OXFORD

(GIGA)n SCIENCE

GigaScience, 2017, 1–7

doi: xx.xxxx/xxxx
Manuscript in Preparation
Paper

PAPER

A single-cell RNA-seq analysis suite using the Galaxy Framework

Mehmet Tekman[1,*,†], Second Author[2,*,†], Third Author[2] and Rolf Backofen[1,*]

[1]Chair of Bioinformatics, University of Freiburg, Freiburg, Germany, and [2]Second Institution

[*]tekman@informatik.uni-freiburg.de;backofen@informatik.uni-freiburg.de
[†]Contributed equally.

Abstract

Background The turbulent ecosystem of single-cell RNA-seq tools has up to recently been plagued by a plethora of diverging analysis strategies, inconsistent file formats, and a wide range of compatibility issues between different software packages. Due to the initial sparseness of the data caused by low coverage and intrinsic cell noise meant that strategies could be developed on very few compute resources to solve the problems of normalisation, confounder removal, and clustering. The differing sequencing protocols produced by the various small labs has produced many different types of datasets, and specific strategies to analyse them. This has led to a saturation of different scRNA tools, many of which see very little use. The rise of 10X Genomics datasets that are now saturating the field has provided a new standard that has refocused the energies of the different analysis pipelines towards analysis consistency, and has once again driven the demand for large computing requirements to cluster the increasingly growing datasets.
Given the cluster-based interpretive nature of an scRNA-seq analysis, it is not always clear to initial researchers whether the steps they have undertaken to get to the final clustering are necessarily correct. The disparity between the statistically-driven algorithmic nature of the analysis to the underlying cell biology is a frightening prospect for many researchers first introduced into the topic.
Results The Galaxy bioinformatic reproducible computing framework addresses both of these paradigms, by providing tools, workflows and trainings that enables users to perform one-click 10X pre-processing, and also empowers them to de-multiplex raw sequencing data for more custom setups. The resulting downstream analysis is supported by a range of high quality interoperable suites separated into common stages of analysis: inspection, filtering, normalization, confounder removal and clustering. The teaching resources cover a host of different concepts with the core focus of unifying the algorithms and statistics with the underlying cell biology. Access to all resources is provided at the https://singlecell.usegalaxy.eu portal.
Conclusions The reproducible and training-oriented ethos of the Galaxy community and framework provides a sustainable HPC environment for users to run flexible analyses on both 10X and smaller datasets. The scRNA-oriented trainings within the Galaxy Training Network paired with the frequent training workshops hosted by the Galaxy Team provides a means to for users to be taught and to teach others to navigate through the complexities of a scRNA-seq analysis.

Key words: scRNA; Galaxy; resources; HPC; single-cell; 10X

Background

The continuing rise in single cell technologies has led to previously unprecedented levels of analysis into cell heterogene-

Compiled on: September 25, 2019.
Draft manuscript prepared by the author.

Key Points

- The analysis of scRNA is currently stabilising towards 10X Genomics datasets
- A rapid
- One last point.

ity within tissue samples, providing new insights into developmental and differentiation pathways for a wide range of different disciplines. Gene expression studies are now performed at a cellular level of resolution, which compared to bulk RNA-seq methods, allows researchers to model their tissue samples as distributions of different expression instead of as an average.

The various expression profiles uncovered within tissue samples infer discrete cell types which are related to one another across an "expression landscape", where relationships between the more distinct profiles are inferred via distance-metrics or manifold learning techniques which aim to model the continuous biological process of cell differentiation from multipotent stem cells to mature cell types, and infer lineage and differentiation pathways between these various distinct and transient cell types [?].

Trajectory analysis can then be performed to measure the up or down regulation of significant genes along lineage branches in order to uncover the factors and extracellular triggers that can coerce a pluripotent cell to become biased towards one cell fate outcome compared to another. This undertaking has created a new frontier of exploration in cell biology, where researchers have assembled reference maps for different cell lines in order to fully record these cell dynamics and characteristics to create a global "atlas" of cells [1].

