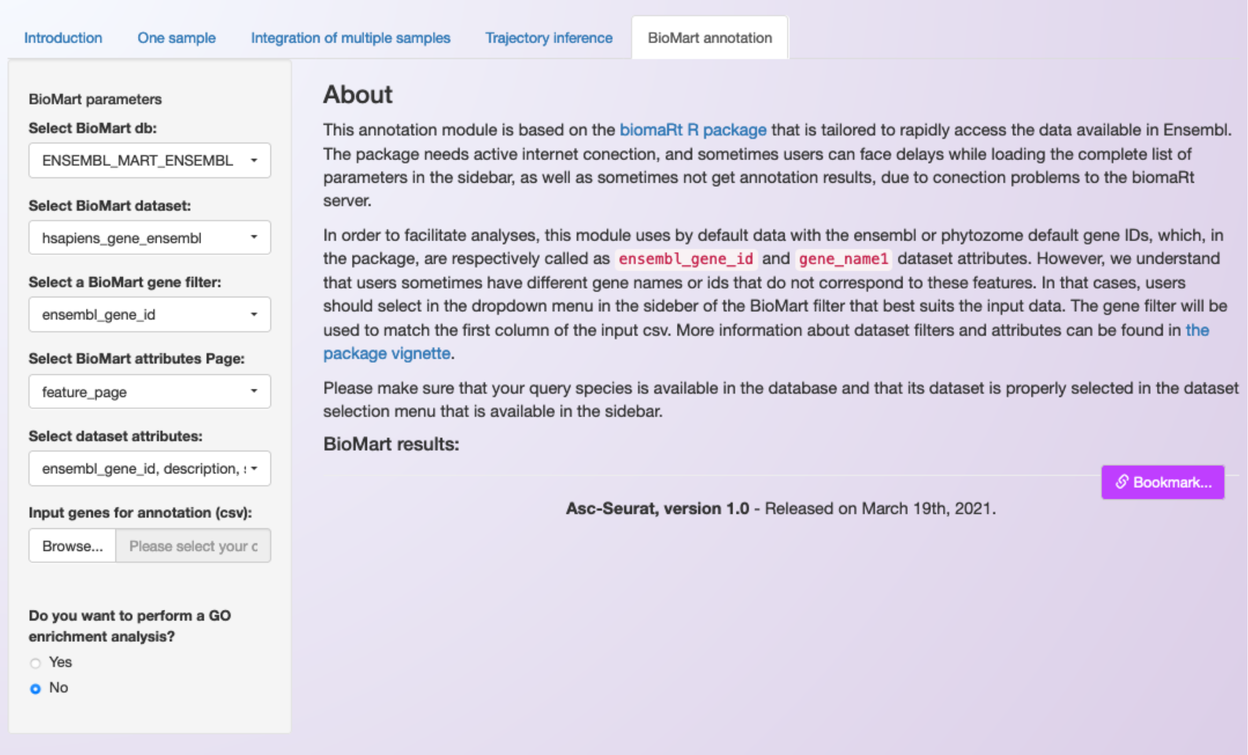
FIFTEEN

BIOMART ANNOTATION

The annotation module of Asc-Seurat is based on the biomaRt package (Bioconductor). BiomaRt is designed to facilitate the functional annotation of genes available for various species through the BioMart databases. To date, the primary databases in BioMart are the ones provided by Ensembl. Fortunately, biomaRt provides direct access to these datasets, and they can all be accessed via Asc-Seurat. Moreover, due to its importance for plant species, we also incorporated access to the Phytozome's BioMart database.

15.1 Functional annotation of genes

The annotation module of Asc-Seurat was designed to be simple to use (See image below). Nonetheless, a basic understanding of how BioMart queries are built is required so that users can select the filters and attributes needed. Please, visit biomaRt's vignettes for an overview.



BioMart parameters

Select BioMart db: ENSEMBL_MART_ENSEMBL

Select BioMart dataset: hsapiens_gene_ensembl

Select a BioMart gene filter: ensembl_gene_id

Select BioMart attributes Page: feature_page

Select dataset attributes: ensembl_gene_id, description, :

Input genes for annotation (csv): Please select your c

Do you want to perform a GO enrichment analysis?

