Bioinformatics, 36(12), 2020, 3910–3912
doi: 10.1093/bioinformatics/btaa269
Advance Access Publication Date: 23 April 2020
Applications Note

Gene expression

alona: a web server for single-cell RNA-seq analysis

Oscar Franzén[1,*] and Johan L. M. Björkegren[1,2]

[1]Department of Medicine, Integrated Cardio Metabolic Centre, Karolinska Institutet, Huddinge 14157, Sweden and [2]Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY 10029, USA

*To whom correspondence should be addressed.

Associate Editor: Jan Gorodkin

Received on February 13, 2020; revised on March 27, 2020; editorial decision on April 15, 2020; accepted on April 16, 2020

Abstract

Summary: Single-cell RNA sequencing (scRNA-seq) is a technology to measure gene expression in single cells. It has enabled discovery of new cell types and established cell type atlases of tissues and organs. The widespread adoption of scRNA-seq has created a need for user-friendly software for data analysis. We have developed a web server, `alona` that incorporates several of the most popular single-cell analysis algorithms into a flexible pipeline. alona can perform quality filtering, normalization, batch correction, clustering, cell type annotation and differential gene expression analysis. Data are visualized in the web browser using an interface based on JavaScript, allowing the user to query genes of interest and visualize the cluster structure. `alona` accepts a compressed gene expression matrix and identifies cell clusters with a graph-based clustering strategy. Cell types are identified from a comprehensive collection of marker genes or by specifying a custom set of marker genes.

Availability and implementation: The service runs at https://alona.panglaodb.se and the Python package can be downloaded from https://oscar-franzen.github.io/adobo/.

Contact: p.oscar.franzen@gmail.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) is a powerful technology to measure gene expression in single cells as it provides more detailed information than bulk RNA-seq (Sandberg, 2014). A typical scRNA-seq experiment generates hundreds to thousands of transcriptomes. The rapid rise of scRNA-seq has created a wealth of scRNA-seq data and parallel to this development an increasing need for user-friendly data analysis software. A web server for analysis of scRNA-seq data unlocks access to researchers without having to learn programming.

Here, we describe `alona`—a public, fully automated web service, with a core written in the Python 3 programing language—that can be used to analyze, annotate and visualize scRNA-seq data. The tool takes advantage of a wide range of state-of-the-art scRNA-seq methods, normalization schemes and clustering algorithms as well as an intuitive web interface for data exploration. The web server accepts a compressed gene expression matrix in plain text format. The uploaded data are queued, processed and analyzed, often within an hour depending on the workload. Results are visualized in the web browser using a light-weight JavaScript library, which allows exploring cell clusters and gene expression using simple interactions. In addition, the analysis script is always provided so that the user can examine the code needed to reproduce the results.

2 Materials and methods

The analysis framework (named adobo; https://oscar-franzen.github.io/adobo/) is written in Python and runs on a virtual private server shared with the PanglaoDB web server (Franzén *et al.*, 2019). The backend is based on the LEMP stack. Jobs are queued and executed serially. The web interface allows the user to upload data and select analysis parameters (the default parameters are sensible and fit most experiments). During the data upload, the web server checks for data consistency and reports problems to the user. A typical experiment of ~3000 cells is processed within 10 min; an optional e-mail address can be specified to send a reminder when the analysis is completed. The web server does not require registration to be used; uploaded data are kept confidential and are automatically deleted after 7 days. Data are only seen within the scope of the present browser session, which is identified using a cookie containing a random string.

Pre-processing of the raw sequencing data (barcode demultiplexing, alignment and deduplication of unique molecular identifiers) is performed using external bioinformatics tools. The input data must be raw read counts in a matrix with genes as rows and cells as columns; the input file must also be compressed with gzip, zip, bzip2 or xz. The matrix can have a header or not. Fields are separated by tabs, spaces or commas. The Matrix Market format is also supported (https://math.nist.gov/MatrixMarket/formats.html); in which

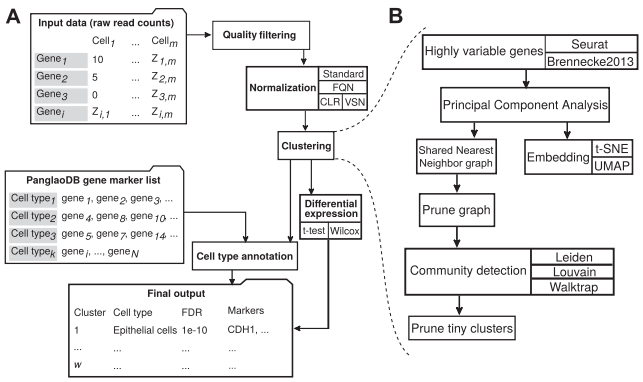

Fig. 1. Flowchart showing the main analysis steps in `alona`. (**A**) Global overview. (**B**) A detailed overview of the cell clustering process

case the input file should be a tar.gz archive containing three files: matrix.mtx.gz, barcodes.tsv.gz and genes.tsv.gz.

