User-friendly, scalable tools and workflows for single-cell RNA-seq analysis

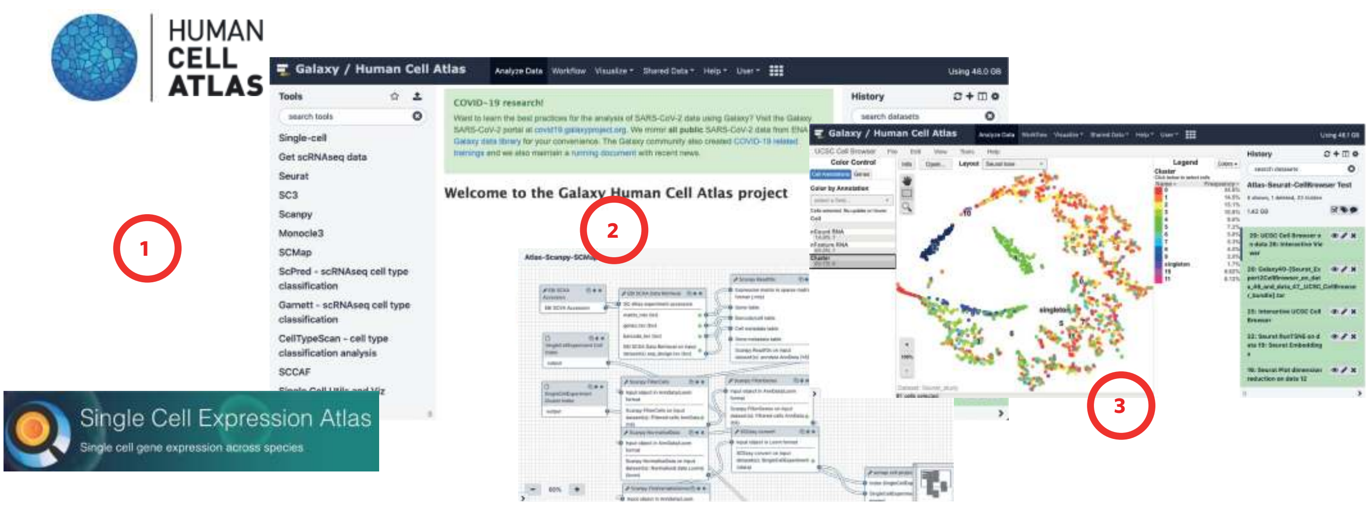
To the Editor — As single-cell RNA sequencing (scRNA-seq) becomes widespread, accessible and scalable computational pipelines for data analysis are needed. We introduce an interactive computational environment for single-cell studies based on Galaxy[1], with functions from established workflows. Single Cell Interactive Application (SCiAp) provides easy access to data from the Human Cell Atlas (HCA) and EMBL-EBI's Single Cell Expression Atlas (SCEA)[2] projects and can be deployed on different computing platforms, making single-cell data analysis of large-scale projects accessible to the scientific community.

Consortia such as the HCA, the Fly Cell Atlas and others are generating large numbers of scRNA-seq datasets that will be available for researchers to reuse alongside the analysis of their own datasets. For instance, the SCEA provides scRNA-seq datasets comprising over 3 million cells from 14 species, including a wide variety of cell types and tissues. This large collection of scRNA-seq data demands adequate computational infrastructure, analysis tools and workflows to help researchers make the most of it.

The Galaxy framework has enabled flexible and scalable deployment across multiple clouds through the Galaxy–Kubernetes integration[3], thereby supporting analysis of large datasets. Galaxy offers a user-friendly framework for building and sharing workflows. It is supported by a vibrant community of bioinformaticians who continually enrich the tool repository with analysis methods for applications such as scRNA-seq[4]. Built on Galaxy, SCiAp facilitates data access (HCA, SCEA and one's own data), downstream analysis, and visualization of scRNA-seq datasets. We share tools and workflows (including those used in the SCEA) in SCiAp that can run through the web interface or the command line. An instance, known as the HCA Galaxy instance, is available at https://humancellatlas.usegalaxy.eu/ (Fig. 1). Further technical details and usability, among many other topics, are covered in the Supplementary Methods.

A key feature of SCiAp is the ability to integrate tools from different workflows, written in different languages. We break monolithic tools into analysis modules, enabling users to try different competing tool sets and, where possible, integrate them into the same workflows. For example, we produced more than 20 modules for Scanpy[5], covering data input, filtering, normalization, variable genes, clustering, dimensionality reductions and trajectory methods, among others. Supplementary Table 1 shows all the tools integrated and the different functional modules into which they were broken; Supplementary Note 1 shows the integration of modules from different tools on analysis workflows. SCiAp provides functionality from Scanpy, Seurat[6], Monocle3[7], SC3[8], SCmap[9], Scater[10], SCCAF[11], SCPred[12], SCEasy and UCSC CellBrowser. Supplementary Figure 1 shows a map of scRNA-seq data analysis functionalities that are covered by tool wrappers contributed as part of this work and external contributions incorporated, shown accordingly.