○ Yes

● No

About

This annotation module is based on the biomaRt R package that is tailored to rapidly access the data available in Ensembl. The package needs active internet conection, and sometimes users can face delays while loading the complete list of parameters in the sidebar, as well as sometimes not get annotation results, due to conection problems to the biomaRt server.

In order to facilitate analyses, this module uses by default data with the ensembl or phytozome default gene IDs, which, in the package, are respectively called as `ensembl_gene_id` and `gene_name1` dataset attributes. However, we understand that users sometimes have different gene names or ids that do not correspond to these features. In that cases, users should select in the dropdown menu in the sideber of the BioMart filter that best suits the input data. The gene filter will be used to match the first column of the input csv. More information about dataset filters and attributes can be found in the package vignette.

Please make sure that your query species is available in the database and that its dataset is properly selected in the dataset selection menu that is available in the sidebar.

BioMart results:

Asc-Seurat, version 1.0 - Released on March 19th, 2021.

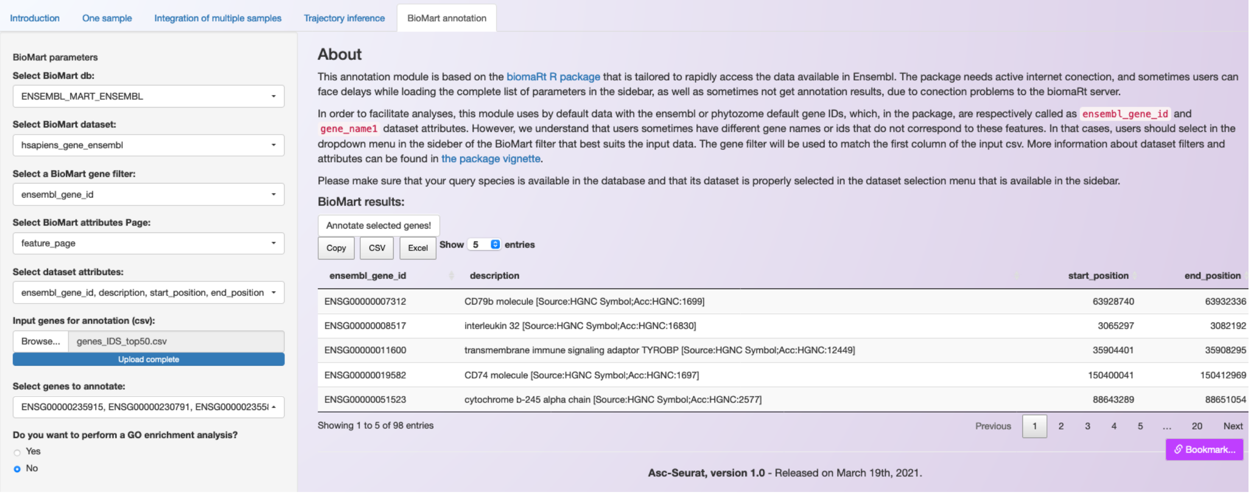
As shown in the image above, Asc-Seurat contains a sidebar on which users can select the best parameters for annotating their genes. Initially, users should select the database to use (Phytozome or one of Ensembl's databases). Then, Asc-Seurat will load it and display the datasets (species) available for the selected database.

After selecting the species' dataset to use, users can define the filter and attributes of the query. In summary, the filter corresponds to the dataset being used as input and, for most cases, will be the gene IDs or the gene names. The attributes are the information users want to extract from the database, e.g., description of the gene function, Gene Ontology (GO) terms, Pfam domains, etc. Please check this section of biomaRt's vignettes for an example.

After defining the filter and the attributes, users can provide a csv file containing a list of gene ids (or gene names) and start the query. Moreover, users can select only a subset of the genes listed in the csv file, reducing the time necessary for the annotation.

Note: The input csv file should contain one or more columns, separated by commas. A header is required, but users are free to use their choice of column(s) name(s). The only required information is the gene ids, or gene names, one entrance per line. Asc-Seurat will ignore other columns that might be present. The csv files generated within Asc-Seurat are adequate as input for the annotation.

To execute the annotation, users need to click on *Annotate selected genes!*. An iterative table containing the requested information will be generated. Also, users can download the list of annotated genes as a csv or an Excel file (see below).