Pitfalls and Technical Challenges

With each new protocol comes a host of new technical problems to overcome. The first wave of software utilities to deal with the analysis of single cell datasets were statistical packages, aimed at tackling the issue of "dropout events" during sequencing, which would manifest as a high prevalence of zero-entries in over 80% of the feature-count matrix. These zeroes were problematic, since they could not be trivially ignored as their presence stated that either the cell did not produce any molecules for that transcript, or that the sequencer simply did not detect them. Traditional normalisation techniques such as *EdgeR* and *Limma* had to be adapted to accommodate for this uncertainty, and new ones were created that harnessed hurdle models, data imputation via manifold learning techniques, or by pooling subsets of cells together and building general linear models [2].

With the downstream analysis packages attempting to solve the dropouts via stochastic methods, the upstream sequencing technologies also aspired to solve the capture efficiency via well, droplet, and flow cytometry based protocols, all of which lend importance to the process of barcoding sequencing reads. In each protocol, cells are tagged with cell barcodes such that any reads derived from them can be unambiguously quantified to originate from a specific cell, and the inclusion of unique molecular identifiers (UMIs) are also employed to mitigate the effects of amplification bias of transcripts within the same cell. The detection, extraction, and (de-)multiplexing of these barcodes is therefore one of the first hurdles researchers encounter when receiving their raw FASTQ data from the sequencing facility.

The Burgeoning Software Ecosystem

Since its conception, over two hundred different packages and many pipelines have been developed to assist researchers in the analysis of scRNA-seq [Citation Needed]. The vast majority of these packages were written for the R programming language since many of the novel normalisation methods developed to handle the dropout events, depended on statistical packages that were primarily R-based [Citation Needed]. Standalone analysis suites emerged as the different authors of these packages rapidly expanded their methods to encapsulate all facets of the single-cell analysis, often creating compatibility issues with previous package versions. The Bioconductor repository provided some much-needed stability by hosting stable releases, but researchers were still prone to building directly from repository sources in order to reap the benefits of the new features in the more developed upstream versions [Citation Needed].

Another issue was the proliferation of the many different and quickly evolving R-based file formats for processing and storing the data, such as `SingleCellExperiment` as used by the *Scater* suite, `SCSeq` from *RaceID*, and `SeuratObject` from *Seurat* [3, 4, 5, 6]. Many new packages would cater only towards one format or suite, leading to the common problem that data processed in one suite could not be reliably processed by methods in another. This incompatibility between packages fuelled a choice of one analysis suite over another, or required researchers to dig deeper into the internal semantics of R to slot components together. These problems only accelerated the rapid development of these suites, leading to further version instability. As a result of this analysis diversity, there are many opinionated tutorials on how to perform scRNA-seq analysis each oriented around one of these pipelines such as those proposed by Hemberg (Scater-oriented), **THING**, **THING**, and the multiple analysis pipelines listed in BioConductor [Citation Needed].

Rise of 10x Genomics

In 2015, 10x Genomics released their *GemCode* product, which was a droplet-seq based protocol capable of sequencing tens of thousands of cells with an average cell quality higher than other facilities [7]. This unprecedented level of throughput steadily gained traction amongst researchers and startups seeking to perform single-cell analysis, and thus 10x datasets began to prevail in the field.

10x Genomics provided software that was able to perform much of the pre-processing, and provided feature-count matrices in a transparent HDF5-based format which provided a means of efficient matrix storage and exchange, and ultimately removed the restriction for downstream analysis modules to be written in R. From the user perspective, literate datasets no longer needed to be analysed in RStudio but were now permitted in less language-restrictive notebooks such as Jupyter[8, 9].

One of the first packages to make use of this language independence was the ScanPy suite, written in Python, a more widely used language [10, 11]. Within a year, ScanPy had become the dominant suite for analysing 10x datasets, and quickly adopted the `HDF5` format, with their own `AnnData` ex-

tension better suited to their pipeline. They were not alone in this effort, as the Seurat team adopted the `LOOM` format for their datasets with similar goals. However, the dominance of ScanPy became much more pronounced as it began to rapidly integrate the methods of other standalone packages into its codebase, making it the natural choice for users who wanted to achieve more without compromising on compatibility between methods [Citation Needed].