An overview of the analysis steps is shown in Figure 1 and the main steps are described here:

1. *Quality filtering*. Low quality cells are initially removed using simple thresholds (minimum number of total reads). Subsequently, the quality filtering approach from Lun *et al.* (2016) is applied. Cells are removed based on two quality metrics: (i) the log of the library size and (ii) the log of the number of detected genes. The median and median absolute deviation (MAD) is computed for (i) and (ii). For any cell, if (i) or (ii) are below a defined number of MAD (default is 3) from the median, the cell is removed. Uninformative genes are removed by requiring each gene to be expressed in a certain percent of cells (default is 1%). Doublet detection is performed in this step using the Scrublet package (Wolock *et al.*, 2019).
2. *Normalization*. Four normalization procedures are supported: (i) standard normalization (simple scaling of counts by library size); (ii) full-quantile normalization; (iii) centered log-ratio normalization; and (iv) variance-stabilizing normalization (Hafemeister and Satija, 2019). The standalone Python package also supports adjustment by gene length (RPKM).
3. *Batch correction (optional)*. The user can supply a list of batches (one per cell) to correct for known batch effects using the ComBat algorithm (Johnson *et al.*, 2007). An alternative to ComBat is to directly regress out batch effects using the function adobo.dr.regress.
4. *Feature selection*. Highly variable genes (HVG) are discovered using either: (i) a Seurat-like strategy, utilizing binning of genes according to average expression (Butler *et al.*, 2018) or (ii) the method described by Brennecke *et al.* (2013). Three additional methods are supported in the standalone package (Andrews and Hemberg, 2019; Chen *et al.*, 2016; Lun *et al.*, 2016). The default is to find 1000 HVG.
5. *Dimensionality reduction*. Principal component (PC) analysis is performed on the HVG with the method described by Baglama and Reichel (2005). The default setting is to identify 40 PCs. The Python package also supports the jackstraw method for identifying the optimal number of PCs to use. The 2D embedding is performed on PCs with *t*-Distributed Stochastic Neighbor Embedding (*t*-SNE) (van der Maaten and Hinton, 2008) (perplexity is set to 30 as default) or Uniform Manifold Approximation and Projection (UMAP) (Becht *et al.*, 2019).
6. *Clustering*. The PCs are searched for *k*-nearest neighbors using the BallTree algorithm. A shared nearest neighbor graph, with weights as the number of shared neighbors, is generated and pruned. Cell clusters are identified from the graph with the Leiden (Traag *et al.*, 2019), Louvain or Walktrap (Pons and Latapy, 2005) algorithms. For Leiden and Louvain, cluster resolution is set to 0.6 as default (decreasing this value gives larger clusters and vice versa).
7. *Cell type annotation*. The method for cell type annotation was described in Franzén *et al.* (2019). Annotation of cell types is performed at the cluster level. Cluster-level analysis is faster than cell-level analysis since not every cell needs to be considered; it also reduces the impact of molecular dropout events and cell doublet artifacts, which frequently contaminate scRNA-seq data. Gene expression in clusters is represented by taking the median across all cells. The procedure estimates gene expression activity of a set of marker genes and then ranks the resulting cell types. Significance is determined by computing a one-sided Fisher's exact test for each cell type and adjusting *P*-values with the Benjamini–Hochberg procedure. An acceptable false-discovery rate was chosen to be 10%. Thus, if the adjusted *P*-value is higher than 0.1, the cell type receives an 'Unknown' annotation. Custom marker genes can be entered or the user can choose to simply use markers from PanglaoDB. The latter option only supports mouse and human data. The present function is implemented in adobo.bio.cell_type_predict.
8. *Differential gene expression*. The first step involves all-versus-all cluster comparisons; i.e. every gene is compared between every pair of clusters. Two methods are available for generating the initial set of comparisons: (i) linear models followed by *t*-tests, similar to the limma R package (Ritchie *et al.*, 2015) or (ii) Wilcox tests, as a non-parametric option. The latter is computationally much slower since *t*-tests were implemented using vectorized operations. To generate a single *P*-value for every gene, pairwise *P*-values are combined for every gene using Fisher's method. Multiple testing correction is then applied with the Benjamini–Hochberg procedure. Tests are subsequently filtered based on two criteria: (i) adjusted *P*-value ¡ 0.01 and (ii) the number of cells expressing the gene in the cluster must be above a specified threshold (default is 80%).

Results can be downloaded as a tar.gz archive as well as visualized in the web browser. Supplementary Figure S1 shows an overview of the interface and contains descriptions of analysis output files.