BioMart parameters

Select BioMart db: ENSEMBL_MART_ENSEMBL

Select BioMart dataset: hsapiens_gene_ensembl

Select a BioMart gene filter: ensembl_gene_id

Select BioMart attributes Page: feature_page

Select dataset attributes: ensembl_gene_id, description, start_position, end_position

Input genes for annotation (csv): Browse... genes_IDS_top50.csv

Upload complete

Select genes to annotate: ENSG00000235915, ENSG00000230791, ENSG000002355

Do you want to perform a GO enrichment analysis?

○ Yes

● No

About

This annotation module is based on the biomaRt R package that is tailored to rapidly access the data available in Ensembl. The package needs active internet conection, and sometimes users can face delays while loading the complete list of parameters in the sidebar, as well as sometimes not get annotation results, due to conection problems to the biomaRt server.

In order to facilitate analyses, this module uses by default data with the ensembl or phytozome default gene IDs, which, in the package, are respectively called as `ensembl_gene_id` and `gene_name1` dataset attributes. However, we understand that users sometimes have different gene names or ids that do not correspond to these features. In that cases, users should select in the dropdown menu in the sideber of the BioMart filter that best suits the input data. The gene filter will be used to match the first column of the input csv. More information about dataset filters and attributes can be found in the package vignette.

Please make sure that your query species is available in the database and that its dataset is properly selected in the dataset selection menu that is available in the sidebar.

BioMart results:

Show 5 entries

ensembl_gene_id	description	start_position	end_position
ENSG00000007312	CD79b molecule [Source:HGNC Symbol;Acc:HGNC:1699]	63928740	63932336
ENSG00000008517	interleukin 32 [Source:HGNC Symbol;Acc:HGNC:16830]	3065297	3082192
ENSG00000011600	transmembrane immune signaling adaptor TYROBP [Source:HGNC Symbol;Acc:HGNC:12449]	35904401	35908295
ENSG00000019582	CD74 molecule [Source:HGNC Symbol;Acc:HGNC:1697]	150400041	150412969
ENSG00000051523	cytochrome b-245 alpha chain [Source:HGNC Symbol;Acc:HGNC:2577]	88643289	88651054

Showing 1 to 5 of 98 entries

Asc-Seurat, version 1.0 - Released on March 19th, 2021.

15.2 GO terms enrichment analysis

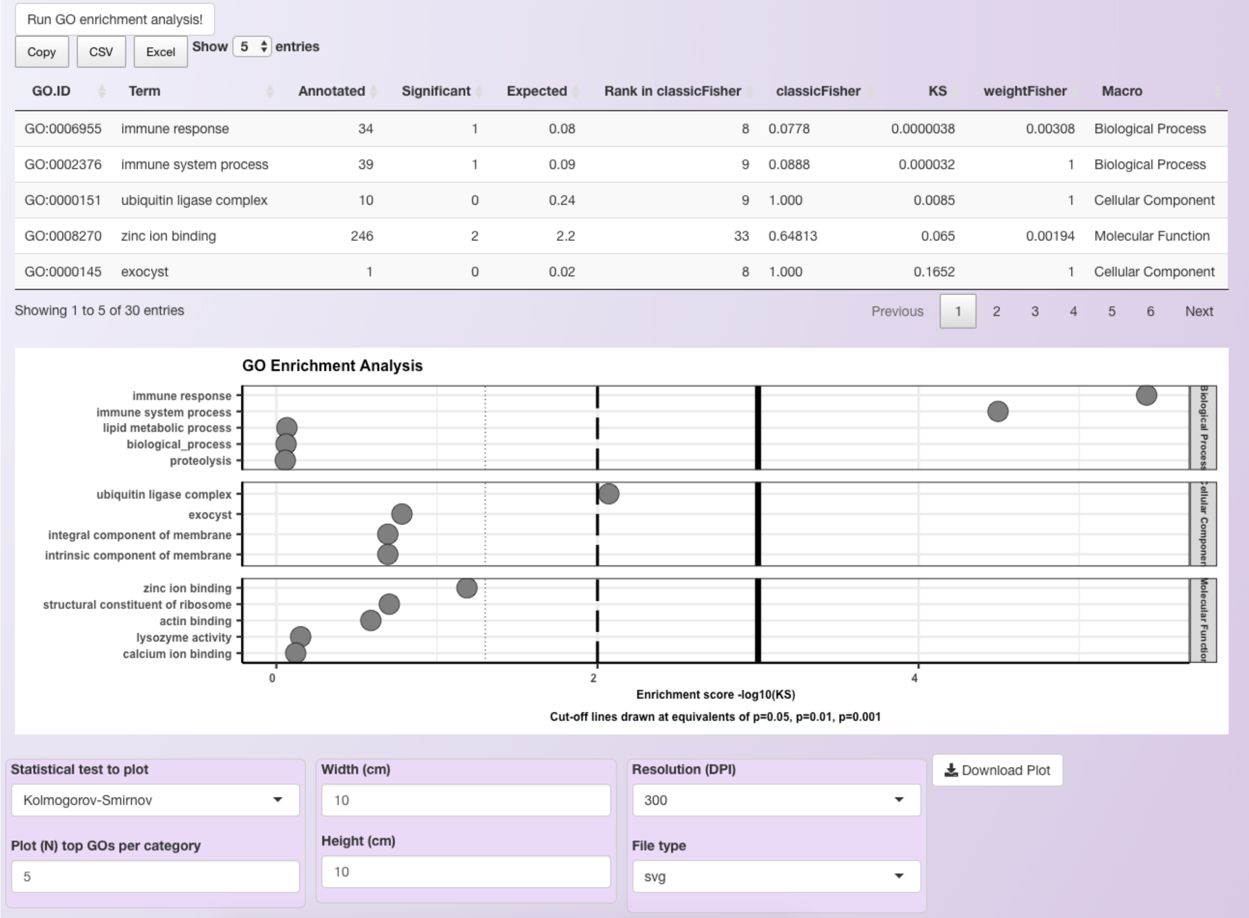
Asc-Seurat also provides an option to execute the GO terms enrichment analysis using topGO, a Bioconductor package.