Solutions in the Cloud

As the size of datasets scaled, so did the computing resources required to analyse them, both in terms of the processing power and in storage. Galaxy is an open source biocomputing infrastructure that encapsulates the three main tenets of science; reproducibility, peer review, and open-access - all freely accessible within the web-browser [12]. It hosts a wide range of highly-cited bioinformatic tools with many different versions, enables users to freely create their own workflows via a seamless drag-and-drop interface.

Galaxy makes use of *Conda* tool environments in order to ensure that the bioinformatics tools will always be able to run, even when the library dependencies for a tool have changed, by building tools under locked version dependencies and bundling them together in a self-contained environment [**?**]. These *Conda* environments provide a concise solution for the package version instability that plagues scRNA-seq analysis notebooks, both in terms of reproducibility and analysis flexibility. A user can run the quality control stages with an older version of ScanPy, whilst reaping the benefits of the upstream improvements in the clustering stage. By allowing the user to select multiple versions of the same tool, and by further permitting different versions of the tools within a workflow, Galaxy enables an unprecedented level of free-flow analysis by letting researchers pick and choose the best aspects of a tool without worrying about the underlying software libraries.

Analyses are not limited to one pipeline either, as the datasets which are passed between tools can easily be interpreted by a different tool which is capable of reading that dataset. In the case of scRNA-seq, Galaxy makes good use of the `LOOM` and `AnnData` which enables the inter-exchange of modules from different tools and further extends flexibility and functionality by again letting the user decide which component of a tool would be best suited for a specific part of an analysis.

Galaxy also provides a wide host of learning resources, with the aim of guiding users step-by-step through an analysis, often reproducing the results of published works. The teaching and training materials are part of the Galaxy Training Network (GTN) which is a worldwide collaborative effort to produce high-quality teaching material in order to educate users in how to analyse their data, and in turn to train others of the same materials via easily deployable workshops. The GTN has grown rapidly since it's conception and gains new volunteers every year who each contribute and coordinate training and teaching events, maintain topic and subtopics, translate tutorials into multiple languages, and provide peer review on new material [**?**].

Methods

The analysis of scRNA-seq within Galaxy was a two-pronged effort concentrated on bringing high quality single-cell tools into Galaxy, and providing the necessary workflows and training to accompany them. As mentioned in the previous section, this effort needed to overcome incompatible file format issues, unstable packages due to rapid development, and standardisation of the actual analysis.

The tutorials are split into two main parts that describe the post-processing and downstream analysis, referring to the distinct stages of constructing a count matrix from initial sequencing data, and then performing cluster analysis on the count matrix. These stages are very far from one another in terms of their information content, since the pre-processing stage requires the researcher to have more wetlab knowledge than a bioinformatician would normally need, and the downstream analysis stage requires the researcher to be familiar with machine learning concepts that a wetlab scientist might not be too familiar with.

The tutorials are designed to appeal to both the biologist and the statistician, such that each can benefit from the other's knowledge and, most importantly, bridge the knowledge gaps that separate the two.

Pre-processing Workflows

The pre-processing scRNA-seq materials tackle the two most common use-cases that researchers will encounter when they first begin the field: processing scRNA-seq data from 10x Genomics, and processing data generated from more custom protocols.

Before the era of 10x Genomics, scRNA-seq data had to be demultiplexed, mapped, and quantified. The demultiplexing stage requires an intimate knowledge of cell barcodes and Unique Molecular Identifiers (UMIs) which are protocol dependent and requires that the bioinformatician knows exactly where and how the data they are processing were generated. One common pitfall at the very first stage is determining exactly how many cells to expect in the FASTQ input data, and this requires threee crucial piece of information: which read contains the barcodes (or which subset of both the forward and reverse reads contains the barcodes); which specific barcodes were used in the analysis, and determining the number of acceptable barcode mismatches/errors and how to resolve or cluster them.