In summary, SCiAp is a suite of components derived from commonly used tools in scRNA-seq analysis. Being based on Galaxy, it can be deployed on large computational infrastructures or on existing Galaxy instances, reducing software engineering complexities for the biological research community. Supplementary Table 2 shows a comparative overview between SCiAp and similar services. SCiAp outperforms in accessibility and the breadth of tool sets provided. We also provide the underlying tools that resolve software dependencies via Bioconda[13] and Biocontainers[14], which are commonly used frameworks in bioinformatics. Lab-based scientists with a deep understanding of a cellular system can use this computational framework to interrogate scRNA-seq data, propose further hypotheses and guide their experiments to explore the translational


HUMAN
CELL
ATLAS
Galaxy / Human Cell Atlas
Analyze Data
Workflow
Visualize
Shared Data
Help
User
Tools
search tools
Single-cell
Get scRNAseq data
Seurat
SC3
Scanpy
Monocle3
SCMap
ScPred - scRNAseq cell type classification
Garnett - scRNAseq cell type classification
CellTypeScan – cell type classification analysis
SCCAF
Single Cell Utils and Viz
History
search datasets
Atlas-Scanpy-SCMap
Single Cell Expression Atlas
Single cell gene expression across species
1
2
3

COVID–19 research!

Want to learn the best practices for the analysis of SARS-CoV-2 data using Galaxy? Visit the Galaxy SARS-CoV-2 portal at covid19.galaxyproject.org. We mirror all public SARS-CoV-2 data from ENA Galaxy data library for your convenience. The Galaxy community also created COVID-19 related trainings and we also maintain a running document with recent news.

Welcome to the Galaxy Human Cell Atlas project

Fig. 1 | SCiAp. (1) Load matrix data from HCA or SCEA directly into SCiAp Galaxy. (2) Run configurable scRNA-seq analysis through SCiAp. (3) Inspect results interactively through UCSC-CellBrowser and plots within Galaxy.

potential of large-scale, single-cell studies using the friendly Galaxy environment.

Data availability

Example input data, in the form of Galaxy histories, are available at http://usegalaxy.eu, with direct links available in Supplementary Note 1. Single Cell Expression Atlas data are directly available from https://www.ebi.ac.uk/gxa/sc and from its FTP site at ftp://ftp.ebi.ac.uk/pub/databases/microarray/data/atlas/sc_experiments/. The Human Cell Atlas data are available from https://data.humancellatlas.org/. In both cases, the appropriate Galaxy modules retrieve data directly from Single Cell Expression Atlas and the Human Cell Atlas.

Code availability

Code contributed here is made available through the GitHub repos, biocontainers, bioconda recipes and Galaxy Toolshed entries shown and linked in Supplementary Table 1 and Supplementary Note 2.

Pablo Moreno[1]✉, Ni Huang[1,2], Jonathan R. Manning[1], Suhaib Mohammed[1], Andrey Solovyev[1], Krzysztof Polanski[2], Wendi Bacon[1], Ruben Chazarra[1], Carlos Talavera-López[1,2], Maria A. Doyle[3,4], Guilhem Marnier[1], Björn Grüning[5], Helena Rasche[5], Nancy George[1], Silvie Korena Fexova[1], Mohamed Alibi[1], Zhichao Miao[1], Yasset Perez-Riverol[1], Maximilian Haeussler[6], Alvis Brazma[1], Sarah Teichmann[2], Kerstin B. Meyer[2] and Irene Papatheodorou[1]✉

[1]*European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, UK.* [2]*Wellcome Sanger Institute, Wellcome Genome Campus, Hinxton, UK.* [3]*Research Computing Facility, Peter MacCallum Cancer Centre, Melbourne, Victoria, Australia.* [4]*Sir Peter MacCallum Department of Oncology, The University of Melbourne, Melbourne, Victoria, Australia.* [5]*Department of Computer Science, University of Freiburg, Freibug, Germany.* [6]*Genomics Institute, University of California at Santa Cruz, Santa Cruz, CA, USA.*
✉*e-mail: pmoreno@ebi.ac.uk; irenep@ebi.ac.uk*