This analysis aims to identify genes over/under-represented in the set of genes being annotated (known as target) compared to a broader set of genes (known as the universe). The universe can be a set of all genes expressed in the dataset or any set of genes that users desire to compare with the set of genes being annotated.

If users choose to execute this analysis, they need to provide a second csv file containing the list of genes to be used as the "universe" of the analysis.

Warning: Both sets of genes should contain the same type of identifier (i.e., gene ID). Also, be aware of extra spaces or any discrepancy between the two sets of genes' IDs.

At the end of the GO enrichment analysis, an iterative table containing all enriched GO terms is generated, which can be downloaded in the csv format or as an Excel file. Moreover, a plot showing the most significant GO terms is generated. Users can adjust the number of significant GO terms shown for each GO category in the plot (see below for an example using 5 GO terms per category).



GO.ID	Term	Annotated	Significant	Expected	Rank in classicFisher	classicFisher	KS	weightFisher	Macro
GO:0006955	immune response	34	1	0.08	8	0.0778	0.0000038	0.00308	Biological Process
GO:0002376	immune system process	39	1	0.09	9	0.0888	0.000032	1	Biological Process
GO:0000151	ubiquitin ligase complex	10	0	0.24	9	1.000	0.0085	1	Cellular Component
GO:0008270	zinc ion binding	246	2	2.2	33	0.64813	0.065	0.00194	Molecular Function
GO:0000145	exocyst	1	0	0.02	8	1.000	0.1652	1	Cellular Component

Showing 1 to 5 of 30 entries

Statistical test to plot: Kolmogorov-Smirnov

Plot (N) top GOs per category: 5

Width (cm): 10

Height (cm): 10

Resolution (DPI): 300

File type: svg

CHAPTER

SIXTEEN

ADVANCED PLOTS

As shown in the sections describing the expression visualization tools (*here* and *here*), Asc-Seurat provides a diversity of plots to explore your dataset. However, it focuses on exploring each gene individually, not providing tools to visualize the expression of multiple genes at once.

Starting on v2.0, Asc-Seurat also provides the capacity of generating dot plots and "stacked violin plots" comparing multiple genes.

Using an rds file containing the clustered data as input, users must provide a csv or tsv file in the same format described in the *expression visualization* section. Next, using the grouping variable, column two of the csv (or tsv), select the sets of genes to be used in the plot. Both violing and dot plot will be generated.