Naive strategies involve using a known barcode template and querying against the FASTQ to profile the number of reads that align to a specific barcode, often employing 'knee' methods to estimate this amount. However, this approach ultimately fails since certain cells are more likely to be over-represented compared to others, and some cell barcodes may contain more un-mappable reads compared to others meaning that the metric of higher read library sizes are not necessarily correlated with a better-defined cell. Ultimately, the bioinformatician must go to the sequencing lab and ask them which cell barcodes were used, as these are often not specific to the protocol but to the technician who designed them, with the idea that they should not align to the reference genome or transcriptome.

One-click Pre-processing

10x Genomics simplified the scRNA package ecosystem by using a language independent file format, and streamlining much of the barcode particularities with their *Cell Ranger* pipeline, allowing researchers to focus more on internal biological variability of their datasets.

The 10x pre-processing workflow given in Galaxy uses *RNA STARsolo* utility as a drop-in replacement for *Cell Ranger*, as it is a feature update of the already exist *RNA STAR* tool already in Galaxy, and because it boasts a ten-fold speedup in comparison [].

Though the memory requirements are also an order of magnitude higher, they are easily catered for using the Galaxy Framework which does not suffer from such memory con-

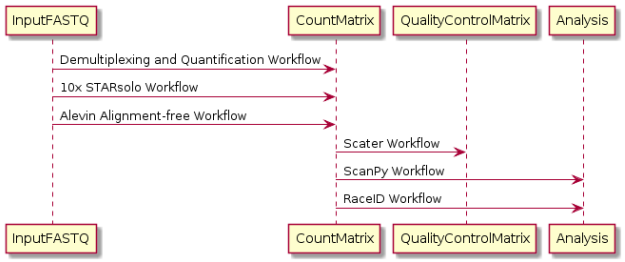

Figure 1. UML of workflows

straints, and has the benefit of not requiring the Illumina lane read information to perform the processing either. The training and workflow for this is therefore quite straightforward and follows the same mode of discovery and analysis. In a similar vein, there is also an Alevin-based workflow which also performs the demultiplexing, alignment (without mapping), and quantification in a single step to produce a count matrix. This is ideally paired with the downstream Monocle workflow to perform the full analysis, but the choice is left to the user due to the interexchangeability of the file formats being generated.

Training and Subdomains

The training materials cover several topics and are written in a flexible markdown format that empowers contributors to write tutorials on their expert topics without resorting to bloated editors, whilst enabling the community to peer-review such contributions easily via git or on Github by using standard diff utilities. Tutorials usually consist of a hands-on workflow that guides the user through an analysis with Galaxy utilising a step-by-step approach, and is often accompanied with a slide deck that either serve to explain standalone concepts more concisely, or are used during workshops and trainings as a way to introduce the user to the topic. In an effort to maintain reproducibility in science, all tutorials require example workflows, and all materials required to run the workflows and tutorials are hosted for free open access at Zenodo with a DOI tag.

The Galaxy tools and the GTN are further tied together by Galaxy subdomains, that better encapsulate the various topics into their own self-contained spheres of influence. These complement the training material by providing only the necessary Galaxy tools from the total set of Galaxy utilities such that the toolbar is not over-cluttered with tools that are not so relevant to the material (e.g. Variant Analysis tools being included in scRNA-seq resources), and also the benefit that smaller specialised Galaxy instances can be deployed with much smaller redundancy since only the core tools (e.g. text manipulation, FASTQ quality control, etc.) and the topic-specific utilities are packaged with it.

In this light, the `singlecellomics.usegalaxy.eu` subdomain hosts the entirety of the single-cell materials, tools, workflows, and single-cell related events, and it is this which is the focus of the paper. Given the extremely broadening scope of scRNA-seq, the Human Cell Atlas also have their own subdomain in this regard and maintain their materials at `humancellatlas.usegalaxy.eu`.

Integrate the Following....

Many of these are computationally and statistically complex for the average biologically-minded researcher, and thus the uptake of many of these packages is dependent on whether the methods come from the main downstream analysis pipelines such as Seurat or ScanPy, or whether the methods from these packages have been incorporated into them.

The reason for this is three-fold: with differing formats and inter-package incompatibility being a major factor, since no one will work with a package that creates its own container format or is not compatible with some of the main formats such as Loom, AnnData, or SingleCellExperiment; platform incompatibilites, where much of the software was written in R and therefore pipelines were driven via R/Bash pipelines, whereas actual pipeline frameworks such as snakemake, Galaxy, or CWL are more python-driven and use more transparent formats; Different pipelines produce different results, where the stochastic nature of the analyses means that any uncertainty in a crucial quality control stage upstream (such as filtering) will propagate forward into the downstream sections to yield different results. This uncertainty, and the statistically-driven methods to overcome them leaves a wide information gap for researchers simply trying to understand the underlying dynamics of a cell.

The custom pre-processing uses the CELSeq2 protocol following the barcoding strategies at the local Max-Planck Freiburg laboratory as its main template, but the training is flexible to accommodate any droplet or well-based protocol. The training pictographically walks users through the concepts of extracting cell barcodes, both at the over-archingly conceptial and basic file previewing level, and elucidates the use of UMIs and its role and significance in the process of read deduplication, and explains and instructs the user in the process of performing further quality controls on their data during the post-mapping process via *RNA STAR* and basic BAM Filter utilities that are native to Galaxy. At each stage, the user's knowledge is queried via expandable Question Box dialogs, and helpful hints for future processing are given via Comment Boxes, all written in transparent Markdown to the specification of the GTN contributing guidelines designed to aid users.

The downstream modules are defined by the five main stages of downstream scRNA-seq analysis as defined by the Hemberg group; filtering, normalisation, confounder removal, clustering, and trajectory inference. There are three workflows to aid in this process, each sporting a different well-established scRNA-seq pipeline tool. The Scater pipeline was contributed by the XXXX group, and follows an intuitive visualise-edit-visualise paradigm which provides a very understandable means to perform further quality control on a count matrix by use of repeated incremental changes on a dataset through the use of PCA and library size based metrics. Once this pre-analysis stage is complete and that the user is now confident with the quality of their data, they can perform the analysis via one or both of two comprehensive workflows: RaceID, and ScanPy.

The RaceID package was developed initially to analyse rare cell transcriptomes whilst being robust against noise, and thus is ideal for working with smaller datasets in the range of 300 to 1000 cells, though due to its impressive cell lineage and fate predictions models it can also be used on larger datasets albeit with some scaling costs. The ScanPy pipeline was developed as the Python counter response to overinflated R package ecosystem for scRNA-seq which was the dominant language for such analyses, and it was one of the first packages with native 10x genomics file format support. Since then it has grown substantially, and has been re-implementing much of the newer methods released in BioConductor, therefore providing a single source to perform many different types of the same analysis and becoming a dominant leader in the field where others have been slow to adapt.

Both workflows guide the user from the initial filtering step (if required) through to the clustering and final trajectory inference and inspection, mostly through a guided linear path because historically different pipelines do not work so well with one another. However, through the use of a convenience wrap-

per, the different modules of each pipeline are now able to communicate with one another via the AnnData datatype. The native RaceID R datatypes are converted into AnnData so that the analysis via one workflow can be continued within another, should the user wish to reap the benefits of one or more modules from another pipeline.

The modules emulate the five main stages of analysis mentioned previously, where filtering, normalisation, and confounder removal are typically separated into distinct stages. During the filtering stage, the initial count matrix removes low-quality or unwanted cells using commonly used parameters such as minimum gene detection sensitivity and minimum library size, and low-quality genes are also removed under similar metrics, where the minimum number of cells for a gene to be included is decided. The Scater workflow also offers a PCA-based method to help with the feature selection so that only the highly variable genes are left in the analysis.

There is always the danger of over-filtering a dataset, and indeed some normalisation methods rely on a background noise model generated from the expression of less variable or housekeeping genes, so it is important that the user first performs a naive analysis and only refines their analysis to boost the signal-to-noise ratio.

The normalisation of a step aims to remove any technical factors that are not relevant to the analysis, such as the library size, where cells of the same time are likely to differ from one another more by the number of transcripts they exhibit due to several factors that can affect it. The first and foremost is cell capture efficiency, where different cells produce more or less transcripts based on the amplification and coverage conditions they are sequenced in. The second is the presence of dropout events which manifest as counts of "zeroes" in the final count matrix but are uncertainly derived from the molecule existing in the cell and simply not being detected, or that the molecule itself was not present. This uncertainty alone led to a wide selection of different normalisation techniques that try to model this expression either via hurdle models, or imputing the data via manifold learning techniques, or working around the issue by pooling subsets of cells together.

In this regard, both the RaceID and ScanPy workflows offer many different options for normalisation and users are encouraged to take advantage of the branching workflow model of Galaxy to explore all possible avenues.

Other sources of variability stem from unwanted biological contributions known as confounder effects, such as cell cycle effects and transcription. Depending on what stage of the cell cycle a cell was sequenced at, two cells of the same type might cluster differently simply because one might have more transcripts due to being in the M-phase of the cell cycle. Library sizes not withstanding, it is the variability in specific cell cycle genes that can be the main driving factor in the overall variability. Thankfully, these effects are linear and are therefore quite easy to regress out, and we replicate an entire standalone ScanPy workflow dedicated to detecting and visualising the effects based on the original notebook.

The transcription effects are harder to model, as these are semi-stochastic and are as of yet still not well understood. In bulk RNA-seq the expression of genes undergoing transcription are averaged to give "high" or "low" signals producing a global effect that gives the false impression that transcription is a continuous process. The reality is more complex; where cells undergo transcription in "bursts" of activity followed by periods of no activity, at irregular intervals. At the bulk level these discrete processes are smoothed to give a continuous effect, but at the cell level it could mean that even two directly adjacent cells of the same type normalised to the same number of transcripts can still have wildly different levels of expression for a gene due to this process. This is not something that can be countered for, but it does educate the users in the factors that they can and cannot control in an analysis, and how much variability they can expect to see.

Once a user has obtained a count matrix they are confident in, they can perform clustering where they are first educated in the use of commonly used clustering techniques such as k-means and hierarchical clustering, as well as dimension reduction techniques, through the use of helpful images and community examples. In the ScanPy tutorial, the louvain clustering approach is explained via a standalone slide deck to assist in the workflow.

The clustering and the cluster inspection are notably separated into distinct utilities here, with the understanding that the same initial clustering can appear differently under different projections, for example tSNE or UMAP. Ultimately the user is encouraged to play with the plotting parameters to yield the best looking clusters for the same static clustering parameters. For tSNE this often means adjusting the perplexity parameter without having to rerun the entire tSNE clustering computation, though UMAP also has such parameters.

The inspection wrappers reflect the modules offered by the native packages, with package-specific information overlayed on top of the map projections in the manner that the package authors assumed was best. However, the AnnData and LOOM specification store this map projection data separately, and so the user is not at the mercy of these plotting tools, with plenty of HTML5-based interactive visualisations available at their disposal such that the user can query individual cell features without having to generate static images via tools such as *cellxgene* which are also available on the Galaxy server. Though these tools are excellent at dynamically displaying map projections, especially 3-dimensional ones, further computation must be performed to perform a full pseudotime analysis.

The cell pseudotime series analysis is often referred to as the trajectory inference stage, since cells are ordered along a trajectory to reflect the continuous changes to gene expression along a pathway under the assumption that the cells are transitioning from one type to another.

For the trajectory inference stage there is the PAGA graph abstraction technique championed by ScanPy, and there is also the FateID and StemID packages for the RaceID workflow. The former provides a level of graph abstraction to the datasets in order to infer a community graph structure which it can use to learn the shape of the data and infer pathways between neighbourhoods. The latter is more intuitive, in that it constructs a minimal spanning tree of related clusters that infer lineage, and cell fate decisions can be explored by individually querying branches in this tree, as a function of the genes which are up or down regulated along the currently explored pathway. The statistical strength and significance of each pathway guides the user along more valid trajectories that would more accurately reflect the biological variation occurring within transitioning cells.

The insights and novel cell types discovered in these analyses can also be integrated into the Human Cell Atlas portal, which is an initiative that aims to classify unique or rare cell types and their transitive properties in order to build a comprehensive map of cells that can be used to inverstigate the various differentiation pathways of multipotent stem cells.

The single-cell materials on the GTN are growing substantially every year, with at first only one pre-processing tutorial in 2018, one downstream tutorial at the start of 2019, and at the current time of writing three pre-processing tutorials and three downstream analysis workflows, further accompanied by visualisations. The first single-cell workshop was given internally within the Freiburg MeInBio consortium, but is now available at the [Feb. meeting thing] and the summer and winter Freiburg Galaxy workshops.

With single-cell RNA firmly within the Galaxy framework, echoes the efforts to standardise the field in order to promote reproducible research. The size of the datasets are substantial, and the computing resources required to handle them are growing as the sequencing technology scales, requiring more and more researchers to migrate towards cloud-based solutions in order to reap the benefits of superior hardware; computing abilities, storage requirements, and most of all data redundancy.

Those looking for stability in the field would have to look hard to find it, due to the non-standardisation of the R package ecosystem in which most of the analysis packages were written. Galaxy abstracts the user from the format specifics, by exposing the user only to the tool and data mindset, where tools simply ingest and produce data.

The community comes together regularly during scheduled CoFests to review, contribute, and actively maintain the training materials, and the number of volunteers is growing every year with named roles assigned to interested parties. This upkeep ensures that the Galaxy resources will continue to outlive other materials which typically decay over time once interest in them declines. By visiting and revisiting material in these scheduled slots, the material can only continue to improve, and now reaches even more audiences via the use of language translation tools which ensures that the international community is never left out.

Overall, the materials and tools presented in this paper represent the first fully comprehensive interactive guide to producing a single-cell RNA-seq analysis from the data capture stage all the way to the final publication, with the GTN community fully supporting the user the entire way. The materials grow with the user's knowledge, and expert users are known to give back to the material, enabling a constructive cycle of knowledge that emulates the ideals of open and transparent science through the use of web-based tools and a strong biocomputing framework to support it.

Each of the trainings seamlessly leads on from topic to the next, with the markdown encouraging authors to set tutorial pre-requisities and follow up tutorials, all encapsulated within the single topic. Not only does this allow users to derive a seamless route through the flurry of tutorials, but it also enables a branching tree structure of topics to be generated so that users can guide their own learning.

Citations and References

This is a 3rd level heading

This is a 4th level heading.

THIS IS A 5TH LEVEL HEADING.

Figures and Tables

Data Description

Analyses

Discussion

Methods

Availability of source code and requirements (optional, if code is present)

Lists the following:

- Project name: e.g. My bioinformatics project
- Project home page: e.g. http://sourceforge.net/projects/mged
- Operating system(s): e.g. Platform independent
- Programming language: e.g. Java
- Other requirements: e.g. Java 1.3.1 or higher, Tomcat 4.0 or higher
- License: e.g. GNU GPL, FreeBSD etc. Any restrictions to use by non-academics: e.g. licence needed

Availability of supporting data and materials

Declarations

List of abbreviations

If abbreviations are used in the text they should be defined in the text at first use, and a list of abbreviations should be provided in alphabetical order.

Ethical Approval (optional)

Manuscripts reporting studies involving human participants, human data or human tissue must:

- include a statement on ethics approval and consent (even where the need for approval was waived)
- include the name of the ethics committee that approved the study and the committee's reference number if appropriate

Studies involving animals must include a statement on ethics approval and have been treated in a humane manner in line with the ARRIVE guidelines.

See our editorial policies for more information.

If your manuscript does not report on or involve the use of any animal or human data or tissue, this section is not applicable to your submission. Please state "Not applicable" in this section.

Consent for publication

If your manuscript contains any individual person's data in any form, consent to publish must be obtained from that person, or in the case of children, their parent or legal guardian. All presentations of case reports must have consent to publish. You can use your institutional consent form. You should not send the form to us on submission, but we may request to see a copy at any stage (including after publication). Please also confirm you have followed national guidelines on data collection and release in the place the research was carried out, for example confirming you have Ministry of Science and Technology (MOST) approval in China.

If your manuscript does not contain any individual person's data, please state "Not applicable" in this section.

Competing Interests

All financial and non-financial competing interests must be declared in this section. See our editorial policies for a full explanation of competing interests. Where an author gives no competing interests, the listing will read 'The author(s) declare that they have no competing interests'. If you are unsure whether you or any of your co-authors have a competing interest please contact the editorial office.

Funding

All sources of funding for the research reported should be declared. The role of the funding body in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript should be declared. Please use FundRef to report funding sources and include the award/grant number, and the name of the Principal Investigator of the grant.

Author's Contributions

The individual contributions of authors to the manuscript should be specified in this section. Guidance and criteria for authorship can be found in our editorial policies. We would recommend you follow some kind of standardised taxonomy like the CASRAI CRediT (Contributor Roles Taxonomy).

Acknowledgements

Please acknowledge anyone who contributed towards the article who does not meet the criteria for authorship including anyone who provided professional writing services or materials.

Authors should obtain permission to acknowledge from all those mentioned in the Acknowledgements section. If you do not have anyone to acknowledge, please write "Not applicable" in this section.

See our editorial policies for a full explanation of acknowledgements and authorship criteria.

Group authorship: if you would like the names of the individual members of a collaboration group to be searchable through their individual PubMed records, please ensure that the title of the collaboration group is included on the title page and in the submission system and also include collaborating author names as the last paragraph of the "Acknowledgements" section. Please add authors in the format First Name, Middle initial(s) (optional), Last Name. You can add institution or country information for each author if you wish, but this should be consistent across all authors.

Please note that individual names may not be present in the PubMed record at the time a published article is initially included in PubMed as it takes PubMed additional time to code this information.

Authors' information (optional)

You may choose to use this section to include any relevant information about the author(s) that may aid the reader's interpretation of the article, and understand the standpoint of the author(s). This may include details about the authors' qualifications, current positions they hold at institutions or societies, or any other relevant background information. Please refer to authors using their initials. Note this section should not be used to describe any competing interests.

References

1. Rozenblatt-Rosen O, Stubbington MJ, Regev A, Teichmann SA. The human cell atlas: from vision to reality. Nature News 2017;550(7677):451.
2. Camara PG. Methods and challenges in the analysis of single-cell RNA-sequencing data. Current Opinion in Systems Biology 2018;7:47–53.
3. Lun A, Risso D, Korthauer K. SingleCellExperiment: S4 classes for single cell data. R package version 2018;1(0).
4. McCarthy DJ, Campbell KR, Lun AT, Wills QF. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. Bioinformatics 2017;33(8):1179–1186.
5. Grün D, Lyubimova A, Kester L, Wiebrands K, Basak O, Sasaki N, et al. Single-cell messenger RNA sequencing reveals rare intestinal cell types. Nature 2015;525(7568):251.
6. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nature biotechnology 2015;33(5):495.
7. Vickovic S, Ståhl PL, Salmén F, Giatrellis S, Westholm JO, Mollbrink A, et al. Massive and parallel expression profiling using microarrayed single-cell sequencing. Nature communications 2016;7:13182.
8. Allaire J. RStudio: integrated development environment for R. Boston, MA 2012;770.
9. Kluyver T, Ragan-Kelley B, Pérez F, Granger BE, Bussonnier M, Frederic J, et al. Jupyter Notebooks-a publishing format for reproducible computational workflows. In: ELPUB; 2016. p. 87–90.
10. StackOverflow, StackOverflow Developer Insights; 2019. https://insights.stackoverflow.com/survey/2019.
11. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome biology 2018;19(1):15.
12. Afgan E, Baker D, Van den Beek M, Blankenberg D, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic acids research 2016;44(W1):W3–W10.