3 Results and discussion

3.1 Test case: PBMC

To demonstrate the utility of `alona`, we applied it on a dataset consisting of 8381 peripheral blood mononuclear cells (PBMC). The

dataset came from a healthy human donor and it was originally generated by 10X Genomics. Cells were clustered with default settings into 20 groups. Supplementary Figure S2 shows a UMAP plot of the data (colors correspond to clusters). Six cell types were identified (number of cells in parenthesis): T memory cells (3404), monocytes (2224), NK cells (1331), B cells (1222), platelets (91) and plasmacytoid dendritic cells (66). The identified cell types are commonly found in blood, and their proportions were consistent with the typical proportions reported in PBMC samples (Bolen *et al.*, 2011).

3.2 Comparison with existing web servers

A number of important web servers for scRNA-seq analysis have been developed, such as ASAP (Gardeux *et al.*, 2017), SCRAT (Ji *et al.*, 2017), iS-CellR (Patel, 2018), Granatum (Zhu *et al.*, 2017) and Single Cell Explorer (Feng *et al.*, 2019). The functionality of `alona` is comparable to the aforementioned services, with some notable differences: `alona` offers more choices in terms of algorithms; the clustering strategy is graph-based; cell type prediction is always performed—a key goal in most single-cell experiments. Finally, the backends of previously published web servers can, in most cases, not be executed standalone. The latter makes it impossible or difficult to reproduce results. Every analysis run by `alona` can be reproduced offline since the Python code for the analysis is always provided. Finally, alona automatically recognizes the Matrix Market format, which is common in NCBI's Gene Expression Omnibus. Supplementary Figure S3 shows a comparison matrix where key features are compared with five other web servers.

4 Conclusions

We have here presented a user-friendly software for scRNA-seq analysis, `alona`. Development of alona will continue and we plan to expand the number of supported algorithms and analysis strategies.

Funding

This work was supported by the Karolinska Institutet & AstraZeneca Integrated Cardio Metabolic Centre (to J.L.M.B.); the Fondation Leducq – Transantlantic PlaqOmics Network (to J.L.M.B.); Hjärt- och Lungfonden [20170265 to J.L.M.B.]; and Vetenskapsrådet [2018-02529 to J.L.M.B].

Conflict of Interest: none declared.

References

Andrews,T.S. and Hemberg,M. (2019) M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics*, **35**, 2865–2867.

Baglama,J. and Reichel,L. (2005) Augmented implicitly restarted lanczos bidiagonalization methods. *SIAM J. Sci. Comput.*, **27**, 19–42.

Becht,E. *et al.* (2019) Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.*, **37**, 38–44.

Bolen,C.R. *et al.* (2011) Cell subset prediction for blood genomic studies. *BMC Bioinformatics*, **12**, 258.

Brennecke,P. *et al.* (2013) Accounting for technical noise in single-cell RNA-seq experiments. *Nat. Methods*, **10**, 1093–1095.

Butler,A. *et al.* (2018) Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.*, **36**, 411–420.

Chen,H.-I.H. *et al.* (2016) Detection of high variability in gene expression from single-cell RNA-seq profiling. *BMC Genomics*, **17**, 508.

Feng,D. *et al.* (2019) Single Cell Explorer, collaboration-driven tools to leverage large-scale single cell RNA-seq data. *BMC Genomics*, **20**, 676.

Franzén,O. *et al.* (2019) PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database*, **2019**.

Gardeux,V. *et al.* (2017) ASAP: a web-based platform for the analysis and interactive visualization of single-cell RNA-seq data. *Bioinformatics*, **33**, 3123–3125.

Hafemeister,C. and Satija,R. (2019) Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biol.*, **20**, 296.

Ji,Z. *et al.* (2017) Single-cell regulome data analysis by SCRAT. *Bioinformatics*, **33**, 2930–2932.

Johnson,W.E. *et al.* (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, **8**, 118–127.

Lun,A.T.L. *et al.* (2016) A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Res.*, **5**, 2122.

Patel,M.V. (2018) iS-CellR: a user-friendly tool for analyzing and visualizing single-cell RNA sequencing data. *Bioinformatics*, **34**, 4305–4306.

Pons,P. and Latapy,M. (2015) In: Yolum,P. *et al.* (eds) *Computer and Information Sciences - ISCIS 2005*. Vol. **3733**. Springer, Berlin Heidelberg, pp. 284–293.

Ritchie,M.E. *et al.* (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.*, **43**, e47.

Sandberg,R. (2014) Entering the era of single-cell transcriptomics in biology and medicine. *Nat. Methods*, **11**, 22–24.

Traag,V.A. *et al.* (2019) From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.*, **9**, 5233.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

Wolock,S.L. *et al.* (2019) Scrublet: computational Identification of cell doublets in single-cell transcriptomic data. *Cell Syst.*, **8**, 281–291.

Zhu,X. *et al.* (2017) Granatum: a graphical single-cell RNA-Seq analysis pipeline for genomics scientists. *Genome Med.*, **9**, 108.