16.1 Stacked Violin plot

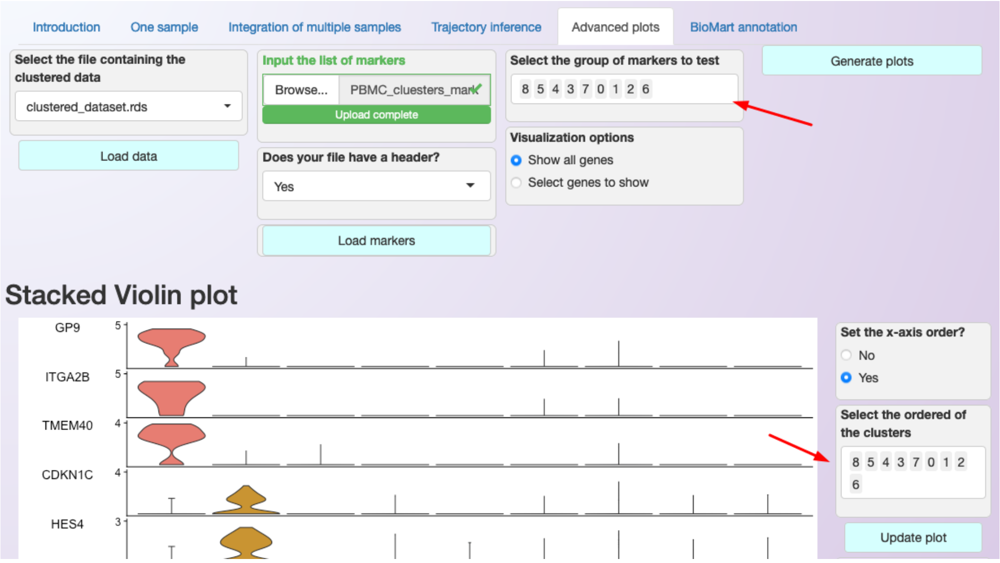
Stacked violin plots are a popular way to represent the expression of gene markers but are not provided by Seurat. Asc-Seurat's version of the stacked violin plot is built by adapting the code initially posted on the blog "DNA CONFESSES DATA SPEAK", by Dr. Ming Tang.

Note that the genes (y-axis) **will be displayed following the order of the grouping variable (column two of your file) selected by the user**. Once the plot is generated, users can choose the order of the clusters to show on the x-axis. For example, we show the expression profile of the three most significant gene markers identified for each cluster of the PBMC dataset.

An arbitrary order of the clusters is used in the plot, demonstrating how users can customize the result.

16.2 Multiple-genes Dot plot

A multiple-genes dot plot will be generated following the same order selected for the stacked violin plot.



Select the file containing the clustered data
clustered_dataset.rds

Input the list of markers
PBMC_cluesters_mark

Does your file have a header?
Yes

Select the group of markers to test
8 5 4 3 7 0 1 2 6

Visualization options
● Show all genes
○ Select genes to show

Set the x-axis order?
○ No
● Yes

Select the ordered of the clusters
8 5 4 3 7 0 1 2
6

Fig. 1: Interface for generating multiple genes plot. Note that users can select the order that genes (y-axis) and clusters (x-axis) are shown; see the red arrows in the image.