Published online: 29 March 2021
https://doi.org/10.1038/s41592-021-01102-w

References

1. Afgan, E. et al. *Nucleic Acids Res.* **46**, W537–W544 (2018). W1.
2. Papatheodorou, I. et al. *Nucleic Acids Res.* **48**, D77–D83 (2020). D1.
3. Moreno, P. et al. Preprint at *bioRxiv* https://doi.org/10.1101/488643 (2018).
4. Tekman, M. et al. *Gigascience* https://doi.org/10.1093/gigascience/giaa102 (2020).
5. Wolf, F. A., Angerer, P. & Theis, F. J. *Genome Biol.* **19**, 15 (2018).
6. Stuart, T. et al. *Cell* **177**, 1888–1902.e21 (2019).
7. Cao, J. et al. *Nature* **566**, 496–502 (2019).
8. Kiselev, V. Y. et al. *Nat. Methods* **14**, 483–486 (2017).
9. Kiselev, V. Y., Yiu, A. & Hemberg, M. *Nat. Methods* **15**, 359–362 (2018).
10. McCarthy, D. J., Campbell, K. R., Lun, A. T. L. & Wills, Q. F. *Bioinformatics* **33**, 1179–1186 (2017).
11. Miao, Z. et al. *Nat. Methods* **17**, 621–628 (2020).
12. Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q. & Powell, J. E. *Genome Biol.* **20**, 264 (2019).
13. Grüning, B. et al. *Nat. Methods* **15**, 475–476 (2018).
14. da Veiga Leprevost, F. et al. *Bioinformatics* **33**, 2580–2582 (2017).

Acknowledgements

The authors acknowledge the invaluable support from the Bioconda, Biocontainers and Galaxy communities. P.M., J.R.M., N.H., K.B.M., I.P.: Silicon Valley Community Foundation 2018-183498. M.H.: Silicon Valley Community Foundation 2018-182809 and NHGRI 5U41HG002371-19.

Author contributions

P.M. designed architecture; P.M. and J.R.M. were lead technical contributors; P.M., N.H., J.R.M., S.M., A.S., K.P., R.C. and G.M. implemented CLIs and tools; N.H., C.T.-L., A.B. and S.T. advised on methods; W.B., P.M., J.R.M., C.T.-L., N.G., S.K.F., Z.M. and M.H. ran training; N.H., P.M. and J.R.M. worked on tool interoperability; M.A.D., Z.M., M.H. implemented tools; B.G., H.R. and P.M. set up and managed the Human Cell Atlas Galaxy instance; W.B. designed training; M.A. and P.M. set up cloud infrastructure for training; W.B., N.G., S.K.F., J.R.M., A.S. and P.M. tested Galaxy tools. Y.P.-R. and B.G. designed and advised on architecture; A.B., S.T. helped conceive the study; I.P. and P.M. designed the study; I.P. and K.B.M. conceived the study.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-021-01102-w.

Chunkflow: hybrid cloud processing of large 3D images by convolutional nets

To the Editor — Automated microscopes with both high resolution and large field of view are generating terascale and even petascale 3D images. A local cluster might not have enough computational resources to process them in reasonable time, but public cloud platforms can provide computational resources on demand. Convolutional networks have become the state-of-the-art approach for 3D biological image analysis[1,2], and cloud processing by 3D convolutional nets has been used for processing independent small image stacks[3–5]. However, cloud computing tools to perform distributed processing of terascale or petascale 3D images by convolutional nets are lacking. Here, we report chunkflow, a framework for distributing computational tasks over both cloud and local computational resources, including both GPUs and CPUs with multiple deep-learning framework back ends, to maximize efficiency, increase flexibility and reduce cost.

In chunkflow's architecture, a task production front end communicates with task consumption back end through a cloud queue (Fig. 1a). Each task is defined as the processing of a subvolume ('chunk') of the entire volume (Supplementary Fig. 1). Task production is the ingestion of tasks to a cloud queue (Amazon Web Services Simple Queue Service) (Fig. 1a,b, Supplementary Note and Supplementary Fig. 1). Task production by itself does not require setting up, or even accessing, a cluster. The task consumption back end is a set of computational workers, which continually fetch and perform tasks from the queue until it is empty. According to the fetched task, a worker 'cuts out' each chunk from the entire volume to read it from cloud storage. Workers may be distributed across multiple cloud vendors and local computers. Each worker communicates only with the cloud queue and cloud storage, without direct dependency.

Chunkflow has several features (Supplementary Note). Chunkflow is fault tolerant using the visibility timeout mechanism of Simple Queue Service; it can utilize the cheap but unstable cloud instances offered by many cloud vendors; it has a modular and extensible design; it