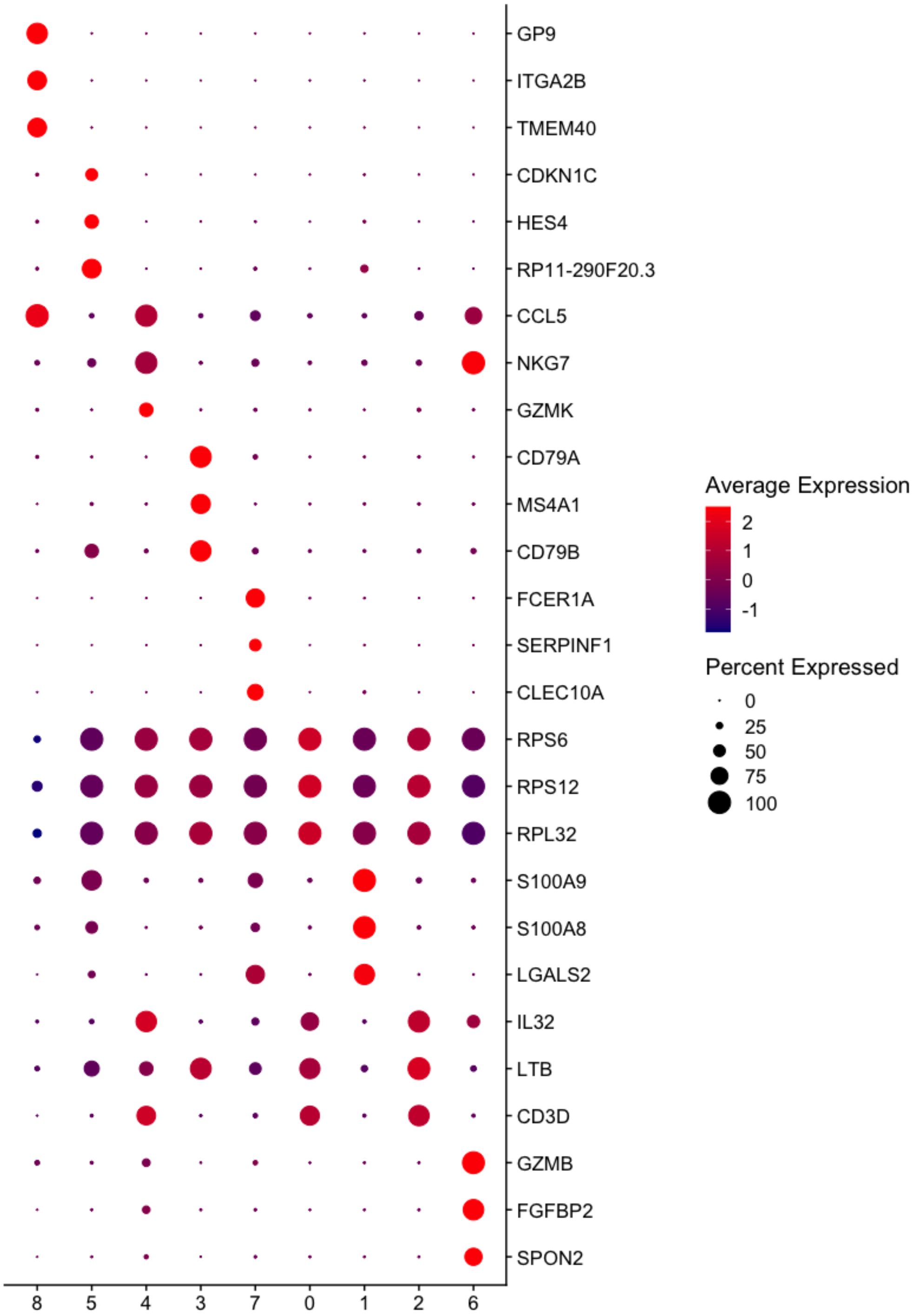


Fig. 3: Multiple-genes dot plot showing the three most significant markers of each cluster of the PBMC dataset. Observe that some of the significant markers are not specific for the cluster but present a higher level of expression than the other clusters.

CHAPTER

SEVENTEEN

RELEASE NOTES

- v2.1 - **Released on May 26th, 2021**.
  - Changes the assay used for differential expression analysis and visualization to "RNA" when using SCTransform normalization. Therefore, "SCT" assay is used for the steps until clustering the data.
  - Changes the output of the differential expression analysis to the format required for the visualization tools.
- v2.0 - **Released on May 19th, 2021**.
  - Inclusion of SCTransform normalization
  - Addition of stacked violin plots
  - Addition of multiple-genes dot plot
  - Improvements on the user interface
  - Improvements in the app stability
  - Fix of minor bugs.
- v1.0 - **Released on March 19th, 2021**.
  - Release of Asc-Seurat.

CHAPTER EIGHTEEN

REFERENCE

[1] Pereira WJ, Almeida FM, Balmant KM, Rodriguez DC, Triozzi PM, Schmidt HW, Dervinis C, Pappas Jr. GJ, Kirst M. Asc-Seurat – Analytical single-cell Seurat-based web application. BioRxiv, 2021.

CHAPTER NINETEEN

SUPPORT CONTACT

Have any questions or suggestions? Please contact us at GitHub.

Footnotes: